OPSWAT.



MetaDefender™ for Secure Storage

MetaDefender Storage Security

4.6.2

Table of Contents

Introduction 11
- Overview 12
- Architecture & Components 13
- Planning your Deployment 16
- Sizing Guide 19
- General System Requirements 22
  - Unix deployment 24
  - Windows deployment 26
- Licensing Model 28
- MetaDefender Storage Security Documentation 29
- Choosing your deployment model 30
  - Basic Deployment 30
  - Basic Deployment with Self-Hosted Shared Services 31
  - Basic Deployment with Managed Services 32
  - Advanced Deployment with Managed Services 34
  - Advanced Deployment with Self-Hosted Shared Services 35
  - Cloud Deployment with Kubernetes (k8s) 37

Installation 39
- Types of deployment 40
- Deploy to cloud service providers 41
  - CSP Deployment Overview 41
  - Architecture Examples in CSPs 41
  - General Cloud Considerations 48
  - Using OPSWAT´s image from CSPs Marketplace 49
  - AWS Deployment 50
  - Azure Deployment 65
  - GCP Deployment 80
  - External Services Configuration 95
- ON-PREMISE DEPLOYMENT 111
  - On-Premise Deployment Overview 111
  - Production Considerations 111
  - Installation Methods 120
  - Upgrade MDSS 137
  - Uninstall 143
  - Troubleshooting On-Premise 145

containerized deployments 146
Containerized Deployment Overview 146
Kubernetes Deployment 146
Portable Package 159

Integrations 160

Integrations 161
Add An Amazon S3 Storage 164
How to Enable Inline Scanning? 172
How to verify if tagging functionality is supported? 173
How to Enable Autodiscovery 174
Add A Cubbit Storage 177
Add An Oracle Cloud Storage 179
Add An Oracle SDK Storage 184
Add A Wasabi Storage 187
Add a Cloudian Storage 191
Add a MinIO Storage 194
Add an OneDrive Storage 196
Add a SharePoint Online Storage 204
How to Enable Autodiscovery 213
Add a Microsoft Teams Storage 215
Add a SharePoint On-Premise Storage 217
Configuring Error Handling and Retry Mechanisms for SharePoint On-Premise 218
Add a Box Storage 223
Add an Azure Files Storage 233
Add an Azure Blob Storage 235
How to Enable Autodiscovery 243
Add an Alibaba Cloud Storage 246
Add a Google Cloud Storage 248
How to Enable Autodiscovery 251
Add an OPSWAT MFT Storage 253
Add an NFS Storage 258
[Linux] How to add an NFS storage unit 259
[Windows] How to add an NFS storage unit 262
How to add an Amazon EFS as an NFS storage unit 264
[Windows] How to Expose an SMB Share as an NFS Storage Unit 267
How to Enable NFS scanning on a Kubernetes Cluster 269
What is User Squashing for Network File System (NFS) 270
Add an NFS with Data Wrapping Storage 272

Add an SMB-Compatible Storage 274
How to scan Windows client SMB shares? 274
Recommended approach when adding SMB units on Linux deployments 275
Add an SMB-Compatible unit in Windows deployments 276
Add an SMB-Compatible unit in Linux deployments 279
Add a NetApp ONTAP Storage 288
Vscan | How to enable 290
Vscan | Best practices 304
Vscan | Scalability 305
AWS FSx for NetApp ONTAP 308
Vscan | How to connect multiple SVMs 310
Vscan | Configuration 315
ONTAP Connector Configuration file 319
Add an SFTP Storage 335
Add a Dell EMC Isilon Storage 339
Add an FTP Storage 341
Configuring a Windows IIS FTP Server for MDSS Scanning 343
Add a Scality RING Storage 349

Operations 354
Onboarding 355
Dashboard 361
Scanning 366
Password Protected Files and Encrypted Archives 375
Reprocess files 377
Identity Scanning 381
Reports 387
Settings 389
Importing/Exporting a configuration 390
Scan Pools 395
Workflows 397
Autodiscovery 406
Inventory 412
Cross-Domain Transfer (Data Diode) 421

Configuration and Management 424
Custom configuration 425
Web interface port 427
Real Time Processing configuration for NAS 428

Temporary local files created during remediation 430
Performance Tuning Guide 433
Configuring HTTPS 437
Custom CA certificates 440
Enable Real-Time Processing (RTP) 442
Polling handling 444
Event-based handling 445
Event-Based request format 446
Event-Based Real-Time Cloud Functions Examples 449
Event-based handling for Box 451
Event-based handling for SharePoint Online 454
Event-Based Real-Time handling for AWS 457
Event-Based Real-Time handling for Azure 464
Event-Based Real-Time handling for OracleSDK 467
Workflows 473
File Tagging 473
Deep Content Disarm and Reconstruction (Deep CDR) 478
Keep, Copy, Move or Delete Files 481
Delete Empty Folders 492
Remediation Process Retries 495
File Definition 495
Technologies 496
Scan File Versions 497
File Filters 501
User Management 509
Reset/forgot password 514
Change your account information 520
Generate an API Key 521
SMTP 524
License 527
Notifications 528
Email reports 529
User registration request notifications 533
Blocked file notifications 537
Scan completed notifications 538
File scanned notifications 541
File scanned RabbitMQ notifications 545
External Loggers Configuration 550
Kafka External Logger 551

Syslog UDP External Logger 554
Syslog TCP External Logger 557
AWS CloudWatch External Logger 561
Single Sign On 564
AD FS Specific Configuration 564
Okta Specific Configuration 572
Azure AD Specific Configuration 582
PingFederate Specific Configuration 595
AD FS SAML2 Specific Configuration 603
Database procedures 612
AWS RDS with PostgreSQL 612
Exporting database 616
Upgrade Guide to PostgreSQL (MDSS v4.0.0) 617
Database Vacuum 623
Database Retention 625
File Discovery Debounce 626
Active-Active configuration 628
Security & Encryption Keys 630
My OPSWAT Central Management 632
Enrolling the product to Central Management 632
Telemetry 636
Automatic Scaling (Autoscaler) 637
Automatic scaling settings 647

Knowledge Base 659

The MDSS API Gateway - What It Is and Which Port It Uses 660
How to create an MDSS Scan Start API: Moving Files Between NFS Storages After Sanitization 664
Why Does MetaDefender Storage Security Intermittently Show "License Information Is Not Available" on RHEL 8? 666
CONFIGURATION _ SETUP 669
Is there a limitation on importing configuration to MetaDefender Storage Security? 669
[Linux] What are MDSS-related resources and how to perform disk cleanup? 669
How to Enable Specific Modules in the Configuration? 671
Which API endpoints from MD Core does MDSS utilize? 674
Advanced File API Filtering and Sorting 674
Is SNMP monitoring supported for MetaDefender Storage Security? 676
[Linux] How to download docker images from alternate location? 676
[Windows] How to create a persistent mount for SMB and NFS? 676

[K8s] How to enable Horizontal Pod Autoscaling based on RabbitMQ queue metrics 681
How to Configure and Use Syslog and Serilog Sinks? 690
Does MetaDefender Storage Security support symlinks scanning? 693
Does a CVE affect MetaDefender Storage Security? 694
How do I Check Port 80 and 443 Usage Before Installing MDSS on Windows and Linux? 695
What's the Best Practice for Managing Logs and Data Retention in MDSS? 698
How to disable the Local PostgreSQL when using an external Database? 700
Why remediation actions are not working? 701
What is the expected MetaDefender Storage Security Remediation Behavior Behind F5 Load Balance? 703
Why connection to SMB share cannot be established? 704
What permissions are needed for the user of the storage of Google Cloud? 705
How to Integrate Cerberus SFTP with MDSS? 707
How to Handle SSL/TLS Issues with OPSWAT MetaDefender Core Webhook Callback? 722
Why does MetaDefender Storage Security reset the RTP StartTime every day? 723
How to Exclude the OPSWAT Folder from Windows Defender? 724
What are the Syslog Facilities and Levels in MetaDefender Storage Security? 727
How to protect the customer.env file in MetaDefender Storage Security? 729
How Can I Improve Folder Discovery and Scanning Performance in MetaDefender Storage Security? 729
How to create a PostgreSQL Database in AWS? 731
How to set up Active Active configuration? 736
AWS Lambda with S3 Trigger for MDSS Real-Time Processing 738
How to configure SMTP server on MetaDefender Storage Security to send notification to Gmail mailbox? 744
What are the Maximum and Minimum Data Retention Periods for MetaDefender Storage Security? 746
How to change PostgreSQL DB port used by MDSS 747
What is the latest MetaDefender Storage Security version? 749
Why is the Discovery Service becoming unresponsive due to RabbitMQ connection issues? 751
Changing the Authority URL in SSO will redirect to initial one 751
Scan Priorities & Throughput Tuning 753
Service Instance Scaling on Linux 759
How to move the location of the MetaDefender Storage Security services_logs folder? 763
How to install RabbitMQ on Windows as an external service? 764
Endpoint Protection Exclusions for MetaDefender Storage Security running on Windows 770
Scanning NetApp Volumes with Tiered (Hot and Cold) data 785
System Health APIs: which one to use 790
Which File Operations Trigger a Real-Time Processing Scan? 806
Errors and Troubleshooting 813

Why does MetaDefender Storage Security get stuck in a remediation loop, causing inaccessibility? 813
How to fix 'Name does not resolve' error in RHEL / Rocky Linux 8 with Podman 814
Why is my MetaDefender Storage Security account disabled due to too many failed login attempts? 816
Troubleshooting errors when configuring HTTPS communication for webserver 817
Why does a report fail to load with error "API took too long to respond"? 819
Why am I receiving the error "PostgreSQL migrations did not finish" when installing MetaDefender Storage Security? 821
Why am I getting a "stat /etc/mdss/docker-compose.onedrive.yml: no such file or directory" error when upgrading MetaDefender Storage Security from 3.6.1 to 4.1.0? 823
Why is RabbitMQ consuming excessive memory in MetaDefender Secure Storage (MDSS) Kubernetes deployment? 825
Is the Scheduled Scan in MetaDefender Storage Security Based on OS Time or an Internal Timer? 826
Why does MDSS installation might fail with “net/http: TLS handshake timeout” when pulling Docker containers? 827
How to troubleshoot slow historical data retrieval from the AVEVA PI Server? 829
How to resolve “Remediation Reached the Configured Timeout” for SMB Storage? 832
Slow 'mdss.ps1' commands on Windows 834
Is MetaDefender Storage Security (MDSS) Affected by PHP2Shell (CVE-2026-109618 and CVE-2026-109619)? 837

Troubleshooting Guide 839

Diagnostic 840
Logs location 840
How to read the logs 840
How to generate a support package 842
How to Enable Extra Logging in the Configuration? 845
Enable debug logs for script commands 849
Resolving error code 1722 when OPSWAT ONTAP Connector tries to connect to ONTAP AV Connector 849
INSTALLATION 856
Unable to remove docker network 856
Running behind a proxy 856
Why I am getting error "No available rule is present for scanning" 858
Why am I getting the 'Configured user limit on inotify instances has been reached' error in MDSS on Linux/Kubernetes? 861
How to handle empty objects in Discovery? 863
Permission denied while uninstalling on Linux 863
INTEGRATIONS 865

NFS storage unit is not detecting any files 865
Why my files are not moving on SharePoint On-Prem with MDSS, despite UI confirmation? 866
Scan stuck in the "processing" state on the SMB Service 866

Release Notes 868

Release notes 869
Changelog 874

MetaDefender Storage Security API 891

Introduction

Overview

MetaDefender Storage Security offers a robust solution for protecting enterprise data stored in various on-premises, cloud, and hybrid environments. It is designed to prevent data breaches, downtime, and compliance violations by scanning files for malware, vulnerabilities, and sensitive data.

A crucial aspect of MetaDefender Storage Security's operation is its **exclusive bundling and integration with MetaDefender Core and its associated engine packages.**

MetaDefender Storage Security itself does not perform the deep file analysis. Instead, it acts as a component that connects to various storage solutions (Amazon S3, Azure Blob, network-attached storage, etc.) and orchestrates the scanning process by sending files to a MetaDefender Core instance.

MetaDefender Core is the underlying analysis engine that utilizes multiple proprietary OPSWAT technologies, including:

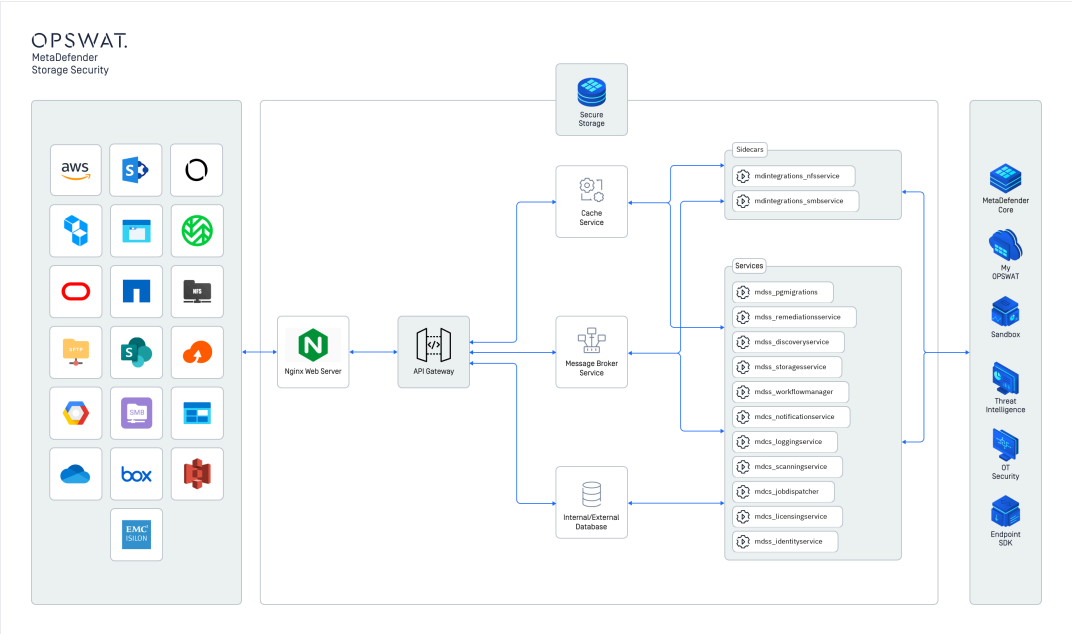
- Multiscanning leverages multiple anti-malware engines to detect a broad range of known threats.
- Deep CDR (Content Disarm and Reconstruction) sanitizes files by removing potentially malicious content and reconstructing safe, usable versions, effectively preventing zero-day attacks.
- Proactive DLP (Data Loss Prevention) identifies and redacts or blocks sensitive information within files to help meet compliance requirements. **NOTE:** MetaDefender Storage Security does not process the redacted file; it only displays the detected sensitive data in a file.
- File-Based Vulnerability Assessment detects known vulnerabilities within software components.

MetaDefender Storage Security functions as the layer that integrates with your storage systems and manages the file processing workflow, while MetaDefender Core, with its various licensed engine packages, performs the actual threat detection, sanitization, and data protection tasks.

This architecture ensures that files are thoroughly inspected before they are stored or accessed, providing a comprehensive security layer for enterprise data. Deployment options may involve a self-hosted MetaDefender Core, a cloud-based MetaDefender Cloud instance (which includes Core functionalities as a service), or MetaDefender Core deployed via AMIs or containers.

Architecture & Components

MetaDefender Storage Security (MDSS) is built for flexibility. It uses containers, which let you customize and adapt your development to match your specific needs. Getting the sizing right is crucial for performance and reliability as your workload directly impacts sizing. Proper sizing ensures smooth operation. This guide will help you understand the key components and make informed decisions.



MDSS Components

MDSS is a set of services that share the infrastructure below. Storage-type specific code (Amazon S3, Azure, Google Cloud, Box, SharePoint, SFTP, FTP, and so on) is loaded in-process by the Storages, Discovery, Remediations and Scanning services; there is no separate container per storage type. Only NFS and SMB have a dedicated sidecar service.

Component	Role	Log folder
Web Client	The management console served by NGINX. Forwards API calls to the API Gateway	`services/webclient`
API Gateway	The REST API used by the web client, the public API and the OPSWAT ONTAP Connector	`mdcs_apigateway`
Identity Service	Users, roles, API keys and single sign-on	`mdss_identityservice`
Licensing Service	License activation and enforcement	`mdcs_licensingservice`
Job Dispatcher	Schedules scans and tracks job progress	`mdcs_jobdispatcher`
Storages Service	Accounts, groups and storage units, including connection tests and autodiscovery	`mdss_storagesservice`
Discovery Service	Enumerates the objects in each storage unit and detects changes for real-time protection	`mdss_discoveryservice`
Scanning Service	Downloads objects and submits them to the configured scan instance (MetaDefender Core, MetaDefender Cluster or MetaDefender Cloud)	`mdcs_scanningservice`
Remediations Service	Applies the remediation actions of the workflow: delete, move, copy, tag, replace with the sanitized copy	`mdss_remediationsservice`
Workflow Manager	Workflows, scan instances and remediation rules	`mdss_workflowmanager`
Notification Service	Email, webhook and RabbitMQ notifications	`mdcs_notificationservice`
Logging Service	Audit log, with optional forwarding to external loggers (Syslog, Kafka)	`mdcs_loggingservice`
NFS Service, SMB Service	Sidecars that mount NFS and SMB shares on behalf of the central services. Started only when the `nfs` or `smb` module is enabled	`mdintegrations_nfsservice`, `mdintegrations_smbservice`

Database migrations	One-shot jobs that upgrade the database schema at startup, then exit	`services/pgmigrations`, `services/mongomigrations`

On Linux, the platform services (Identity, Licensing, Job Dispatcher, Logging, Scanning, Notification, Discovery, Remediations, Storages, Workflow Manager) run by default in one bundled container (`mdss_platform`). Setting `COMMON_SERVICES_ENABLED=no` runs each of them in its own container instead. The log folders are the same in both modes. On Windows, the services run as native processes managed by a single Windows service, `mdss`.

Shared Services

MDSS depends on the following stateful services to ensure high performance and data consistency:

- **PostgreSQL (Database)** – Stores configuration, job metadata, and system state. Proper sizing ensures transactional integrity and I/O performance at scale.
- **RabbitMQ (Message Broker)** – Coordinates communication across system components. It must be horizontally scalable to handle high volumes of concurrent file scan tasks.
- **Redis (Cache)** Provides low-latency access to ephemeral data and scan results. Effective caching reduces load on other components and speeds up workflow processing.
- **MongoDB** – Present only on installations upgraded from versions that used it. New installations do not deploy it.

Why These Services Matter:

- **Scalability** – Each service must scale independently to accommodate varying workloads.
- **Reliability** – As stateful services, downtime or data loss can impact the entire system's operation.
- **Performance** – Throughput and responsiveness of the system depend on proper resource allocation for these services.

Planning your Deployment

MetaDefender Storage Security (MDSS) offers versatile deployment options to align with your organization's specific infrastructure and security requirements. This includes support for air-gapped environments. Understanding the different deployment models will help you choose the most suitable approach.

Regardless of the deployment type, make sure you have reviewed the **recommended system requirements** for MetaDefender Storage Security.

MetaDefender Storage Security can be deployed across various platforms:

- **Physical Servers** for direct hardware installations.
- **Virtualization Platforms** compatible with popular platforms like VMware, Hyper-V, and XenServer.
- Can be deployed on **Infrastructure as a Service** from major cloud providers such as AWS, Azure, and GCP.
- Supports containerized deployments in **Kubernetes** clusters.

The Basic and Advanced Deployment types can be broadly categorized based on how the shared services (database, message broker, and cache) are managed:

Deployments with Self-Hosted Shared Services

- Usage Cases
  - Shared services are segregated from the main MetaDefender Storage Security instance, often onto a single dedicated VM (Linux is preferred). This is suitable for allocating dedicated resources to these services. (Basic Deployment)
  - Each shared service is installed on separate instances to enhance reliability, scalability, and high availability, crucial for managing larger workloads. The key difference from the basic setup is the implementation of high availability for one or more services and potentially more advanced scalability features like database clustering (sharding) if required. (Advanced Deployment)
- High-Level Steps
  1. Prepare the system(s) by updating packages
  2. Deploy PostgreSQL (Database)
  3. Deploy RabbitMQ (Message Broker)
  4. Deploy Redis (Cache)
  5. Configure MetaDefender Storage Security to connect to these self-hosted shared services
  6. Start MetaDefender Storage Security

7. (For advanced deployments) Implement and configure scalability and high availability for the shared services

Deployments with Managed Shared Services

This model leverages cloud-provider-managed services for the database, message broker, and cache, reducing the operational overhead of managing these components yourself.

- Usage Scenarios
  - A single MetaDefender Storage Security instance utilizes managed cloud services for its shared components (Basic Deployment)
  - Multiple MetaDefender Storage Security instances are deployed, all connecting to the same managed cloud services. This enhances capacity, reliability, and high availability (Advanced Deployment)
- Characteristics
  - Easily deploy multiple MetaDefender Storage Security instances. Each new instance is configured to use the same managed shared services.
  - For deployments with multiple instances, a **load balancer** is recommended, especially for event-based, real-time scanning, to ensure high availability and distribute traffic.
    - Cloud - use the cloud provider's load balancing services.
    - On-Premises - manually deploy load balancers like NGINX, HAProxy, or Traefik.
  - Managed services are configured within MetaDefender Storage Security similarly to self-hosted ones, though specific terminology might vary by cloud provider.
  - Examples of Managed Services:
    - Database (PostgreSQL alternatives): Amazon RDS, Azure Database for PostgreSQL, Google Cloud SQL
    - Cache (Redis alternatives): AWS ElastiCache for Redis, Azure Cache for Redis
    - Message Broker (RabbitMQ alternatives): AWS Amazon MQ, Azure MP Cloud AMQP and Bitnami package for RabbitMQ in Azure MP.

Kubernetes Deployment

This approach involves deploying MetaDefender Storage Security within a Kubernetes cluster using a Helm chart.

- Prerequisites
  - An existing Kubernetes cluster (Managed Kubernetes Services like EKS, AKS, GKE are common for production, while self-managed Kubernetes offers more control

but requires more maintenance).

- For deployments using the OPSWAT-provided convenience script (`metadefenderk8s.sh`):
  - Familiarity with the chosen Cloud Service Provider (AWS or Azure)
  - An account with the CSP to create the necessary resources
  - Linux shell environment
  - Tools: Terraform, Helm, AWS-CLI (for AWS), Kubectl

- Deployment Method
  - The OPSWAT **convenience script** (`metadefenderk8s.sh`) is the simplest way to provision a Kubernetes cluster (if needed) and install MetaDefender Storage Security. It guides users through the required steps.

Choosing the right deployment type depends on your existing infrastructure, technical expertise, scalability needs, and budget.

Carefully evaluate these factors and the prerequisites for each model before proceeding with your MetaDefender Storage Security installation.

Sizing Guide

This table summarizes the various deployment options available for MDSS. It helps you understand the differences between each option and choose the one that best fits your needs.

		Small scale deployments	
Feature	**Basic Deployment (Single Instance)**	**Basic Deployment with Self-Hosted Shared Services**	**Basic Deployment with Managed Services**
Best for	*Small and predictable workloads*	*Small workloads, some flexibility, HA desired*	*Small workloads, with managed services*
Scalability	NO	NO	NO
High Availability	NO	NO	PARTIAL
Auto-Scale Ready	NO	NO	NO
Files / hour (Objects / hour)	**5,000** (125,000)	**20,000** (500,000)	**50,000** (1,250,000)
Recommended MD Core #	2-4	4 or more	4 or more
Recommended MDSS #	1	1	1
Infrastructure complexity	1/5	4/5	3/5

	Medium scale deployments		Large scale deployments	
Feature	**Advanced Deployment with Self-Hosted Shared Services**	**Advanced Deployment with Managed Services**	**Cloud Deployment with Kubernetes (k8s)**	**MDSS Cloud (SaaS)**
Best for	*Medium workloads, manual scaling*	*Medium to Large workloads, no auto-scaling*	*Large workloads, dynamic scaling, cost-optimized*	Scalable, secure, efficient cloud-based storage security solution
Scalability	YES	YES	YES	YES
High Availability	YES	YES	YES	YES
Auto-Scale Ready	NO	NO	YES	YES
Files / hour (Objects / hour)	**100,000** (2,500,000)	**100,000** (2,500,000)	**100,000** or more	**500,000** or more
Recommended MD Core #	4 or more	4 or more	Tailored per customer	Tailored per customer
Recommended MDSS #	2 or more	2 or more	Tailored per customer	Tailored per customer
Infrastructure complexity	5/5	4/5	1/5	1/5

How we tested and what it means

We tested MDSS with a mix of everyday files, like documents, images, and videos, to get realistic performance numbers. We started small and then scaled up to see how it handled larger workloads. However, remember that these are just guidelines. Your actual performance will depend on your files, network, and hardware. For critical systems, it's always best to test it out yourself.

The numbers in the deployment table are based on rigorous testing using a diverse set of real-world files. We didn't just use one type of file; we included a mix of:

- **Documents** - PDFs, Word files (DOC/X), Excel files (XLS/X), PowerPoint files (PPT/X)
- **Executables** - EXEs, MSIs (installation files)
- **Images -** BMP, JPG, PNG
- **Media** - MP3, MP4 (audio and video)

- **Text and Archives** - various text files and compressed archives

The Numbers Explained

- **our starting point was 5,000 files**, totaling 7,728.5 MB. The average file size was 1.55 MB.
- to see how MDSS handles **bigger workloads**, we replicated this dataset to create scenarios with 50,000 files and over 100,000 files. This helped us understand performance scaling.

The numbers in the table are meant to be *guidelines*. Real-world performance can vary due to:

- **File Type Variations** - some file types are processed faster than others.
- **Network Speed** - a slow network can impact throughput.
- **Hardware Specs** - the performance of your servers matters.

For mission-critical applications we strongly recommend running your own performance tests (benchmarks) in your specific environment *before* deploying MDSS in production. This will give you the most accurate picture of how MDSS will perform with your unique setup.

General System Requirements

This page outlines the necessary prerequisites for deploying MetaDefender Storage Security (MDSS), covering supported operating systems, system requirements, and specific details for Unix and Windows deployments.

Operating system-specific system requirements are available at:

- Unix deployment
- Windows deployment

Prerequisites

Hardware Configurations

Depending on your deployment type, please ensure your hardware meets the following minimum requirements:

Basic Deployment (Minimum System Requirements)

Hardware	MDSS Dedicated Instance	MDSS + MD Core Shared Instance
CPU	16 cores (vCPUs)	32 cores (vCPUs)
RAM	32 GB	64 GB
Free Disk Space	500 GB*	900 GB*

Advanced Deployment (Minimum System Requirements)

For advanced usage, particularly when you intend to process a very large number of files, it is not recommended to run MD Core on the same machine as MDSS.

Hardware	MDSS Dedicated Instance	PostgreSQL Database Instance
CPU	16 cores (vCPUs)	16 cores (vCPUs)
RAM	32 GB	32 GB
Free Disk Space	150 GB*	380 GB*

*** Disk Space Planning (applicable to all deployments)**

> **Info**

> Please note that the database disk space requirements outlined in this document are calculated based on processing **10 million files***.
>
> **Important:** If you anticipate discovering and processing more than 10 million files, you will need to allocate additional disk space for the database proportionally.
>
> The disk space requirements for the application and logs remain constant regardless of the number of files processed.
>
> If you need assistance calculating the appropriate database disk size for your specific file volume, please don't hesitate to reach out.
>
> *Each file discovery creates a separate database entry, even if it's the same file.

Careful disk space planning is crucial for the optimal performance and reliability of the MetaDefender Storage Security platform.

- Base Installation (database excluded):
  - MDSS Application: **20 GB**
  - Default install path:
    - Windows: `C:\Program Files\OPSWAT\MetaDefender Storage Security`
    - Linux: `/etc/mdss`
- Operational Requirements
  - A minimum of **100 GB** is recommended for **log retention and analysis**
    - Default install path:
      - Windows:
        - `C:\Program Files\OPSWAT\MetaDefender Storage Security\logs`
        - `C:\Windows\System32\opswat.mdss.windows.service`
        - Installer logs: `C:\Users\<user>\AppData\Local\Temp\`
      - Linux: `/var/log/mdss` and `/etc/mdss/sevices_logs`
  - For the various MDSS services (database included), an additional **380 GB** is required
    - Default install path:
      - Windows: `C:\Program Files\OPSWAT\MetaDefender Storage Security\services`
      - Linux: `/etc/mdss`
- This heavily depends on the actual dataset and configuration in a production environment.
- More details on the exact logging paths can be found at Logs location

Note: Insufficient disk space may negatively impact system performance and reliability, particularly during high-volume remediation operations. Regular monitoring of disk usage

is strongly advised to ensure optimal system performance.

Unix deployment

Supported Operating Systems

Distribution	Supported versions	Architecture
CentOS Stream	9.x, 10.x	x86_64 (64-bit)
Red Hat Enterprise Linux	8.x, 9.x, 10.x	x86_64 (64-bit)
Rocky Linux	8.x, 9.x, 10.x	x86_64 (64-bit)
Alma Linux	9.x, 10.x	x86_64 (64-bit)
Debian	11.x, 12.x, 13.x	x86_64 (64-bit)
Ubuntu	22.04, 24.04, 26.04	x86_64 (64-bit)
Amazon Linux	2, 2023	x86_64 (64-bit)

Installation details

The default installation path for MetaDefender Storage Security resources is the `/etc/mdss` folder.

The installation package will automatically install the following dependencies if they are not already present on the system:

Debian / Ubuntu	Red Hat Enterprise / CentOS
- `curl` - `openssl` (uses the default version on the host) - `openssl` (the container uses version 3.2.0) - `ca-certificates` - `sudo` - `jq` - `iproute2` - `apt-utils` - `gnupg-agent` - `apt-transport-https` - `software-properties-common`	- `curl` - `openssl` (uses the default version on the host) - `openssl` (the container uses version 3.2.0) - `ca-certificates` - `sudo` - `jq` - `iproute` - `yum-utils` - `gnupg2` - `wget` - `device-mapper-persistent-data` - `lvm2`

Installation components

MetaDefender Storage Security automatically deploys the following components:

Name	Version	Type
Docker Engine	system defined*	Containerization technology
Docker Compose	system defined*	Container orchestration
PostgreSQL	v17.0	Database
RabbitMQ	v4.3.5	Message broker
NGINX	v1.29.2	Web server
Redis	v8.6.6-alpine	Data Cache

Our system uses the existing Docker version if available. If not, it automatically downloads the latest stable version using the docker-install script.

> **On RHEL, Rocky Linux, AlmaLinux, CentOS Stream and Amazon Linux, install Docker before installing MDSS.** The installer cannot reliably install it for you on these distributions.

> On **RHEL 10 and AlmaLinux 10**, reboot after installing Docker. On a freshly updated system the running kernel and the installed kernel modules can be out of step, and `docker.service` fails to start until the machine restarts.

Using Podman instead

On RHEL and Rocky Linux, Podman is a supported alternative, and Red Hat supports it natively on those platforms. To use it, install `podman`, `podman-docker` and a compose tool, then set the following in `/etc/mdss/customer.env` **before** starting MDSS:

```
SKIP_DOCKER_CHECKS=yes
```

Additional prerequisite for offline and portable deployments

The offline and portable packages bundle Docker as static binaries, which carry no package dependencies of their own. `iptables` must be present or the bundled Docker daemon fails to start:

```
sudo dnf install -y iptables-nft      # RHEL family, Amazon Linux
```

Windows deployment

Supported Operating Systems

OS	Version	Architecture
Windows Server	2019, 2022, 2025 (latest)	64-bit

> **Operating System Requirements and Licensing Considerations**
>
> Microsoft's licensing terms prohibit the use of Windows Desktop operating systems in server environments. As a result, MetaDefender Storage Security is not compatible with Windows Desktop editions.
>
> For organizations concerned about Windows Server licensing costs, we recommend deploying on Unix-based systems instead. Unix platforms deliver enterprise-level reliability and performance without additional licensing fees.

Port Requirements

> **The following ports must be available for our services to listen on.**
>
> Please ensure that no other services are using these ports before starting the installation. NGINX port can be customized during the installation process.
>
> For the MDSS Windows deployment, every port in the below table is **TCP**, and the rule you need is an **inbound "Allow" rule** for each so the components can talk to each other and clients can reach the services. Open these before installing:

Component	Port
PostgreSQL	5432
RabbitMQ	5672
NGINX	80 (and 443 if SSL enabled)
API Gateway	8005
Licensing Service	5001
SMB Service	6002, 6003, 6004, 6005, 6006, 7002, 7003, 7004, 7005, 7006
SFTP Service	5003
NFS Service	5004

Installation components

MetaDefender Storage Security automatically deploys the following components:

Name	Version	Type
PostgreSQL	v17.0	Database
RabbitMQ	v4.3.0	Message Broker
NGINX	v1.29.2	Web Server
Garnet	v1.1.8	Data Cache (Redis compatible)

Licensing Model

Licensing for MetaDefender Storage Security is directly integrated with MetaDefender Core, as both solutions are delivered together. The licensing framework includes several key elements:

Licenses are provided through a subscription model, with agreements structured for annual or multi-year terms.

Pricing details, entitlements, and usage terms are outlined in a formal agreement with OPSWAT. For enterprise-level implementations, tailored quotes are provided to align with organisational requirements.

Activation is managed via a license key linked to a unique deployment ID generated by the associated MetaDefender Core instance. Both online and offline activation options are available.

A valid and appropriately tiered MetaDefender Core license is required to support MetaDefender Storage Security. This includes coverage for necessary engine packages such as anti-malware engines, Deep CDR, and Proactive DLP to meet the intended security standards.

A Bring Your Own License (BYOL) model is supported for deployments using cloud marketplaces like AWS Marketplace. In this scenario, customers acquire the license directly from OPSWAT and apply it within the cloud platform.

Organizations planning to implement MetaDefender Storage Security must consult OPSWAT or their sales representatives to receive personalized pricing and licensing guidance based on deployment size, security objectives, and selected engine packages.

MetaDefender Storage Security Documentation

MetaDefender Storage Security (MDSS) protects your storage against malware, zero-day threats, and hidden sensitive data. It scans files at rest and in real time across cloud and on-premises storage, using multiscanning with 30+ anti-malware engines, Deep CDR (Content Disarm and Reconstruction), Proactive DLP, and vulnerability assessment.

Use this documentation to plan, deploy, size, license, and operate MDSS across supported storage targets including Amazon S3, Azure Blob, Dell EMC Isilon, NetApp, SMB/CIFS and NFS shares, and Kubernetes-based cloud deployments.

What you'll find here

- Overview and how MDSS fits into an OPSWAT MetaDefender deployment
- Architecture and components, including MD Core scanning backend and shared services
- Planning your deployment and choosing a deployment model (basic, advanced, managed, self-hosted, or cloud/Kubernetes)
- Sizing guide for on-demand and real-time scanning, instance counts, and machine specs
- General system requirements and supported storage integrations
- Licensing model

Choosing your deployment model

MDSS supports basic and advanced deployments, with either managed or self-hosted shared services, plus a Kubernetes (k8s) option for cloud. Pick the model that matches your scale, storage type, and operational preferences. See the deployment sections in the left menu for step-by-step guidance.

Download the full manual

Prefer a single file? Download link for the complete guide in PDF format for OPSWAT MetaDefender Storage Security (MDSS) 4.6.2 Documentation

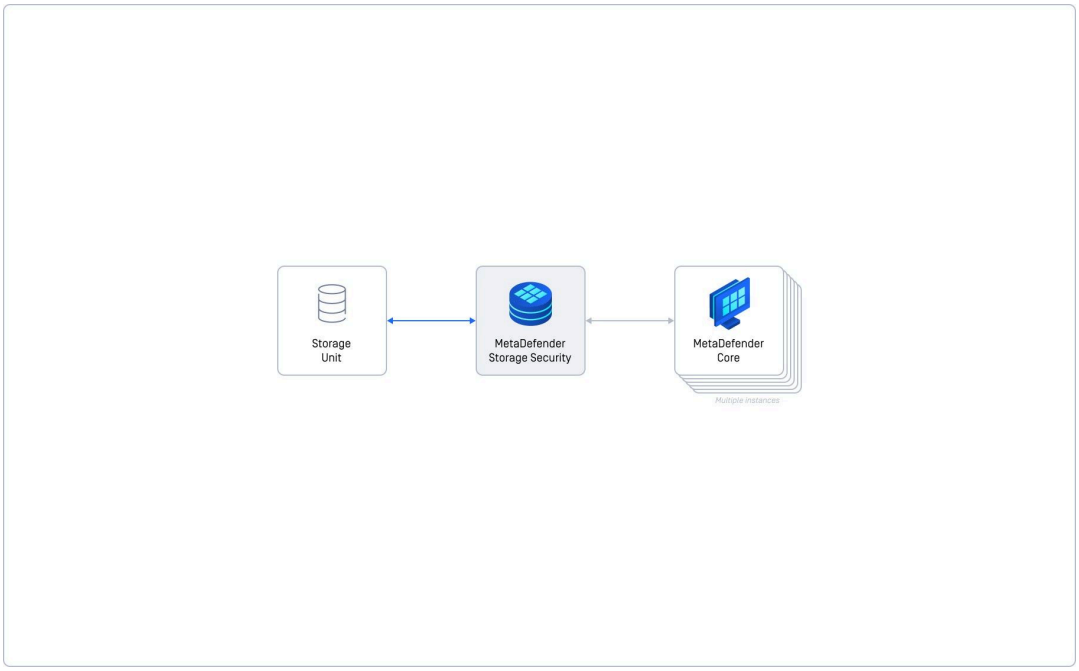
SHA256: 88FC91478CEA9A43E0191B9F5FEEAF4C3BE095F3BFD6EE44BC4F855F0A121805

Choosing your deployment model

Basic Deployment

The basic deployment is the fastest way to get **MetaDefender Storage Security (MDSS)** up and running. All system components — MDSS, Metadefender Core, and shared services — are installed on a single machine.

Imagine running MDSS on a single computer. It's fast to set up and good for small tasks. But if you need to handle a lot of work or can't afford any downtime, this isn't the right choice.



Suitable For

- Small workloads (perfect for testing or light usage)
- Quick setup with minimal configuration

Key Points

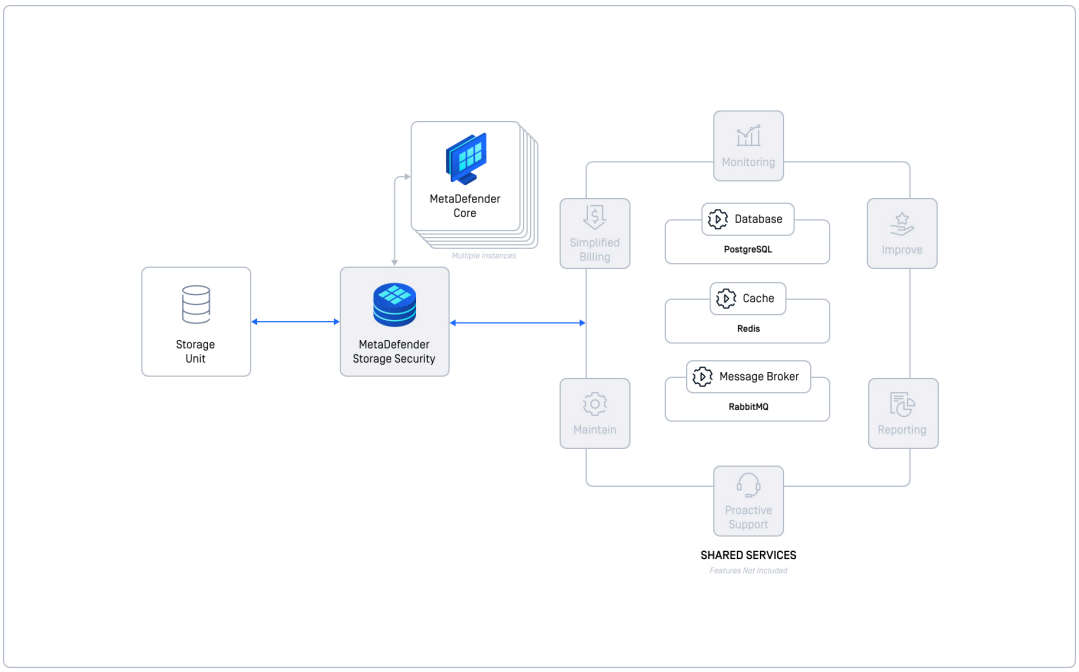
- All components run on one server.
- You can't add more servers to handle more load.
- If the server goes down, MDSS goes down.
- You can still reconfigure or upgrade MDSS on this setup.

Important Considerations

- Performance will vary depending on your server and file types.
- This setup is *not recommended* for live, critical systems where you need high availability or the ability to handle a lot of data.
- You get simplicity, but you lose the ability to scale and ensure continuous operation.

Basic Deployment with Self-Hosted Shared Services

This deployment takes the basic setup a step further by separating the core services (database, cache, and message broker) from the main MDSS instance. This gives you more control and improves performance, but adds complexity.



Instead of running everything on one machine, you're now running the database, memory cache, and communication hub on separate machines. This gives you more control and better performance, but you're also responsible for setting everything up and keeping it running smoothly.

Suitable For

- **Small to Medium Workloads** as it can handle up to 20,000 files per hour with proper resources.
- **Organizations with IT Expertise** as it requires in-house expertise to manage the infrastructure.

Key Components & Their Roles

- PostgreSQL (Database) stores data.
- Redis (Cache) speeds up data access.
- RabbitMQ (Message Broker) manages communication between components.

Key Points

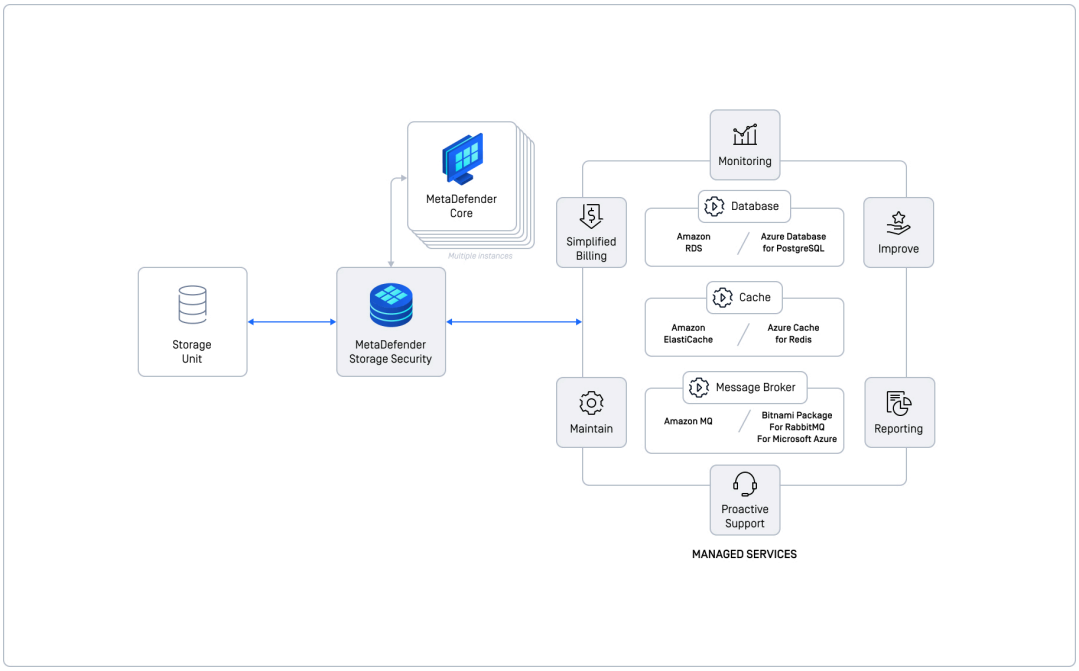
- Each shared service runs on its own server/VM.
- You are responsible for installing, configuring, and maintaining these services.
- Requires more technical knowledge and effort.
- Can handle larger workloads compared to a single instance.
- You can scale individual services, but it's not as seamless as a fully cloud-native setup.

Important Considerations

- You need to configure replication for high availability.
- Make sure sufficient resources are allocated to each shared service.
- Running all shared services on a single VM defeats the purpose of this setup and impacts performance.
- While better than a single instance, it's still possible to have single points of failure if not configured correctly.

Basic Deployment with Managed Services

Instead of managing the database, cache, and message broker yourself, you let the cloud provider handle it. You focus on setting up and scaling MDSS, while the cloud takes care of the underlying infrastructure. Think of it as renting managed services instead of buying and maintaining them yourself.



This deployment builds on the basic model by using *managed cloud services* for the shared components (database, cache, message broker). This means you leverage the cloud provider's expertise to handle the infrastructure, simplifying your setup.

Suitable For

- Cloud-First Organizations, organizations heavily reliant on cloud infrastructure.
- Organizations Seeking Simplicity as it reduces the burden of managing infrastructure.
- Workloads up to 50,000 Files/Hour - achievable with sufficient cloud resources and MDSS and MD Core instances.

Key Advantages

- Cloud provider handles maintenance, monitoring, and scaling of shared services.
- Ideal for organizations already using cloud services like AWS, Azure, or Google Cloud.
- Reduces operational overhead compared to self-hosting.
- Benefits from cloud-provided features like enhanced monitoring, reporting, and proactive support.

Example: Amazon AWS Integration

- Database - use Amazon RDS.
- Cache - use Amazon ElastiCache for Redis.
- Message Broker - use Amazon MQ.

Adaptable to Other Clouds

This approach can be adapted to Azure and Google Cloud, using their respective managed services. Please follow this article for more information regarding integrating with various cloud providers.

Key Points

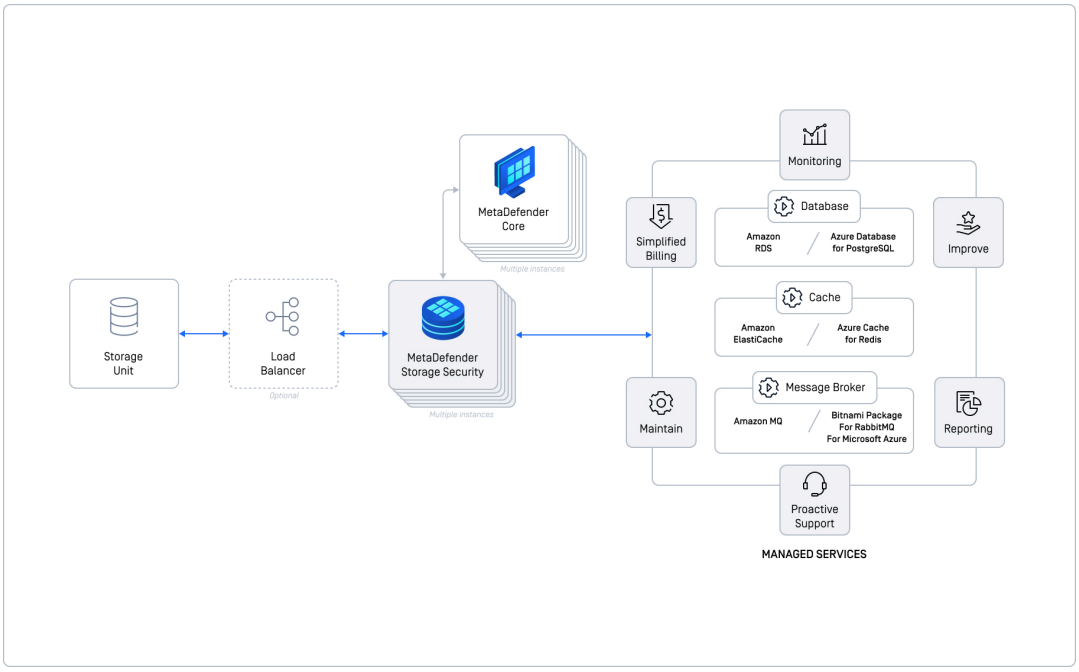
- You don't manage the shared services directly.
- Your primary focus is on configuring and scaling MDSS itself.
- Leverage the cloud provider's scalability for shared services.
- Benefit from the cloud provider's redundancy and reliability.

Important Considerations

- Managed services have associated cloud costs, which need to be factored in.
- Requires understanding cloud service integrations and their complexity

Advanced Deployment with Managed Services

This setup uses **multiple** MetaDefender Storage Security (MDSS) instances and leverages your cloud provider's managed services for the database, cache, and message broker. It's designed for high throughput and simplifies management.



Imagine having a team of experts (your cloud provider) handle the backend infrastructure, allowing you to focus on scaling and managing your MDSS instances. This setup is powerful and efficient for high-volume workloads.

Suitable For

- Organizations processing a lot of files.
- Cloud-Centric Businesses who already leveraging cloud infrastructure.

Why It's Powerful

- Multiple MDSS instances handle large workloads efficiently.
- Your cloud provider handles the underlying infrastructure (database, etc.).
- Ideal if you're already using cloud services.
- Get monitoring, reporting, and proactive support from your cloud provider.

Example (AWS)

- Database: use Amazon RDS
- Cache: use Amazon ElastiCache
- Message Broker: use Amazon MQ

Adaptable to Azure & Google Cloud: You can use similar services from those providers. For more information on integrating with various cloud providers, please follow this page: CSP Deployment Overview.

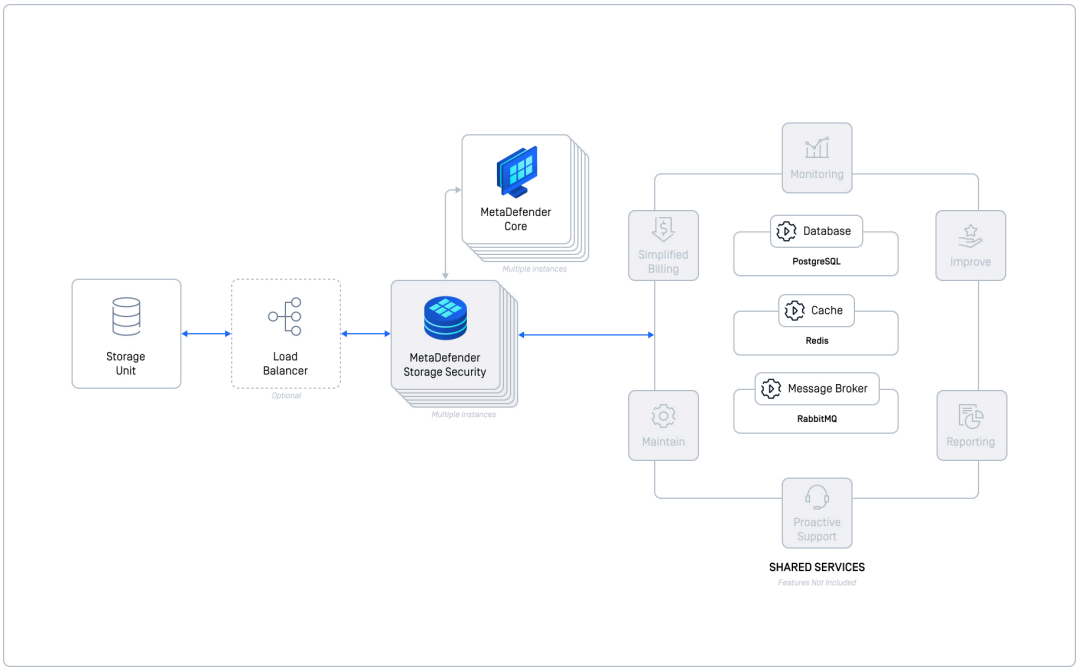
Key Benefits

- You manage the MDSS instances, not the database, cache, etc.
- Cloud providers ensure redundancy and availability.
- Tested to handle 100,000 files/hour (with 2+ MDSS and 4+ MDSS Core instances).

Important Note: Cloud managed services have associated costs.

Advanced Deployment with Self-Hosted Shared Services

This deployment is for high-performance needs, using multiple MDSS and MD Core instances and self-hosted shared services (database, cache, message broker). It's significantly more complex than the basic setups but offers superior performance and reliability.



This setup is like building a powerful engine. You're running multiple MDSS and MD Cores and managing the database, cache, and communication hub yourself. This gives you high performance, but it's also more complex to set up and maintain.

Suitable For

- High-Throughput Environments, organizations processing large volumes of files.
- Organizations with Strong IT Infrastructure as it requires in-house expertise to manage the complexity.

Key Components & Their Roles

- Multiple MDSS Core Instances - allows parallel processing for higher throughput.
- PostgreSQL (Database) - stores data.
- Redis (Cache) - speeds up data access.
- RabbitMQ (Message Broker) - manages communication between components.

Key Points

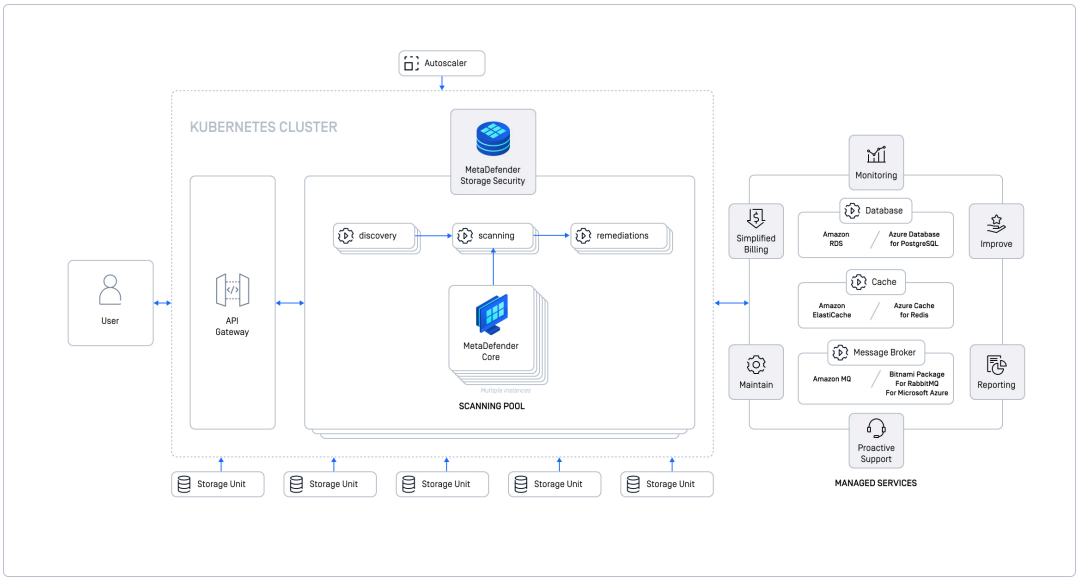
- Requires more planning and technical expertise.
- Can handle up to 100,000 files per hour with sufficient resources.
- You are responsible for installing, configuring, and maintaining all components.
- Allows scaling through multiple MDSS Core instances and shared services.
- Achieved through database replication and careful configuration.

Important Considerations

- Requires significant hardware resources for all components.
- Effective monitoring is essential due to the increased complexity.
- While better than basic setups, careful planning is needed to avoid single points of failure.
- To fail over between two nodes automatically, a reference active-passive failover script is available; it monitors the active node and starts a scan on the passive one when the active node stops responding. Usage is documented in its README.
- Our testing showed 2 MDSS instances with 4 MDSS Core instances yielded substantial performance improvements.
- Once all services are deployed and ready to be used, please follow Installation Methods to connect MDSS to them.

Cloud Deployment with Kubernetes (k8s)

This is our recommended option for maximum scalability, high availability, and handling large volumes of data. It uses Kubernetes to automate deployment, scaling, and management of MDSS in a cloud environment.



Kubernetes is like an automated control system for your cloud deployment. It lets you scale MDSS easily, ensures it stays running, and optimizes resource usage. If you need high performance and availability, this is the way to go.

Suitable For

- Organizations handling massive amounts of data.

- Organizations Prioritizing High Availability, needing minimal downtime.
- Cloud-Savvy Teams familiar with Kubernetes.

Why Use Kubernetes?

- Dynamically adjust resources based on workload.
- Distributes MDSS across multiple nodes to minimize downtime.
- Optimize resource allocation for each component.
- Kubernetes handles load balancing, autoscaling, and more.

Key Components & Cloud Integration

- Kubernetes (k8s) orchestrates the deployment.
- Managed Cloud Services for database, cache, message broker (e.g., AWS EKS, Azure AKS, Google GKE).

Examples

- For AWS - use Amazon RDS, Amazon ElastiCache, Amazon MQ
- Users of Azure or Google Cloud can adapt this model to employ their respective managed services with Azure Kubernetes Service (AKS) or Google Kubernetes Engine (GKE). For detailed guidance on integrating with various cloud providers, please follow:
  - Azure AKS
  - Azure Database for PostgreSQL
  - Azure Cache for Redis
  - Google Kubernetes Engine (GKE)

Key Points

- Kubernetes automates many deployment tasks.
- Easily scale up or down as needed.
- Kubernetes ensures high availability.
- Integrates with cloud monitoring, reporting, etc.

Important Considerations

- Requires understanding Kubernetes concepts.
- You need a Kubernetes cluster set up before deploying MDSS.
- MDSS is deployed using a Helm chart available on GitHub.

Installation

Types of deployment

Whether you require the rapid scalability of the cloud, the control of on-premises infrastructure, or the simplicity of SaaS, MetaDefender Storage Security ensures seamless integration and robust protection across all platforms.

Pre-Deployment Checklist

Before continue with the deployment of MetaDefender Storage Security

1. Verify prerequisites met (System, Network, Database).
2. Gather necessary credentials and OPSWAT license (Service accounts, API keys).
3. Confirm chosen deployment architecture.

Deployment Options

Aiming to facilitate the deployments of the OPSWAT´s products to different environments, there are documented examples of architectures and how to deploy them for the following options.

- Deploy using Cloud Service Providers services
- Deploy to your On-Premise Deployment Overview
- Deploy using containerized technologies in all environments

Deploy to cloud service providers

CSP Deployment Overview

This section explains the various deployment options for MetaDefender Storage Security using services from Cloud Service Providers (CSPs) such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

Before consulting the documentation specific to each CSP, please review the following general documents:

- Architecture Examples in CSPs illustrate different deployment options within each CSP.
- General Cloud Considerations cover access and configuration considerations applicable to cloud environments.
- Using OPSWAT´s image from CSPs Marketplace, detail using the pre-configured image designed to facilitate Proof of Concepts (POCs).

This guide also includes dedicated sections for the main Cloud Service Providers:

- AWS Deployment
- Azure Deployment

Architecture Examples in CSPs

This page outlines the recommended architecture for running MetaDefender Storage Security (MDSS) with any Cloud Service Provider (CSP).

We provide diagrams for AWS, Azure, and GCP, where our product is available in their respective marketplaces. However, our products are cloud-agnostic and can be deployed to any CSP.

Infrastructure Options for MetaDefender Storage Security using Compute Services

The appropriate infrastructure will vary depending on the number of instances where MetaDefender Storage Security will be installed.

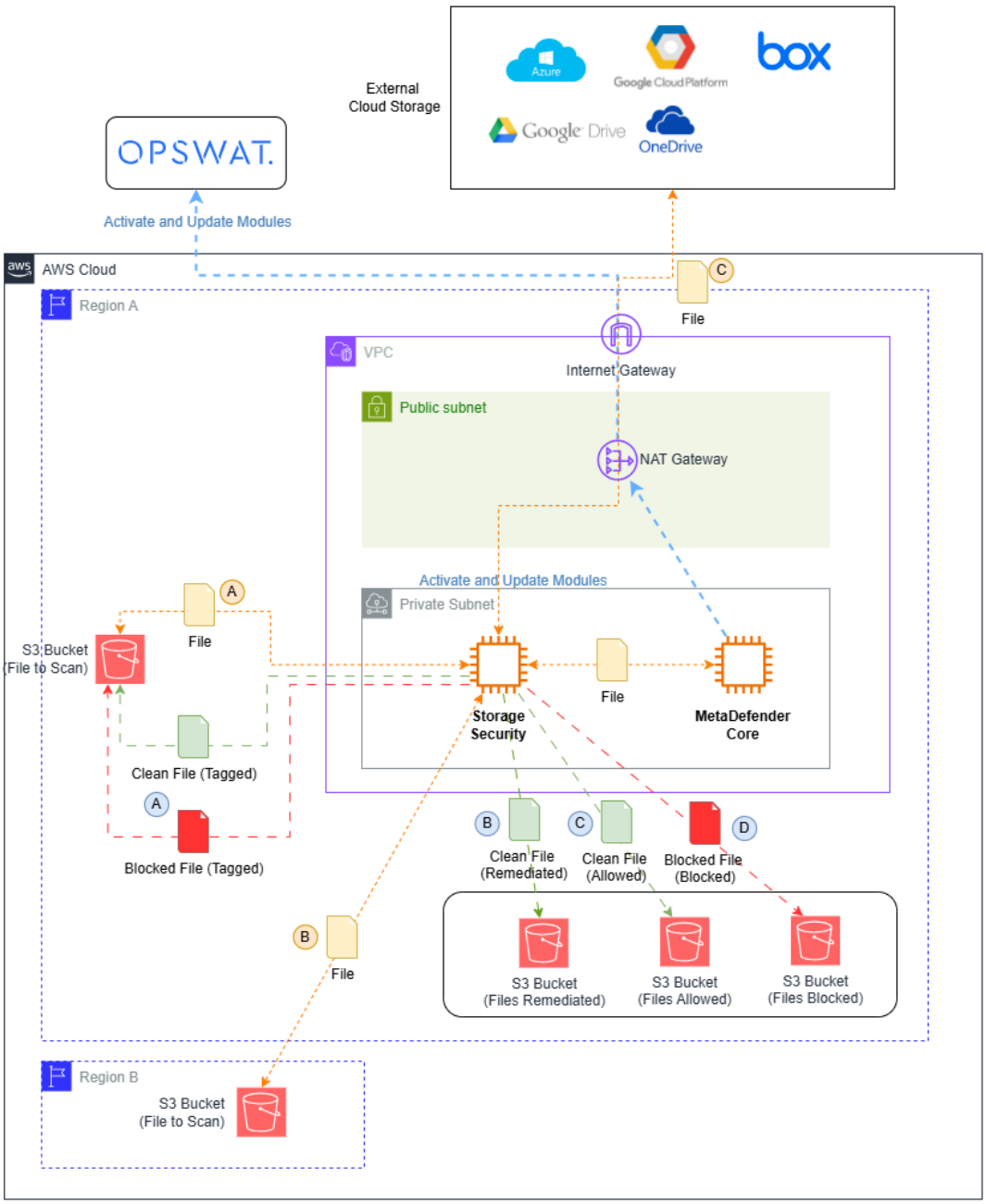
Options:

1. Single Instance
2. Multi-Instance with Autoscaling

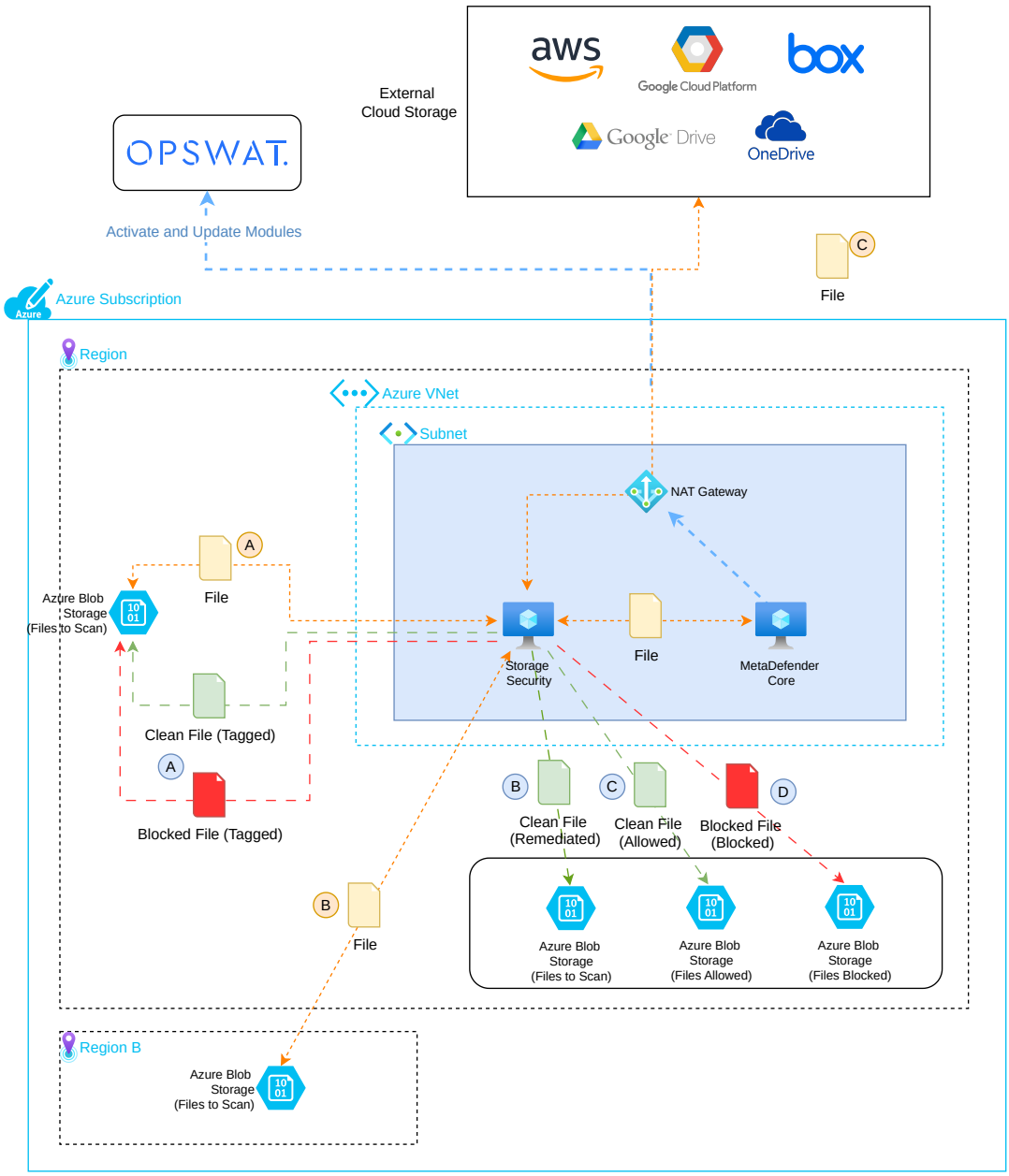
Single Instance

To proceed, select the tab corresponding to the CSP where you intend to deploy MetaDefender Storage Security.

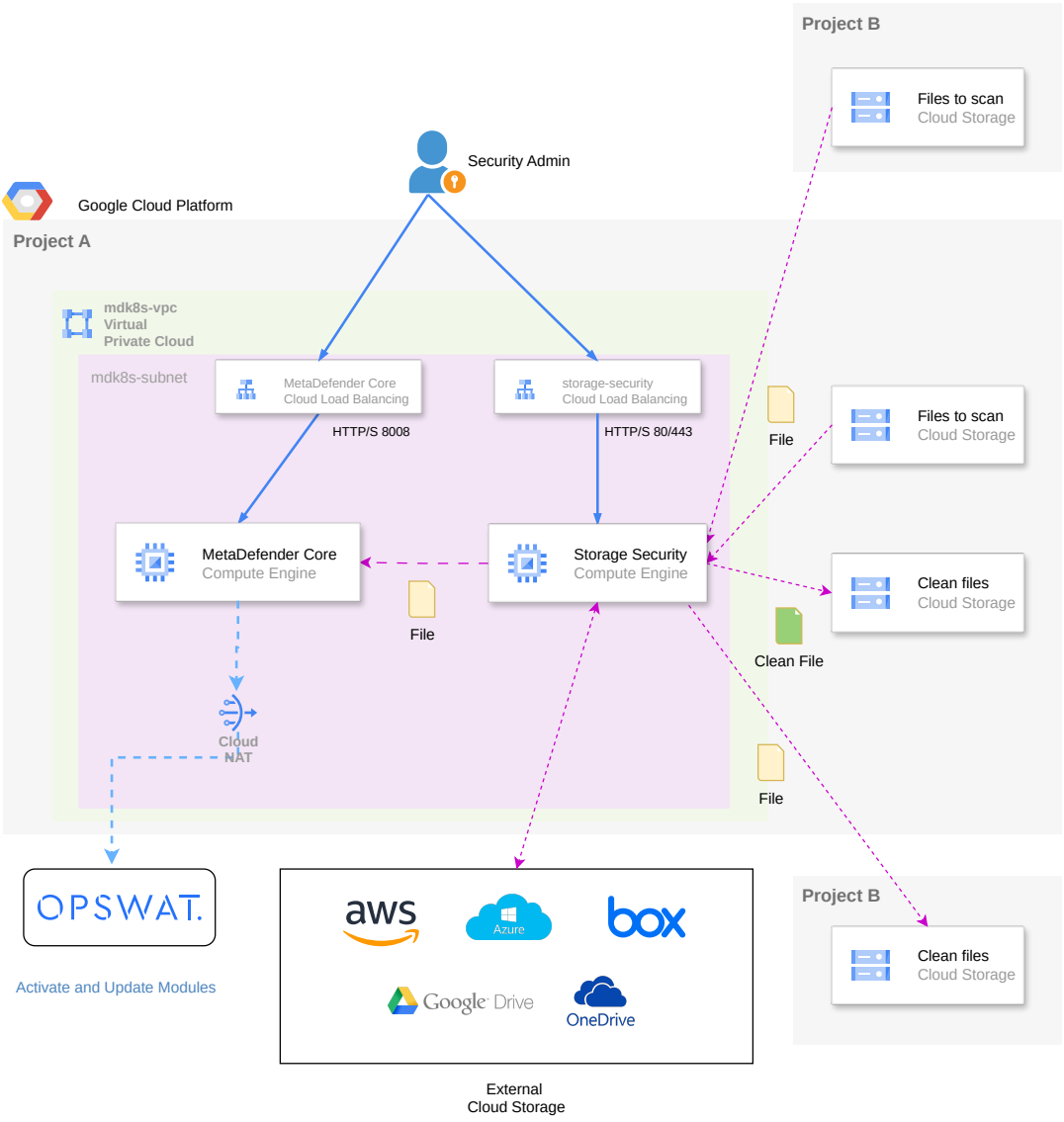
AWS



Azure


External
Cloud Storage
aws
Google Cloud Platform
box
Google Drive
OneDrive
OPSWAT.
Activate and Update Modules
C
File
Azure Subscription
Region
Azure VNet
Subnet
NAT Gateway
A
File
Azure Blob
Storage
(Files to Scan)
Storage
Security
File
MetaDefender
Core
Clean File (Tagged)
A
Blocked File (Tagged)
B
C
D
Clean File
(Remediated)
Clean File
(Allowed)
Blocked File
(Blocked)
B
File
Azure Blob
Storage
(Files to Scan)
Azure Blob
Storage
(Files Allowed)
Azure Blob
Storage
(Files Blocked)
Region B
Azure Blob
Storage
(Files to Scan)

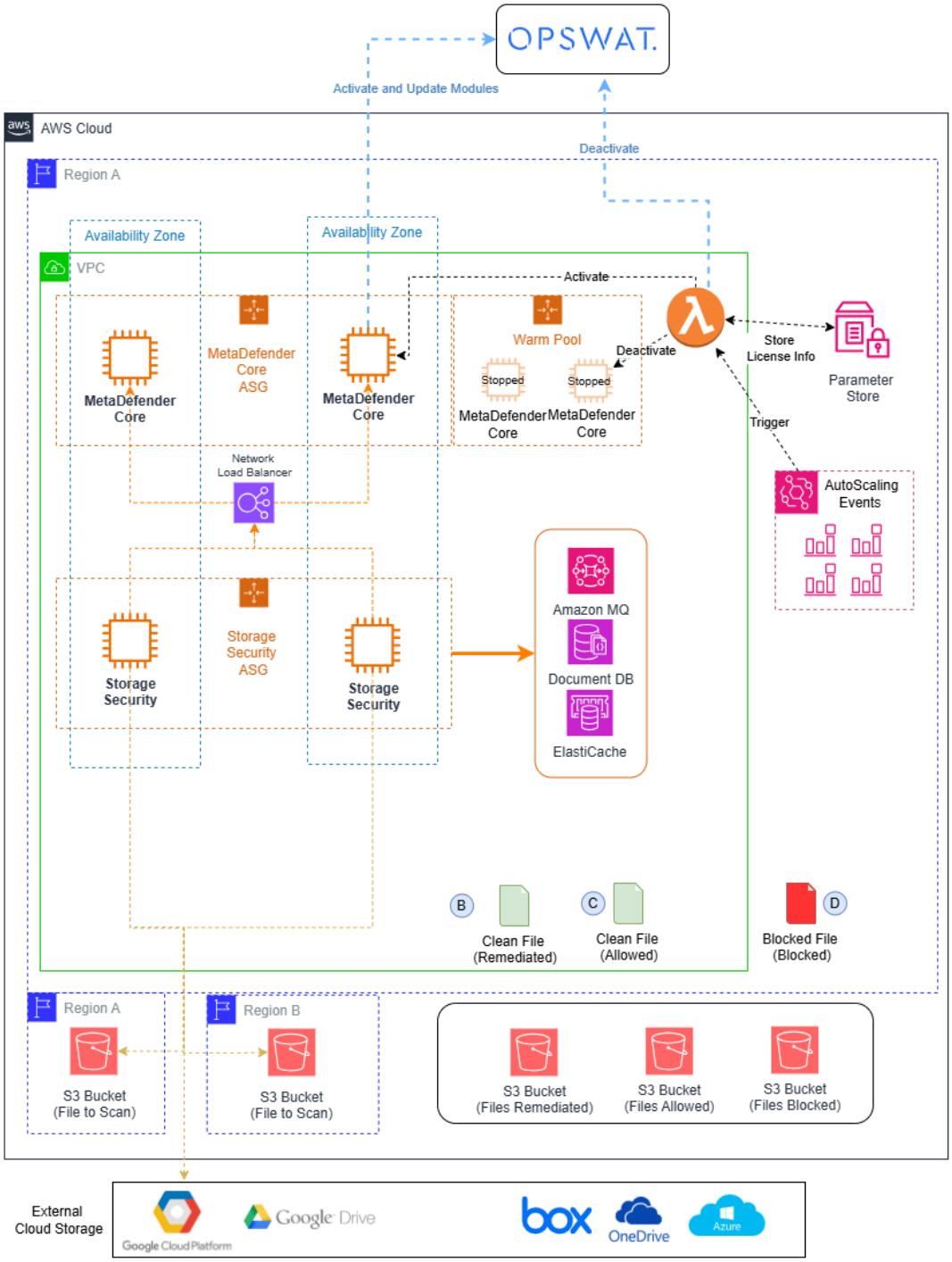
GCP



Multi-Instance with Autoscaling

Select the tab for the CSP where you want to deploy MetaDefender Storage Security.

AWS



Infrastructure Options for MetaDefender Storage Security using Container Services

The possible infrastructure **varies** depending on the Cloud services used for hosting MetaDefender Storage Security and how the **required third-party tools** are provisioned.

Options:

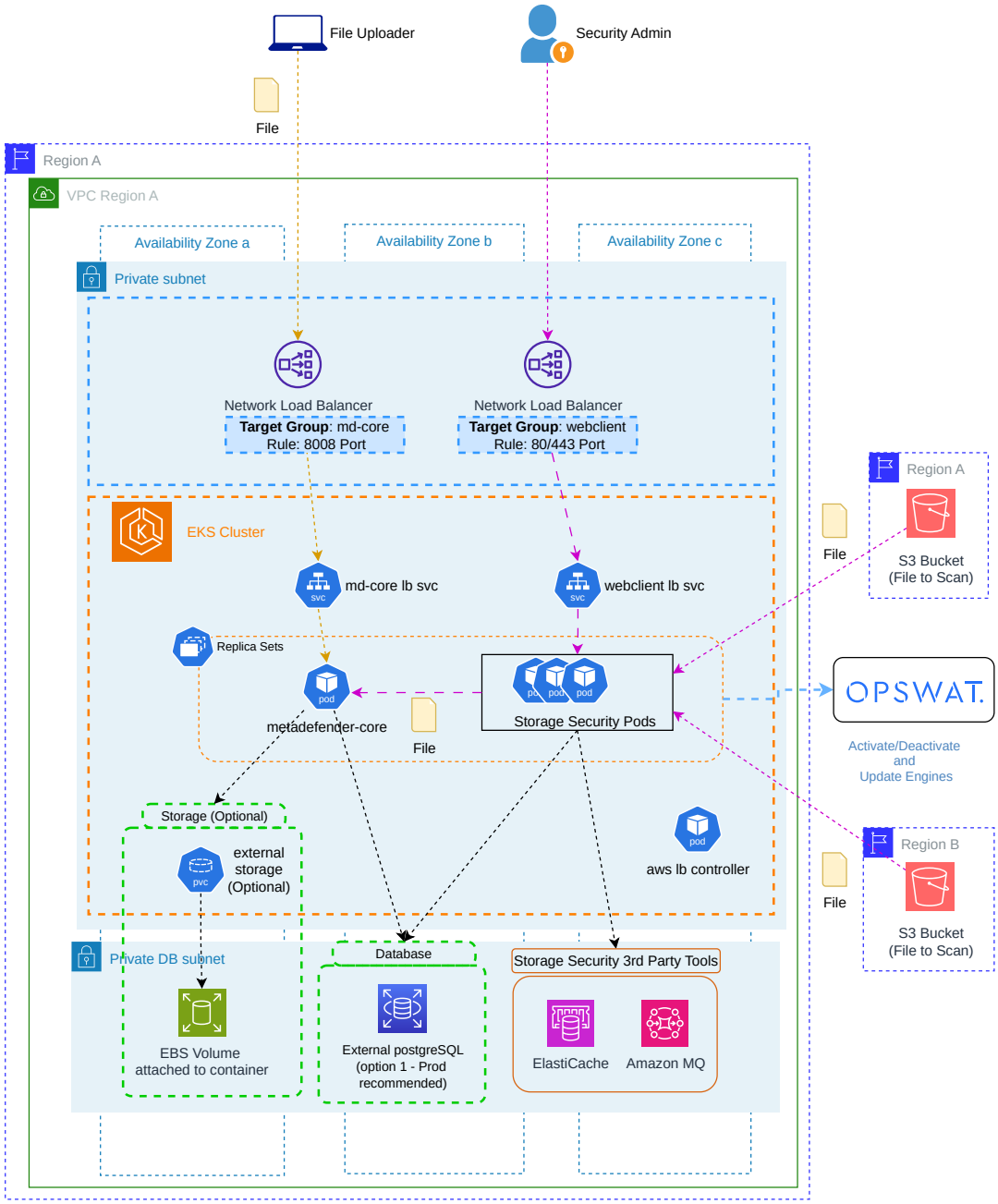
1. Kubernetes Cluster with **Customer-Managed** Worker Nodes, utilizing either a Cloud PostgreSQL Service or **third-party tools as Pods**.
2. Kubernetes Cluster with **CSP-Managed** Worker Nodes, utilizing a Cloud PostgreSQL Service

Customer Managed Worker Nodes

Select the **tab** for the CSP where you want to deploy MetaDefender Storage Security.

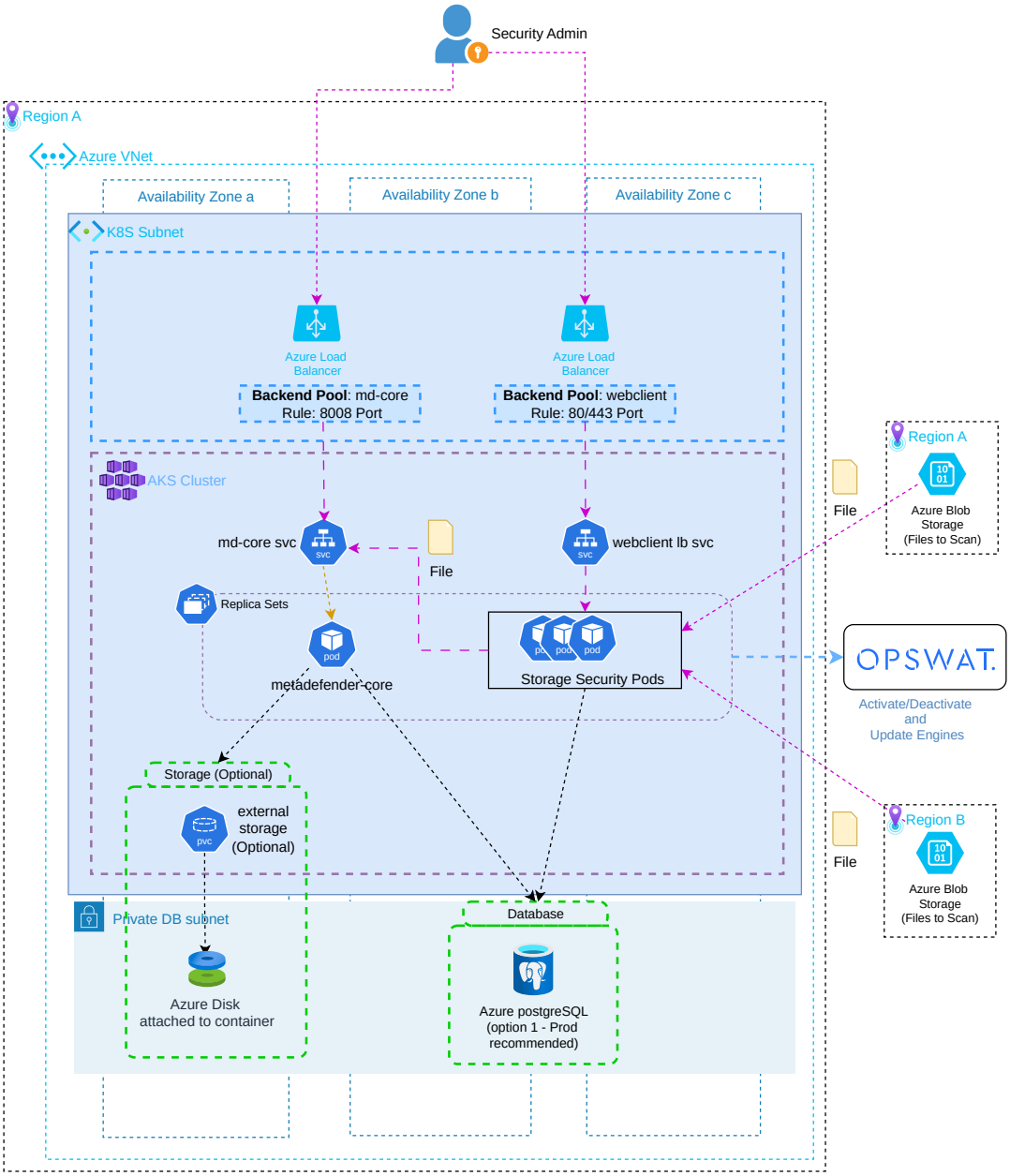
AWS

EC2 Instances + AWS Document DB or 3rd Party tools as Pod



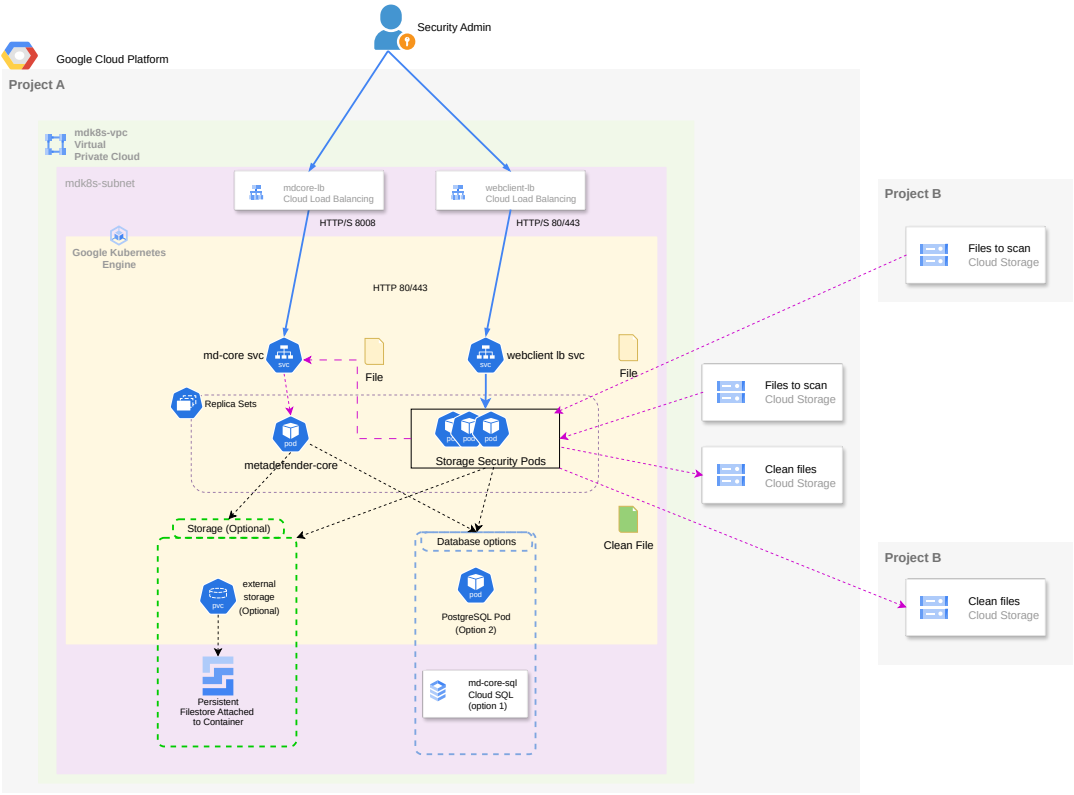
Azure

Azure VM (Worker Nodes) + Azure PostgreSQL DB Service or 3rd Party tools as Pod



GCP

Google Compute Engines Instances (Worker Nodes) + Google Cloud SQL or 3rd Party tools as Pod



CSP-Managed Worker Nodes

Above examples include **AWS Fargate or Azure Container Apps options for computing.**

These are serverless compute engines for containers that allow you to run containerized applications without managing the underlying infrastructure.

These services run containers with certain **limitations**, such as CPU limits per instance or ephemeral storage, which may be required for some Linux packages.

Please check the **system requirements** before choosing this architecture.

> **IMPORTANT
>
> We recommend deploying a database service from your Cloud Service Provider (CSP) for MetaDefender Core and MetaDefender Storage Security.**

General Cloud Considerations

When deploying to any Cloud Service Provider (CSP), consider these general recommendations:

Roles/Permissions

The Principle of Least Privilege (PoLP) is a foundational cybersecurity concept. It entails granting users, applications, and systems only the minimum access necessary to perform their tasks.

This approach minimizes the potential attack surface, mitigates risks associated with insider threats, and helps prevent privilege creep. When deploying OPSWAT products, **OPSWAT recommends** adhering to this principle.

Regarding access to applications or API endpoints, generate **strong passwords** for UI access.

Store the **API keys** required to access the API endpoints of the various products in a secure secrets manager.

VPC/Networking

OPSWAT recommends following networking best practices when deploying products:

- Multi-AZ Deployment
  - Distribute resources across multiple Availability Zones (AZs) to enhance fault tolerance and availability. OPSWAT products are designed to run in different AZs simultaneously.
- Security Groups
  - These act as virtual firewalls to control inbound and outbound traffic. Follow the system requirements for each OPSWAT product to open the necessary ports.
- IP Address Planning
  - Allocate CIDR blocks thoughtfully to avoid overlaps, especially when integrating with on-premises networks.
- Subnet Segmentation
  - Use distinct subnets to separate public-facing resources (MDSS UI) from private resources (MDSS PostgreSql database server).

Using OPSWAT´s image from CSPs Marketplace

Usage of image published in CSPs Marketplaces

OPSWAT has published images of MetaDefender Storage Security in **AWS, Azure and GCP** only **Linux OS**. OPSWAT provides a sample AMI (updated on a regular cadence).

> **Info**

> It is recommended for the customer to **run only POCs** as it is configured with the default options.

For production cases it is recommended to generate an image, after following the installation steps as in On-Premise Deployment. This process is always good to get familiar with the product and configure it to be adapted to the customer use case.

> **Info**
>
> In case the image published to the marketplace with the default configuration works for the customer, the overall and ongoing responsibility of the OS security patch status for that AMI is with the customer.

CSPs Marketplace Images

AWS

- MetaDefender Storage Security on Linux

Azure

- MetaDefender Storage Security on Linux

Google Cloud

- MetaDefender Storage Security – Marketplace – OPSWAT Public – Google Cloud console

AWS Deployment

This Deployment Guide provides step-by-step instructions for deploying MetaDefender Storage Security on Amazon Web Services (AWS) infrastructure.

This guide is intended for IT infrastructure architects, administrators, and DevOps professionals seeking to prevent malicious traffic from entering their AWS Cloud environment. It outlines how MetaDefender Core enhances threat prevention by inspecting uploaded productivity files and identifying known vulnerabilities in running services, applications, and unpatched operating systems deployed within AWS.

MetaDefender is provided in the following formats:

- An Amazon Machine Image (AMI) available through the AWS Marketplace (MetaDefender products) (Recommended for PoC)
- A packaged installer available for download from the My OPSWAT portal.

- Helm charts for MetaDefender products on GitHub with corresponding container images accessible from the OPSWAT Docker Hub Repository

Currently, there are no restrictions on the AWS regions where MetaDefender can be deployed, with the exception that MetaDefender is not available in AWS GovCloud at this time.

For installing our solution, please review the guidelines listed below to deploy the different infrastructure needed to install the product:

- Deploy using EC2 Instances
- Deployment on EKS

Database Configuration

MetaDefender Storage Security currently supports AWS RDS, an AWS-managed service compatible with PostgreSQL.

EC2 Key Management

An EC2 key pair is used exclusively for SSH or RDP access to an EC2 instance, a scenario typically reserved for debugging.

You can either create a new key pair or use an existing one.

Deploy using EC2 Instances

The deployment options vary depending on what are the number of instances where MetaDefender Storage Security will be installed.

Options:

1. **Single Instance** deployment of MetaDefender Core and MetaDefender Storage Security
2. **Multi-Instance** with Autoscaling for MetaDefender Core and single instance of MetaDefender Storage Security.

OPSWAT provides a terraform project to deploy MetaDefender Storage Security and MetaDefender Core in different instances using AWS services.

Prerequisites

AWS knowledge: This guide assumes familiarity with AWS Services

AWS account: needs permission to create AWS Services depending on the examples listed in Architecture Examples in CSPs

Tools installation: Terraform, AWS CLI

Single Instance Deployment

Deploy using Terraform

OPSWAT provides a terraform project to create the infrastructure needed to deploy the Architecture Examples in CSPs

- Locally clone the metadefender-csp repository and go to AWS/single-ec2

```
git clone git@github.com:OPSWAT/metadefender-csp.git
cd AWS/single-ec2
```

- Modify terraform.tfvars with the desired options
  - LICENSE_KEY required if wanted to have the MetaDefender Core instance activated automatically
  - APIKEY_GENERATION=true to have the apikey generated by terraform
  - There is one general section and one section for each product that is supported using this Terraform project

```
# General variables

MD_ENV_NAME             = "metadefender" # Prefix to add to all the
resources
MD_REGION               = "eu-central-1" # Region for all the resources
EC2_KEY_NAME            = "" # Key pair to attach to EC2 instances
(Optional)
#ACCESS_KEY_ID          = "<ACCESS_KEY_ID>"
#SECRET_ACCESS_KEY      = "<SECRET_ACCESS_KEY>"  # To give access to
terraform (Optional, can use other ways to authenticate)
PUBLIC_ENVIRONMENT      = true
APIKEY_GENERATION       = true

# MetaDefender Core variables

DEPLOY_CORE             = true
MD_VPC_CIDR             = "192.168.0.0/16"  # VPC CIDR where to create
the MetaDefender products
CORE_PRODUCT_ID           = "ani6v4vb5z4t87cymrfg3m451" #MetaDefender
Core ID in AWS Marketplace || For Windows it is
"9s8powksm1cj7fuafdnv0sfj9"
CORE_INSTANCE_TYPE      = "c5.2xlarge"   # Instance type for
MetaDefender Core
LICENSE_KEY_CORE        = ""
```

- To deploy MetaDefender Storage Security, modify the section for Storage Security

```
# MetaDefender Storage Security variables

DEPLOY_MDSS             = true               # true to deploy
MetaDefender Storage Security together with Core
MDSS_PRODUCT_ID         = "3mup1qubt6hwmup405eljau0k" # MetaDefender
Storage Security LINUX ID in AWS Marketplace
MDSS_INSTANCE_TYPE      = "c5.2xlarge"    # Instance type for
MetaDefender Storage Security

DEPLOY_MDSS_ELASTICACHE = true  # true to deploy  MetaDefender Storage
Security with a managed instance of Elasticache Redis
MDSS_ELASTICACHE_NODE_TYPE = "cache.m5.large"
```

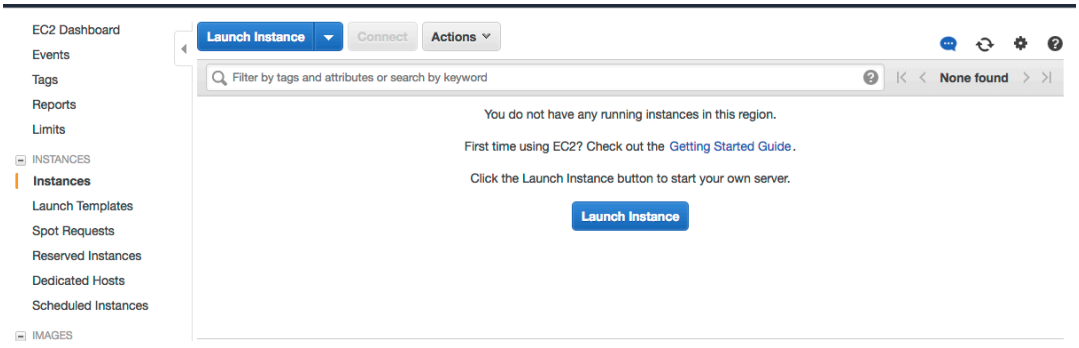
- Run terraform init and apply. Check the resource to be created, after that enter “y”

```
terraform init
terraform apply
```

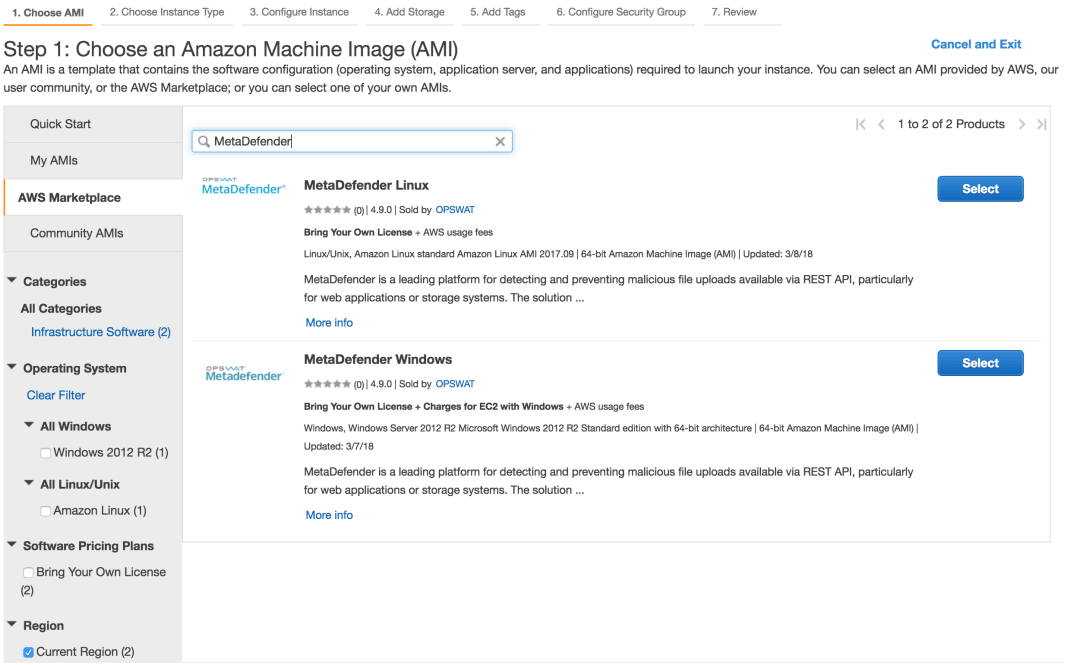
Deploy using AWS Console

Select MetaDefender Storage Security AMI

- Go to the EC2 Management Console in AWS and select Launch Instance

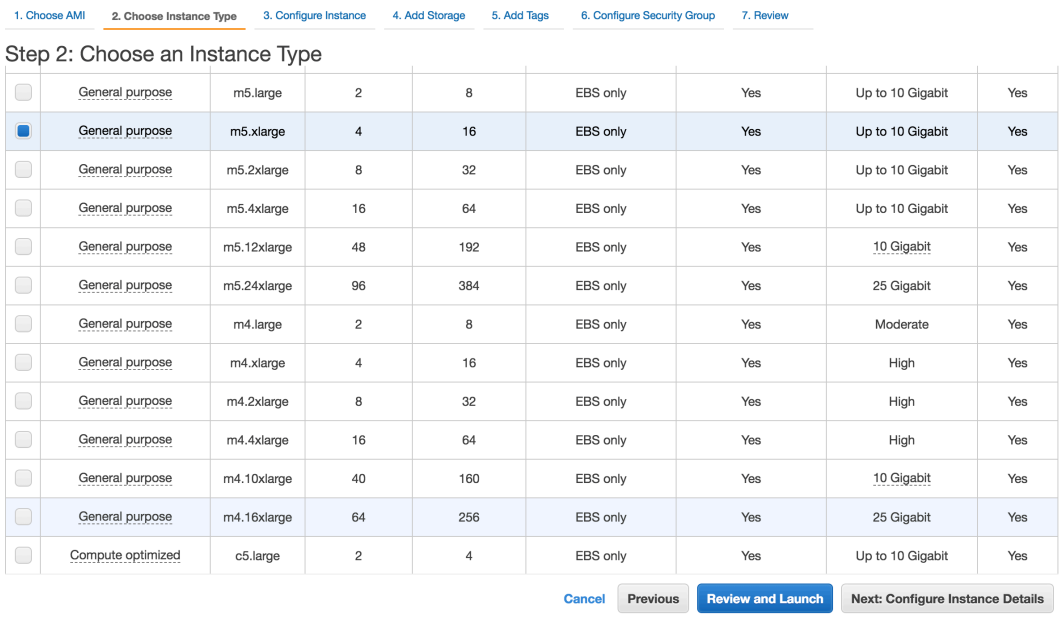


- Select the desired MetaDefender Storage Security, based on the OS support:



- Choose the Instance type

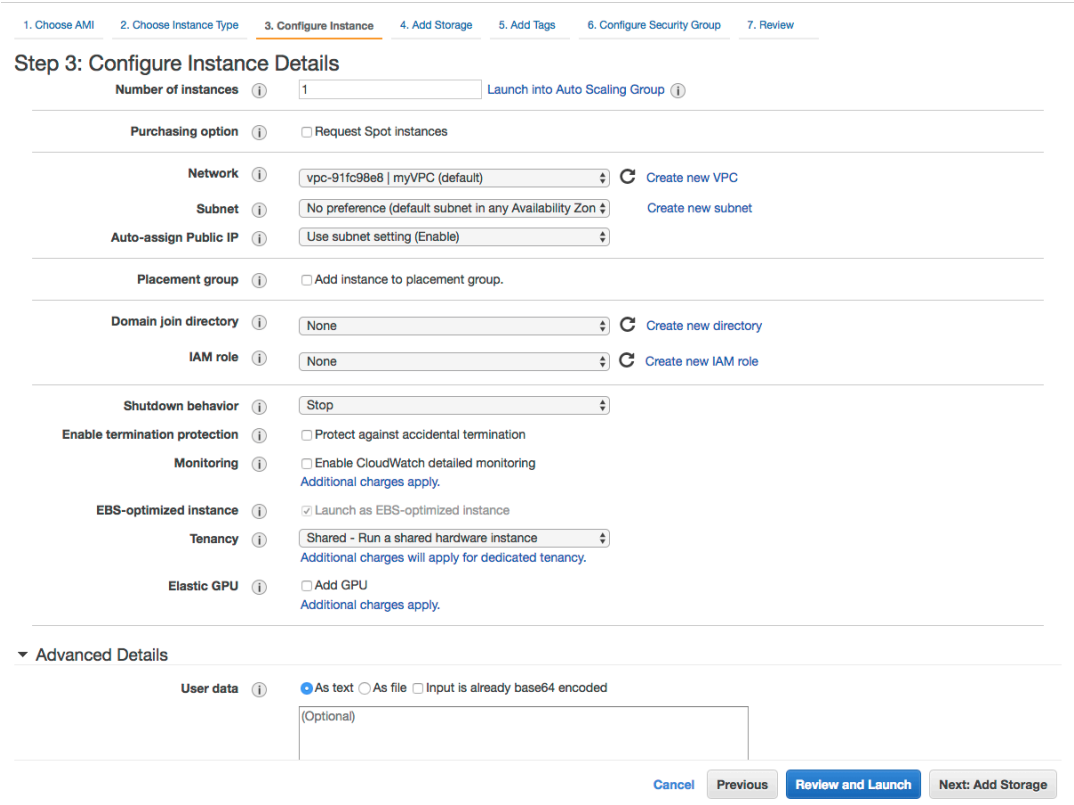
Please review MetaDefender Storage Security's general system requirements before choosing the desired AMI and instance type.



	General purpose	m5.large	2	8	EBS only	Yes	Up to 10 Gigabit	Yes
	General purpose	m5.xlarge	4	16	EBS only	Yes	Up to 10 Gigabit	Yes
	General purpose	m5.2xlarge	8	32	EBS only	Yes	Up to 10 Gigabit	Yes
	General purpose	m5.4xlarge	16	64	EBS only	Yes	Up to 10 Gigabit	Yes
	General purpose	m5.12xlarge	48	192	EBS only	Yes	10 Gigabit	Yes
	General purpose	m5.24xlarge	96	384	EBS only	Yes	25 Gigabit	Yes
	General purpose	m4.large	2	8	EBS only	Yes	Moderate	Yes
	General purpose	m4.xlarge	4	16	EBS only	Yes	High	Yes
	General purpose	m4.2xlarge	8	32	EBS only	Yes	High	Yes
	General purpose	m4.4xlarge	16	64	EBS only	Yes	High	Yes
	General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit	Yes
	General purpose	m4.16xlarge	64	256	EBS only	Yes	25 Gigabit	Yes
	Compute optimized	c5.large	2	4	EBS only	Yes	Up to 10 Gigabit	Yes

Networking for EC2

- Select the desired VPC and subnet you would like to have MetaDefender Storage Security deployed.



Depending on the deployment model, the recommendation would be to deploy MetaDefender Storage Security in a private subnet.

Storage configuration

Storage step can be skipped

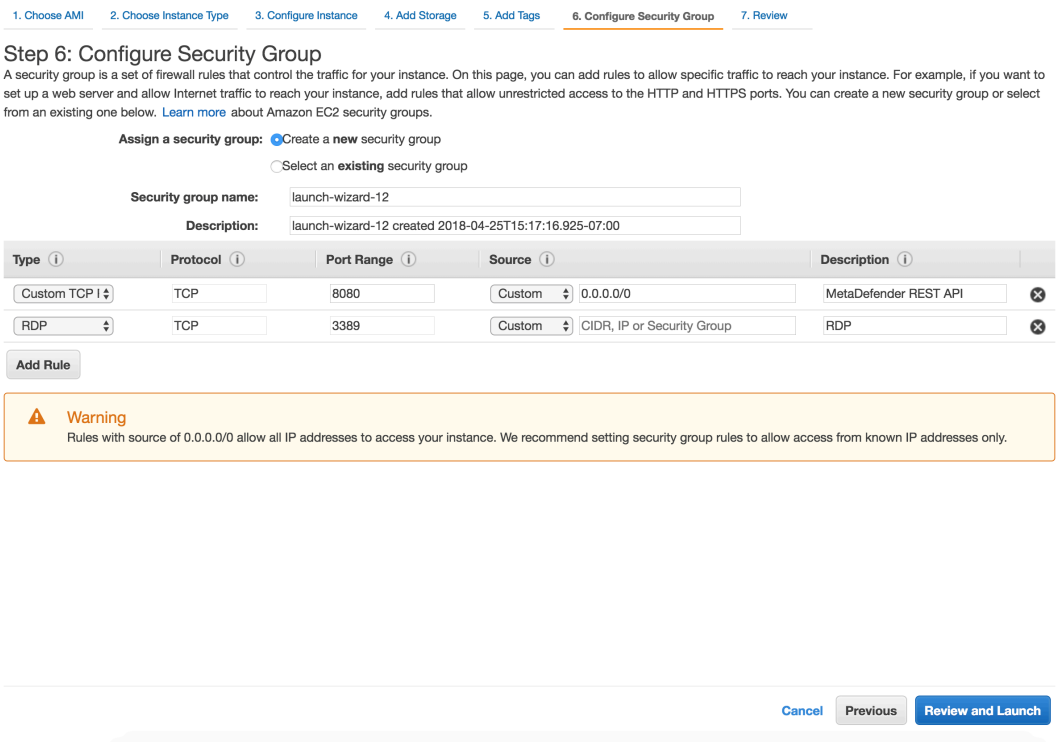
In general there's no need for additional storage by MetaDefender Storage Security. Follow the system requirements for Storage.

Security Groups configuration

There are 2 rules that are needed to be opened during the installation phase, after which only one rule should remain:

- Custom TCP: 80/443
  - MetaDefender Storage Security exposes the web UI by default to port 80 and 443 with SSL. However this port can be changed during the installation phase or updated during it's lifetime.
- RDP / SSH
  - In order to install MetaDefender Storage Security on Windows add access for RDP and for Linux add support for SSH

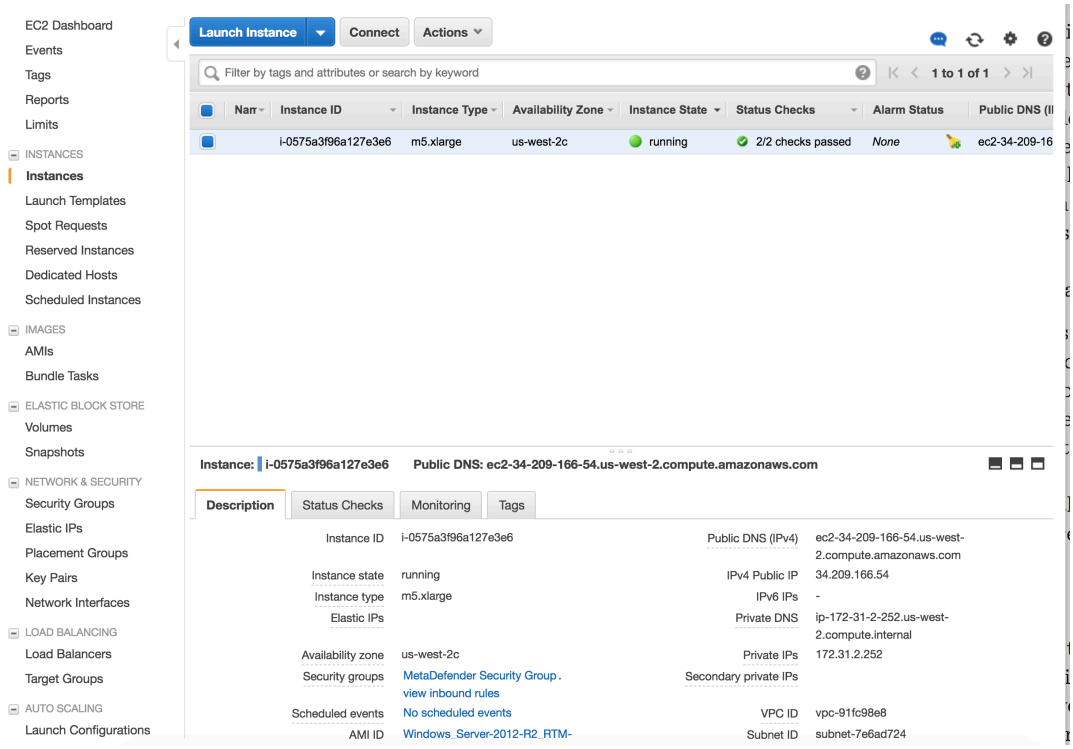
- Highly recommended would be not to allow traffic to RDP or SSH from anywhere, but to limit to your IP address



> **Warning**
>
> *After the installation remove RDP or SSH from this Security Group!*

Launch the instance

After you review the settings, hit Launch. In a few minutes the instance should become available



Multi-Instance Deployment

Deploy using Terraform

OPSWAT provides a terraform project to create the infrastructure needed to deploy the Architecture Examples in CSPs

- Locally clone the metadefender-csp repository and go to AWS/autoscaling-group

```
git clone git@github.com:OPSWAT/metadefender-csp.git
cd AWS/autoscaling-group
```

- Modify terraform.tfvars with the desired options

```
# General variables

MD_ENV_NAME             = "metadefender"
MD_REGION               = "eu-central-1"
#ACCESS_KEY_ID          = "<ACCESS_KEY_ID>"
#SECRET_ACCESS_KEY      = "<SECRET_ACCESS_KEY>"
MD_VPC_CIDR             = "192.168.0.0/16"
PUBLIC_ENVIRONMENT      = true
WARM_POOL_ENABLED       = true

# MetaDefender Core variables

DEPLOY_CORE             = "false"
CORE_PRODUCT_ID            = "ani6v4vb5z4t87cymrfg3m451" # For Windows
it is "9s8powksm1cj7fuafdnv0sfj9"
CORE_INSTANCE_TYPE      = "c5.2xlarge"
LICENSE_KEY_CORE        = ""
CORE_PWD                = "admin"
CORE_PORT               = 8008

# MetaDefender Storage Security variables

DEPLOY_MDSS             = true               # true to deploy
MetaDefender Storage Security together with Core
MDSS_PRODUCT_ID         = "3mup1qubt6hwmup405eljau0k" # MetaDefender
Storage Security LINUX ID in AWS Marketplace
MDSS_INSTANCE_TYPE      = "c5.2xlarge"   # Instance type for
MetaDefender Storage Security

DEPLOY_MDSS_DOCUMENTDB  = true  # true to deploy  MetaDefender Storage
Security with a managed instance of Amazon DocumentDB
MDSS_DOCUMENTDB_INSTANCE_CLASS = "db.r5.large"
DEPLOY_MDSS_ELASTICACHE = true  # true to deploy  MetaDefender Storage
Security with a managed instance of Elasticache Redis
MDSS_ELASTICACHE_NODE_TYPE = "cache.m5.large"
```

- Run terraform apply and check the resource to be create, after that enter “y”

```
terraform apply
```

Deployment on EKS

This guide explains how to use the provisioning tools provided by OPSWAT to create an AWS EKS and generate all the Kubernetes components needed to run MetaDefender Core and MetaDefender Storage Security.

> **Info**

> In case of having the K8S cluster already created, please check the Containerized Deployment Overview
>
> This page is purely for provisioning the EKS cluster and install the products at the same time with tools that OPSWAT has prepared.

Deploy using Terraform

- Locally clone the metadefender-k8s repository and go to terraform folder

```
git clone git@github.com:OPSWAT/metadefender-k8s.git
cd terraform/aws/
```

- Modify variables/variables.tfvars file
  - PERSISTENT_DEPLOYMENT=true -> To create K8S cluster with EC2 Worker Nodes
  - DEPLOY_FARGATE_NODES=false -> To true if wanted to create a Fargate profile
  - DEPLOY_RDS_POSTGRES_DB=true -> To create RDS instance for PostgreSQL server
  - POSTGRES_USERNAME -> if DEPLOY_RDS_POSTGRES_DB=true, set up admin database username
  - POSTGRES_PASSWORD -> if DEPLOY_RDS_POSTGRES_DB=true, set up admin database password
  - MQ_USERNAME -> if DEPLOY_RABBITMQ=true
  - MQ_PASSWORD -> if DEPLOY_RABBITMQ=true

```
MD_CLUSTER_NAME         = "md-k8s"
MD_CLUSTER_REGION       = "eu-central-1"
ACCESS_KEY_ID           = "<ACCESS_KEY_ID>"
SECRET_ACCESS_KEY       = "<SECRET_ACCESS_KEY>"
PERSISTENT_DEPLOYMENT   = true
DEPLOY_FARGATE_NODES    = false
DEPLOY_RDS_POSTGRES_DB  = true
POSTGRES_USERNAME       = "<POSTGRES_USERNAME>"
POSTGRES_PASSWORD       = "<POSTGRES_PASSWORD>"

DEPLOY_REDIS            = true
DEPLOY_RABBITMQ         = true
MQ_PASSWORD             = "<MQ_USERNAME>"
MQ_USERNAME             = "<MQ_PASSWORD>"
```

- Run terraform init and apply. Check the resource to be created, after that enter “y”

```
terraform init
terraform apply
```

MetaDefenderK8S script details

- Script path: %provisioning_script_path%
- **Programming Language:** Bash
- Installation Pre-requisites for provisioning:
  - Terraform
  - Helm
  - AWS CLI
  - Kubectl
  - eksctl (When using the lb controller script)
- AWS Credentials
  - Set in your local environment credentials the IAM user credentials under ***AWS_ACCESS_KEY_ID*** & ***AWS_SECRET_ACCESS_KEY***
- MetaDefender Core License Key (Required with --mdcore parameter)
  - Set it in your local environment credentials under ***MDCORE_LICENSE_KEY***

> **Warning**
>
> As security measure do not use the root user for any deployment or operation. Follow the principle of least privilege for all access granted.

How to run the script

```
./metadefenderk8s.sh provision -l AWS --mdcore --mdss
```

Script Parameters

Parameter	Flags	Options	Default	Description	Required/Optional
Action		- provision - install		Action to indicate the script if we want to provision (Create resources + install Core) or install (Install Core)	**Required**
Location	-l or --location	- AWS - Azure - GCP		Where is going to be the K8S cluster	**Required**
MetaDefender Flag Installation	Combination of - --mdcore - --mdss - --icap		-	Install MetaDefender Core in the cluster provisioned	**Required**
Image Version	–image	- latest -	latest	MetaDefender Core image version to install	Optional
Region	–region	AWS Regions	eu-central-1	AWS region where all the resources will be provisioned	Optional
Cluster Name	–name	Not uppercase allowed	md-k8s	Name of the cluster that will be used for naming all the resources	Optional
Number of Replicas	–replicas	[0-9]*	1	Number of replicas for MetaDefender Core service	Optional
Namespace	–namespace	[A-Za-z] {1,10}		Namespace where MetaDefender products will be installed in the K8S Cluster	Optional Max Characters: 10

Limitations using parameters

- When using Fargate
  - the pods should be installed in the default namespace.
  - if using AWS Load Balancer controllers need a worker node to be installed in.

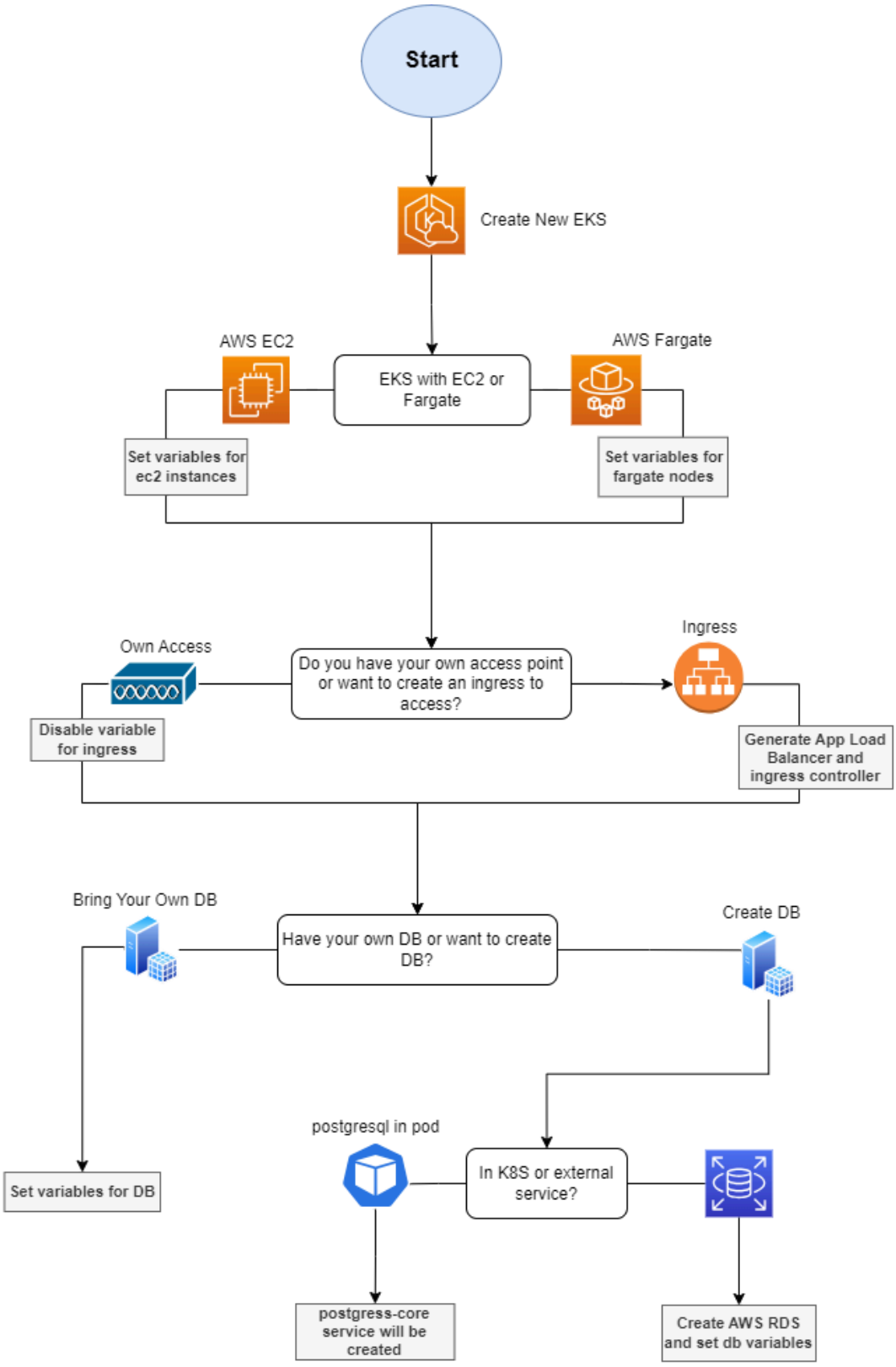
> **Info**
>
> The script will deploy a single Worker Node for the cluster with enough space for 1 replica of MD Core. AWS EC2 Instance size is **c5.2xlarge (8 vCPU & 16 GiB Memory).** Each pod would request 7 vCPU and 8 GiB Memory. To change the request to adapt each pod to the specific case go to values.yml To change the node group go to the terraform file terraform/aws/eks_cluster_module

MetaDefender Core Flowchart Provisioning in AWS

The following flowchart represents how the provisioning script will configure the environment based on the options selected for provisioning AWS EKS.

Summary options to be selected

1. EKS with **EC2** or **Fargate**
2. Access to the K8S cluster. Generate **Ingress** or provide **own** access.
   1. **An Ingress** will be create per each product flag added as parameter to the script
   2. **Own Access**, you decide how to access to the cluster so it won't generate any load balancer for accessing
3. Have your **own database** or create **new database**
   1. **Own database,** will be asked if you want the script to set up the credentials and database host url for you or the script will just indicate the secrets to edit, later on by you, for connecting the MetaDefender Core with your database.
   2. Create new **DB in K8S** or **external DB** that for AWS we will provision a ***RDS instance***


Start
Create New EKS
AWS EC2
AWS Fargate
EKS with EC2 or Fargate
Set variables for ec2 instances
Set variables for fargate nodes
Own Access
Do you have your own access point or want to create an ingress to access?
Ingress
Disable variable for ingress
Generate App Load Balancer and ingress controller
Bring Your Own DB
Have your own DB or want to create DB?
Create DB
postgresql in pod
In K8S or external service?
Set variables for DB
postgress-core service will be created
Create AWS RDS and set db variables

OPSWAT Licensing on AWS

To activate the instances where the MetaDefender products are running, it is needed to activate these products providing the LICENSE_KEY.

> **Info**
>
> LICENSE_KEY needs to be requested to Sales before starting to install the product.

MetaDefender Storage Security ask for the LICENSE_KEY when doing the on-boarding process. It is needed to assure the persistence of the database, not to have to reactivate the product on every restart.

> **Info**
>
> MetaDefender Storage Security works together with MetaDefender Core, to see how works the licensing for MetaDefender Core check this page

Azure Deployment

Introduction

This Deployment Guide provides step-by-step instructions for deploying MetaDefender Storage Security on Azure Cloud Services infrastructure.

This guide is for IT infrastructure architects, administrators and DevOps professionals who are seeking to prevent potential malicious traffic being allowed in their Azure Cloud deployment. Threat Prevention is ensured for both productivity files that might be uploaded and for known vulnerabilities that can be identified for all running services/applications deployed in Azure. The vulnerability scanning is checking known vulnerabilities for unpatched OS and running applications.

MetaDefender is provided as:

- AMI through the Azure Marketplace [MetaDefender products]
- A packaged installer available for download from the My OPSWAT portal.
- Helm charts for MetaDefender products on GitHub with corresponding container images accessible from the OPSWAT Docker Hub Repository

There are currently no restrictions on Azure regions in which you can deploy MetaDefender. The only restriction is that MetaDefender is not available Azure for Government at this time.

For deploying our solution in Azure, please review the guideline listed below:

- Deploy using Azure VMs
- Deployment on AKS

Private Key Management

The Private key pair is only used to SSH or RDP to the Azure VM. This is a very uncommon scenario, used primarily for debugging. You can create a new key pair, or use an existing pair.

Deploy using Azure VMs

The deployment options vary depending on what are the number of instances where MetaDefender Storage Security will be installed.

Options:

1. **Single Instance** deployment of MetaDefender Core and MetaDefender Storage Security
2. **Multi-Instance** with Autoscaling for MetaDefender Core and single instance of MetaDefender Storage Security.

OPSWAT provides a terraform project to deploy MetaDefender Storage Security and MetaDefender Core in different instances using Azure services.

> **Warning**
>
> MetaDefender Storage Security running in VMs only supports single instance as the 3rd party tools cannot be all exported with Azure services. However, in case of using VMs separately for these 3rd party tools then MDSS can be replicated multiple times.

Prerequisites and Requirements

Azure knowledge: This guide assumes familiarity with Azure Services

Azure account: needs permission to create Azure Services listed in Architecture Examples in CSPs for Azure.

Tools installation: Terraform

Single Instance Deployment

For a single deployment of MetaDefender Core in Azure you can use this example to test this deployment type, besides the MetaDefender VM instance, additional resources are being generated and set up.

Deploy using Terraform

OPSWAT provides a terraform project to create the infrastructure needed to deploy the Architecture Examples in CSPs

- Locally clone the metadefender-csp repository and go to Azure/single-vm-deployment

```
git clone git@github.com:OPSWAT/metadefender-csp.git
cd Azure/single-vm-deployment
```

Modify terraform.tfvars with the desired options

- LICENSE__KEY_CORE_ required if wanted to have the MetaDefender Core instance activated automatically
- APIKEY_GENERATION=true to have the apikey generated by terraform
- There is one general section and one section for each product that is supported using this Terraform project

```
# General variables

RG_NAME                 = "metadefender" # Prefix to add to all the
resources
MD_REGION               = "eastus" # Region for all the resources
MD_VNET_CIDR            = "192.168.0.0/16"  # VPC CIDR where to create
the MetaDefender products
PUBLIC_ENVIRONMENT      = true
APIKEY_GENERATION       = true
IMPORT_RG               = false
VM_PWD                  = "<SET_UP_VM_PWD>"

# MetaDefender Core variables

DEPLOY_CORE             = true
CORE_INSTANCE_TYPE      = "Standard_D8s_v5"   # Instance type for
MetaDefender Core
LICENSE_KEY_CORE        = ""
OFFER_PRODUCT_CORE      = "opswat-mdcore-linux"  # Windows opswat-
mdcore-windows
SKU_CORE                = "opswat-mdcore-linux"  # Windows opswat-
mdcore-windows


# MetaDefender Storage Security variables

DEPLOY_MDSS             = true                          # true to deploy
MetaDefender Storage Security together with Core
MDSS_INSTANCE_TYPE      = "Standard_D8s_v5"   # Instance type for
MetaDefender Storage Security
OFFER_PRODUCT_MDSS      = "opswat-mdss-ubuntu"
SKU_MDSS                = "opswatmdssubuntu"
DEPLOY_MDSS_COSMOSDB     = false
```

- Run terraform init and apply. Check the resource to be created, after that enter “y”

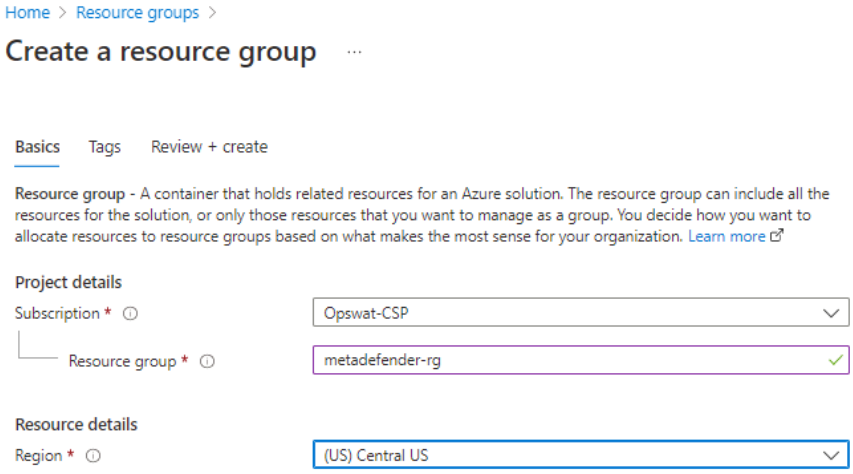
```
terraform init
terraform apply
```

Deploy using Azure portal

For having MetaDefender Core in a Single Azure VM as showed in Single Instance VM Architecture for Azure, it is needed to follow the next steps:

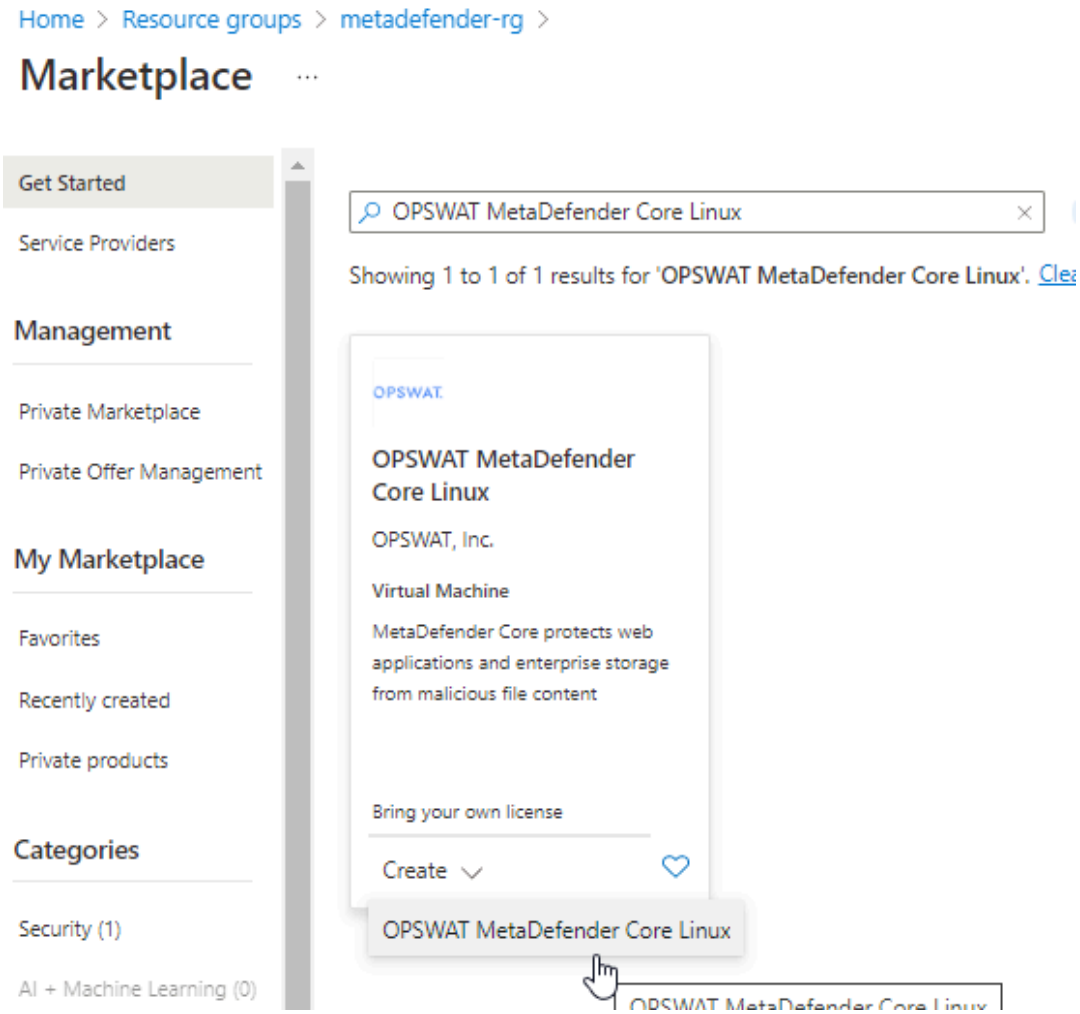
1. Create Resource Group

- Go to you Azure account and create a new resource group (we will refer to it as “metadefender-rg”)

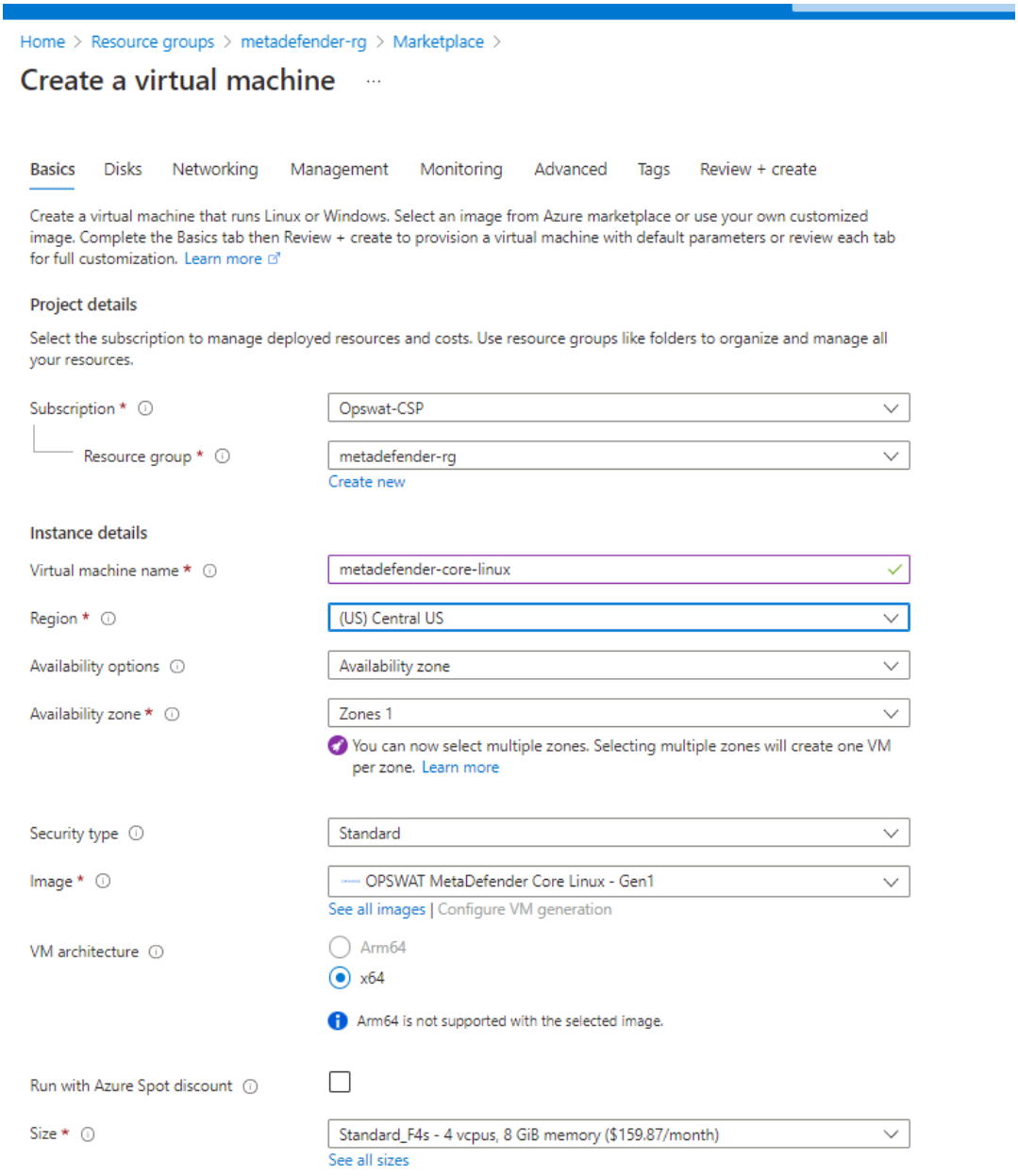


2. Create Azure VM

- Inside the resource group click on create resource and search for ***"OPSWAT MetaDefender Core Linux"***



- Input the vm name and select the size based on General System Requirements
- Under Administrator Account section select the desired way to access to the VM. For this guideline we will generate a public key to access to it



3. Storage Configuration

Storage step can be skipped.

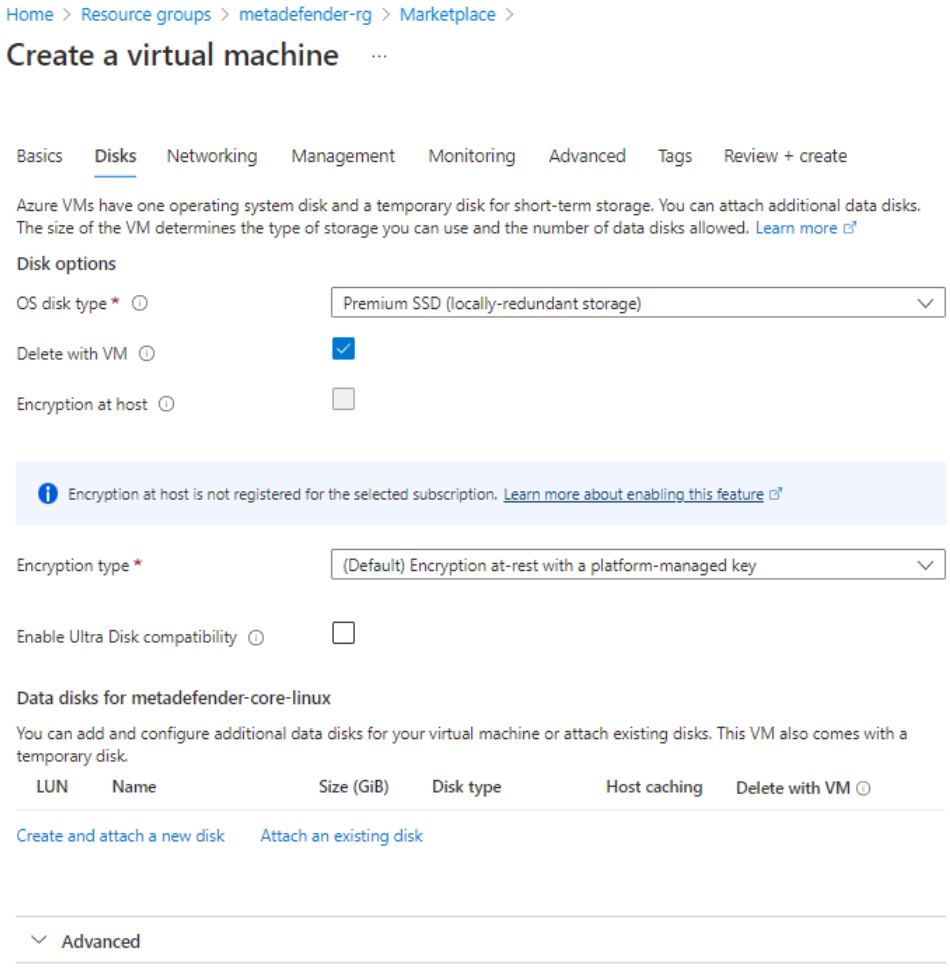
In general there's no need for additional storage by MetaDefender. However, there are 2 situations where additional local storage might be required:

- MetaDefender will process large files or a high volume of files which submitted in MetaDefender's queue will need over 10GB files storage

- MetaDefender is configured to store files in the Quarantine section which will eventually fill the entire root volume.

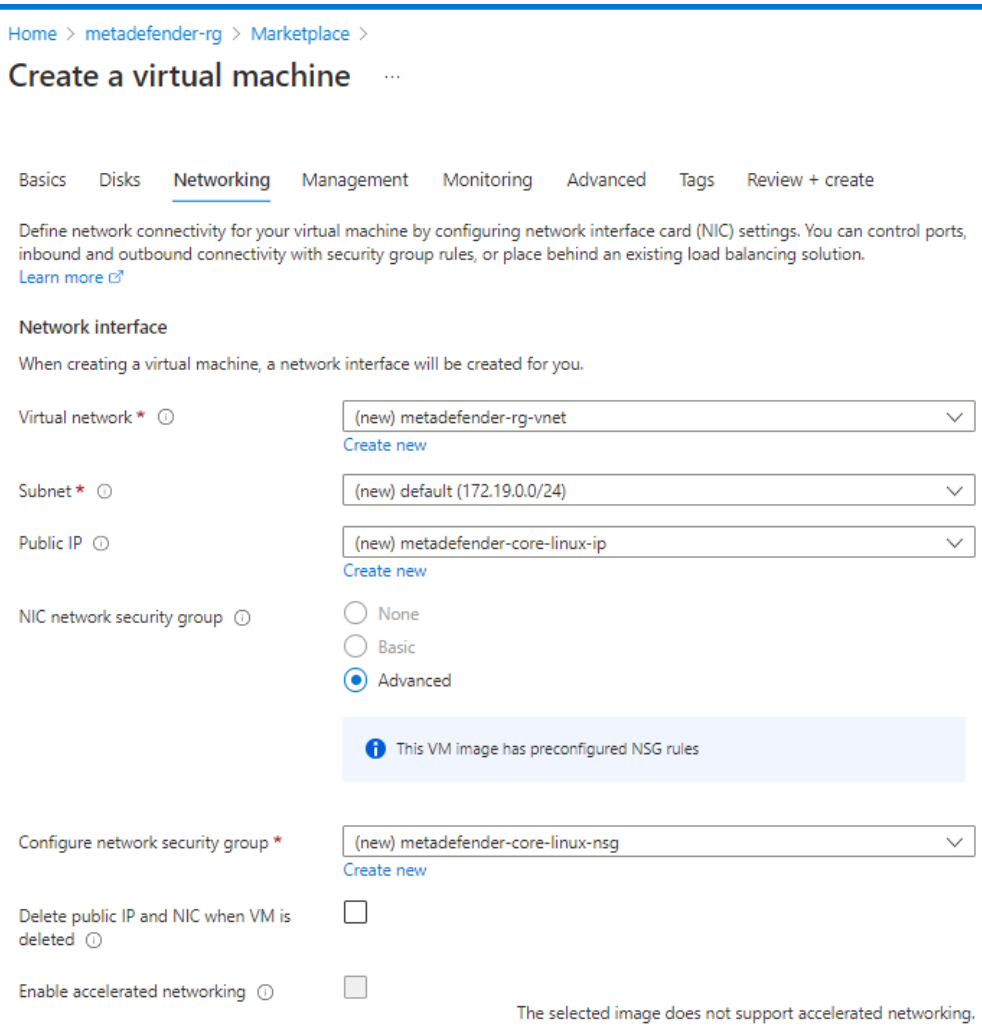
In case quarantining the files in the MetaDefender instance or analyzing high volumes of files is a requirement, please consider adding an additional Azure Managed Disk

- Premium SSD is recommended for having the best performance

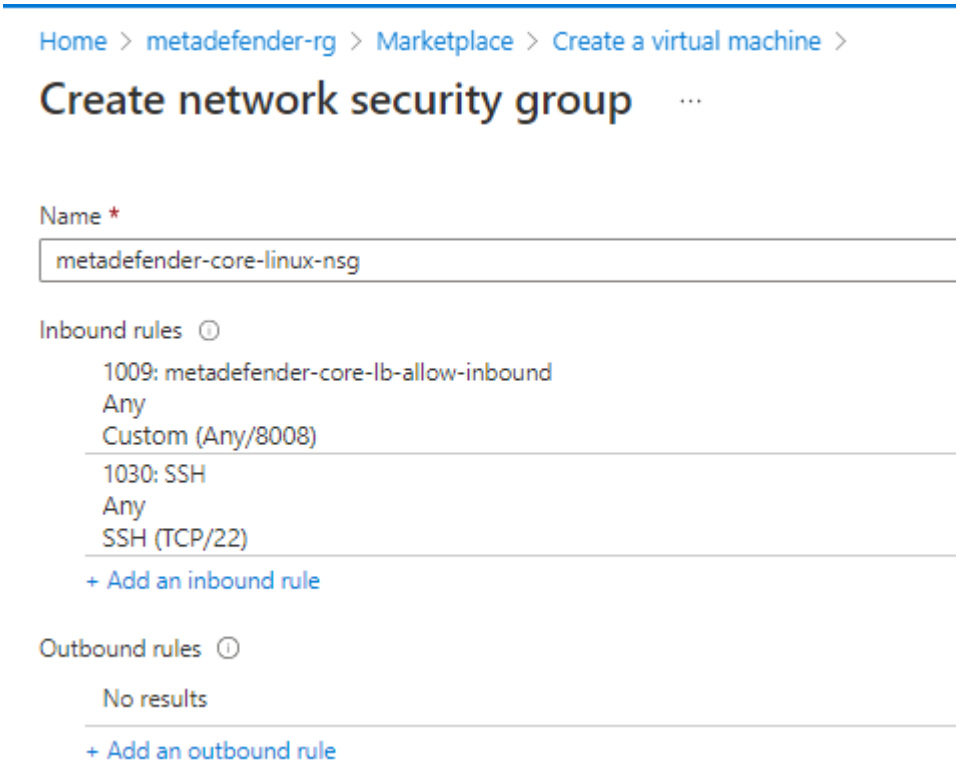


4. Network Configuration

- Change public IP to None in case you do not want to access from internet.
- Select the Virtual network and Subnet in case you already have them created, if not leave it to create them



- Create Security Group adding rules for 8008 and SSH



- Under Load Balancing section as it is a single vm it is not really needed but it can be used to set up a health check using the /readyz endpoint

5. Review & Create the VM

- Under Management, Monitoring, Advanced and Tags there are not any configuration different from the default ones that are needed to be selected. So customize it as it is needed for each use case.
- After creating it will show you a window to download the private key to access to the VM in case that option was selected for accessing

Deployment on AKS

This guide explains how to use the provisioning tools provided by OPSWAT to create an Azure AKS and generate all the Kubernetes components needed to run MetaDefender Core and MetaDefender Storage Security.

> **Info**
>
> In case of having the K8S cluster already created, please check the Containerized Deployment Overview

> This page is purely for provisioning the AKS cluster and install the products at the same time with tools that OPSWAT has prepared.

Deploy using Terraform

- Locally clone the metadefender-k8s repository and go to terraform folder

```
git clone git@github.com:OPSWAT/metadefender-k8s.git#OPTIONAL, IT CAN
BE SET UP WITH ENV VARIABLES
cd terraform/azure/
```

- Modify variables/variables.tfvars file
  - PERSISTENT_DEPLOYMENT=true -> To create K8S cluster with EC2 Worker Nodes
  - DEPLOY_FARGATE_NODES=false -> To true if wanted to create a Fargate profile
  - DEPLOY_RDS_POSTGRES_DB=true -> To create RDS instance for PostgreSQL server
  - POSTGRES_USERNAME -> if DEPLOY_RDS_POSTGRES_DB=true, set up admin database username
  - POSTGRES_PASSWORD -> if DEPLOY_RDS_POSTGRES_DB=true, set up admin database password`

```
aks_service_principal_app_id = "" #OPTIONAL, IT CAN BE SET UP WITH ENV
VARIABLES
aks_service_principal_client_secret = "" #OPTIONAL, IT CAN BE SET UP
WITH ENV VARIABLES
aks_service_principal_object_id = "" #OPTIONAL, IT CAN BE SET UP WITH
ENV VARIABLES

postgres_admin = ""
postgres_password = ""
```

- Run terraform init and apply. Check the resource to be created, after that enter “y”

```
terraform init
terraform apply

# IF Variables are not setup you can use this command
terraform apply \
  -var="aks_service_principal_app_id=$ARM_CLIENT_ID" \
  -var="aks_service_principal_client_secret=$ARM_CLIENT_SECRET"
```

MetaDefenderK8S Script Details

- **GitHub Project** metadefender-k8s --> **Script path**: ./metadefenderk8s.sh
- **Programming Language:** Bash

- Installation **Pre-requisites** for provisioning:
  - Terraform
  - Helm
  - AZ CLI
  - Kubectl
- **Azure Credentials**
  - Set in your local environment variables the credentials of the IAM user under
    - **ARM_CLIENT_ID**
    - **ARM_CLIENT_SECRET**
    - **ARM_SUBSCRIPTION_ID**
    - **ARM_TENANT_ID**
  - ssh public key to used to access the cluster with kubectl. Default [~/.ssh/id_rsa.pub]. Change from here
  - az login before running the script for the step of downloading the kubeconfig file for you
- MetaDefender **Core License Key** [Required with --mdcore parameter]
  - Set it in your local environment variables the credentials under ***MDCORE_LICENSE_KEY***

How to run the script

```
./metadefenderk8s.sh provision -l Azure --mdcore
```

```
./metadefenderk8s.sh provision -l Azure --mdcore --mdss
```

```
./metadefenderk8s.sh provision -l Azure --mdcore --icap
```

> **Info**
>
> The script will deploy a single Worker Node for the cluster with enough space for 1 replica of MD Core. Azure VM size is **Standard_F8s_v2 [8 vCPU & 16 GiB Memory].** Each pod would need a minimum of 4 vCPU and 8 GiB Memory. To change the request to adapt each pod to the specific case go to values.yml To change the size of the node pool for having more MD Core replicas or install additional MetaDefender products go to terraform file terraform/azure/main.tf

Script Parameters

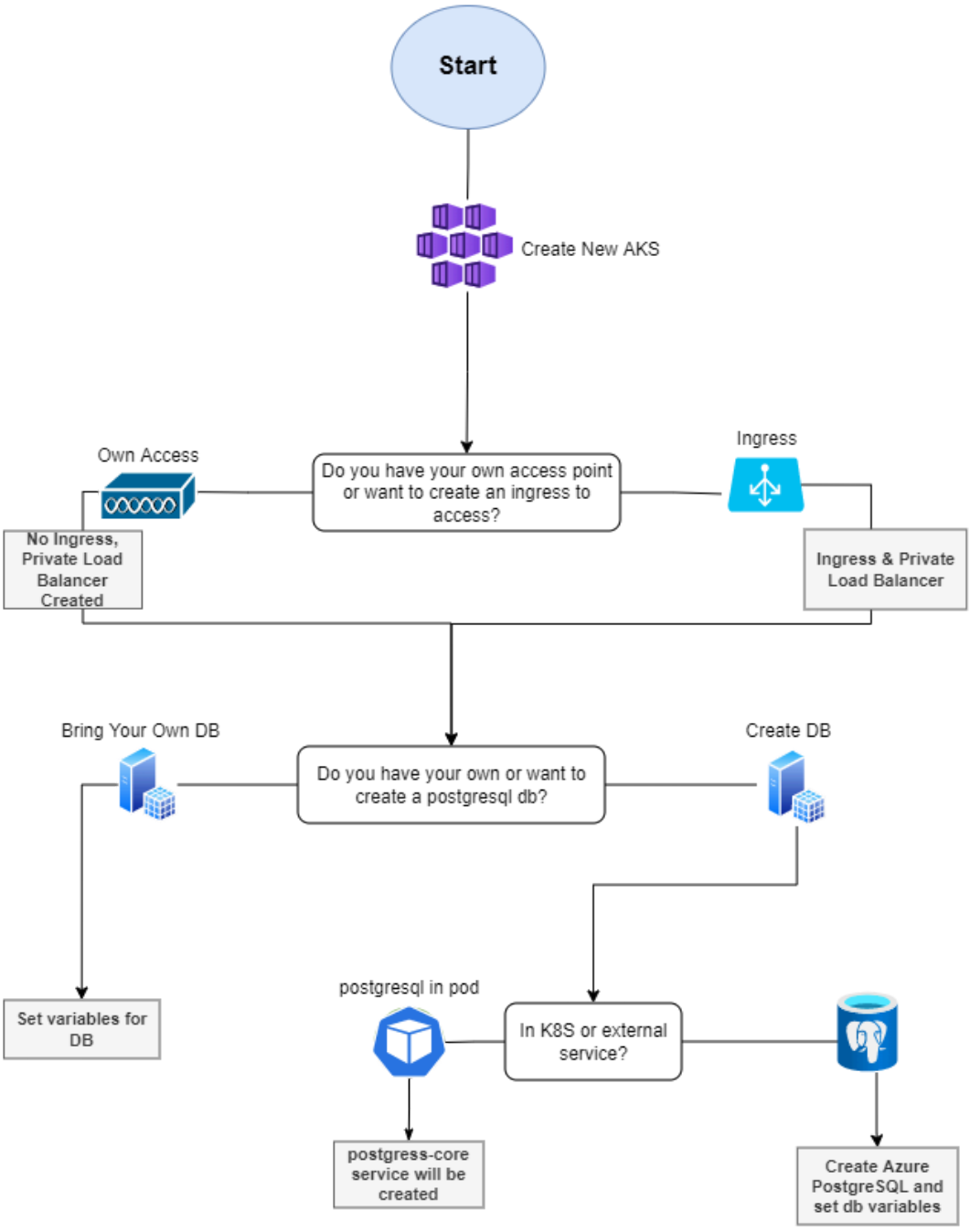
Parameter	Flags	Options	Default	Description	Required/Optional
Action		- provision - install		Action to indicate the script if we want to provision (Create resources + install Core) or install (Install Core)	**Required**
Location	-l or --location	- AWS - Azure - GCP		Where is going to be the K8S cluster	**Required**
MetaDefender Flag Installation	Combination of - --mdcore - --mdss - --icap		-	Install MetaDefender Core in the cluster provisioned	**Required**
Image Version	–image	latest 5.0.1	latest	MetaDefender Core image version to install	Optional
Region	–region	Azure Regions	centralus	AWS region where all the resources will be provisioned	Optional
Cluster Name	–name		md-k8s	Name of the cluster that will be used for naming all the resources	Optional
Number of Replicas	–replicas	[0-9]*	1	Number of replicas for MetaDefender Core service	Optional
Namespace	–namespace	[A-Za-z] {1,10}		Namespace where MetaDefender products will be installed in the K8S Cluster	Optional Max Characters: 10

MetaDefender Core Flowchart Provisioning in Azure

The following flowchart represents how the provisioning script will configure the environment based on the options selected for provisioning Azure AKS.

Summary options to be selected

1. Access to the K8S cluster. Generate **Ingress** or provide **own** access.
    1. **An Ingress** and an internal load balancer will be created per each product flag added as parameter to the script
    2. **Own Access**, you decide how to access to the cluster so it won't generate any ingress but will still create the internal load balancer for the product service deployed
2. Have your **own database** or create **new database**
    1. **Own database,** will be asked if you want the script to set up the credentials and database host url for you or the script will just indicate the secrets to edit, later on by you, for connecting the MetaDefender Core with your database.
    2. Create new **DB in K8S** or **external DB** that for Azure we will provision a **Azure *PostgreSQL Flexible Server***_ **** ___ _ ******._



OPSWAT Licensing on Azure

All the AMI published in the Azure Marketplace are Bring Your Own License, that need to be provided by an OPSWAT Sales representative.

To activate the instances where the MetaDefender products are running, it is needed to activate these products providing the LICENSE_KEY.

> **Info**

> LICENSE_KEY needs to be requested to Sales before starting to install the product.

MetaDefender Storage Security ask for the LICENSE_KEY when doing the on-boarding process. It is needed to assure the persistence of the database, not to have to reactivate the product on every restart.

> **Info**
>
> MetaDefender Storage Security works together with MetaDefender Core, to see how works the licensing for MetaDefender Core check this page

GCP Deployment

Introduction

This Deployment Guide provides step-by-step instructions for deploying MetaDefender Storage Security on Google Cloud Services infrastructure.

This guide is for IT infrastructure architects, administrators and DevOps professionals who are seeking to prevent potential malicious traffic being allowed in their Google Cloud deployment. Threat Prevention is ensured for both productivity files that might be uploaded and for known vulnerabilities that can be identified for all running services/applications deployed in Google. The vulnerability scanning is checking known vulnerabilities for unpatched OS and running applications.

MetaDefender is provided as:

- A packaged installer available for download from the My OPSWAT portal.
- Helm charts for MetaDefender products on GitHub with corresponding container images accessible from the OPSWAT Docker Hub Repository
- AMI through the Google Marketplace [**On Roadmap for Storage Security, for MetaDefender Core it is available**]

There are currently no restrictions on Google regions in which you can deploy MetaDefender.

For deploying our solution in Google, please review the guideline listed below:

- Deploy using Google Compute Engines
- Deployment on GKE

Deploy using Google Compute Engines

The deployment options vary depending on what are the number of instances where MetaDefender Storage Security will be installed.

Options:

1. **Single Instance** deployment of MetaDefender Core and MetaDefender Storage Security
2. **Multi-Instances** with Autoscaling for MetaDefender Core and MetaDefender Storage Security.

OPSWAT provides a terraform project as example to deploy MetaDefender Storage Security and MetaDefender Core in different instances using Google services.

> **Warning**
>
> MetaDefender Storage Security running on Compute Engine only supports multiple instances if the third-party tools are externalized from the Storage Security instance. Google Cloud provides managed services for PostgreSQL and Redis Cache, but not for RabbitMQ. If RabbitMQ is deployed externally on a separate Compute Engine instance, then a multi-instance deployment is supported.

Prerequisites and Requirements

Google knowledge: This guide assumes familiarity with Google Services

Google project: needs permission to create Google Services listed in Architecture Examples in CSPs for Google.

Tools installation: Terraform

Single Instance Deployment

For a single deployment of MetaDefender Core in Google you can use this example to test this deployment type, besides the MetaDefender VM instance, additional resources are being generated and set up.

Deploy using Terraform

OPSWAT provides a terraform project to create the infrastructure needed to deploy the Architecture Examples in CSPs

- Locally clone the metadefender-csp repository and go to GCP/single-vm-deployment

```
git clone git@github.com:OPSWAT/metadefender-csp.git
cd GCP/single-compute-engine-deployment
```

Modify terraform.tfvars with the desired options

- LICENSE__KEY_CORE_ required if wanted to have the MetaDefender Core instance activated automatically
- APIKEY_GENERATION=true to have the apikey generated by terraform
- There is one general section and one section for each product that is supported using this Terraform project

```
# General variables

RG_NAME                 = "metadefender" # Prefix to add to all the
resources
MD_REGION               = "eastus" # Region for all the resources
MD_VNET_CIDR            = "192.168.0.0/16"  # VPC CIDR where to create
the MetaDefender products
PUBLIC_ENVIRONMENT      = true
APIKEY_GENERATION       = true
IMPORT_RG               = false
VM_PWD                  = "<SET_UP_VM_PWD>"

# MetaDefender Core variables

DEPLOY_CORE             = true
CORE_INSTANCE_TYPE      = "Standard_D8s_v5"   # Instance type for
MetaDefender Core
LICENSE_KEY_CORE        = ""
OFFER_PRODUCT_CORE      = "opswat-mdcore-linux"  # Windows opswat-
mdcore-windows
SKU_CORE                = "opswat-mdcore-linux"  # Windows opswat-
mdcore-windows


# MetaDefender Storage Security variables

DEPLOY_MDSS             = true                          # true to deploy
MetaDefender Storage Security together with Core
MDSS_INSTANCE_TYPE      = "Standard_D8s_v5"   # Instance type for
MetaDefender Storage Security
OFFER_PRODUCT_MDSS      = "opswat-mdss-ubuntu"
SKU_MDSS                = "opswatmdssubuntu"
DEPLOY_MDSS_COSMOSDB     = false
```

- Run terraform init and apply. Check the resource to be created, after that enter “y”

```
terraform init
terraform apply
```

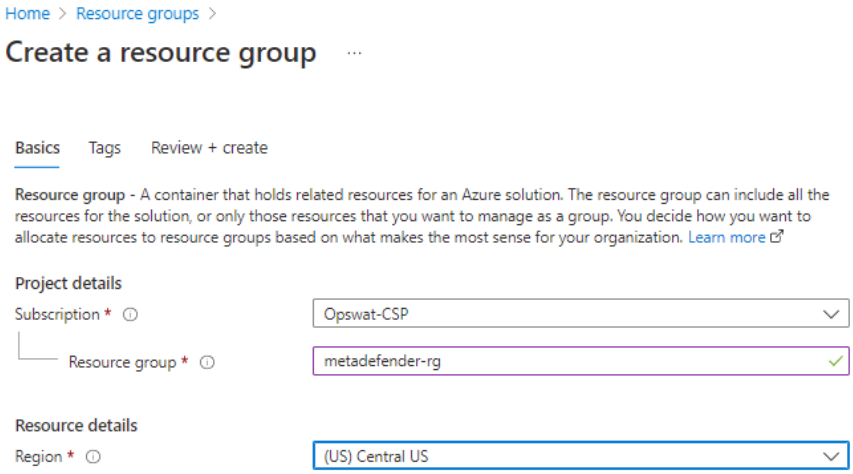
Deploy using Google Cloud Console

Before starting with the installation, Storage Security needs that MetaDefender Core is installed, if you haven´t already installed it please follow this page

For having MetaDefender Storage Security in a Single Compute Engine VM as showed in Single Instance VM Architecture for GCP, it is needed to follow the next steps:

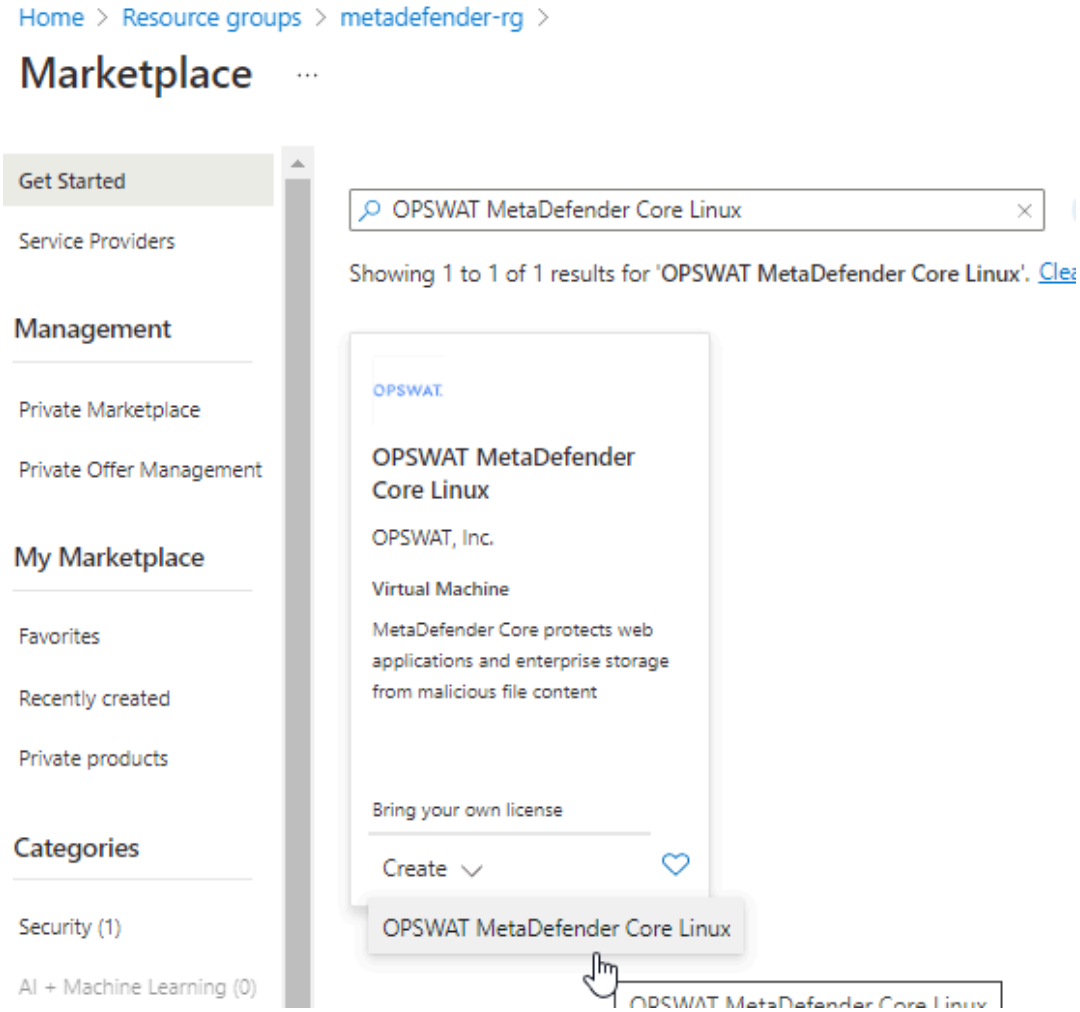
1. Create Resource Group

- Go to you Azure account and create a new resource group (we will refer to it as "metadefender-rg")

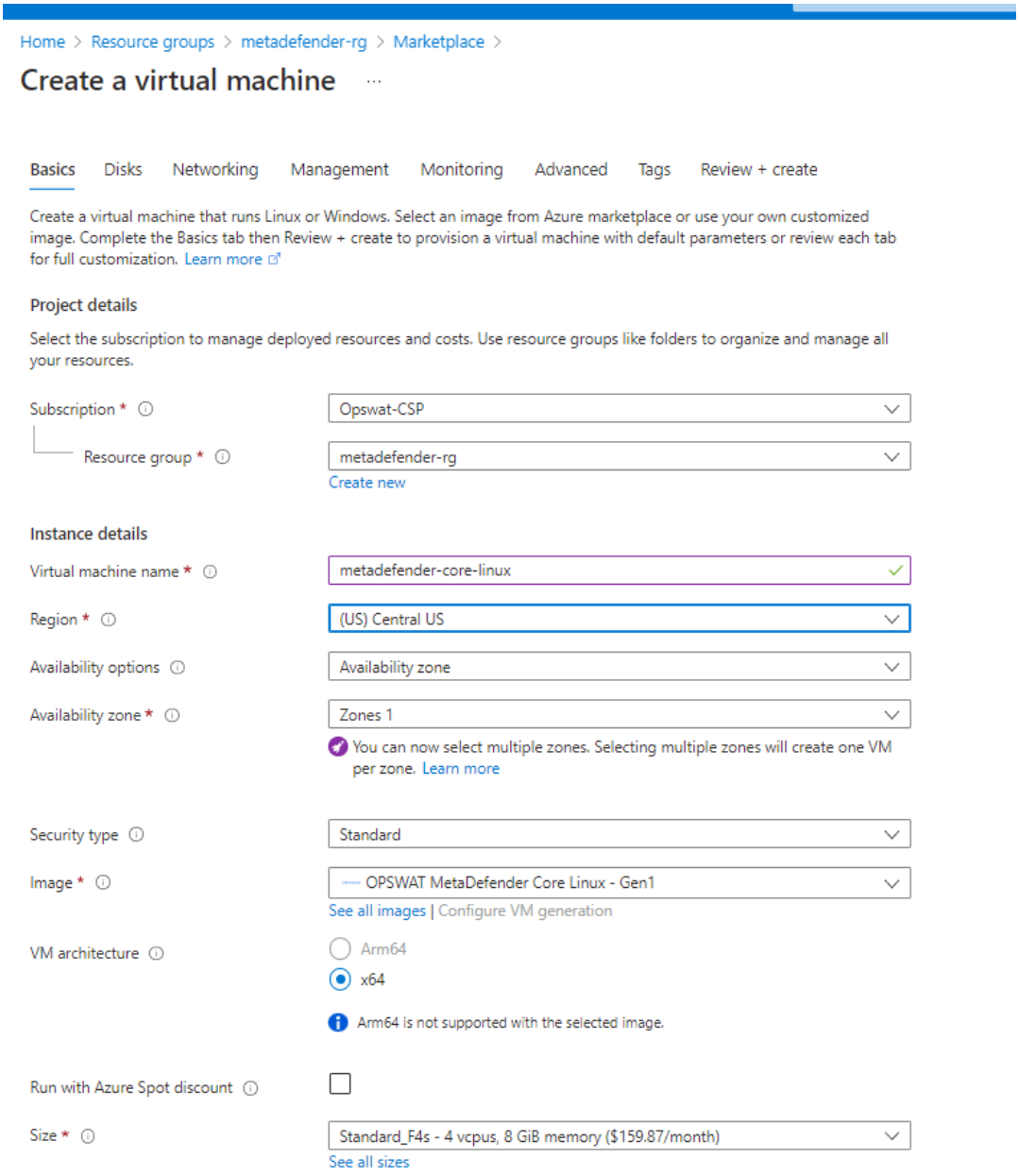


2. Create Azure VM

- Inside the resource group click on create resource and search for ***"OPSWAT MetaDefender Core Linux"***



- Input the vm name and select the size based on General System Requirements
- Under Administrator Account section select the desired way to access to the VM. For this guideline we will generate a public key to access to it



3. Storage Configuration

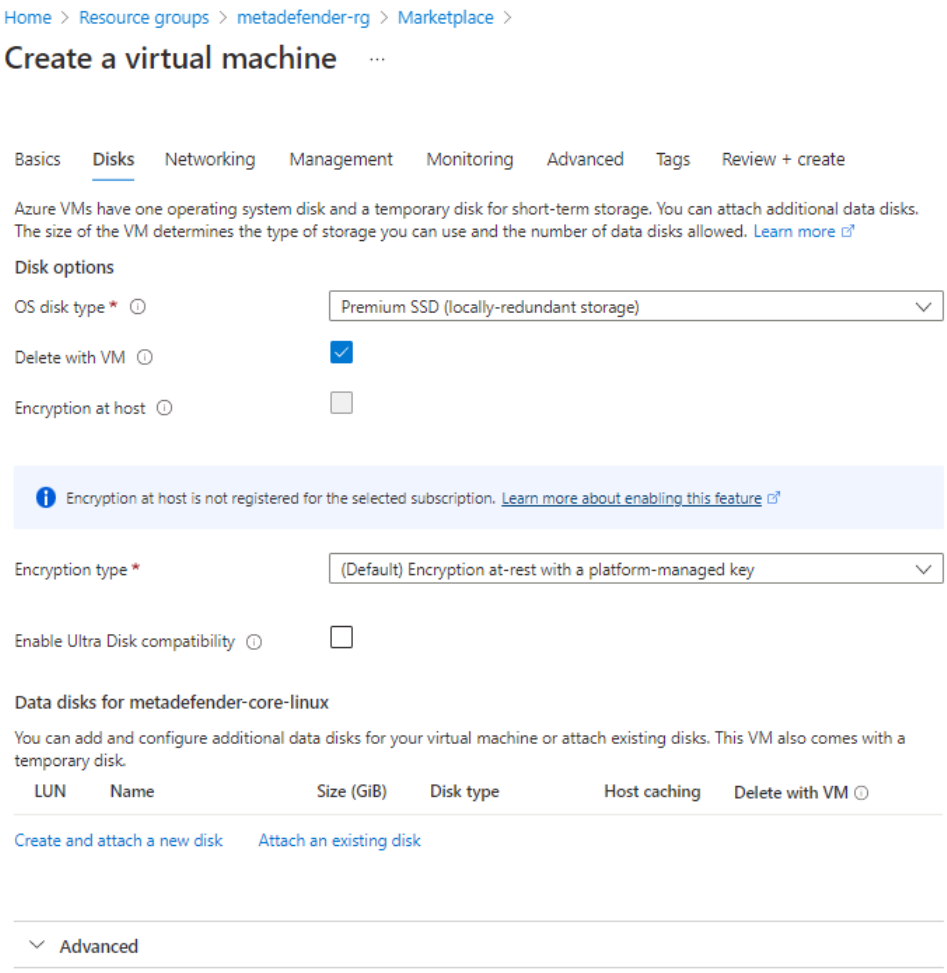
Storage step can be skipped.

In general there's no need for additional storage by MetaDefender. However, there are 2 situations where additional local storage might be required:

- MetaDefender will process large files or a high volume of files which submitted in MetaDefender's queue will need over 10GB files storage
- MetaDefender is configured to store files in the Quarantine section which will eventually fill the entire root volume.

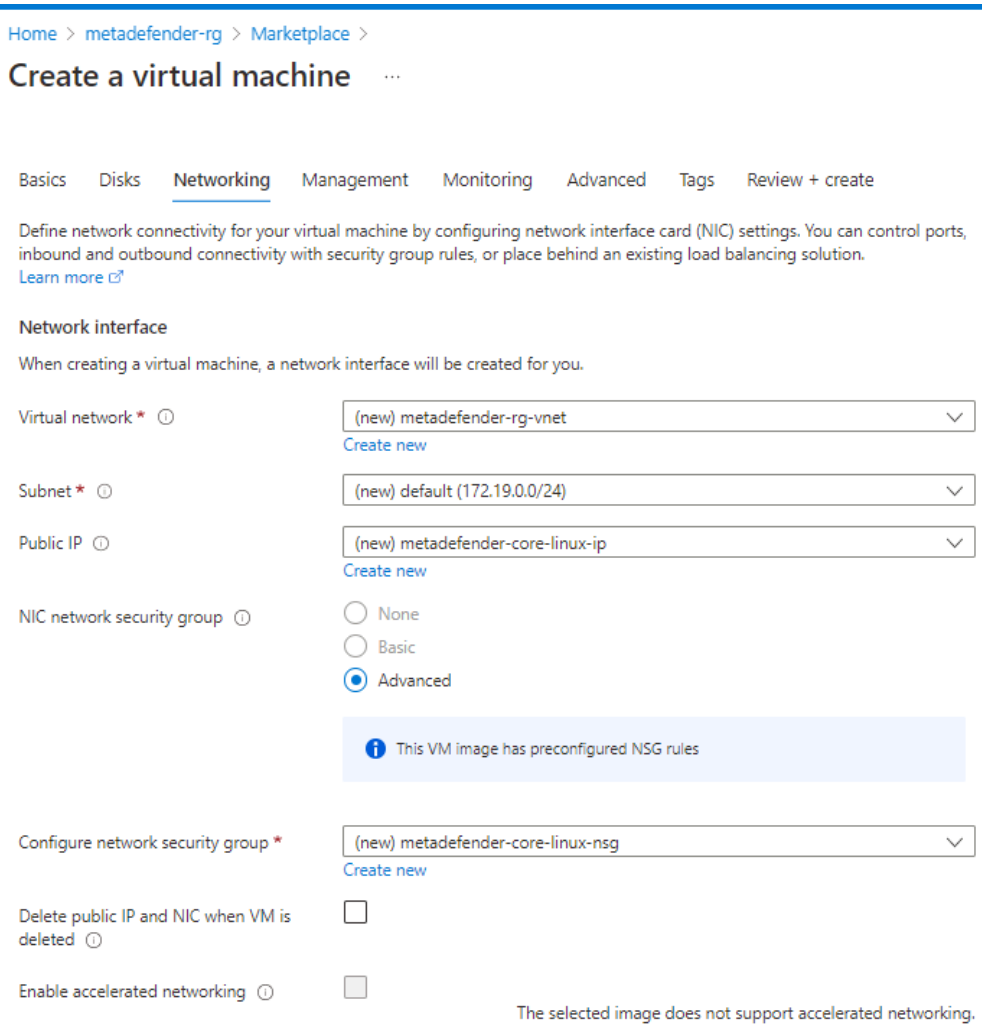
In case quarantining the files in the MetaDefender instance or analyzing high volumes of files is a requirement, please consider adding an additional Azure Managed Disk

- Premium SSD is recommended for having the best performance

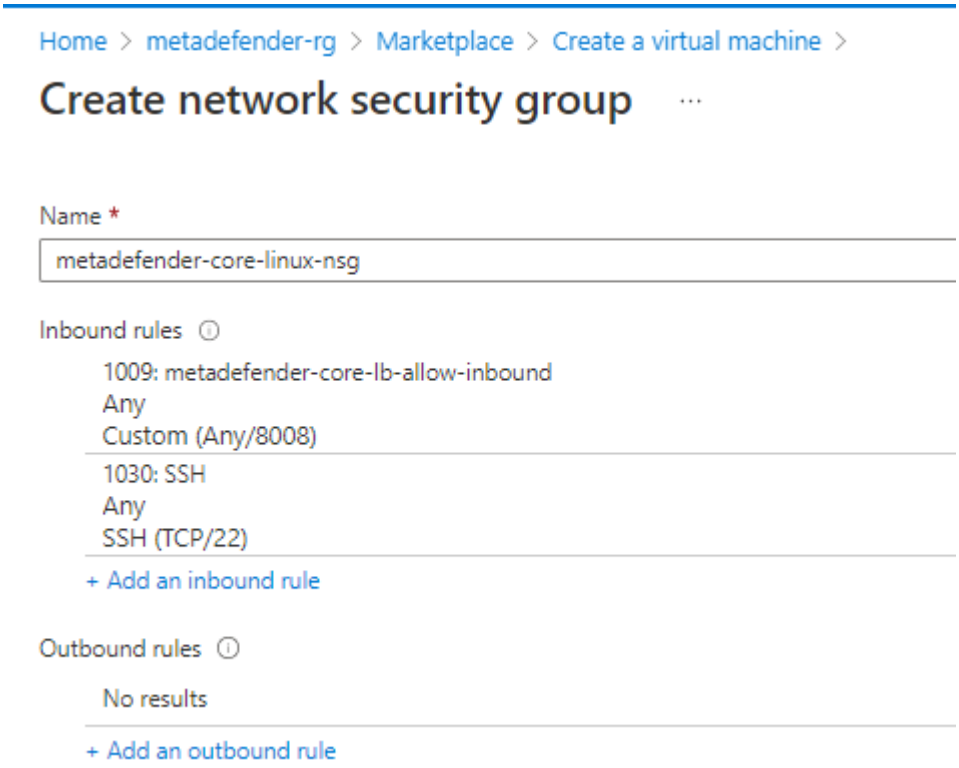


4. Network Configuration

- Change public IP to None in case you do not want to access from internet.
- Select the Virtual network and Subnet in case you already have them created, if not leave it to create them



- Create Security Group adding rules for 8008 and SSH



- Under Load Balancing section as it is a single vm it is not really needed but it can be used to set up a health check using the /readyz endpoint

5. Review & Create the VM

- Under Management, Monitoring, Advanced and Tags there are not any configuration different from the default ones that are needed to be selected. So customize it as it is needed for each use case.
- After creating it will show you a window to download the private key to access to the VM in case that option was selected for accessing

Deployment on GKE

This guide explains how to use the provisioning tools provided by OPSWAT to create an AWS GKE and generate all the Kubernetes components needed to run MetaDefender Core and MetaDefender Storage Security.

> **Info**
>
> In case of having the K8S cluster already created, please check the Deployment Using Helm page

> This page is purely for provisioning the EKS cluster and install the products at the same time with the tools that OPSWAT has prepared.

Deploy using Terraform

- Locally clone the metadefender-k8s repository and go to terraform folder

```
git clone git@github.com:OPSWAT/metadefender-k8s.git
cd terraform/gcloud/
```

- Modify terraform.tfvars file
  - PERSISTENT_DEPLOYMENT=true -> To create K8S cluster with EC2 Worker Nodes
  - DEPLOY_FARGATE_NODES=false -> To true if wanted to create a Fargate profile
  - DEPLOY_RDS_POSTGRES_DB=true -> To create RDS instance for PostgreSQL server
  - POSTGRES_USERNAME -> if DEPLOY_RDS_POSTGRES_DB=true, set up admin database username
  - POSTGRES_PASSWORD -> if DEPLOY_RDS_POSTGRES_DB=true, set up admin database password
  - MQ_USERNAME -> if DEPLOY_RABBITMQ=true
  - MQ_PASSWORD -> if DEPLOY_RABBITMQ=true

```
# project_id = "<ADD_YOUR_PROJECT_ID>"
# region = "europe-central2"
# cluster_location = "europe-central2"
gcloud_json_key_path = "/path/to/json"
# node_count = "1"
# machine_type = "e2-standard-8"
deploy_cloud_sql = true
AUTOPILOT_GKE = false
# private_ip_cloud_sql = true
# cloud_sql_user = postgres
# cloud_sql_password = <ADD_YOUR_POSTGRES_PASS>
```

- Run terraform init and apply. Check the resource to be created, after that enter “y”

```
terraform init
terraform apply
```

MetaDefenderK8S script details

- Script path: %provisioning_script_path%
- **Programming Language:** Bash

- Installation Pre-requisites for provisioning:
  - Terraform
  - Helm
  - AWS CLI
  - Kubectl
  - eksctl (When using the lb controller script)
- AWS Credentials
  - Set in your local environment credentials the IAM user credentials under ***AWS_ACCESS_KEY_ID*** & ***AWS_SECRET_ACCESS_KEY***
- MetaDefender Core License Key (Required with --mdcore parameter)
  - Set it in your local environment credentials under ***MDCORE_LICENSE_KEY***

> **Warning**
>
> As security measure do not use the root user for any deployment or operation. Follow the principle of least privilege for all access granted.

How to run the script

```
./metadefenderk8s.sh provision -l AWS --mdcore --mdss
```

Script Parameters

Parameter	Flags	Options	Default	Description	Required/Optional
Action		- provision - install		Action to indicate the script if we want to provision (Create resources + install Core) or install (Install Core)	**Required**
Location	-l or --location	- AWS - Azure - GCP		Where is going to be the K8S cluster	**Required**
MetaDefender Flag Installation	Combination of - --mdcore - --mdss - --icap		-	Install MetaDefender Core in the cluster provisioned	**Required**
Image Version	–image	- latest -	latest	MetaDefender Core image version to install	Optional
Region	–region	AWS Regions	eu-central-1	AWS region where all the resources will be provisioned	Optional
Cluster Name	–name	Not uppercase allowed	md-k8s	Name of the cluster that will be used for naming all the resources	Optional
Number of Replicas	–replicas	[0-9]*	1	Number of replicas for MetaDefender Core service	Optional
Namespace	–namespace	[A-Za-z] {1,10}		Namespace where MetaDefender products will be installed in the K8S Cluster	Optional Max Characters: 10

Limitations using parameters

- When using Fargate
  - the pods should be installed in the default namespace.
  - if using AWS Load Balancer controllers need a worker node to be installed in.

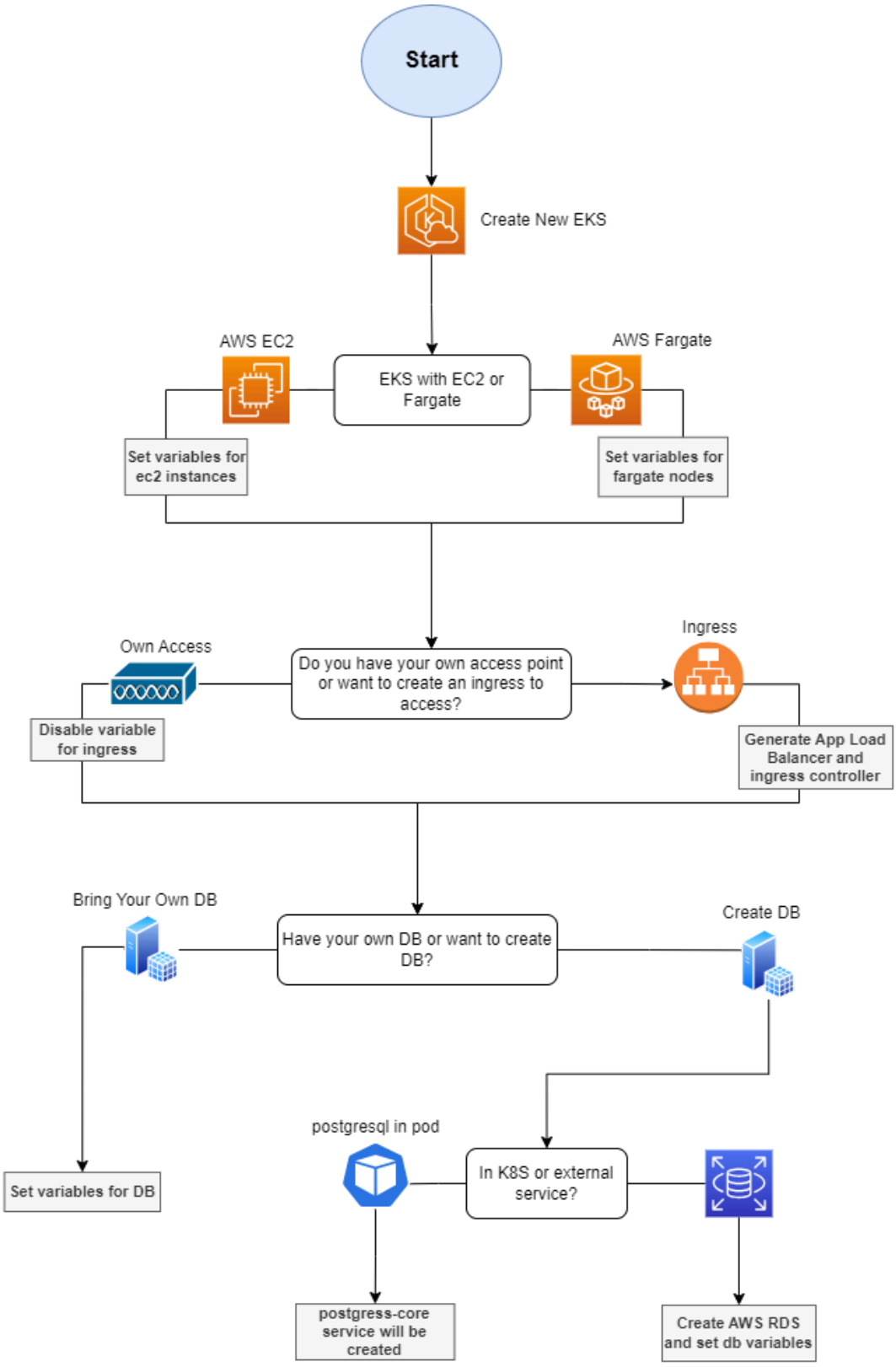
> **Info**
>
> The script will deploy a single Worker Node for the cluster with enough space for 1 replica of MD Core. AWS EC2 Instance size is **c5.2xlarge (8 vCPU & 16 GiB Memory).** Each pod would request 7 vCPU and 8 GiB Memory. To change the request to adapt each pod to the specific case go to values.yml To change the node group go to the terraform file terraform/aws/eks_cluster_module

MetaDefender Core Flowchart Provisioning in AWS

The following flowchart represents how the provisioning script will configure the environment based on the options selected for provisioning AWS EKS.

Summary options to be selected

1. EKS with **EC2** or **Fargate**
2. Access to the K8S cluster. Generate **Ingress** or provide **own** access.
   1. **An Ingress** will be create per each product flag added as parameter to the script
   2. **Own Access**, you decide how to access to the cluster so it won't generate any load balancer for accessing
3. Have your **own database** or create **new database**
   1. **Own database,** will be asked if you want the script to set up the credentials and database host url for you or the script will just indicate the secrets to edit, later on by you, for connecting the MetaDefender Core with your database.
   2. Create new **DB in K8S** or **external DB** that for AWS we will provision a ***RDS instance***


Start
Create New EKS
AWS EC2
AWS Fargate
EKS with EC2 or Fargate
Set variables for ec2 instances
Set variables for fargate nodes
Own Access
Do you have your own access point or want to create an ingress to access?
Ingress
Disable variable for ingress
Generate App Load Balancer and ingress controller
Bring Your Own DB
Have your own DB or want to create DB?
Create DB
postgresql in pod
In K8S or external service?
Set variables for DB
postgress-core service will be created
Create AWS RDS and set db variables

OPSWAT Licensing in GCP

All the AMI published in the GCP Marketplace are Bring Your Own License, that need to be provided by an OPSWAT Sales representative.

To activate the instances where the MetaDefender products are running, it is needed to activate these products providing the LICENSE_KEY.

> **Info**
>
> LICENSE_KEY needs to be requested to Sales before starting to install the product.

MetaDefender Storage Security ask for the LICENSE_KEY when doing the on-boarding process. It is needed to assure the persistence of the database, not to have to reactivate the product on every restart.

> **Info**
>
> MetaDefender Storage Security works together with MetaDefender Core, to see how works the licensing for MetaDefender Core check this page

External Services Configuration

While this documentation focuses on self-managed, locally deployed services for maximum control and customization, MDSS supports any compatible external services — whether self-managed on your own VMs or consumed as managed services from a Cloud Service Provider (CSP).

Protocol Compatibility

MDSS communicates with its external stateful dependencies using **standard open-source protocols only**:

- **PostgreSQL:** standard PostgreSQL wire protocol
- **RabbitMQ:** AMQP 0-9-1 (the standard RabbitMQ protocol)
- **Redis:** standard Redis protocol (RESP)

Any service — managed or self-deployed — that exposes these standard protocols is compatible with MDSS. This is why managed services not explicitly listed in this guide (for example, managed PostgreSQL on Oracle Cloud) also work without compatibility issues. Conversely, managed services that only expose proprietary or different protocols (for example, **Azure Service Bus**, which does not implement AMQP 0-9-1 semantics compatible with RabbitMQ clients) are **not** compatible.

If your cloud provider does not offer a protocol-compatible managed service for a given component, either:

- **Self-deploy the service on a Linux VM** in your cloud environment, following the External Components Setup instructions below, or
- **Use a CSP marketplace alternative** — for example, CloudAMQP provides managed RabbitMQ available through the Azure Marketplace and Google Cloud Marketplace, and can be recommended across any CSP.

The configuration principles outlined in this guide remain the same regardless of whether you choose self-managed or cloud-managed services — simply update the connection URLs and credentials in your `customer.env` file to point to your chosen service endpoints.

Supported Managed Services by Cloud Provider

Component	AWS	Azure	Google Cloud
PostgreSQL	Amazon RDS for PostgreSQL	Azure Database for PostgreSQL	Cloud SQL for PostgreSQL
RabbitMQ	Amazon MQ for RabbitMQ	No native compatible managed service — use CloudAMQP (Azure Marketplace) or self-deploy on a Linux VM	No native compatible managed service — use CloudAMQP (Google Cloud Marketplace) or self-deploy on a Linux VM
Redis	Amazon ElastiCache for Redis	Azure Cache for Redis	Memorystore for Redis

PostgreSQL

Amazon RDS for PostgreSQL

1. In the AWS Console, create an **RDS for PostgreSQL** instance, choosing an instance class that meets the recommended specs.
2. Deploy it in the same VPC as your MDSS hosts and configure the security group to allow inbound traffic on port `5432` from the MDSS hosts only.
3. Connect with the master user and create the `mdss` user, the `MDSS` database, and grants as described in PostgreSQL Configuration (steps 1–3).
4. Copy the instance endpoint into `POSTGRESQL_URL` in `customer.env` on the MDSS host and disable the embedded PostgreSQL container (`DISABLED_SERVICES=postgres`).

Reference: Creating and connecting to a PostgreSQL DB instance (AWS)

Azure Database for PostgreSQL

1. In the Azure Portal, create an **Azure Database for PostgreSQL – Flexible Server** instance sized per the recommended specs.
2. Use private access (VNet integration) or configure firewall rules so only MDSS hosts can reach port `5432`.
3. Connect with the admin user and create the `mdss` user, the `MDSS` database, and grants as described in PostgreSQL Configuration (steps 1–3).
4. Copy the server FQDN into `POSTGRESQL_URL` in `customer.env` and disable the embedded PostgreSQL container (`DISABLED_SERVICES=postgres`).

Reference: Create an Azure Database for PostgreSQL flexible server (Microsoft)

Google Cloud SQL for PostgreSQL

1. In the Google Cloud Console, create a **Cloud SQL for PostgreSQL** instance sized per the recommended specs.
2. Enable a private IP on the VPC shared with your MDSS hosts (or use authorized networks) so only MDSS hosts can reach port `5432`.
3. Connect with the default `postgres` user and create the `mdss` user, the `MDSS` database, and grants as described in PostgreSQL Configuration (steps 1–3).
4. Copy the instance IP into `POSTGRESQL_URL` in `customer.env` and disable the embedded PostgreSQL container (`DISABLED_SERVICES=postgres`).

Reference: Create a Cloud SQL for PostgreSQL instance (Google)

RabbitMQ

Amazon MQ for RabbitMQ (AWS)

1. In the AWS Console, create an **Amazon MQ** broker with the **RabbitMQ** engine type, deployed in the same VPC as your MDSS hosts.
2. Configure the broker's security group to allow inbound traffic on port `5671` (AMQPS) from the MDSS hosts.
3. Create the broker admin credentials and note the AMQPS endpoint.
4. Set `RABBITMQ_URI` (using the `amqps://` scheme), `RABBITMQ_HOST`, `RABBITMQ_PORT`, `RABBITMQ_DEFAULT_USER`, and `RABBITMQ_DEFAULT_PASS` in `customer.env`, and disable the embedded RabbitMQ container (`DISABLED_SERVICES=rabbitmq`).
5. Review the broker's consumer timeout configuration — MDSS long-running scan consumers require the consumer timeout to be disabled or increased, as described in RabbitMQ Configuration.

Reference: Getting started with Amazon MQ for RabbitMQ (AWS)

> **Amazon MQ for RabbitMQ 4.3 or higher** is not supported yet. Create the broker on RabbitMQ 4.2 and turn off automatic minor version upgrades. AWS still upgrades a broker once its version reaches end of support, so check the Amazon MQ version support calendar before that date.

CloudAMQP (Azure and Google Cloud)

Azure and Google Cloud do not offer a native managed service compatible with the standard RabbitMQ (AMQP 0-9-1) protocol — Azure Service Bus is **not** RabbitMQ-compatible. For these providers, use **CloudAMQP** (managed RabbitMQ available via the CSP marketplaces) or self-deploy RabbitMQ on a Linux VM as described in RabbitMQ Configuration.

1. Subscribe to CloudAMQP through the Azure Marketplace or Google Cloud Marketplace, or sign up directly at cloudamqp.com.
2. Create an instance in the same cloud region as your MDSS deployment; for production, choose a dedicated plan and enable VPC peering to keep traffic private.
3. From the instance details page, copy the AMQPS URL and credentials.
4. Set `RABBITMQ_URI` (using the `amqps://` scheme, including the vhost from the CloudAMQP URL), `RABBITMQ_HOST`, `RABBITMQ_PORT`, `RABBITMQ_DEFAULT_USER`, and `RABBITMQ_DEFAULT_PASS` in `customer.env`, and disable the embedded RabbitMQ container (`DISABLED_SERVICES=rabbitmq`).
5. Disable the consumer timeout for the instance (configurable via the CloudAMQP console) as described in RabbitMQ Configuration.

Reference: CloudAMQP documentation

Self-deployed RabbitMQ on a CSP Linux VM (any provider)

If a marketplace service is not an option, provision a Linux VM (e.g., Azure Virtual Machines, Google Compute Engine, Amazon EC2) in the same network as your MDSS hosts and follow the RabbitMQ Configuration instructions below.

Redis

Amazon ElastiCache for Redis (AWS)

1. In the AWS Console, create an **ElastiCache for Redis** cluster in the same VPC as your MDSS hosts.
2. Configure the security group to allow inbound traffic on port `6379` from the MDSS hosts.

3. Enable authentication (Redis AUTH) and, for production, in-transit encryption (TLS).
4. Copy the primary endpoint into `CACHE_SERVICE_URI`, `CACHE_SERVICE_URL`, and `CACHE_SERVICE_PORT` in `customer.env`, and disable the embedded Redis container (`DISABLED_SERVICES=redis`).

Reference: Getting started with Amazon ElastiCache (AWS)

Azure Cache for Redis

1. In the Azure Portal, create an **Azure Cache for Redis** instance in the same region as your MDSS deployment.
2. Use private endpoint / VNet integration or firewall rules so only MDSS hosts can reach the cache.
3. From **Access keys**, copy the host name, port, and access key.
4. Set `CACHE_SERVICE_URI` (host, port, and the access key as the password), `CACHE_SERVICE_URL`, and `CACHE_SERVICE_PORT` in `customer.env`, and disable the embedded Redis container (`DISABLED_SERVICES=redis`).

Reference: Create an Azure Cache for Redis instance (Microsoft)

Google Cloud Memorystore for Redis

1. In the Google Cloud Console, create a **Memorystore for Redis** instance on the VPC shared with your MDSS hosts.
2. Enable AUTH and, for production, in-transit encryption (TLS).
3. Note the instance IP address, port, and AUTH string.
4. Set `CACHE_SERVICE_URI`, `CACHE_SERVICE_URL`, and `CACHE_SERVICE_PORT` in `customer.env` accordingly, and disable the embedded Redis container (`DISABLED_SERVICES=redis`).

Reference: Create a Memorystore for Redis instance (Google)

Prerequisites

- Multiple VMs in the same network for MDSS deployment
- Separate VMs or managed services for external components
- Administrative access to all systems

Recommended specs:

- PostgreSQL: minimum 8 CPU and 16GB of RAM for the average workload
- RabbitMQ: RabbitMQ System Requirements

- Redis: Redis System Requirements
- MDSS VM: General System Requirements

1. External Components Setup

RabbitMQ Configuration

Installation: Follow the official RabbitMQ installation guide for your distribution.

Configure credentials for remote access:

```
sudo rabbitmqctl add_user <username> <password>
sudo rabbitmqctl set_permissions -p / <username> ".*" ".*" ".*"
sudo rabbitmqctl set_user_tags <username> administrator
```

Enable management interface for monitoring:

```
sudo rabbitmq-plugins enable rabbitmq_management
```

Disable consumer timeout (important for long processing consumers)

Edit RabbitMQ config:

```
sudo nano /etc/rabbitmq/rabbitmq.conf
```

Add:

```
consumer_timeout = undefined
```

Restart RabbitMQ

```
sudo systemctl restart rabbitmq-server
```

PostgreSQL Configuration

Installation: Follow the PostgreSQL installation guide.

Configure for remote access:

1. **Create the Database User and Database**

Connect to PostgreSQL as a superuser and run:

```
-- Create the MDSS user (replace <your_password> with a strong
password)
CREATE ROLE mdss LOGIN PASSWORD '<your_password>';

-- Create the database
CREATE DATABASE "MDSS" OWNER mdss;
```

2. ** Grant Required Permissions

**

Still connected as superuser, switch to the `MDSS` database and run:

```
\c "MDSS"

-- Database connection
GRANT CONNECT ON DATABASE "MDSS" TO mdss;

-- Schema permissions (required for migrations to create tables)
GRANT CREATE, USAGE ON SCHEMA public TO mdss;

-- Full privileges on existing objects
GRANT ALL PRIVILEGES ON ALL TABLES    IN SCHEMA public TO mdss;
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO mdss;
GRANT ALL PRIVILEGES ON ALL FUNCTIONS IN SCHEMA public TO mdss;
GRANT ALL PRIVILEGES ON ALL PROCEDURES IN SCHEMA public TO mdss;

-- Default privileges for new objects created by migrations
ALTER DEFAULT PRIVILEGES IN SCHEMA public
    GRANT ALL PRIVILEGES ON TABLES    TO mdss;
ALTER DEFAULT PRIVILEGES IN SCHEMA public
    GRANT ALL PRIVILEGES ON SEQUENCES TO mdss;
ALTER DEFAULT PRIVILEGES IN SCHEMA public
    GRANT ALL PRIVILEGES ON FUNCTIONS TO mdss;

-- pg_catalog access (For encryption extensions if needed)
GRANT USAGE ON SCHEMA pg_catalog TO mdss;
```

3. **Enable the pgcrypto and pg_trgm Extensions**

MDSS requires the pgcrypto and pg_trgm extensions to be available in the `MDSS` database. The embedded PostgreSQL container enables them automatically, so this step applies only when you use an external or remote database.

On a self-managed server, `pgcrypto` and `pg_trgm` ship in the PostgreSQL *contrib* package. Install it if it is not already present:

```
# Debian / Ubuntu
sudo apt-get install postgresql-contrib

# RHEL / Rocky / Alma (replace 16 with your PostgreSQL major version)
sudo dnf install postgresql16-contrib
```

Then, still connected as a superuser, create the extensions **inside the MDSS database**:

```
\c "MDSS"

CREATE EXTENSION IF NOT EXISTS pgcrypto;
CREATE EXTENSION IF NOT EXISTS pg_trgm;
```

Verify that they are installed:

```
SELECT extname, extversion FROM pg_extension WHERE extname IN
('pgcrypto', 'pg_trgm');
```

> **Note:** On PostgreSQL 13 and later, `pgcrypto` and `pg_trgm` are trusted extensions: the `mdss` user, as owner of the `MDSS` database, can create them, and MDSS creates the ones it needs during the database migration. Creating them beforehand as a superuser is still recommended, and it is required on services that restrict extensions, such as Azure [see below].

Managed PostgreSQL services differ slightly:

- **Amazon RDS / Aurora for PostgreSQL:** run the statement as the master user. No additional configuration is required.
- **Azure Database for PostgreSQL – Flexible Server:** add `PGCRYPTO` and `PG_TRGM` to the `azure.extensions` server parameter, save, and then run the statement.
- **Google Cloud SQL for PostgreSQL:** run the statement as the default `postgres` user. No additional configuration is required.

4. **Configure postgresql.conf:**

```
listen_addresses = '*'

# PostgreSQL Performance Configuration
max_connections = 1000

# Memory settings for optimal database performance
shared_buffers = 16GB
work_mem = 256MB
temp_buffers = 64MB
maintenance_work_mem = 2GB
effective_cache_size = 24GB

# WAL (Write-Ahead Logging) configuration
wal_level = minimal
max_wal_senders = 0
max_replication_slots = 0
max_wal_size = 2GB
min_wal_size = 160MB
```

5. **Increase file limits in sysctl.conf:**

```
fs.file-max = 50000
```

6. Update systemd service file `/usr/lib/systemd/system/postgresql.service` to increase file limits for PostgreSQL user.
7. Restart PostgreSQL:

```
sudo systemctl daemon-reload && sudo systemctl restart
postgresql.service
```

8. **Configure customer.env on the MDSS Host**

On the MDSS host, edit /etc/mdss/customer.env (in your MDSS installation directory) and add:

```
POSTGRESQL_URL=Host=<pg-host-ip>;Port=5432;Username=mdss;Password=
<your_password>;Database=MDSS;Timeout=60;CommandTimeout=60
```

Replace with the IP or hostname of your PostgreSQL server and <your_password> with the password you set in Step 1.

9. **Disable the Embedded PostgreSQL Container**

Add the following to customer.env:

```
DISABLED_SERVICES=postgres
```

Redis Configuration

> **Note:** Use external Redis only when deploying MDSS services on multiple nodes. For single-node deployments, the local Redis instance provides better latency.

Installation: Follow the Redis installation guide.

Configure password and remote access in `/etc/redis/redis.conf`:

```
requirepass your_password_here
bind 0.0.0.0
```

2. MDSS Configuration

Configure External Services

Edit `customer.env` file with your external service configurations:

```
# Configure external RabbitMQ
RABBITMQ_URI=amqp://mdss:mdss@<rabbitmq-host>:5672
RABBITMQ_HOST=<rabbitmq-host>
RABBITMQ_PORT=5672
RABBITMQ_DEFAULT_PASS=mdss
RABBITMQ_DEFAULT_USER=mdss

# Optional: Configure external Redis instance
CACHE_SERVICE_URI=<redis-
host>:6379,user=default,password=mdss,syncTimeout=10000
CACHE_SERVICE_URL=<redis-host>
CACHE_SERVICE_PORT=6379

# Configure external database
POSTGRESQL_URL=Host=<pg-host-ip>;Port=5432;Username=mdss;Password=
<your_password>;Database=MDSS;Timeout=60;CommandTimeout=60

# Disable all the local services
DISABLED_SERVICES=postgres,redis,rabbitmq

# Example performance tuning parameters for higher workloads
RABBITMQ_SCANNING_PREFETCH_COUNT=5
RABBITMQ_LOAD_BALANCER_PREFETCH_COUNT=60
RABBITMQ_NFS_DISCOVERY_PREFETCH_COUNT=60
RABBITMQ_DISCOVERY_PREFETCH_COUNT=40
DISCOVERY_THROTTLER_MAX_PENDING_FILES=10000
```

3. Performance Monitoring

Monitor RabbitMQ Queues

Access the RabbitMQ management interface at `http://<rabbitmq-host>:15672` to monitor queue sizes.

Key queue to monitor:

- `object_ready_for_scan_queue`: Shows files waiting to be scanned

If this queue accumulates messages, consider increasing the number of scanning services.

Prefetch Count Optimization

When running multiple scanning services, decrease the scanning prefetch count to distribute load more evenly: RABBITMQ_SCANNING_PREFETCH_COUNT=5

Adjust other prefetch counts based on your queue sizes and performance requirements.

5. Best Practices

1. **Resource Allocation:** Ensure adequate CPU and memory for both MDSS and external service VMs
2. **Network Latency:** Keep all components in the same network to minimize latency
3. **Monitoring:** Regularly monitor RabbitMQ queues and system resources
4. **Scaling:** Start with fewer MDSS instances and scale up based on queue monitoring
5. **Engine Selection:** Use minimal engines required for your use case to improve performance

Troubleshooting

Common Issues:

- **High queue buildup:** Increase scanning service containers or optimize prefetch counts
- **External service timeouts:** Check network connectivity and service configuration
- **Resource exhaustion:** Monitor CPU/memory usage and scale resources accordingly

Log Locations:

- MDSS logs: `/etc/mdss/services_logs/`

This configuration enables horizontal scaling of MDSS deployments while maintaining shared state through external services, providing improved performance for high-volume scanning workloads.

Production Considerations

Security

Encrypted Connections: Always use secure connection protocols (TLS/SSL) for all communications between MDSS instances and external services. Configure RabbitMQ with AMQPS, and Redis with TLS encryption.

Network Security: Implement network-level security by restricting access to external services only from authorized MDSS hosts using firewalls, VPCs, or security groups. Deploy services in private subnets when possible.

Authentication & Authorization: Create dedicated service accounts with minimal required privileges for MDSS access.

High Availability

Configure external services for high availability using their native clustering or replication features:

- **PostgreSQL**: Use replication to a secondary database
- **RabbitMQ:** Configure clustering with HA policies
- **Redis:** Use Redis Sentinel or Redis Cluster

For managed cloud services, enable multi-AZ deployments and automatic failover capabilities provided by your cloud provider.

Monitoring & Alerting

Implement comprehensive monitoring for:

- Service availability and response times
- Resource utilization (CPU, memory, disk, network)
- Queue depths and processing rates
- Connection pool health
- Error rates and failed operations

Set up alerting for critical thresholds that could impact MDSS performance.

Backup & Disaster Recovery

Establish regular backup procedures and test recovery processes for all external services. For managed services, configure automated backups with appropriate retention policies.

Performance Optimization

- Configure connection pooling based on your concurrent load requirements
- Set appropriate resource limits and file descriptors
- Monitor and tune service-specific performance parameters
- Plan capacity based on expected MDSS throughput

Documentation References

For detailed configuration guidance, refer to the official production documentation:

- PostgreSQL Documentation
- RabbitMQ Production Checklist
- Redis Security Guidelines

For managed cloud services, consult your cloud provider's best practices documentation for the specific services you're using.

Automated Deployment on a Dedicated Machine

Use this option when you want the external services on a separate VM in the same trusted network and you do not need a cloud-managed offering. The manual and cloud-managed paths described earlier on this page remain fully supported.

Supported services

Selector	Service	Default published port	Port override variable
db	PostgreSQL	5432	POSTGRES_EXTERNAL_PORT
cache	Redis	6379	REDIS_EXTERNAL_PORT
queue	RabbitMQ	5672	RABBITMQ_EXTERNAL_PORT

Any subset can be selected — for example only `db` if Redis and RabbitMQ stay on the product machine. Port overrides are set in `/etc/mdss/customer.env` on the dedicated

machine before running the command.

Deployment steps

1. Install the MDSS package

Install the same DEB or RPM package used for the product on the dedicated machine for external services, following the standard installation guide. Do not start the product on this machine.

2. Start the external services

```
mdss -c start -e db,cache,queue
```

The command starts only the selected services, publishes their ports, and generates credentials for them (a Redis password and a RabbitMQ `mdss` user with a random password; the PostgreSQL password is the one already configured in `customer.env`). It then prints a settings block:

```
================================================================================
 The following services are now running on this machine (10.0.20.14):
   db, cache, queue

 Add the lines below to your customer.env file (default location
/etc/mdss/customer.env)
 on the machine running MetaDefender Storage Security, then run 'mdss -
c restart' there:
--------------------------------------------------------------------------------
DISABLED_SERVICES=postgres,redis,rabbitmq
POSTGRESQL_URL=Host=10.0.20.14;Port=5432;Username=mdss;Password=...;Dat
abase=MDSS;Timeout=60;CommandTimeout=60
CACHE_SERVICE_URI=10.0.20.14:6379,syncTimeout=10000,user=default,passwo
rd=...
CACHE_SERVICE_URL=10.0.20.14
CACHE_SERVICE_PORT=6379
RABBITMQ_HOST=10.0.20.14
RABBITMQ_PORT=5672
RABBITMQ_DEFAULT_USER=mdss
RABBITMQ_DEFAULT_PASS=...
RABBITMQ_URI=amqp://mdss:...@10.0.20.14:5672
--------------------------------------------------------------------------------
 A commented copy was saved in /etc/mdss/customer.env on this machine.
================================================================================
```

Example output — addresses and passwords are generated per deployment. A commented copy of the block is kept at the end of `customer.env` on the dedicated machine, so it can be retrieved later without regenerating anything.

3. Apply the printed settings

1. Append the printed block to `/etc/mdss/customer.env` on the main machine running MetaDefender Storage Security.
2. Run `mdss -c restart`.

The product now connects to the dedicated machine, and the embedded PostgreSQL, Redis, and RabbitMQ containers are no longer started there (this is what the generated `DISABLED_SERVICES` line controls).

Configuration variables

Variable	Set on	Purpose
`EXTERNAL_SERVICES_HOST`	Dedicated machine	The address written into the printed settings. Set it before running the command when the machine has several addresses and the auto-detected one is not the address the product should connect to.
`EXTERNAL_SERVICES_BIND_ADDRESS`	Dedicated machine	Binds the published ports to a single interface instead of all interfaces (`0.0.0.0` by default).
`POSTGRES_EXTERNAL_PORT`, `REDIS_EXTERNAL_PORT`, `RABBITMQ_EXTERNAL_PORT`	Dedicated machine	Override the published port of each service.

Post-install operations

After the first start, the dedicated machine remembers its role. The usual commands keep working there without repeating `-e`:

```
mdss -c status
mdss -c stop
mdss -c restart
```

Running `mdss -c start -e ...` again is safe: it re-prints the settings block using the existing credentials, so it can be used at any time to recover the values.

The published ports are not filtered by ufw or firewalld. Docker publishes them at the NAT level, so they bypass host firewalls that filter the INPUT chain and are reachable from every network that can route to the machine. Restrict access to the ports from the `DOCKER-USER` iptables chain, from the network firewall, or by setting `EXTERNAL_SERVICES_BIND_ADDRESS` to a single interface.

The connections are not encrypted. Use this deployment model only over a trusted network, with firewall rules that allow the published ports solely from the machine running MetaDefender Storage Security. For TLS-encrypted connections, use the manual or cloud-managed configurations described earlier on this page.

ON-PREMISE DEPLOYMENT

On-Premise Deployment Overview

Before installation

Before you begin the installation, ensure that your On-premise environment meet the system requirements. If you are installing MetaDefender Storage Security on the same server as MetaDefender Core, the server must meet the cumulative system requirements of all the products.

Installation General Steps

1. Download the package of your choice from the My OPSWAT
2. Check the Production Considerations
3. Install the package on your computer based on Installation Methods
4. Open a web browser and point to http:// [by default, port 80 is used]
5. Complete the installation and proceed with the onboarding

Installation notes

- **Internet Connectivity:** The installation requires internet access to download packages from the container registry
- **.NET Framework:** On Windows, the installer will download and apply .NET Framework 4.8 if it is not already installed
- **Support:** If you encounter issues, open a support case and include the "Metadefender*" files located in `C:\Users\<user>\AppData\Local\Temp`

Offline Installation

If you need to perform an offline installation, please refer to the Linux Offline Installation for detailed instructions on how to install MetaDefender Storage Security without an active internet connection. This guide includes steps for preparing necessary dependencies.

Production Considerations

This section of the user guide focuses on essential production considerations. When deploying in a production environment, it is recommended to follow this guide to ensure optimal security and reliability. Key steps include:

1. **Setting up HTTPS**: Ensure that HTTPS is configured to secure communication between clients and servers.
2. **Using an External Database**: Instead of the built-in database, configure an external database to handle your data for better performance and scalability.
3. **OS-Specific Tasks**: Perform operating system-specific tasks that enhance the security and reliability of your system.
4. **Planning for High Availability**: Add redundancy with an active-active deployment, or use the reference active-passive failover script, which monitors an active instance and, when it stops responding, starts a real-time scan of one chosen storage on a standby instance. Its usage is documented in its README.

Following these guidelines will help you maintain a secure and efficient production environment.

Depending on your deployment type, please choose between:

- Production considerations for Unix-based deployments
- Production considerations for Windows-based deployments

Windows Deployments

This page describes in detail how you can achieve the following tasks:

- Use an external PostgreSQL server instead of the built-in database
- Configure HTTPS communication for the web server

Use an external database

Production deployments of MetaDefender Storage Security should ensure high availability and/or load-balancing at the database level by using an external PostgreSQL (or any compatible service) instead of the built-in database that is included with the basic installation.

> **Info**
>
> We recommend using a version of PostgreSQL that is the same as the built-in database version from the requirements page.

How to setup an external PostgreSQL in MetaDefender Storage Security

> **Info**

> These steps should be performed immediately after the installation (before completing the onboarding or any other UI configuration) because any existing data will be lost.

1. Ensure that your external PostgreSQL is properly configured and ready to accept remote connections

Note: it is recommended to setup access control for your external database as well as taking the necessary network security measures to ensure that only the server running MetaDefender Storage Security can access the database.

1. Locate the configuration file `customer.env` created by the installer (by default it is located at *C:\Program Files\OPSWAT\MetaDefender Storage Security\config*)
2. Add the following configuration line:

```
POSTGRES_USER=user
POSTGRES_PASSWORD=password
POSTGRES_DB=dbName
POSTGRESQL_URL=Host=postgres;Port=5432;Username=user;Password=password;
Database=dbName
```

4. Please check the connection string documentation for more information. Make sure you add “mdss” as the target database as shown in the example above.
5. Save the file.
6. Reinstall the internal components by running the following PowerShell command

```
PS> .\mdss.ps1 -c refresh
```

7. Continue to the onboarding process.

PostgreSQL SSL Configuration

To enable SSL/TLS certificate validation for the PostgreSQL connection, update the `POSTGRESQL_URL` variable in the `customer.env` file.

File Location

By default, the `customer.env` file is located at:

C:\Program Files\OPSWAT\MetaDefender Storage Security\config

The installer creates this file automatically.

Configuring POSTGRESQL_URL

The `POSTGRESQL_URL` variable can include additional parameters for SSL/TLS. By default SSL Mode is set to “prefer”.

```
POSTGRESQL_URL=Host=postgres;Port=5432;Username=user;Password=password;
Database=dbName;SSL Mode=VerifyFull;Root Certificate=<path to root CA
certificate>
```

SSL Mode Options

- **prefer** (default): Attempts to connect using SSL. If SSL fails, falls back to unencrypted. **No certificate validation.**
- **require**: Requires SSL encryption. **Certificate validation is skipped** — allows self-signed, expired, or otherwise invalid certifications.
- **verify-full**: Requires SSL and performs **full certificate validation**, including hostname match and trusted CA.

Important Notes

- If **certificate validation is required**, use `verify-full` and provide a valid CA-signed certificate via the `Root Certificate` parameter.
- If using a **self-signed or otherwise invalid certificate**, use `require`, which skips validation but still encrypts the connection.
- The `Root Certificate` must point to the CA file if using `verify-ca` or `verify-full`.

How to modify it

Set the parameter in your **customer.env** file. Please find here more details on how to update the configuration file.

PostgreSQL advanced setup

For special use cases you may need to backup your database or restore it. MetaDefender Storage Security allows you to do this by using the following commands

```
PS> .\mdss.ps1 -u export_db
```

Running this command will create a dump file in the install directory. This file can be use afterwards as a restore file using the import command:

```
PS> .\mdss.ps1 -u import_db
```

> **Info**
>
> The import_db utility will use the latest file as restore point

> **CRITICAL: Stop MDSS Service Before Database Import**
>
> Data corruption may occur if the MDSS service is running during database import.
>
> Follow these steps:
>
> **1.** Stop the service: mdss -c stop **2.** Verify it's stopped: mdss -c status **3.** Import your database **4.** Restart the service: mdss -c start
>
> **Allow 30 seconds between stop and start commands.**
>
> Failure to follow this procedure may result in data loss.

Configure HTTPS communication for the web server

Production deployments should ensure that any HTTP traffic between clients (browsers or other API client) and MetaDefender Storage Security is encrypted using HTTPS.

Click here for detailed instructions on how to setup a certificate for NGINX web server.

Linux Deployments

This page describes in detail how you can achieve the following tasks:

- Use an external PostgreSQL server instead of the built-in database
- Configure HTTPS communication for the web server
- Install MetaDefender Storage Security on a different disk

Use an external database

Production deployments of MetaDefender Storage Security should ensure high availability and/or load-balancing at the database level by using an external PostgreSQL (or any compatible service) instead of the built-in database that is included with the basic installation.

> **Info**

> We recommend using a version of PostgreSQL that is the same as the built-in database version from this page.

How to setup an external PostgreSQL in MetaDefender Storage Security

The following steps should be performed before initializing MetaDefender Storage Security for the first time (i.e before the **mdss -u init** command from the Installation Methods).

> **Info**
>
> These steps can be performed after the installation or at any other time but the existing data (if any) will be lost. In this case, the last step should be replaced with a restart instead of a start command.

1. Ensure that your external PostgreSQL is properly configured and ready to accept remote connections

Note: it is recommended to setup access control for your external database as well as taking the necessary network security measures to ensure that only the server running MetaDefender Storage Security can access the database.

2. Locate the configuration file created by the installer in **/etc/mdss/customer.env**
3. Add the following configuration line:

```
POSTGRES_USER=user
POSTGRES_PASSWORD=password
POSTGRES_DB=dbName
POSTGRESQL_URL=Host=postgres;Port=5432;Username=user;Password=password;
Database=dbName
```

4. Please check the connection string documentation for more information. Make sure you add “mdss” as the target database as shown in the example above.
5. Save the file.
6. Initialize MetaDefender Storage Security by running the following command:

```
sudo mdss -u init
```

7. Return to the steps in Installation Methods and continue the installation.

PostgreSQL SSL Configuration

To enable SSL/TLS certificate validation for the PostgreSQL connection, update the `POSTGRESQL_URL` variable in the `customer.env` file.

File Location

By default, the `customer.env` file is located at:

C:\Program Files\OPSWAT\MetaDefender Storage Security\config

The installer creates this file automatically.

Configuring POSTGRESQL_URL

The `POSTGRESQL_URL` variable can include additional parameters for SSL/TLS. By default SSL Mode is set to “prefer”.

```
POSTGRESQL_URL=Host=postgres;Port=5432;Username=user;Password=password;
Database=dbName;SSL Mode=VerifyFull;Root Certificate=<path to root CA
certificate>
```

SSL Mode Options

- **prefer** [default]: Attempts to connect using SSL. If SSL fails, falls back to unencrypted. **No certificate validation.**
- **require**: Requires SSL encryption. **Certificate validation is skipped** — allows self-signed, expired, or otherwise invalid certifications.
- **verify-full**: Requires SSL and performs **full certificate validation**, including hostname match and trusted CA.

Important Notes

- If **certificate validation is required**, use `verify-full` and provide a valid CA-signed certificate via the `Root Certificate` parameter.
- If using a **self-signed or otherwise invalid certificate**, use `require`, which skips validation but still encrypts the connection.
- The `Root Certificate` must point to the CA file if using `verify-ca` or `verify-full`.

How to modify it

Set the parameter in your **customer.env** file. Please find here more details on how to update the configuration file.

PostgreSQL advanced setup

For special use cases you may need to backup your database or restore it. MetaDefender Storage Security allows you to do this by using the following commands

```
sudo mdss -u export_db
```

Running this command will create a dump file in `/etc/mdss/db_[...].tar` this file can be use afterwards as a restore file using the import command:

```
sudo mdss -u import_db
```

> **Info**
>
> Product versions 3.4.2 and above do not support importing databases from earlier versions (less than 3.4.2). When using the `import_db` utility on these versions, it will automatically restore the database from the most recent backup file named `db_[...].tar`.

> **CRITICAL: Stop MDSS Service Before Database Import**
>
> > Data corruption may occur if the MDSS service is running during database import.
> >
> > Follow these steps:
> >
> > **1.** Stop the service: mdss -c stop **2.** Verify it's stopped: mdss -c status **3.** Import your database **4.** Restart the service: mdss -c start
> >
> > **Allow 30 seconds between stop and start commands.**
> >
> > Failure to follow this procedure may result in data loss.

Configure HTTPS communication for the web server

Production deployments should ensure that any HTTP traffic between clients (browsers or other API client) and MetaDefender Storage Security is encrypted using HTTPS.

Click here for detailed instructions on how to setup a certificate for NGINX web server.

Install MetaDefender Storage Security on a different disk

The following steps should be performed before initializing MetaDefender Storage Security for the first time (i.e before the **mdss -u init** command from the Linux Installation).

> **Info**
>
> These steps should be performed right after the installation of Docker so that no data is lost.

1. First make sure Docker is not running and the default directory is clean
2. Remove existing images and volumes

```
sudo docker rm -f $(docker ps -aq); docker rmi -f $(docker images -q)
```

3. Stop Docker

```
sudo systemctl stop docker
```

4. Remove the default Docker location

```
sudo rm -rf /var/lib/docker
```

5. Create a new empty directory in the default location

```
sudo mkdir /var/lib/docker
```

6. Create a new empty directory on the new partition

```
sudo mkdir /<PATH_TO_NEW_LOCATION>
```

7. Mount the new location in the default directory

```
sudo mount --rbind //<PATH_TO_NEW_LOCATION> /var/lib/docker
```

8. Start back the docker service

```
sudo systemctl start docker
```

9. Initialize MetaDefender Storage Security by running the following command:

```
sudo mdss -u init
```

Installation Methods

The different installation methods depend by the OS and if they are installed using the command line, the wizard or the offline option.

- Deploy to a Windows server
- Deploy to a Linux Server
- Deploy to a Linux Server offline

Windows Installation

Windows package using Command Line (.msi)

From the command line interface, it is possible to install the product by executing this PowerShell command in an **Administrator** session:

```
PS>msiexec /i metadefender-storage-security.msi
```

where the possible keys and their default values are the following:

Key	Description
/quiet or /passive	Suppress the installation wizard
–% INSTALLFOLDER	Customize the installation directory
WEB_CLIENT_PORT	80 (Web client (UI) port)
API_GATEWAY_PORT	8005
LICENSING_SERVICE_PORT	5001
POSTGRESQL_PORT	5432
SMB_PORT_RANGE_START	6002(SMB starting port. Reserves 10 ports total: 5 consecutive from this value and 5 more offset by 1000)
SFTP_SERVICE_PORT	5003
NFS_SERVICE_PORT	5004
PG_ADMIN_PASSWORD	PostgreSQL superuser password
MODULE_AMAZON_S3	Enable Amazon S3 module (1=enabled, 0=disabled)
MODULE_AZURE_BLOB	Enable Azure Blob module (1=enabled, 0=disabled)
MODULE_AZURE_FILES	Enable Azure Files module (1=enabled, 0=disabled)
MODULE_GOOGLE_CLOUD	Enable Google Cloud module (1=enabled, 0=disabled)
MODULE_ALIBABA_CLOUD	Enable Alibaba Cloud module (1=enabled, 0=disabled)
MODULE_SMB	Enable SMB module (1=enabled, 0=disabled)
MODULE_BOX	Enable Box module (1=enabled, 0=disabled)
MODULE_ONEDRIVE_SHAREPOINT	Enable OneDrive / SharePoint Online module (1=enabled, 0=disabled)
MODULE_SFTP	Enable SFTP module (1=enabled, 0=disabled)
MODULE_NFS	Enable NFS module (1=enabled, 0=disabled)
MODULE_FTP	Enable FTP module (1=enabled, 0=disabled)
MODULE_OPSWAT_MFT	Enable OPSWAT MFT module (1=enabled, 0=disabled)
MODULE_ORACLE_SDK	Enable Oracle SDK module (1=enabled, 0=disabled)
MODULE_SHAREPOINT_ONPREM	Enable SharePoint On-Prem module (1=enabled, 0=disabled)

Step by step using Install Wizard

The Install Wizard is only for the Windows installer (.msi file).

Before you begin, please make sure to read the following pages in order to install the necessary requirements for each system.

1. Navigate to MetaDefender Storage Security installer location
2. Run it with Administrator privileges.

Important: If a local antivirus or endpoint protection solution is active, you must configure an exclusion for the MetaDefender Storage Security installation directory before running the installer. Failure to do so may result in blocked components, incomplete installation, or unstable runtime behavior. Refer to How to Exclude the OPSWAT Folder from Windows Defender? for exclusion instructions.

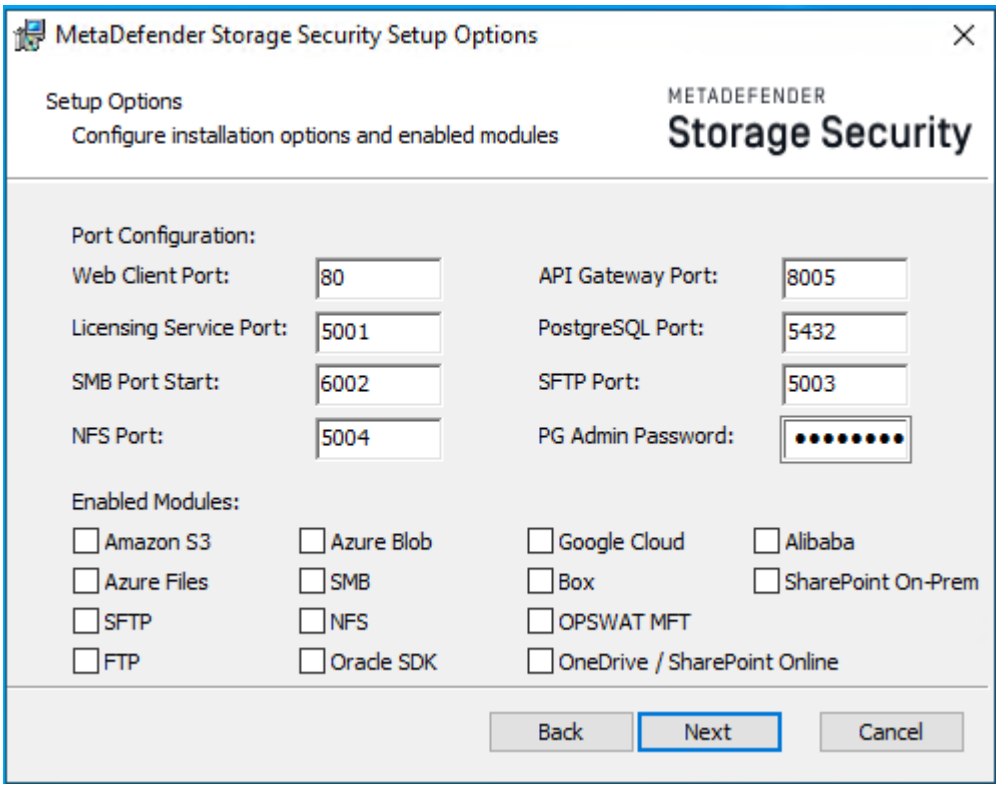
2. To proceed with the installation, read the License Agreement and check **I accept the terms in the License Agreement** if you accept it.



3. (Optional) **Configure Installation Options**
    1. Review the installation options presented in the setup wizard.
    2. Configure the **Port Configuration** section:
        1. Web Client Port: Default is 80
        2. API Gateway Port: Default is 8005

3. Licensing Service Port: Default is 5001
4. PostgreSQL Port: Default is 5432
5. SMB Port Range Start: Default is 6002. This is the starting port of a range. When set, 10 ports will be reserved: 5 consecutive ports starting from this value (e.g., 6002-6006) and another 5 consecutive ports offset by 1000 (e.g., 7002-7006).
6. SFTP Service Port: Default is 5003
7. NFS Service Port: Default is 5004

3. When installing MetaDefender Storage Security, you can choose which storage connection modules to enable. Each module allows integration with different storages. Select the desired **Enabled Modules** by checking the corresponding boxes
4. Click Next to proceed with the installation

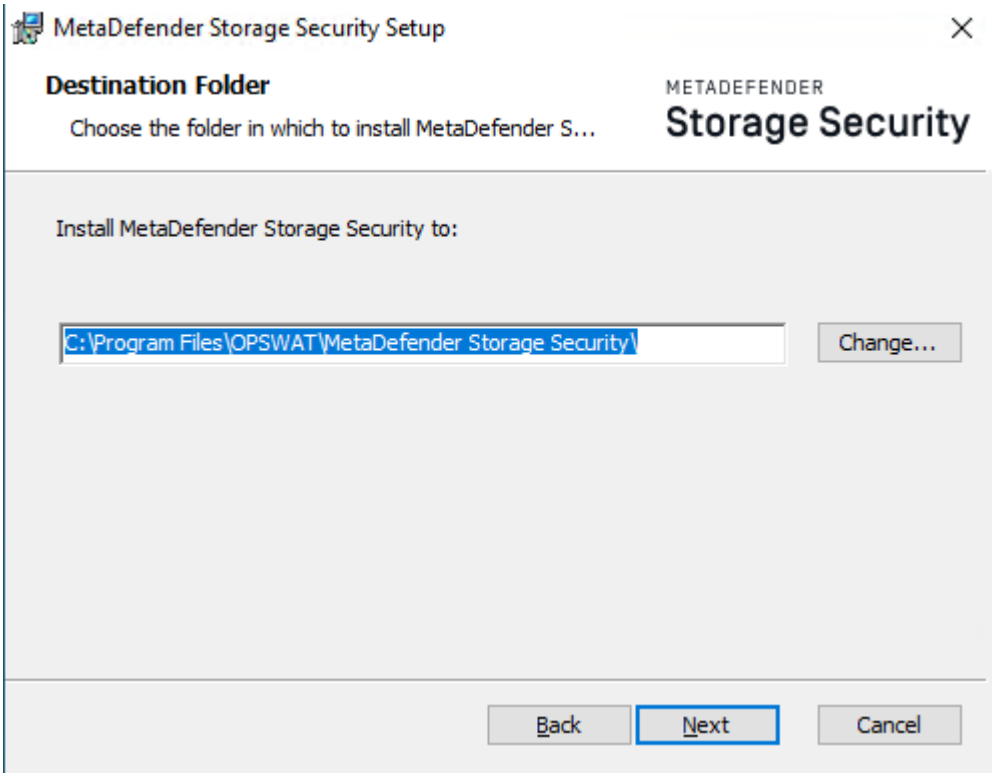


Info — SharePoint On-Prem Module

Enabling this module requires the installation of additional SharePoint Client Components. These will be installed automatically on the next system reboot, or immediately by running the following command from a PowerShell session with Administrator privileges, from the scripts folder, once the main installation has completed: `.\mdss.ps1 -c refresh`

Info

> Port configuration and additional modules can be added later by modifying the customer.env file. See: Custom configuration

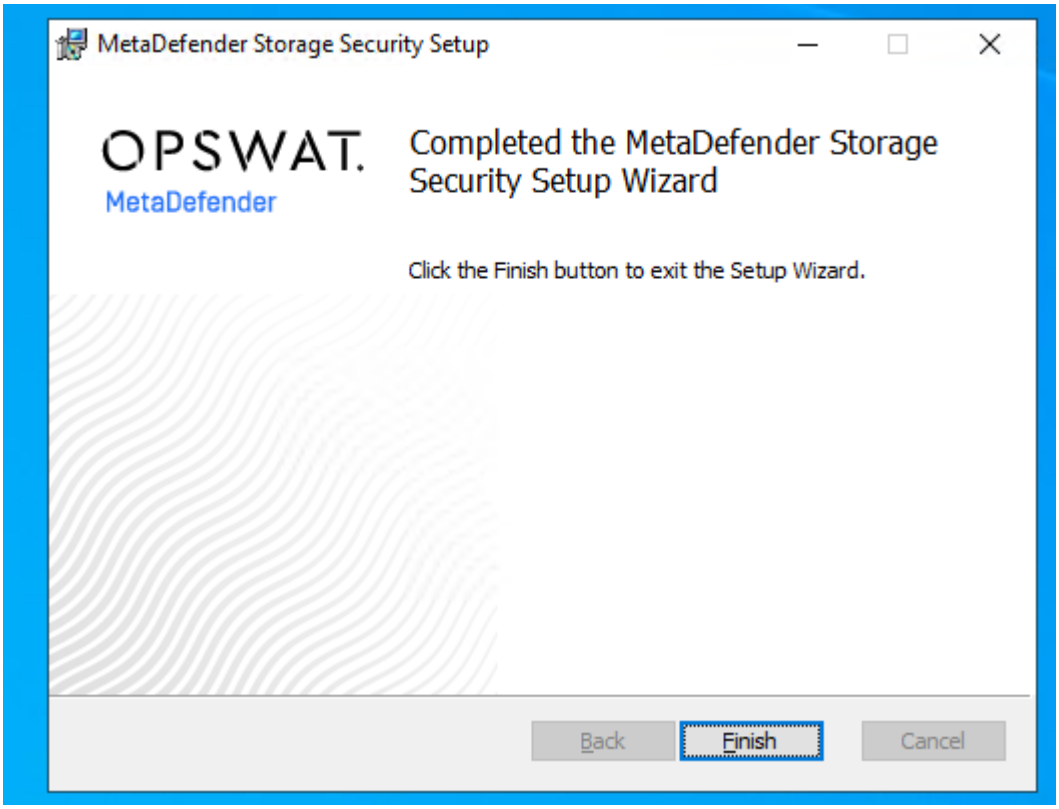


4. To choose another installation path click **Change** and then **Browse** to the desired location. Confirm the new location by clicking **OK**.
5. Click **Next**.
6. Click **Install**.
7. Please wait while the product is being configured.

Note: this operation may take more than 15 minutes to complete



8. Installation completed successfully.



9. Press **Finish.**

Next steps

Production deployments should follow the Production Considerations in order to set up a remote PostgreSQL database and enable HTTPS.

After that, please follow the setup in the newly opened browser window in order to create a local administrator and complete the installation.

Please see the onboarding page for more details.

Linux Installation

Preliminary notes

Before you begin the installation, please ensure that the system requirements are met.

If the MetaDefender Storage Security installer package dependencies are not installed on your system the installer will attempt to automatically download them over the internet.

Acquire the package

In order to acquire the package please visit https://www.opswat.com/solution

> **Info**
>
> To limit the required resources, use the `ENABLED_MODULES` and the `DISABLED_SERVICES` environment variables in the `customer.env` file to select which modules to install when deploying MetaDefender Storage Security, you can follow these steps:
>
> 1. Locate the `customer.env` file in your MetaDefender Storage Security deployment. This file contains environment variables used for customization.
> 2. Open the `customer.env` file in a text editor.
> 3. Add the `ENABLED_MODULES` environment variable to the file. The variable should be in the format `ENABLED_MODULES=<module_list>`, where `<module_list>` is a comma-separated list of the modules you want to enable. For example, if you want to enable modules Amazon S3, Google Cloud, and Azure Blob storage units, your `customer.env` file should include the following line: ENABLED_MODULES=azureblob,amazonsdk,googlecloud
> 4. Save the `customer.env` file.
> 5. Proceed with the deployment of MetaDefender Storage Security.
>
> Recommended Article: How to Enable Specific Modules in the Configuration?

Please note that an online installation involves downloading the required `docker` images from Docker Hub. If image pulls fail, you may have reached Docker Hub's rate limits, especially when multiple services pull from the same IP address.

To address this, please consider one of the following solutions:

- Use `docker login` to increase your pull limit
- Implement image caching or registry proxies
- Stagger deployments to avoid concurrent pulls

Advanced installation

We recommend that production deployments use an external database and ensure that web traffic is encrypted. For detailed information on how to configure an external database or HTTPS see advanced installation section:

- Production Consideration Linux Deployments
- Production Considerations Windows Deployments

Debian / Ubuntu package (.deb)

1. Update your packages

```
sudo apt update
```

2. Install the necessary package dependencies

```
sudo apt install jq curl gnupg-agent software-properties-common
```

3. Move to the folder containing the installer and install it

```
sudo dpkg -i <name of the file>.deb
```

4. After the installation is completed, check if everything went fine

```
man mdss
```

or using the help menu

```
sudo mdss -h
```

5. Run the following command in order to initialize MetaDefender Storage Security

```
sudo mdss -u init
```

6. Start the service

```
sudo mdss -c start
```

7. Optionally, you can check the status using the following command:

```
sudo mdss -c status
```

8. Open a browser and navigate to the web interface http:// to configure your deployment.

Red Hat Enterprise Linux / CentOS / Rocky package (.rpm)

> **Info**
>
> Please note that Docker Engine is officially supported only on RHEL s390x architecture, but you may be able to install it manually following the CentOS installation steps: https://docs.docker.com/engine/install/centos/
>
> On Rocky Linux, it is recommended to use Podman , as Podman is officially supported by Red Hat.

> **Info**
>
> On RHEL 8, the default iptables FORWARD chain policy is set to DROP. Podman's CNI networking does not automatically add forwarding rules for external interface to bridge traffic, which will cause the web interface to be unreachable from outside the host even if the correct ports are open in the firewall or security group.

In case the FORWARD chain policy is set to DROP, here are the commands to switch it to ACCEPT:

```
# External → container (new connections)
iptables -I FORWARD -i eth0 -o cni-podman1 -j ACCEPT

# Container → external (return traffic for established connections)
iptables -I FORWARD -i cni-podman1 -o eth0 -m state --state
RELATED,ESTABLISHED -j ACCEPT
```

1. Update your packages

```
sudo yum -y update
```

2. Move to the folder containing the installer and install it

```
sudo yum install -y <name of the file>.rpm
```

3. After the installation is completed, check if everything went fine

```
sudo man mdss
```

or using the help menu

```
sudo mdss -h
```

4. Run the following command in order to initialize MetaDefender Storage Security

```
sudo mdss -u init
```

Note: If you don't have docker already installed, the installer will do that for you, but you will be prompted to sign out and sign back in. This is necessary because your current user needs to be added to the docker group.

5. Start the service

```
sudo mdss -c start
```

6. Optionally, you can check the status using the following command:

```
sudo mdss -c status
```

7. Open a browser and navigate to the web interface http:// to configure your deployment.
8. If you are using firewall [on CentOS8 and RHEL 8] you may need to update it's policies:

```
# Check what interface docker is using, e.g. 'docker0'
ip link show

# Check available firewalld zones, e.g. 'public'
sudo firewall-cmd --get-active-zones

# Check what zone the docker interface it bound to, most likely 'no
zone' yet
sudo firewall-cmd --get-zone-of-interface=docker0

# So add the 'docker0' interface to the 'public' zone. Changes will be
visible only after firewalld reload
sudo nmcli connection modify docker0 connection.zone public

# Masquerading allows for docker ingress and egress (this is the juicy
bit)
sudo firewall-cmd --zone=public --add-masquerade --permanent
# Optional open required incomming ports (wasn't required in my tests)
# sudo firewall-cmd --zone=public --add-port=443/tcp
# Reload firewalld
sudo firewall-cmd --reload
# Reload dockerd
sudo systemctl restart docker
```

Podman Configuration

MDSS can also run using the Podman container engine instead of the default Docker one on RHEL operating systems. `Podman`, `podman-docker` and `docker-compose` need to be installed and running before installing MDSS.

```
# stop MDSS if is already installed
sudo mdss -c stop

# install podman, podman-docker and docker-compose
sudo yum install podman
sudo yum install podman-docker
sudo curl -SL
https://github.com/docker/compose/releases/download/v2.15.1/docker-
compose-linux-x86_64 -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose

sudo systemctl enable podman.socket
sudo systemctl start podman.socket

# to disable selinux permanetly 'SELINUX=disabled' needs to be set in
the following config file
sudo vim /etc/selinux/config

# reboot machine for changes take effect
sudo reboot
```

Once the Podman and Docker components are installed, proceed with the installation of MetaDefender Storage Security (MDSS). You can skip the installation if MDSS is already installed.

After installation, MDSS must be configured to skip the default Docker engine checks. This is achieved by setting `SKIP_DOCKER_CHECKS=yes` in the `/etc/mdss/customer.env` configuration file.

```
# install additional MDSS dependencies (skip this if MDSS is already
installed)
sudo subscription-manager repos --enable codeready-builder-for-rhel-
8-$(arch)-rpms
sudo yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-
latest-8.noarch.rpm
sudo yum install lvm2
sudo yum install wget

# install MDSS (skip this if MDSS is already installed)
sudo rpm -i mdss-x.x.x-1.noarch.rpm

# edit customer.env and add SKIP_DOCKER_CHECKS=yes
sudo vim /etc/mdss/customer.env

# start MDSS
sudo mdss -c start
```

Configure Automatic Service Startup at Boot on RHEL and Rocky Linux with Podman

Due to a technical limitation, on **RHEL and Rocky Linux**, MDSS will not start automatically after machine restarts. To enable this option, please follow these steps:

1. Create a service file

```
sudo vi /etc/systemd/system/mdss.service
```

2. Paste the following content

```
[Unit]
    Description=MDSS Stack
    After=network-online.target
    Wants=network-online.target

    [Service]
    Type=oneshot
    RemainAfterExit=yes
    ExecStart=mdss -c start

    [Install]
    WantedBy=multi-user.target
```

3. Save and exit the file (Esc + :wq!)
4. Enable the service

```
sudo systemctl daemon-reload
sudo systemctl enable mdss.service
```

That's it. To confirm the changes were applied successfully, please restart your VM and then run **sudo mdss -c status** to make sure MDSS is up and running after reboot.

Linux Offline Installation

This guide provides detailed instructions for installing MetaDefender Storage Security and its dependencies on machines with limited or no internet connectivity. A machine connected to the internet will be needed to download the dependencies manually and copy them to the offline machine.

Installing Binary Packages

You will need to download and install the following binary packages for your respective operating system:

Debian/Ubuntu

```
curl, openssl, sudo, apt-utils, ca-certificates, jq, gnupg-agent,
iproute2, apt-transport-https,  software-properties-common
```

RedHat/CentOS

```
curl, openssl, sudo, yum-utils, ca-certificates, jq, gnupg2, iproute,
lvm2, wget, device-mapper-persistent-data
```

> **Info**
>
> **Note:** For detailed instructions, refer to the guide for Debian/Ubuntu or CentOS/RHEL

Option 1: Using installers

This option uses the the official installation methods for docker and docker-compose to create an offline docker environment where the MDSS installer from the MyOPSWAT portal can be used to deploy MDSS containers. If for some reason this method cannot be used on the target deployment machine then skip to **Option 2** below.

Installing Docker and Docker Compose

To be able to install the dependencies offline, you need a machine connected to the internet to download the required packages. Later, the packages will be copied to the offline machine.

To download the dependencies, follow the official guide for Docker Engine and Docker Compose

The compose plugin for docker can be installed only manually when in an offline environment using the following commands (also available in the official instructions in the link above):

```
mkdir -p /usr/local/lib/docker/cli-plugins/
curl -SL
https://github.com/docker/compose/releases/download/v2.28.0/docker-
compose-linux-x86_64 -o /usr/local/lib/docker/cli-plugins/docker-
compose

sudo chmod +x /usr/local/lib/docker/cli-plugins/docker-compose
```

Installing MetaDefender Storage Security package

After the prerequisites above have been set up, you can run the MetaDefender Storage Security installer to setup the compose files and the mdss command line.

Debian/Ubuntu

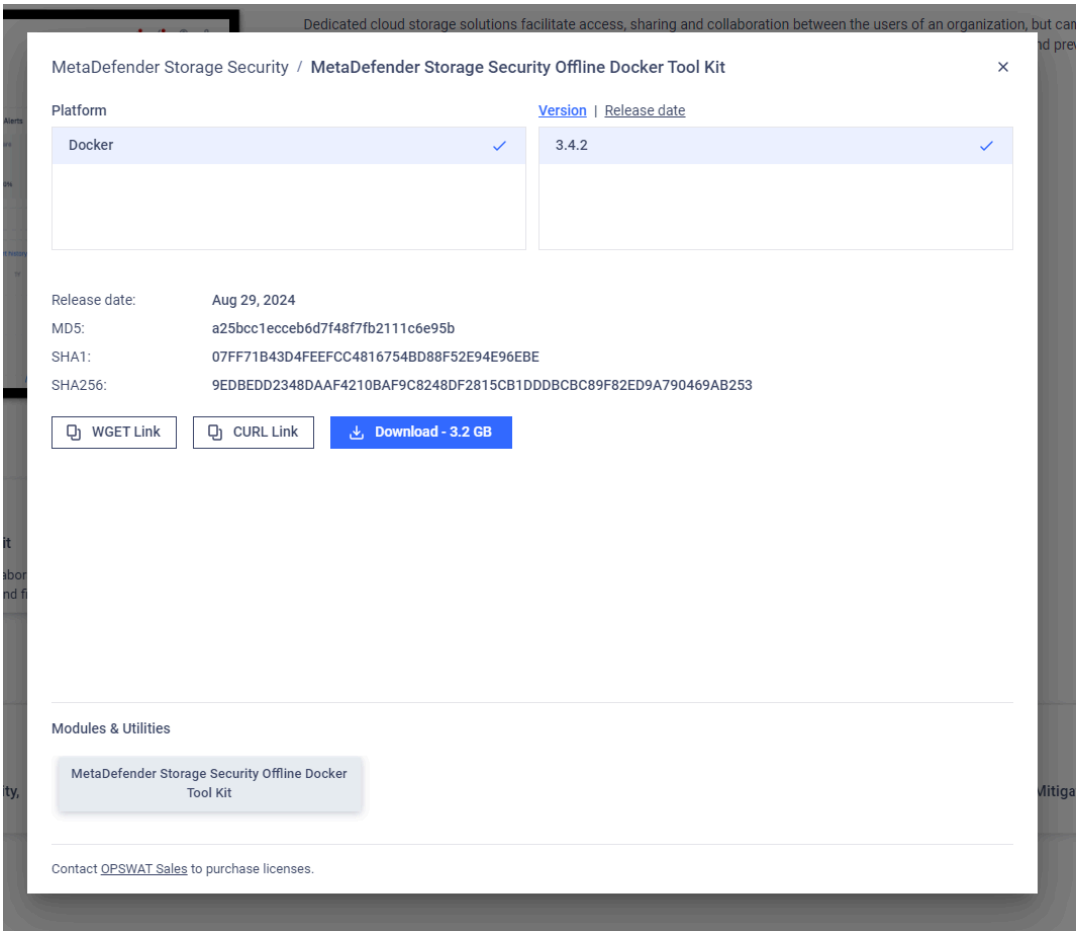
```
sudo dpkg -i <name of the file>.deb || sudo apt-get install -f
```

RedHat/CentOS

```
sudo yum install -y <name of the file>.rpm
```

Loading Docker images

MetaDefender Storage Security is composed of several Docker images that need to be pulled from an online repository. For offline installations, these images are available on the MyOPSWAT portal under **MetaDefender Storage Security** in the **MetaDefender Storage Security Offline Docker Toolkit** section:



Next, you need to extract the `mdss.tar` archive and transfer it to the offline machine. Once the file is on the offline machine, use the following command to load the Docker images:

```
docker load -i <archive_path>\mdss.tar
```

Starting MetaDefender Storage Security

Finally, MDSS can be started using the regular startup command:

```
sudo mdss -c start
```

Option 2: using bundled binaries

Using this method you will only need to download the same MetaDefender Storage Security Offline Docker Toolkit as above without needing to pre-install docker and docker-compose or use the MDSS installer. It's still recommended to have all prerequisites installed before continuing.

Download and extract the offline toolkit

The offline toolkit is available on the **MyOPSWAT** portal under **MetaDefender Storage Security** in the **MetaDefender Storage Security Offline Docker Toolkit** section. The package needs to be extracted to the desired path on the target offline machine and the following steps should all be performed from the same path where the package is extracted on the offline machine.

Load docker images

To load the MDSS docker images run the following command from the same path where the offline package is extracted:

```
sudo ./mdss.sh -u load_images
```

Starting MetaDefender Storage Security

Finally, MDSS can be started using the regular startup command but specifying the local path to the `mdss.sh` script from the offline package:

```
sudo ./mdss.sh -c start
```

Using this method, all mdss commands (start, stop etc.) will need to be executed from the same path where the package is extracted.

Updating MetaDefender Storage Security Offline

> **Info**
>
> To update MetaDefender Storage Security while in offline mode the same steps can be followed but with the latest installer and offline package, detailed instructions are outlined in the Offline Update Guide

Portable Package

This guide provides instructions on how to deploy MetaDefender Storage Security docker images without using the the installer. This method is intended for deployment scenarios where installing docker and docker-compose is not possible using official installation methods.

> **Info**

> It's recommended to have the same dependencies as the ones required for the offline installation preinstalled on the target host machine.

Downloading the portable toolkit

The toolkit is available on the **MyOPSWAT** portal under **MetaDefender Storage Security** in the **MetaDefender Storage Security Portable Toolkit** section. The package needs to be extracted to the desired path on the target machine and the following steps should all be performed from the same path where the package is extracted on the target machine.

Starting MetaDefender Storage Security

From the path where the portable package is extracted, MDSS can be started using the regular startup command but specifying the local path to the `mdss.sh` script:

```
sudo ./mdss.sh -c start
```

Using this method, all mdss commands (start, stop etc.) will need to be executed from the same path where the package is extracted.

> **Info**
>
> When running in portable mode, the `mdss.sh` script detects if docker is already installed or if the bundled docker binaries should be used in which case the following should be taken into consideration:
>
> - Docker commands will not work from the command line by default unless using the docker binarys directly from the bundled `docker` folder.
> - The bundled docker binary is not running as a service, the `dockerd` process is started automatically when running `mdss` commands.
> - Since docker is not running as a service it does not automatically start at boot

Upgrade MDSS

> **Warning**
>
> Before proceeding with the upgrade process, it's highly recommended that a backup is performed of the original deployment in order for recovery purposes.

> The contents of the `customer.env` file (contains credentials and custom settings) and the PostgreSQL database files (contain all MDSS data) are critical in order to prevent any data loss.

Upgrading is typically a two-step process:

1. Upgrade the **mdss** package by installing the newer version.
2. Update the codebase by downloading and applying new container images.

> **NOTE**
>
> > The behavior of the update process depends on the current version:
>
> - **Before v3.5.3:** Can only upgrade to v3.5.3
> - **From v3.5.3:** Can upgrade to version **v3.6.0** for database validation
> - **From v3.6.0**: Can upgrade to the latest compatible version
>
> > ⚠️ MDSS v3.6.0+ requires **MongoDB 6.0 or higher**
> >
> > ⚠️ MDSS v4.0.0+ requires **PostgreSQL 16.0 or higher**

Upgrading to version 4.0.0 or newer

Starting with version 4.0.0 of MDSS the database will be migrated from MongoDB to PostgreSQL. Please follow the Upgrade Guide to PostgreSQL to ensure the upgrade completes successfully and all data is migrated.

> ## Unix-Based Systems
>
> ### Step 1: Download the Installer
>
> Visit My OPSWAT to download the latest `.deb` or `.rpm` package.

> **Warning**
>
> > If you have overridden `ENABLED_MODULES` in `/etc/mdss/customer.env`, review the list before upgrading. Module names must match the ones in `/etc/mdss/.env` of the new version: `onedrive` was renamed to `graph` in 4.0.0

> [OneDrive and SharePoint Online], and newer modules such as `mft`, `ftp` and `oraclesdk` are only enabled if they are in your list. Names are case-sensitive and must be separated by commas without spaces.
>
> Starting with 4.6.0, unknown module names are ignored and the corresponding integrations are not available after the upgrade. Versions 4.0.0 to 4.5.2 stop instead with an error such as `open /etc/mdss/docker-compose.onedrive.yml: no such file or directory`. If you do not need a custom list, remove the line so the default from `.env` applies.

> **Offline Deployments**
>
> In case of an offline deployment, the offline steps need to be followed: Offline Upgrade

Debian / Ubuntu package (.deb)

1. Update your packages

```
sudo apt-get update
```

2. Move to the folder containing the installer and install it

```
sudo dpkg -i <name of the file>.deb || sudo apt-get install -f
```

3. Start the service

```
sudo mdss -u update
```

4. Optionally, you can check the status using the following command:

```
sudo mdss -c status
```

5. Open a browser and navigate to the web interface: **http://** to configure your deployment.

Red Hat Enterprise Linux / CentOS package (.rpm)

1. Update your packages

```
sudo yum -y update
```

2. Move to the folder containing the installer and install it

```
sudo yum upgrade <name of the file>.rpm
```

3. Start the service

```
sudo mdss -u update
```

4. Optionally, you can check the status using the following command:

```
sudo mdss -c status
```

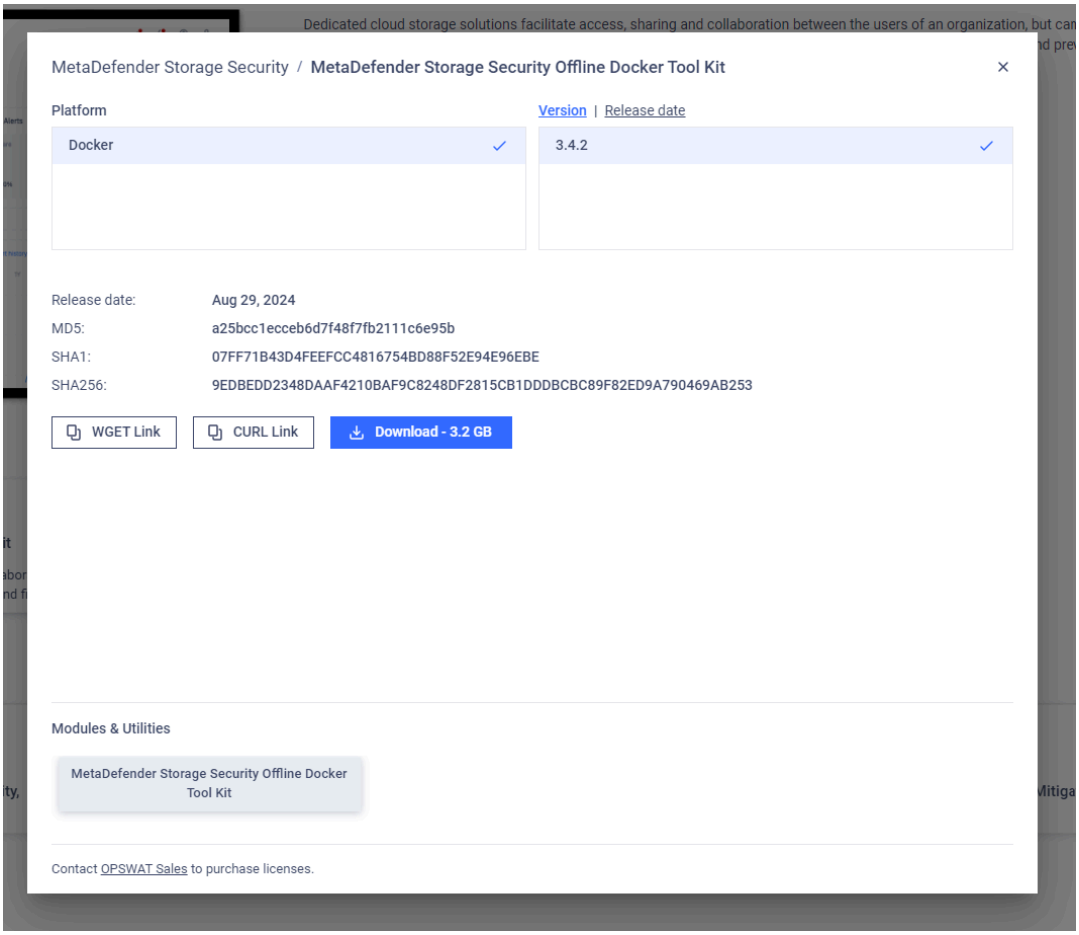
5. Open a browser and navigate to the web interface: http:// to configure your deployment.

Offline Upgrade

> **Warning**
>
> The following steps only apply when MDSS is installed using the official installer (.deb/.rpm). For any other deployment methods like when using the portable package, bundled binaries or the helm chart, then you should consider a clean deployment and migrating the original data if necessary.

> **Info**
>
> Instructions to be followed for upgrading after installing the product using the Installing Offline on Linux documentation

1. Download the latest MetaDefender Storage Security **Installation Package** from My OPSWAT portal.
2. Download the latest MetaDefender Storage Security **Offline Docker Tool Kit** from My OPSWAT portal.



3. Stop the service

```
sudo mdss -c stop
```

4. Run the installation script

```
sudo dpkg -i <name of the file>.deb
```

```
sudo yum install -y <name of the file>.rpm
```

5. Extract deployment.zip folder and load the container images

```
unzip deployment.zip -d <extracting_path>/
cd <extracting_path>/
sudo docker load -i ./mdss.tar
```

6. Start the service

```
sudo mdss -c start
```

Upgrading from a previous version on Windows-based deployments

Upgrading from v3.4.0 or newer

To upgrade from a previous version, running the new installer is enough. The installer will walk you through the Windows Installation Wizard Section

Upgrading from v3.3.4 or earlier to v3.4.0 or newer

Upgrading to v.3.4.0 from an older version requires a special procedure due to major architectural changes. If a clean install with configuration import is not possible, please see the steps below.

Caution: This operation modifies a live database. Errors may cause irreversible data loss. Proceed with caution and ensure adequate backups. We advise on performing this operation only if absolutely necessary.

All commands should be run in a PowerShell terminal.

1. Update MDSS to the latest version

```
.\mdss.ps1 -u update
```

2. Edit the following file: C:\Program Files\OPSWAT\MetaDefender Storage Security\docker\docker-compose.main.linux.yml
3. Under the mongodb container modify the image to use version 6.0. It should look like this:

```
version: '3'
services:
    mongodb:
        platform: linux/amd64
        image: "mongo:6.0"
        volumes:
            - mongo_db:/data/db
```

4. Repeat procedure for C:\Program Files\OPSWAT\MetaDefender Storage Security\docker\docker-compose.main.all.linux.yml
5. Run the upgrade command from the installation directory (C:\Program Files\OPSWAT\MetaDefender Storage Security\docker):

```
.\mdss.ps1 -u upgrade_db
```

Depending on your DB size, this operation may take more than 30 minutes

6. After upgrade is finished, export the DB

```
.\mdss.ps1 -u export_db
```

7. Copy the exported DB archive to a different location, e.g.: C:\temp
8. In that different location, extract the archive

```
tar -xf .\db_{timestamp}.tar
```

9. Compress the files in a zip format

```
Compress-Archive .\dump\  db_{timestamp}.zip
```

10. Uninstall MetaDefender Storage Security 3.3.4 as documented Uninstall.
11. Run the latest installer for MetaDefender Storage Security as documented Installation Methods
12. Run the stop command from the installation directory (C:\Program Files\OPSWAT\MetaDefender Storage Security\scripts)

```
.\mdss.ps1 -c stop
```

13. Copy the .zip file you created earlier (db__{timestamp}_.zip) to the installation directory (by default, (C:\Program Files\OPSWAT\MetaDefender Storage Security)
14. Import the DB

```
.\mdss.ps1 -u import_db
```

15. Run the start command:

```
.\mdss.ps1 -c start
```

Uninstall

NOTE

> If the current MetaDefender Storage Security (MDSS) deployment (4.0.0 or later) was upgraded from 3.6.1 or earlier, any existing MongoDB backups will be deleted during uninstallation. Manually back up the data beforehand if it's still needed.

Uninstall via Command Line

Debian / Ubuntu (.deb package)

To remove all files installed by MetaDefender Storage Security:

```
sudo mdss -u clean
```

To remove the `mdss` package from the system:

```
sudo apt-get remove mdss
```

Red Hat Enterprise Linux / CentOS (.rpm package)

To remove all files installed by MetaDefender Storage Security

```
sudo mdss -u clean
```

To remove the `mdss` package from the system:

```
sudo yum remove mdss
```

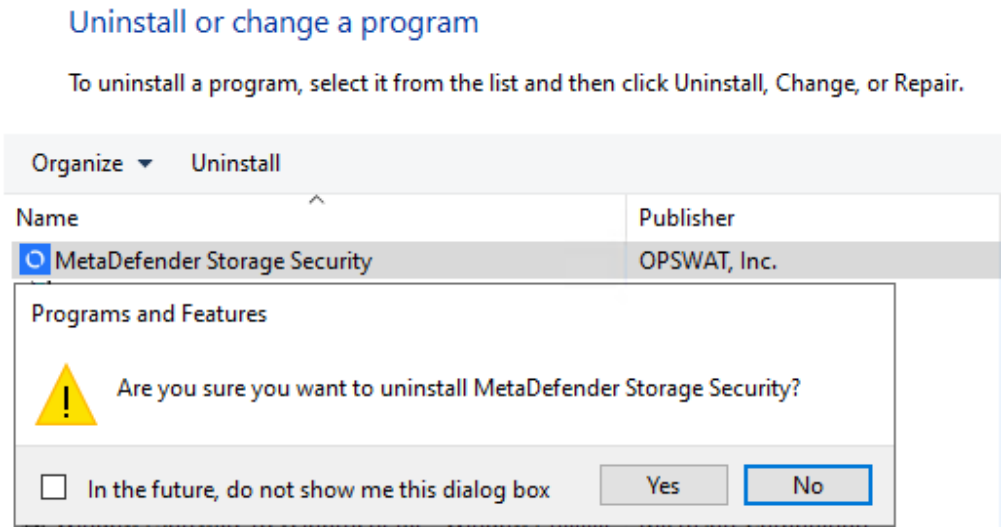
Uninstall via Uninstall Wizard (Windows only)

> **NOTE**
>
> The Uninstall Wizard is available only for the Windows installer [`.msi`].

Step-by-Step Instructions

1. Open **Start** → **Control Panel** → **Uninstall a program**
2. Locate **MetaDefender Storage Security** in the list
3. Double-click the entry
4. Click **Uninstall and Yes** to begin the process

n or



5. Wait for the uninstallation to complete
6. Click **Finish** to finish.

Troubleshooting On-Premise

To troubleshoot when deploying to your On-Premise environment we have created a specific section, where we have divided the articles in:

- Diagnostic
- Deployment
- Integration

containerized deployments

Containerized Deployment Overview

This section provides comprehensive guides for deploying and managing MetaDefender Storage Security in both Docker and Kubernetes environments. Containerization offers significant advantages for flexibility, scalability, and efficiency in managing your security infrastructure.

Docker

Deploying MetaDefender Storage Security with Docker provides a lightweight and portable solution. This guide covers:

- Overview of the MetaDefender Storage Security Docker images published on OPSWAT Docker Hub.

Kubernetes

Leveraging Kubernetes for deploying MetaDefender Storage Security offers enhanced scalability, high availability, and simplified management. This guide covers essential aspects, including:

- Overview of Kubernetes components and their roles in the deployment.
- Upgrade and deployment strategies for seamless transitions and updates.
- Step-by-step instructions for installing MetaDefender Storage Security within your Kubernetes cluster.
- Licensing information specific to Kubernetes deployments.
- Storage configuration options to optimize performance and data management.
- High availability configurations for ensuring continuous operation.
- Guidance on scaling your Kubernetes cluster to adapt to changing demands.
- Troubleshooting tips for common issues and challenges.
- Release information and updates for MetaDefender Storage Security on Kubernetes.

By following these guides, you can effectively deploy and manage MetaDefender Storage Security in containerized environments, enhancing your security posture and ensuring the protection of your critical data.

Kubernetes Deployment

This Installation Guide provides step-by-step instructions for installing MetaDefender Storage Security (MDSS) in your own Kubernetes cluster in whatever environment you have

it deployed.

This guide is for IT infrastructure architects, administrators and DevOps professionals who are seeking to prevent potential malicious traffic being allowed in their environments. Threat Prevention is ensured for both productivity files that might be uploaded and for known vulnerabilities that can be identified for all running services/applications. The vulnerability scanning is checking known vulnerabilities for unpatched OS and running applications.

Using Kubernetes technology provides us with some features that distinct it from other type of deployments. These are the main features:

- **Service discovery and load balancing.**
- **Storage orchestration.** Allows you to automatically mount a storage system.
- **Automated rollouts and rollbacks.**
- **Automatic bin packing.** You tell Kubernetes how much CPU and memory (RAM) each container needs.
- **Self-healing.** Kubernetes restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.
- **Secret and configuration management.** Kubernetes lets you store and manage sensitive information.
- Scale your application up and down

There are recommended configuration for each of the features to use by default when deploying MetaDefender Core in a Kubernetes cluster.

Kubernetes Guide Structure

1. Deployment Using Helm
2. Replication & High Availability on K8S
3. Custom certificates on K8S
4. OpenShift Considerations

Deployment Using Helm

The Kubernetes deployment for MetaDefender Storage Security can be performed directly using the provided Helm chart, which is available in our public GitHub repository here.

Example configuration files for different environments are provided in the `helm_charts` directory within the repository.

Using the helm repository

The GitHub repository can be used directly as a Helm repository:

```
helm repo add mdk8s https://opswat.github.io/metadefender-k8s/
helm repo update mdk8s

#Example installation command
helm install mdss mdk8s/metadefender_storage_security -f
<CUSTOM_VALUES_FILE.yml>
```

Alternatively, the repository can be cloned locally:

```
git clone https://github.com/OPSWAT/metadefender-k8s.git metadefender
cd metadefender/helm_carts

#Example installation command
helm install mdss ./mdss -f <CUSTOM_VALUES_FILE.yml>
```

Storage

MDSS containers are stateless and do not require persistent storage.

However, if deploying the PostgreSQL database within the cluster, it is recommended to use persistent storage managed by your cloud provider.

An external database can be configured using the following values:

```
mdss-common-environment:
  POSTGRESQL_URL:
"Host=postgres;Port=5432;Username=user;Password=password;Database=dbNam
e"
```

When using an external database not deployed via the MDSS chart, the `deploy_with_mdss_db` value must be set to `false` to prevent the chart from deploying an additional database.

Exposing MDSS

By default, the Helm chart deploys a `ClusterIP` service for the MDSS deployment.

This can be changed to any service type supported by your Kubernetes cluster.

For example, to create a `LoadBalancer` service, overwrite the `service_type` value in the `webclient` component:

```
mdss_components:
  webclient:
    service_type: LoadBalancer
```

MDSS can also be exposed using an Ingress:

```
mdss_ingress:
  host: <APP_NAMESPACE>-mdss.k8s  # Hostname for the publicly
accessible ingress, the `<APP_NAMESPACE>` string will be replaced with
the namespace where the chart is deployed
  enabled: true                   # Enable or disable the ingress
creation
  class: nginx                    # Sets the ingress class depending
on the installed ingress controller
```

Flexible deployment

By default, the Helm chart deploys MDSS with support for the following storage units: `azureblob`, `amazonsdk`, `googlecloud`, `alibabacloud`, `azurefiles`, and `box`.

For more efficient resource utilization, you can specify only the required storage units by modifying the `ENABLED_MODULES` value.

For instance, to enable support for only Azure, AWS, and GCP:

```
mdss-common-environment:
  ENABLED_MODULES: "azureblob,azurefiles,amazonsdk,googlecloud"
```

Supported modules

Currently supported modules include:

- `azureblob`
- `amazonsdk`
- `googlecloud`
- `alibabacloud`
- `azurefiles`
- `smb`
- `box`
- `graph`

Replication & High Availability on K8S

When installing MetaDefender Storage Security in Kubernetes as this technology provide the system with its self-healing feature, it might not be strictly necessary to set it up with at least 2 replicas for each of the pods in case of wanting to have **High Availability,** but it is always a best practice to have 2 replicas of each pod running in different worker nodes. For those components that could not be running within the cluster, we will provide different solutions that are supported by our application.

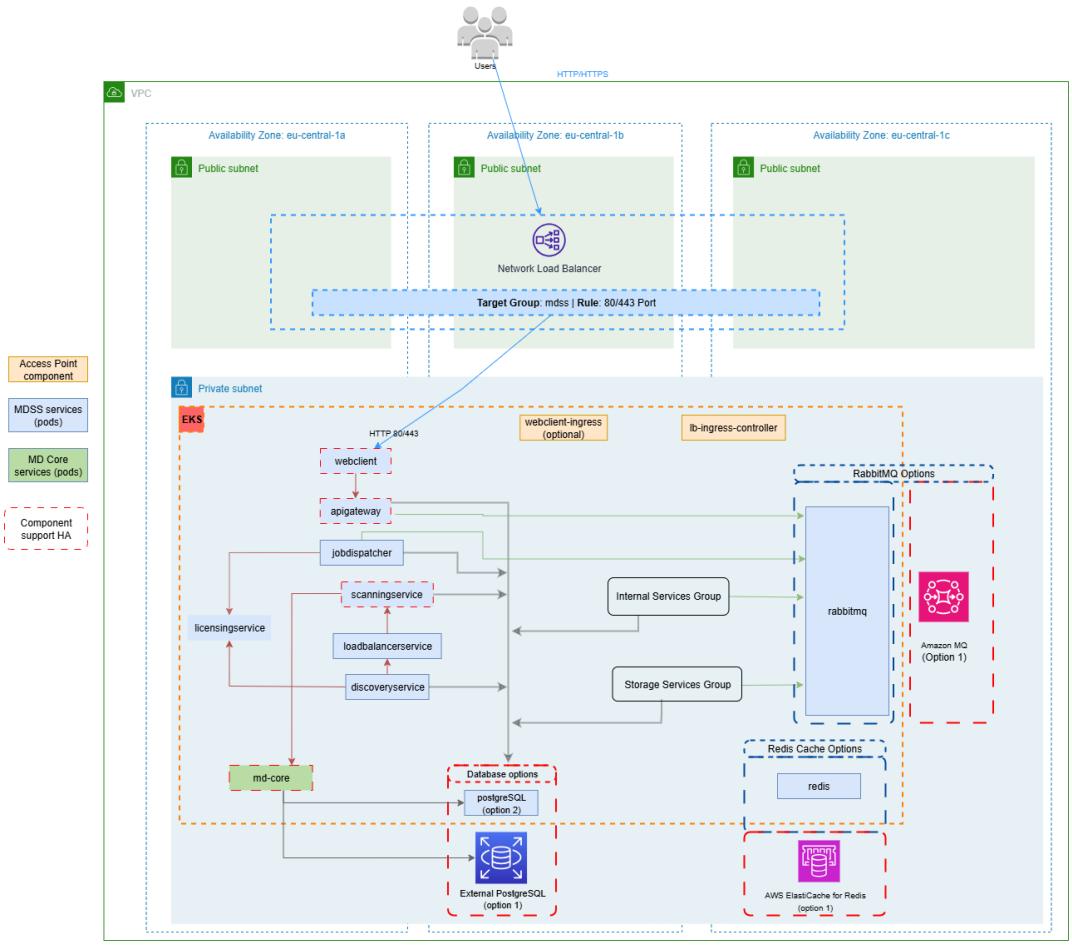
However, there are a few components that may incur in high load in some situations, so for those components it is recommended to have more than 1 replica. These components are the following.

MDSS services

- `Web Client` -> to provide high availability to outside requests
- `API Gateway` -> to provide high availability to outside requests
- `Scanning Service` -> to provide high availability when having to handle big number of files to scan

3rd Party components

- `Database` -> Deploy an external service for PostgreSQL with HA or PostgreSQL operator running on K8S cluster
- `RabbitMQ` -> Use external service for RabbitMQ with HA
- `Redis Cache` -> Use external service for Redis cache with HA



HA solutions for PostgreSQL

For CSPs provided service it has been tested and is supported by the following:

- Amazon RDS
- Azure Database for PostgreSQL

If a highly available database is required inside the k8s cluster, then there are publicly available solutions that can deploy a PostgreSQL Replica Set. For a k8s cluster, Zalando provides an operator for deploying a Replica Set: https://github.com/zalando/postgres-operator/blob/master/docs/quickstart.md

HA Solution for Redis Cache

OPSWAT have tested the following services to provided HA to a redis service.

- AWS ElasticCache for Redis
- Azure Cache for Redis

HA Solution for RabbitMQ

OPSWAT have tested the following services to provided high availability to a RabbitMQ service.

- Amazon MQ for RabbitMQ
- Bitnami package for RabbitMQ in Azure MP

RabbitMQ Cluster Operator

1. Install Cert-Manager (required for operator webhooks)

```
kubectl apply -f https://github.com/cert-manager/cert-
manager/releases/download/v1.13.3/cert-manager.yaml
```

2. Install RabbitMQ Cluster Operator:

You can download the file using this link: RabbitMQ Cluster Operator

```
kubectl apply -f <path-to-cluster-operator.yml>
```

3. Install RabbitMQ Messaging Topology Operator:

You can download the file using this link: RabbitMQ Messaging Topology Operator

```
kubectl apply -f <path-to-messaging-topology-operator-with-
certmanager.yaml>
```

Update your Helm values to enable the operator:

```
rabbitmq-cluster-operator:
  enabled: true
  replicas: 3
  resources:
    requests:
      cpu: "250m"
      memory: "512Mi"
```

The operator will automatically create a 3-node RabbitMQ cluster with quorum-based HA. During node failures, the cluster continues operating and services automatically reconnect to healthy nodes.

HA deployment for MDSS

MDSS containers can be scaled independently depending on the availability and performance requirements. For example, just the `webclient`and `apigateway` pods can be replicated to provide high availability to outside requests and for the web interface.

```
mdss_components:
  webclient:
    replicas: 3
  apigateway:
    replicas: 3
```

The `mongomigrations`service is used to migrate the data from older versions of MDSS at startup if required and provides no benefits if replicated .

Create environment with HA components

For deploying all the components that provide a high availability, OPSWAT have prepared a terraform module to deploy all the 3rd parties applications (Redis, RabbitMQ & PostgreSQL DB).

There are 2 ways of using that terraform module

- Applying changes directly using the terraform project
- Using the metadefenderk8s.sh to deploy MDSS with it.

Automatic replicas using horizontal pod autoscaling (HPA)

The HPA can be enabled on all MDSS services from the values:

```
hpa:
  enabled: true
  minReplicas: 2
  maxReplicas: 3
  cpuTargetUtilization: 85
```

In the example above the number of replicas are adjusted within the specified limits depending on the measured CPU usage. The HPA is applied for all MDSS services, for a more granular autoscaling it's recommended to manually create a HPA separately from the helm deployment just for the desired services.

Custom certificates on K8S

Custom certificates need to be loaded in MDSS when connecting to services that use a private TLS/SSL certificate like MD Core, storage units, external databases etc. These certificates need to be loaded into the K8S cluster as a secret which can be mounted into the appropriate services by adding the following section to the values file:

```
mdss_components:
  <SERVICE_NAME>:
    update_ca: true
    customMounts:
      - mountPath: "/usr/local/share/ca-certificates/my-custom-
cert.crt"
        name: my-cert
        subPath: cert-data
    customVolumes:
      - name: my-cert
        secret:
          secretName: my-custom-cert
```

An example values file where certificates are loaded into multiple MDSS services can be found on our GitHub repo here: https://github.com/OPSWAT/metadefender-k8s/blob/main/helm_charts/mdss-custom-certs-example.yml

OpenShift Considerations

Introduction

This guide explains how to deploy MetaDefender Storage Security (MDSS) on OpenShift using Helm with the required OpenShift-specific configuration.

Prerequisites

- Access to OpenShift cluster with admin privileges
- Helm 3.x installed
- `oc` command-line tool configured
- Ability to grant privileged SCC permissions

Deployment Steps

1. Login to OpenShift

```
oc login --token=<your_token> --server=<cluster_url>
```

2. Create Project

```
oc new-project mdss
```

3. Create Service Account and Grant Permissions

The MDSS components require privileged access to function properly on OpenShift:

```
# Create a service account for MDSS
oc create sa mdss -n mdss

# Grant privileged security context constraints to the service account
oc adm policy add-scc-to-user privileged -z mdss -n mdss
```

Important: The privileged SCC is necessary because several MDSS components require elevated permissions.

4. Add Helm Repository

```
helm repo add mdk8s <https://opswat.github.io/metadefender-k8s/>
helm repo update mdk8s
```

5. Prepare OpenShift Values File

Create a file named `mdss-openshift.yml` with the OpenShift-specific configuration:

```
# Download the OpenShift values file directly from GitHub # Run this
command to fetch the file:
curl -O <https://raw.githubusercontent.com/OPSWAT/metadefender-
k8s/refs/heads/main/helm_charts/mdss-openshift.yml
```

6. Deploy the MDSS Helm Chart

```
helm install mdss mdk8s/metadefender_storage_security -f mdss-
openshift.yml
```

Note: The OpenShift values file configures the SecurityContext to run as user 0 (root), assigns the mdss service account to pods, and adjusts command arguments for

compatibility.

7. Verify Deployment

```
oc get pods -n mdss
oc get services -n mdss
```

Wait until all pods are in Running state. This may take a few minutes as containers initialize.

Post-Deployment Configuration

Access MDSS Console

```
# Create a route to access the MDSS web interface
oc expose svc/webclient -n mdss
# Get the route URL
oc get route webclient -n mdss
```

Access the web interface using the hostname provided by the route.

Maintenance

Upgrading MDSS

```
helm repo update mdk8s helm upgrade mdss
mdk8s/metadefender_storage_security -f mdss-openshift.yml
```

Uninstalling MDSS

```
helm uninstall mdss oc delete project mdss
```

Upgrade/Deployment Strategy on K8S

The helm upgrade command can be used to upgrade the mdss services using the latest helm chart:

```
helm upgrade my_mdss <path_to_chart>
```

General recommendations

Before performing any upgrade, review the following best practices:

- **Back up the database**: Always take a full backup of your PostgreSQL database before upgrading. For in-cluster deployments this means snapshotting the persistent volume; for externally managed databases use the provider's native backup/snapshot mechanism. Keep the backup until the new version has been validated in production.
- **Read the release notes**: Check for any breaking changes or required configuration updates in the target release before upgrading.
- **Blue-green deployment**: For production environments where rollback safety is critical, consider a blue-green upgrade strategy:
  1. Clone your existing PostgreSQL database to a separate instance (e.g. a snapshot-restored replica or a managed DB clone).
  2. Deploy the new chart version into a separate namespace (e.g. `mdss-green`), pointing it at the cloned database:`helm install my_mdss_green path_to_new_chart --namespace mdss-green --set POSTGRESQL_URL="connection_string_to_cloned_db"`
  3. Validate functionality of the new deployment against the cloned data without affecting production traffic.
  4. Switch ingress or DNS to the new namespace once satisfied.
  5. If issues arise, revert by routing traffic back to the original namespace — the original database remains untouched.
  6. Decommission the old deployment and cloned database after a suitable validation period.

Database upgrades

This step is not required when using an external, managed database

The helm chart is configured by default to use the latest compatible version of PostgreSQL. Before upgrading an existing deployment with a persistent in-cluster database, make sure the database version matches the version required by the corresponding release:

- MDSS (latest) - PostgreSQL 17

Upgrading from MongoDB (versions prior to v4.0.0)

Starting with MDSS v4.0.0, PostgreSQL replaces MongoDB as the primary database backend. If upgrading from an older MongoDB-based deployment, follow the official Upgrade Guide to PostgreSQL which covers three upgrade options:

- **Option 1 – Standard upgrade with migration**: Recommended for small to medium deployments where historical data is required. Data is migrated automatically; no extra configuration needed. Note that migration takes approximately 2–3 hours per 1 million files.
- **Option 2 – Skip data migration**: Recommended for large deployments or when historical data is not critical. The chart sets `SKIP_MONGO_TO_PG: "yes"` by default, so migration is skipped unless explicitly removed from the values. This provides immediate availability of the new version without waiting for data migration.
- **Option 3 – Blue-green deployment**: Recommended for production environments requiring zero downtime. Deploy v4.0.0 in parallel, test it, then cut over storage processing with easy rollback capability.

Schema migrations

Database schema migrations are handled automatically by the `pgmigrations` pod, which runs as an init step before any other MDSS containers start up. No manual migration steps are required — ensure the `pgmigrations` pod completes successfully before troubleshooting other service startup issues.

Upgrading with an externally managed database

When using an external (managed) database, the database itself holds all persistent state. In this case it is **strongly recommended to perform a clean reinstall** rather than a rolling upgrade, to avoid schema or configuration conflicts:

```
helm delete my_mdss
helm install my_mdss <path_to_chart>
```

The following components are non-persistent and can be updated to the latest compatible version by setting the respective image tag:

- RabbitMQ: `rabbitmq:4.1.0-alpine`
- Redis Cache: `redis:7.4.1-alpine`

Operational Notes

The entire deployment can be customized by overwriting the chart's default configuration values. Here are a few point to look out for when changing these values:

- By default, a PostgreSQL 17 database is deployed alongside the MDSS deployment
- In a production environment it's recommended to use an external service for the database and set `deploy_with_mdss_db` to false in order to not deploy an in-cluster database

- When using an external database, prefer a clean reinstall (`helm delete` / `helm install`) over `helm upgrade`, since the database holds all persistent state and a clean install avoids potential schema or configuration conflicts

Portable Package

This guide provides instructions on how to deploy MetaDefender Storage Security docker images without using the the installer. This method is intended for deployment scenarios where installing docker and docker-compose is not possible using official installation methods.

> **Info**
>
> It's recommended to have the same dependencies as the ones required for the offline installation preinstalled on the target host machine.

Downloading the portable toolkit

The toolkit is available on the **MyOPSWAT** portal under **MetaDefender Storage Security** in the **MetaDefender Storage Security Portable Toolkit** section. The package needs to be extracted to the desired path on the target machine and the following steps should all be performed from the same path where the package is extracted on the target machine.

Starting MetaDefender Storage Security

From the path where the portable package is extracted, MDSS can be started using the regular startup command but specifying the local path to the `mdss.sh` script:

```
sudo ./mdss.sh -c start
```

Using this method, all mdss commands (start, stop etc.) will need to be executed from the same path where the package is extracted.

Integrations

Integrations

MetaDefender Storage Security integrates with various cloud and on-premises storage solutions, represented as **storage units**.

We provide a list (see table below) of officially supported integrations that are continuously tested for full functionality.

> **IMPORTANT Please get in touch with us if your specific storage provider isn't shown.**
>
> We likely already support storage systems that use S3-compatible interfaces or standard protocols such as SMB, NFS, or SFTP.

Cloud Object Storages

Storage Name	Integration Type	Tagging	Event-Based	Metadata Copy	Autodiscovery
Amazon S3	Native	Yes	Yes	Yes	Yes
Wasabi	S3 Compatible	Yes	Yes	Yes	Yes
Cloudian®	S3 Compatible	Yes	Yes	Yes	Yes
Oracle Cloud	S3 Compatible	No	No	Yes	Yes
Oracle Cloud	Native	No	Yes	Yes	No
MinIO	S3 Compatible	Yes	Yes	Yes	Yes
Cubbit	S3 Compatible	Yes	No	Yes	Yes
Azure Blob	Native	No	Yes	No	Yes
Google Cloud	Native	Yes	Yes	No	Yes
Alibaba Cloud	Native	No	Yes	No	No
Scality RING	S3 Compatible	Yes	Yes	Yes	Yes

File Storages

Storage Name	Integration Type	Event-Based	Autodiscovery	Replace Original	Scan File Versions
Box	Native	Yes	No	No	No
OneDrive	Native	No	No	Yes	No
SharePoint Online	Native	Yes	Yes	Yes	No
Microsoft Teams	Native	No	No	Yes	No
SharePoint On-Premise	Native	No	No	No	Yes
Azure Files	Native	No	No	No	No
OPSWAT Managed File Transfer (MFT)	Native	No	No	No	No

Network Attached Storages (NAS)

Storage Name	Integration Type	Event-Based	Delete empty folders	Autodiscovery
NFS	Native	Yes	Yes	No
NFS with Data Wrapping	NFS	Yes	No	No
SMB	NFS (recommended) Native	Yes	Yes	No
NetApp ONTAP	SMB	Yes	No	No
Amazon EFS	NFS	Yes	Yes	No
Dell Isilon	SMB	Yes	No	No
SFTP	Native	Yes	Yes	No
FTP	Native	Yes	No	No

Extra functionalities

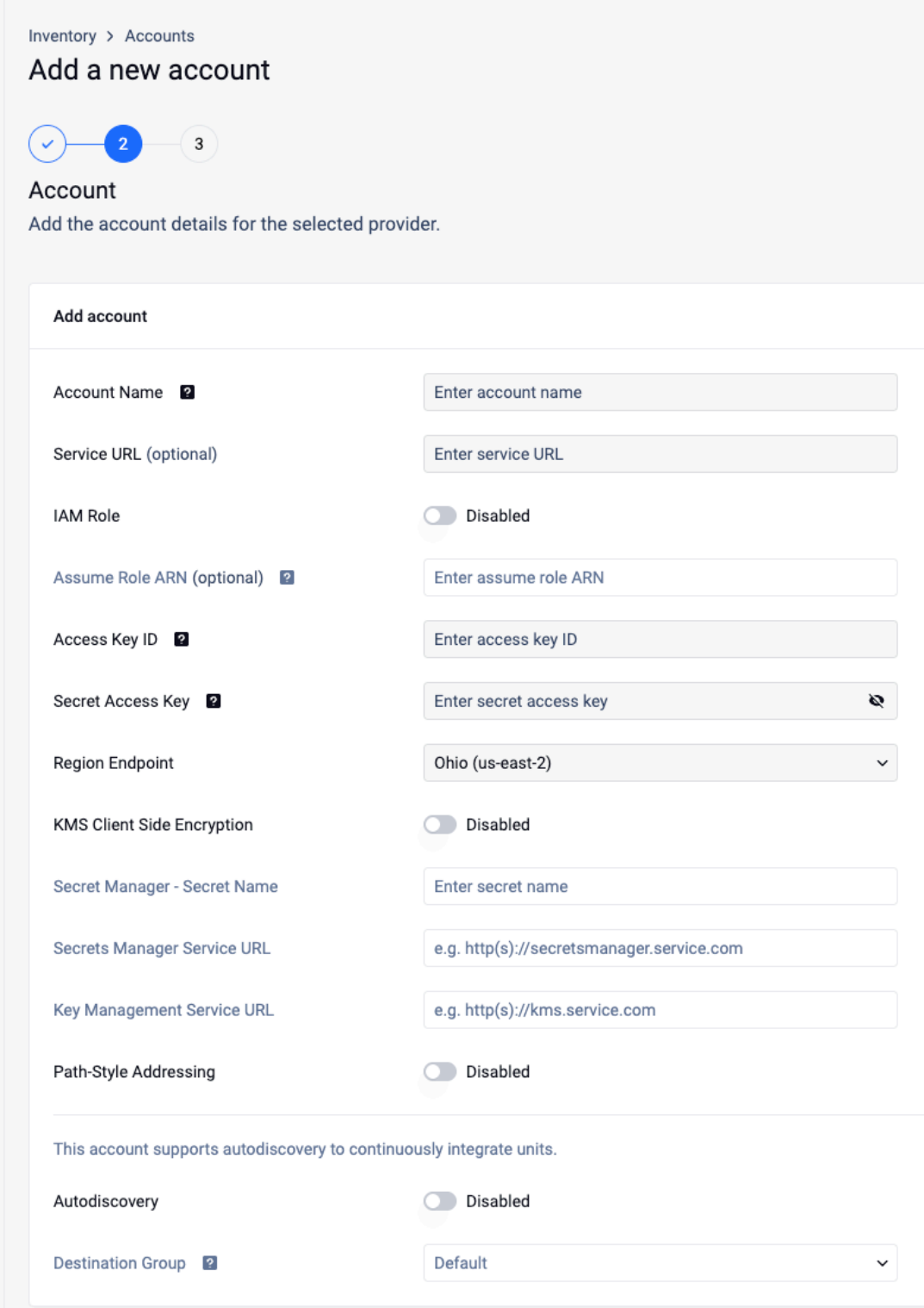
> **How Replace Original works**

> With Replace Original enabled, MetaDefender Storage Security overwrites the content of the original file with the sanitized content. No new file is created: the file keeps its name, location and sharing, so users keep opening the same file and now get the sanitized version.

Add An Amazon S3 Storage

Enabling this option may hinder product performance due to the increased number of actions required when communicating with the S3 server. In order to integrate Amazon S3 with MetaDefender Storage Security please follow these steps:

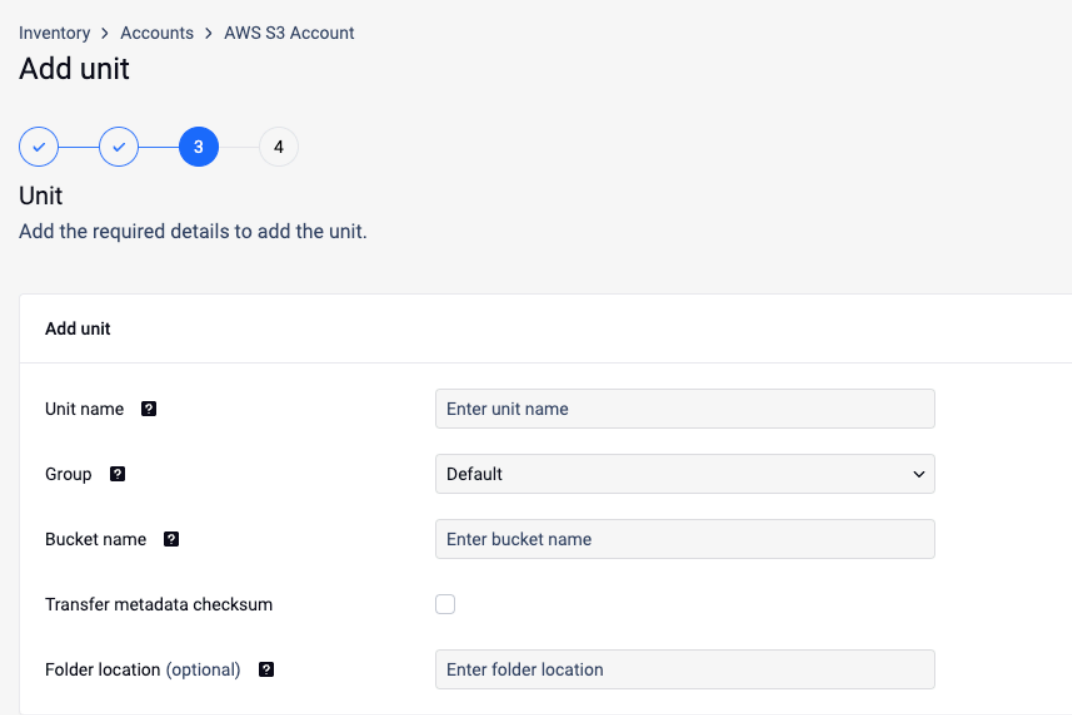
1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **Amazon S3** and click **Next**



2. Give your account a name so you can easily identify it later
3. Optionally, add a custom **Service URL** for your S3 Service, if you have one configured.
4. There are two ways to connect:

- **via the IAM role from the EC2 instance (skip to step 7)**
- **using security credentials**

5. Enter your **Access Key ID**. Scroll to the end of this page for more details.
6. Enter your **Secret Access Key.** Scroll to the end of this page for more details.
7. Choose the **Region endpoint** where your S3 endpoint is located.
8. If you want to enable **client-side encryption**, please read the below section named *Using client-side encryption*
9. *Enable Path-Style Addressing if needed.*
10. *Enable AutoDiscovery if needed*
11. *Choose the destination Group*
12. *Once the account is added successfully, proceed with adding the storage unit by navigating to* **Inventory** tab -> **Add New** -> **Unit -> Choose account previously created**



13. Give your storage a unique name so you can easily identify it later.
14. Assign a Group to your storage unit
15. Enter a valid S3 Bucket name
16. Enable Transfer metadata checksum if needed.
17. Add folder location (optional).
18. Complete the changes and add your storage unit.

Where can I find the Access Key ID and Secret Access Key?

To generate an Access Key ID / Secret Access Key pair please follow these steps:

1. Log in to **IAM Console** and go to **Users**

2. In the **Details** pane, click the appropriate IAM user or create one if none exists
3. From the **Security Credentials,** tab click **Create Access Key**

> **Info**
>
> If you already have a maximum of two access keys—active or inactive—you must delete one first before proceeding. If you need more than two root access keys, IAM users (each of whom can be assigned their own access keys) would probably better suit your requirements.

Using IAM role for Amazon S3 integration

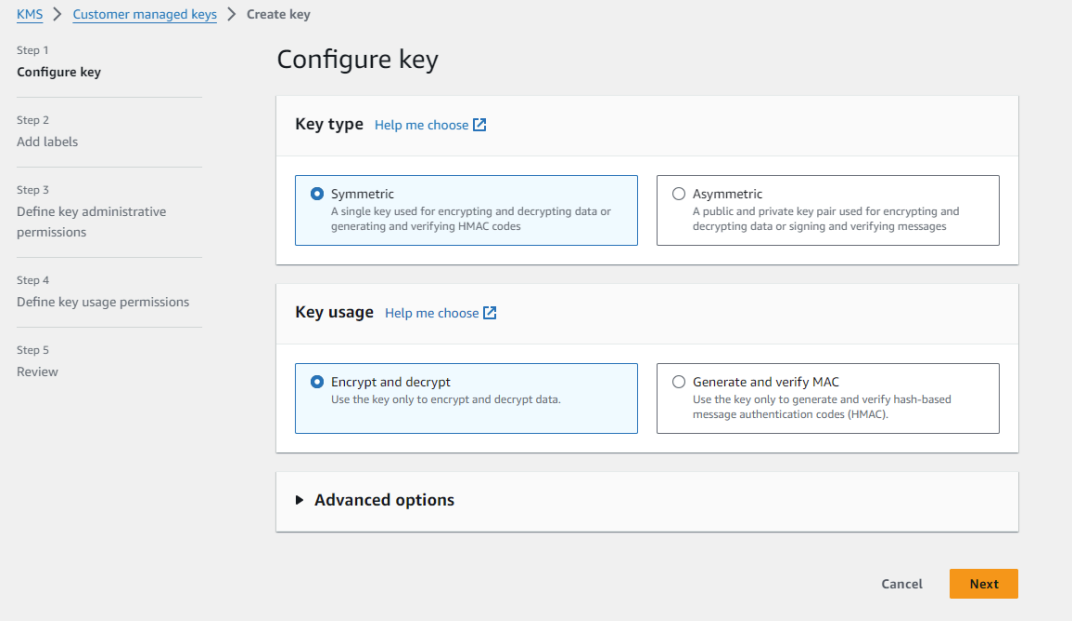
On S3 integration with IAM, the role configured on the EC2 machine will be used to authenticate.

With the optional field of Assume Role ARN, MetaDefender Storage Security allows integration with other buckets that are in different accounts from the current AWS account. This is possible by configuring Assume Role ARN field, with the ARN of the role that has access to the other account's S3 bucket.

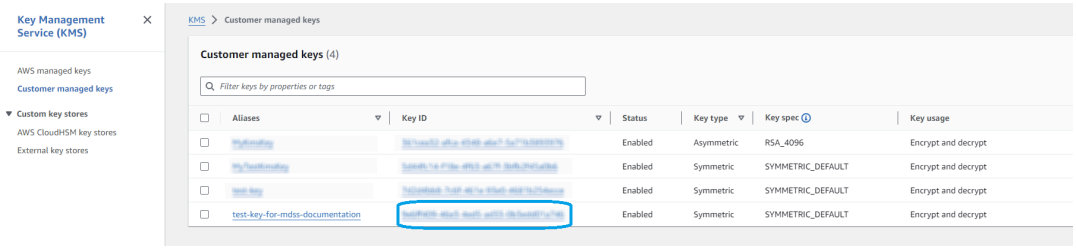
> **Using IAM Role: Modifying the default hop limit**
>
> If you are deploying MetaDefender Storage Security on an EC2 instance you will need to run the following AWS CLI command in order to increase the default `hop-limit`
>
> ```
> aws ec2 modify-instance-metadata-options --instance-id <INSTANCE_ID> --http-put-response-hop-limit 3 --http-endpoint enabled
> ```
>
> This is **required** because our services run within Docker and IMDSv2 restricts the number of "hops" each request to the metadata service can make.

Using client-side encryption

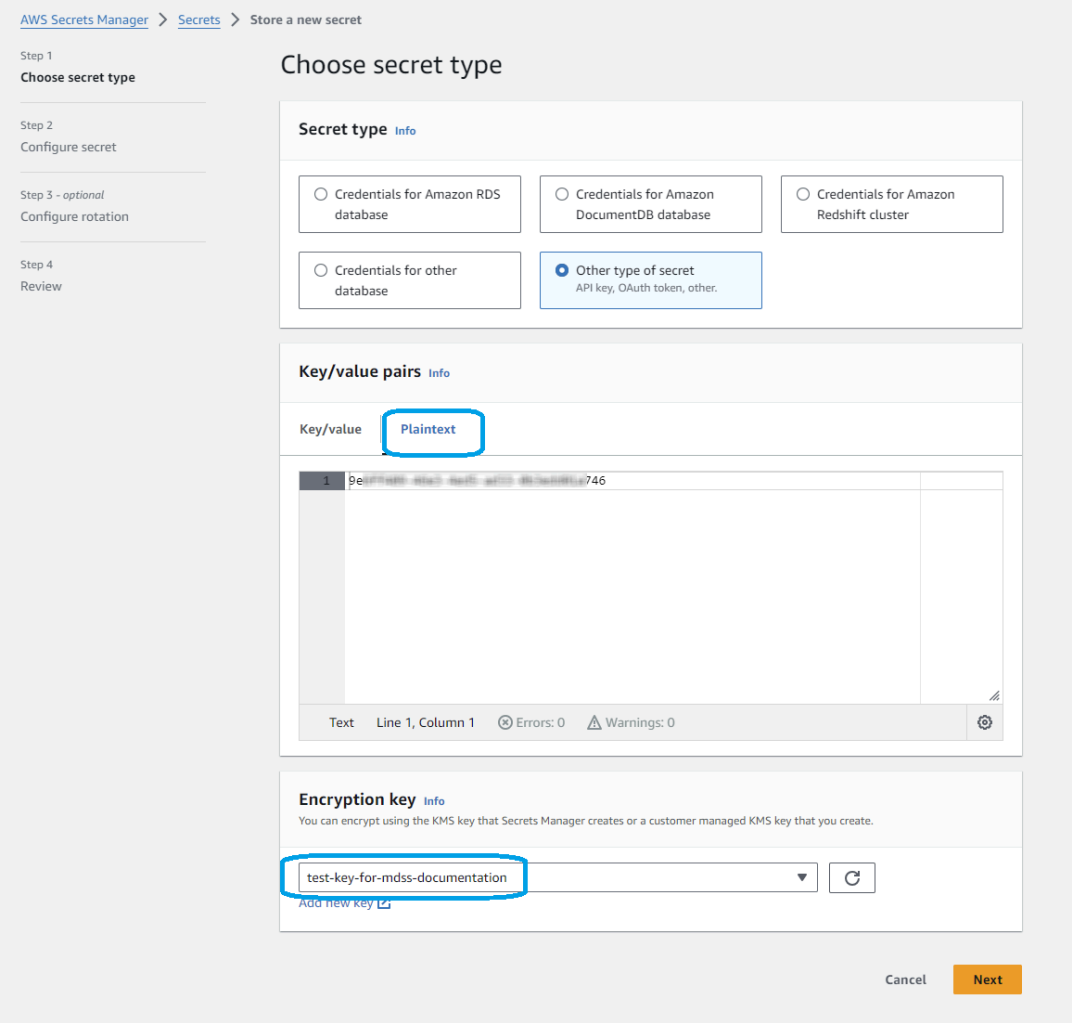
To use client-side encryption, you will need to do the following. From AWS Console, navigate to Key Management Service (KMS) and click on **Create a key.** The key should be a symmetric key and used to encrypt and decrypt.



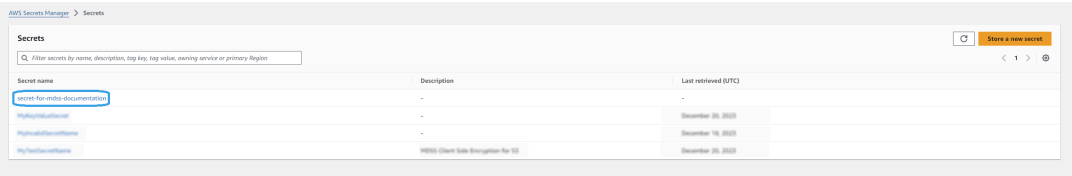
Click **Next,** give a name (Alias) to the key and finish the creation process. Once complete, the key should look like this:



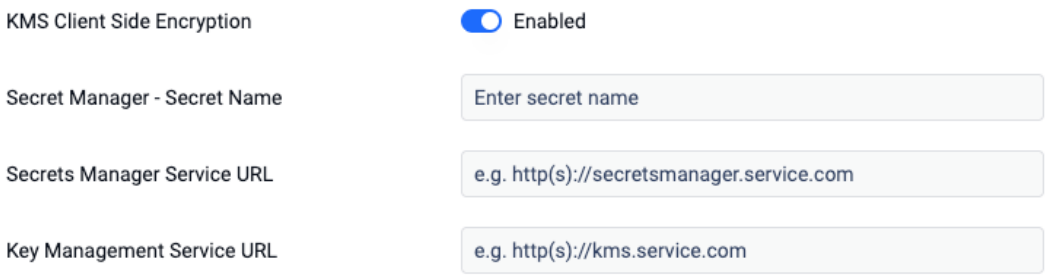
From AWS Console, navigate to Secrets Manager and click on **Store a new secret**. Secret type should be Other type of secret and Encryption key should be the name of the key from step 2. The Secret value needs to be in plain text and should be the ID of the key created at step 1. Please note that we do not support **Key/value**.



Click **Next,** give a name to the secret and finish the process. Once complete, the Secret should look like this:



Navigate to MetaDefender Storage Security UI and introduce the name of the secret created at step 4. If your Secret name is not valid or its value does not point to a valid key ID in Key Management Service, an error will be displayed on top of the UI. Also, optionally, you can add a custom **Service URL** for your Secrets Manager Service, if you have one configured.



Necessary AWS policies

These are the necessary AWS policies required in order to add an Amazon S3 Storage:

- `s3:GetObject`
- `s3:PutObject`
- `s3:GetObjectVersion`
- `s3:ListBucket`
- `s3:ListBucketVersions`
- `s3:GetObjectTagging`
- `s3:GetObjectVersionTagging`
- `s3:PutObjectTagging`
- `s3:PutObjectVersionTagging`
- `s3:DeleteObject`
- `s3:DeleteObjectVersion`

To be able to use client side encryption, additional AWS policies are required:

- `secretsmanager:GetSecretValue`
- `kms:DescribeKey`
- `kms:Decrypt`

To be able to use **server-side encryption with AWS Key Management Service (SSE-KMS)**, additional AWS policies are required:

- `kms:GenerateDataKey`
- `kms:DescribeKey`
- `kms:Decrypt`

To be able to use **autodiscovery**, additional AWS policies are required:

- s3:ListAllMyBuckets

S3 Compatible integrations

In order to use an S3 compatible storage unit the following requirements should be met:

- A valid TLS endpoint. since we are enforcing HTTPS for all S3 integrations. A valid certificate is also required. Please note that there is an option to add a custom certificate

We can integrate with any provider that meets the requirements and implements the S3 compatible open standard.

Configure UserAgent

In order to have a good tracking in AWS of every requests that MDSS does, a environment variable can be set that will be used to configure user agent.

Parameter	Value
MDSS_LOG_IDENTIFIER	string

By default this is not configured and the dotnet SDK value is used.

Transfer file metadata and checksum on created or moved files

> **Warning**
>
> MDSS does not support checksums CRC32C and CRC64NVME for any kind of operation.

This option enables the storage functionality to retrieve the metadata and checksum of files, and upon moving, copying or sanitizing them, the information is kept in the destination bucket for the files if it supports metadata or file checksum.

It can be enabled by checking the "**Transfer file metadata and checksum on created files**" and saving the storage. This also performs a test on the storage to verify if both functionalities are supported. If either of them is not supported either by the storage or due to permissions, the option cannot be enabled and saved.

If this option is enabled, the **File API** will provide information about the Checksum and Metadata of a file on the **ObjectStorageMetadata** field.

Files resulting from **DeepCDR** and **Encrypted Files** will have checksums recalculated if this option is enabled. At the same time these files will have the same metadata as the original file.

If a file has an unsupported checksum algorithm, or it does not have a checksum at all, the default **CRC32** algorithm will be used to compute a checksum for data integrity **by default**. This is the case only if the transfer option is enabled.

Checksum algorithm	Transfer support in MDSS
CRC-32 (used by default)	✅
SHA-1	✅
SHA-256	✅
CRC-32C	❌
CRC-64NVME	❌

> **Warning**
>
> If this option is enabled scanning performance may be reduced as more queries are performed to AWS services, and also the checksum needs to be recalculated during scanning in certain scenarios which also involves a performance penalty.

> **Warning**
>
> To move checksum and metadata to storages created in versions prior to v3.5.3, these need to be retested for this functionality. This can be done with a simple **Edit** action on the storage, with no changed fields.

Server-Side Encryption Override

How to enable: Set the environment variable **AMAZONSDK_USE _AWS _MANAGED _SSE** to **true**. When set, all S3 uploads send the *x-amz-server-side-encryption: aws:kms* header, so S3 uses the AWS-managed KMS key (aws/s3) instead of the bucket's default KMS key.

When to use: Enable this when S3 buckets are configured with SSE-KMS encryption using a KMS key that is not accessible to the service (e.g., cross-account keys, deleted keys, or keys without the required kms:GenerateDataKey permission). This is commonly needed in multi-account setups where the destination bucket's KMS key belongs to a different AWS account.

Note: Objects are encrypted with the AWS-managed key, not the bucket's customer-managed key. This can affect key rotation and compliance that rely on a specific KMS key.

How to Enable Inline Scanning?

In order to prevent access to files on AWS S3 that have not been scanned yet, a bucket policy can be used in combination with the file tagging feature in MDSS to only allow get requests on files with the "No Threat Detected" tag.

For example, a bucket policy allowing full access just to scanned files would look like this:

```
{
  "Effect": "Allow",
  "Principal": "*",
  "Action": [
    "s3:*"
  ],
  "Resource": [
    "arn:aws:s3:::<BUCKET_NAME>/*",
    "arn:aws:s3:::<BUCKET_NAME>"
  ],
  "Condition": {
    "StringEquals": {
      "s3:ExistingObjectTag/MetaDefenderResult": "No Threat Detected"
    }
  }
}
```

In the above example `<BUCKET_NAME>` should be replaced with the S3 bucket name configured in MDSS or with the corresponding folder path that is subject to scanning.

The “Action” list from the example above can also be customized to only allow specific actions on the objects in question. To allow just get operations on scanned objects the `s3:GetObject` action can be used instead of `s3:*`.

This approach is particularly useful when using event-based real time scanning as it prevents any uploaded file from being accessed while it’s still processing in MDSS.

How to verify if tagging functionality is supported?

Object tagging functionality is supported for AWS S3 and other S3-compatible storage providers. This document explains the permission requirements and troubleshooting steps for tagging functionality.

MDSS verifies the following permissions during storage integration or modification:

```
"s3:GetObject",
"s3:PutObject",
"s3:DeleteObject"
"s3:PutObjectTagging",
"s3:GetObjectTagging",
```

Status Conditions

Disabled Tagging Functionality

Tagging functionality may be unavailable due to:

1. Missing core permissions (`s3:GetObject`, `s3:PutObject`, `s3:DeleteObject`)
   - When these are missing, tagging functionality cannot be tested and will be disabled
2. Missing tagging permissions (`s3:PutObjectTagging`, `s3:GetObjectTagging`)
   - When these are missing, tagging functionality will not be supported
3. Storage provider limitations
   - If the S3-compatible storage does not support object tagging

Verification Process

Permission verification occurs:

- When storage is first integrated with MDSS
- When storage configuration is edited

Troubleshooting

If tagging functionality is disabled but required permissions have been added:

1. Edit the storage configuration in MDSS
2. Save changes without modifying any settings

This action will trigger a re-verification of permissions and enable tagging functionality if all requirements are met.

How to Enable Autodiscovery

This guide provides step-by-step instructions for enabling Autodiscovery for Amazon S3 storage in MetaDefender Storage Security.

Autodiscovery automatically detects and adds S3 buckets as storage units, saving time and reducing configuration errors.

> **S3-COMPATIBLE STORAGES**
>
> The same steps apply to the **S3 Compatible**, **Wasabi**, **Oracle S3-Compatible**, **Cubbit** and **Scality RING** storage types. Select that provider instead of Amazon S3 in Method 1, and provide the endpoint and credentials described on its “Add a ... Storage” page.

Prerequisites

Before enabling Autodiscovery, ensure you have:

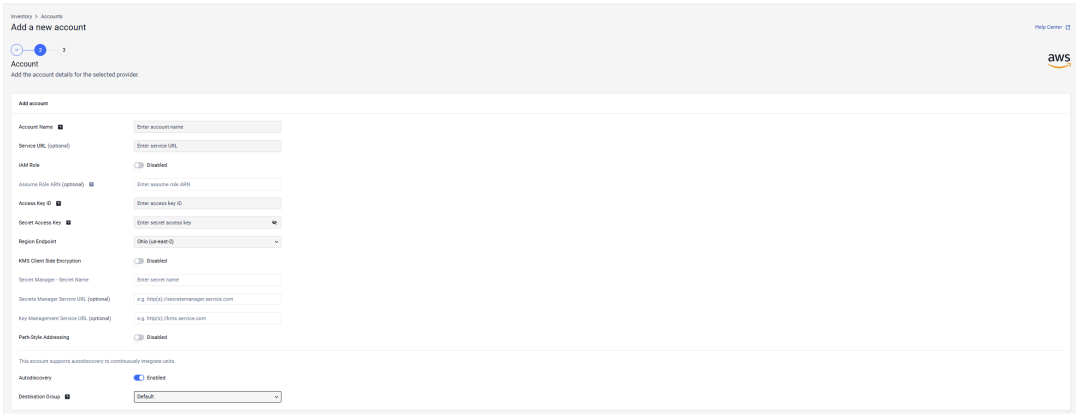
1. Valid AWS access key ID and secret access key with sufficient permissions

2. Permissions as stated in Add An Amazon S3 Storage

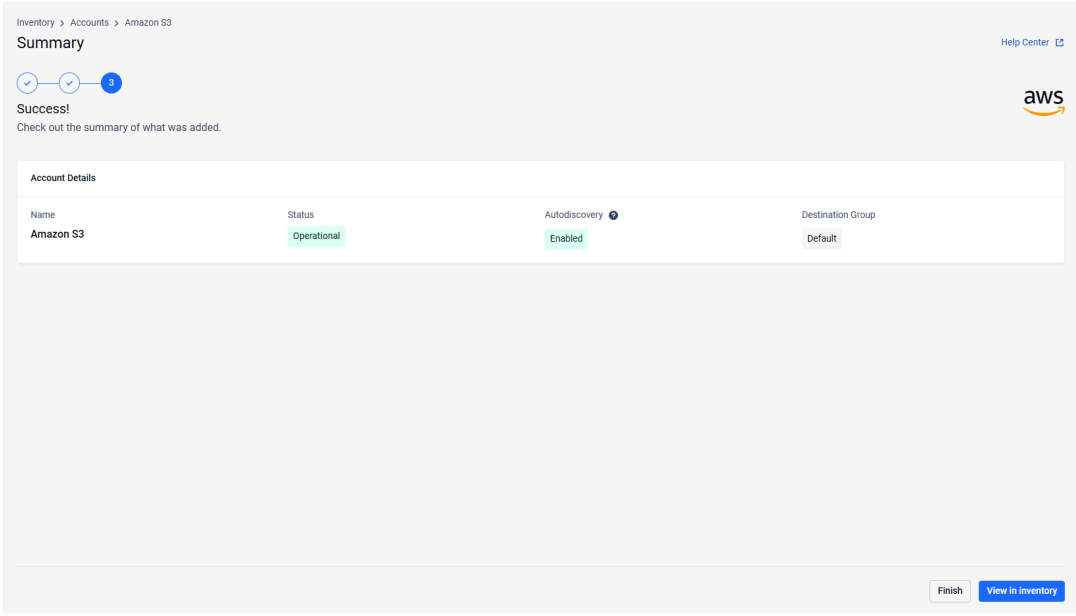
Method 1: Enable Autodiscovery When Adding an Account

- Navigate to **Inventory** › **Accounts** and click **Add account**
- Select the **Amazon S3** provider
- Fill out the form with valid Amazon S3 credentials
- Turn on the **Autodiscovery** switch and select the **Destination Group** where the discovered buckets will be added

> **REGION-SPECIFIC ENDPOINTS**
>
> Only the storages from the selected region endpoint will be discovered.



- Continue to the **Summary** step. Autodiscovery starts in the background and the discovered buckets are added as storage units to the destination group. Click **View in inventory** to open the account, or **Finish**

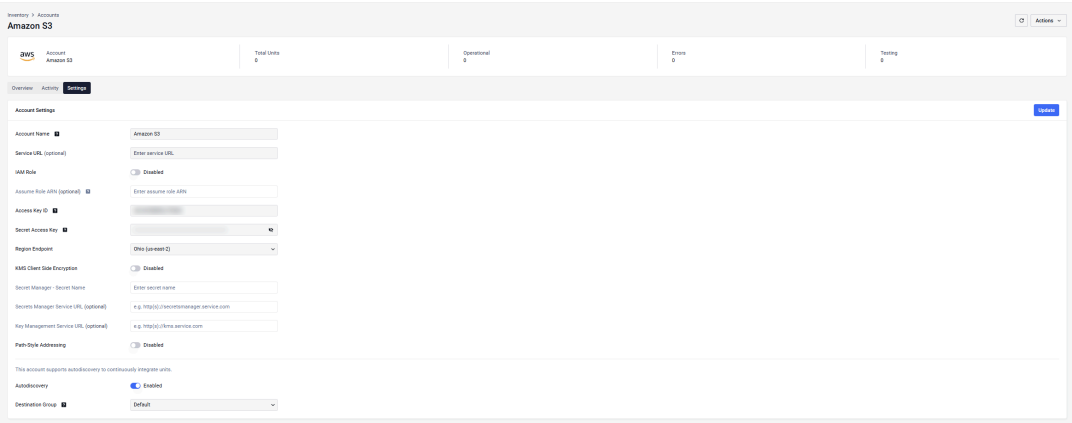


To learn more, please consult the Autodiscovery documentation.

> **ADDING SPECIFIC BUCKETS ONLY**
>
> To add only specific buckets, leave **Autodiscovery** disabled, click **Add unit** on the **Summary** step, and choose **Single unit** or **Multiple units**.

Method 2: Enable Autodiscovery for an Existing Account

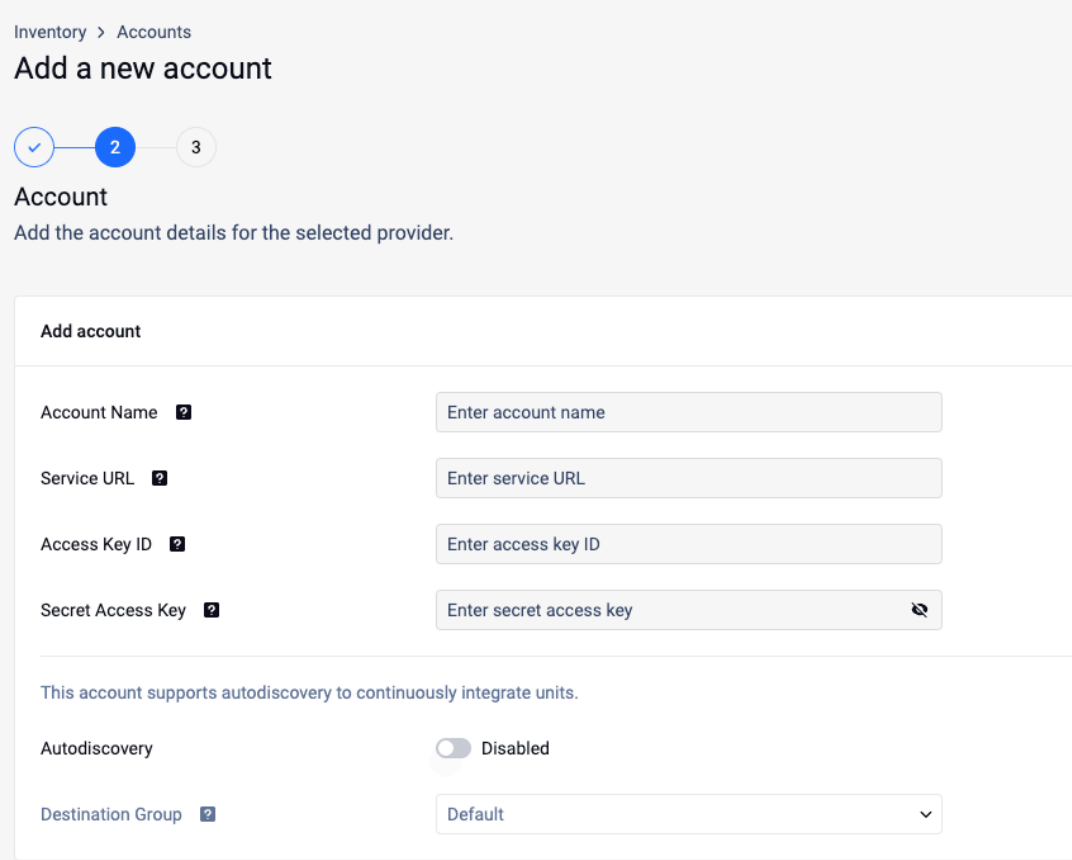
- Navigate to **Inventory** › **Accounts** and open the account you want to start autodiscovery for
- Open the **Settings** tab
- Turn on the **Autodiscovery** switch and select the **Destination Group**
- Enter the **Secret Access Key**
- Click **Update**



Add A Cubbit Storage

1. Prerequisites

In order to Add a **Cubbit** storage in MDSS, you will need the **Cubbit Service URL, Access Key Id, Secret Access Key** and the **bucket name**.



2. Obtaining the Service URL

1. Enter **s3.cubbit.eu** as the REST Endpoint or if you have a custom tenant **s3.[your-tenant].cubbit.eu.**

3. Obtaining Access Keys

1. **Access Console:** Go to the Cubbit console dashboard.
2. **Find Access Keys:** Look for **API keys** in the side menu
3. **Generate Keys:** Create a new set of keys, following any provided guidelines.
4. **Record Keys:** Securely note down your **Access Key ID** and **Secret Access Key**.

4. Obtaining the bucket name

1. **Login:** Sign into Cubbit and navigate to the dashboard.
2. **Identify Bucket:** Find your bucket in the list and note its name.

5. Adding Cubbit storage in MDSS

Once you have the details above, go to your MDSS web page, **Inventory** tab, click **Add New** > **Account**, choose **Cubbit** and fill the form.

6. Limitations

> **Bucket Versioning not supported**
>
> Due to a known issue with the Cubbit, any action that MDSS does to delete a version of the file will not work as expected. This will not affect in any way a user that doesn't use bucket versioning

Add An Oracle Cloud Storage

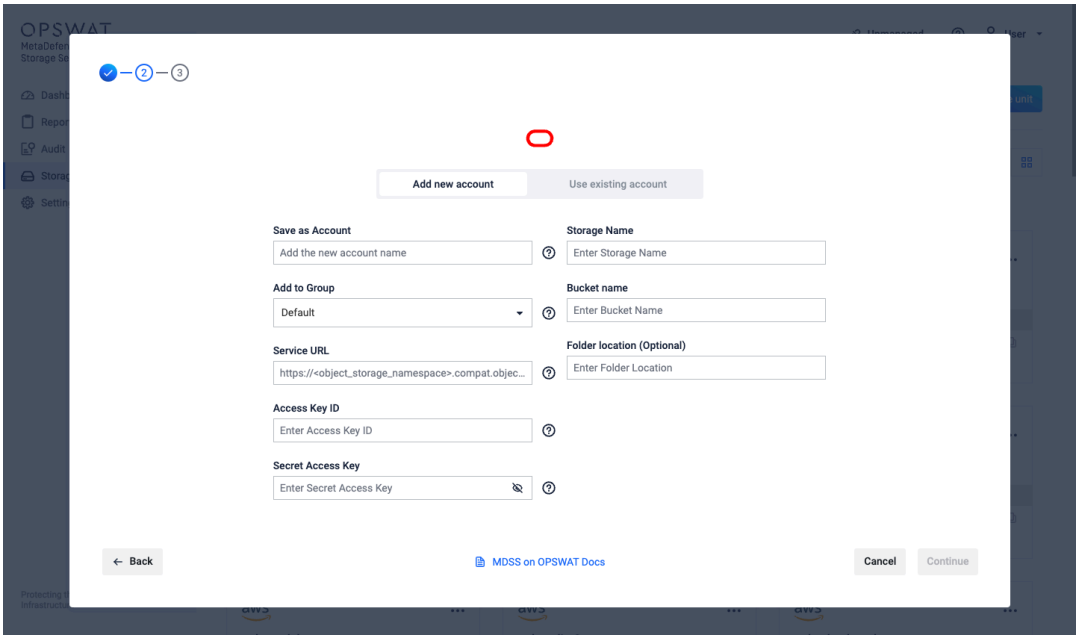
Oracle Cloud Storage can be integrated into MetaDefender Storage Security (MDSS) in **two different ways**:

1. **Using the S3-Compatible API** – This method allows MDSS to connect to Oracle Object Storage using S3-style access credentials (Access Key and Secret Key).
2. **Using the Oracle Native SDK (OCI SDK)** – This method uses Oracle's native authentication system with API keys or IAM roles for a more direct and secure integration.

Integration via S3-Compatible

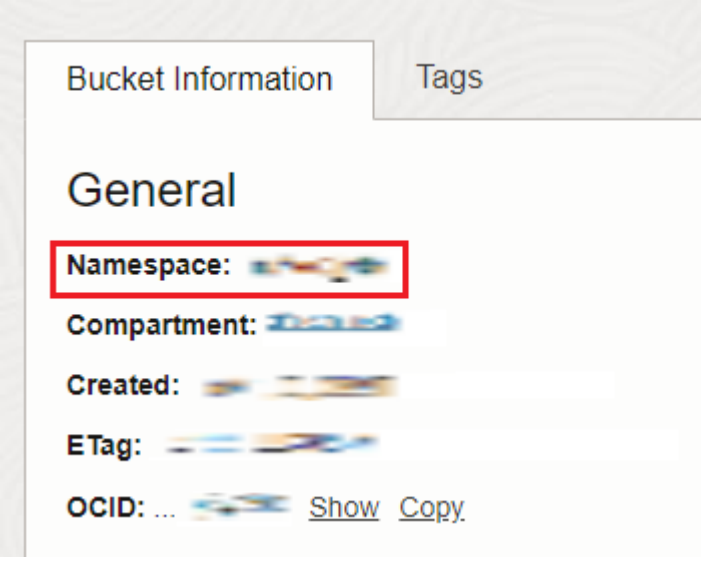
1. Prerequisites

In order to Add an Oracle Cloud storage in MDSS, you will need the **Oracle Cloud Service URL**, **Access Key Id / Secret Access Key** and the **bucket name**.



2. Obtaining the Service URL

1. To construct the URL, you will need to have you bucket's object storage namespace and region. The namespace can be found the navigating to one of your existing buckets and opening it. Afterwards, construct the URL like this: **https://<your_object_storage_namespace>.compat.objectstorage.<your_region>.oraclecloud.com**



3. Obtaining Access Key ID

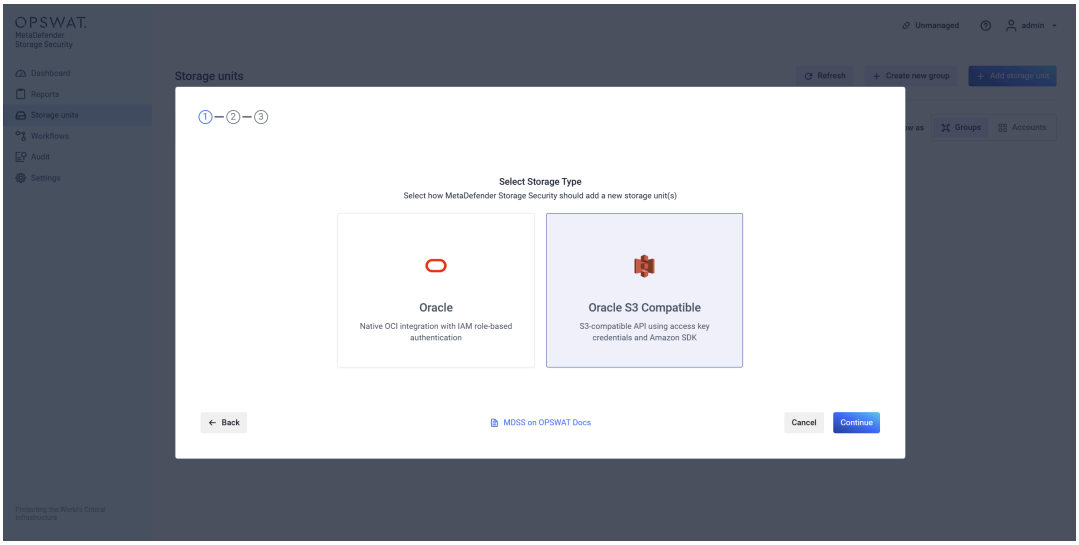
1. Log in to your Oracle Cloud dashboard at **cloud.oracle.com**
2. In the top-right corner, open the **Profile** menu by clicking on your profile icon and afterwards, click on **My profile**
3. From the **Resources** tab**,** navigate to **Customer secret keys** and either click **Generate secret key** or copy an existing one from the list

4. Obtaining Secret Access Key

1. Log in to your Oracle Cloud dashboard at **cloud.oracle.com**
2. In the top-right corner, open the **Profile** menu by clicking on your profile icon and afterwards, click on **My profile**
3. Click **Generate secret key** and give the key any name you like
4. Save the generated key secret somewhere safe. You can only see it at creation

5. Adding the Oracle Cloud storage in MDSS as an S3 Compatible

Once you have the details above, go to your MDSS web page, **Inventory** tab, click **Add New** > **Account**, choose **Oracle S3 Compatible** and fill the form.

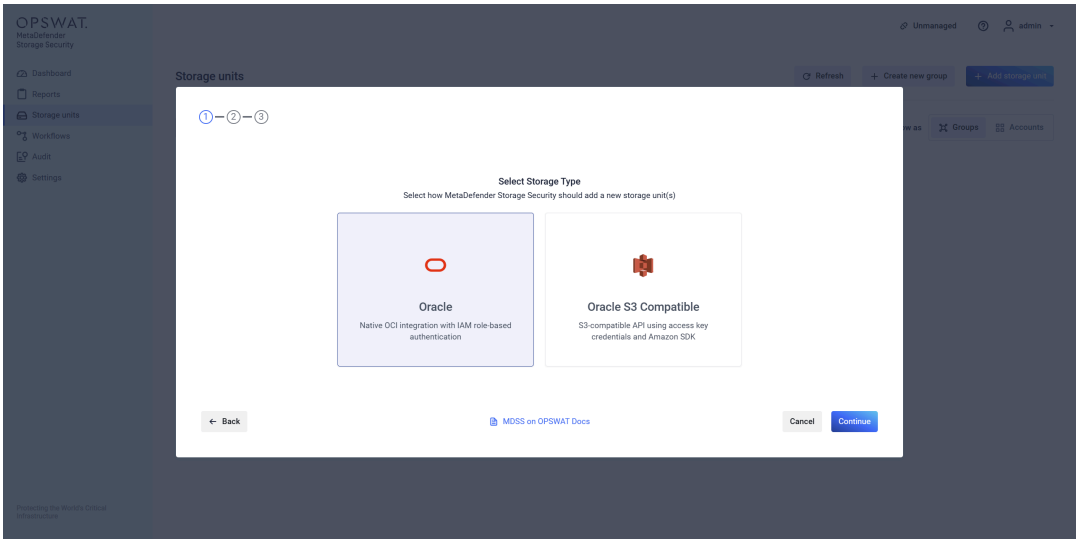


Integration via Oracle Native SDK (OCI SDK)

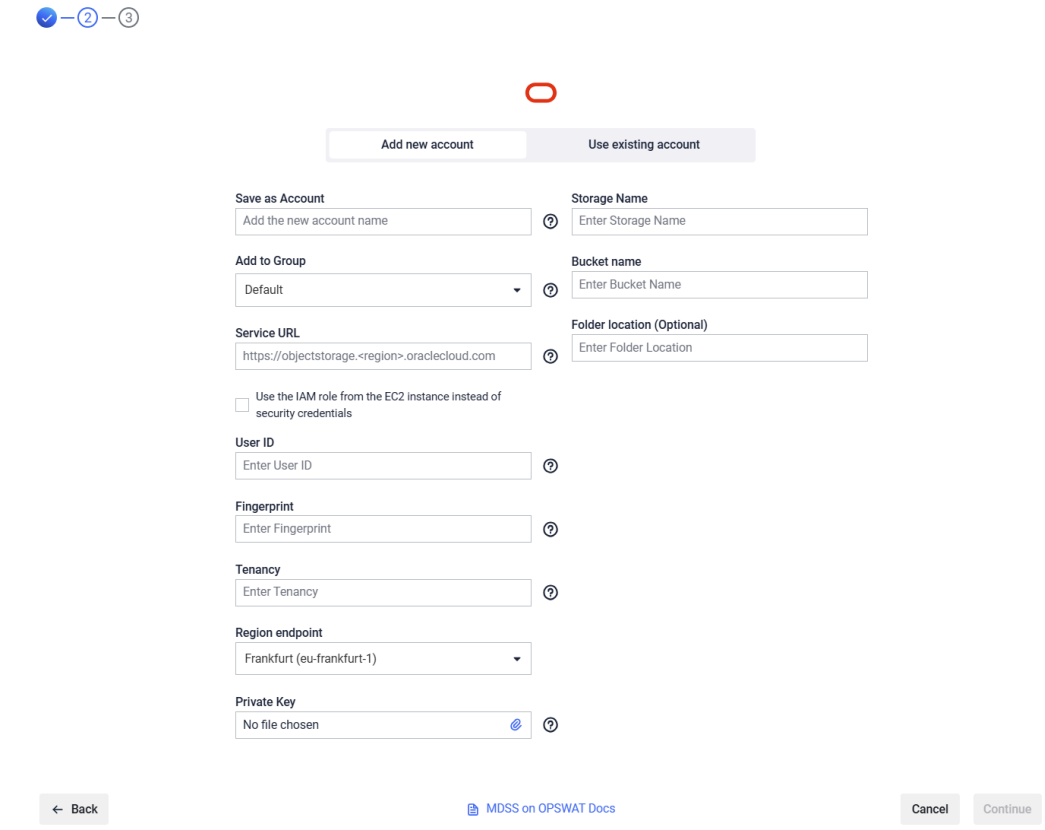
1. Prerequisites

In order to Add an Oracle SDK storage in MDSS, you will need an **Oracle API Key including UserId, Fingerprint, Tenancy, Region and Private Key, and the bucket name**

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **Oracle** and click **Next**



2. Choose the **Oracle** option



3. Give your storage a name so you can easily identify it later
4. There are two ways to connect:
   - **via the IAM role from the instance:** please skip to using IAM Role for authentication
   - **using security credentials**
5. Enter your **User Id**, **Fingerprint**, **Tenancy**, **Region** and **Private Key**
6. Enter the **name of the bucket** you wish to process with **MetaDefender Storage Security**.
7. If you wish to only process a particular folder enter the name or path of the folder in the **Folder location** field. Leave this field empty if you wish to process the entire bucket.
8. Select **Continue** in order to finish the process.

Where can I find the Credentials?

To generate the required credentials please follow these steps:

1. Log in to Oracle Console and go to **My Profile** in the top right
2. From there go to **Token and Keys** tab
3. Click **Add API Key**
4. Download the **Private Key** and Click **Add**

5. After that all the required credentials will be generated and can be copied into **MetaDefender Storage Security**

Using IAM Role for authentication

Allow your **MetaDefender Storage Security** instance to use the **Instance Principals** to access Object Storage (buckets and objects) securely without embedding credentials.

1. Create a **Dynamic Group** with at least the following matching rule `instance.id = <your instance id>`
2. Create an **IAM Policy** with the following statements
    1. Allow dynamic-group to manage buckets in tenancy
    2. Allow dynamic-group to manage objects in tenancy

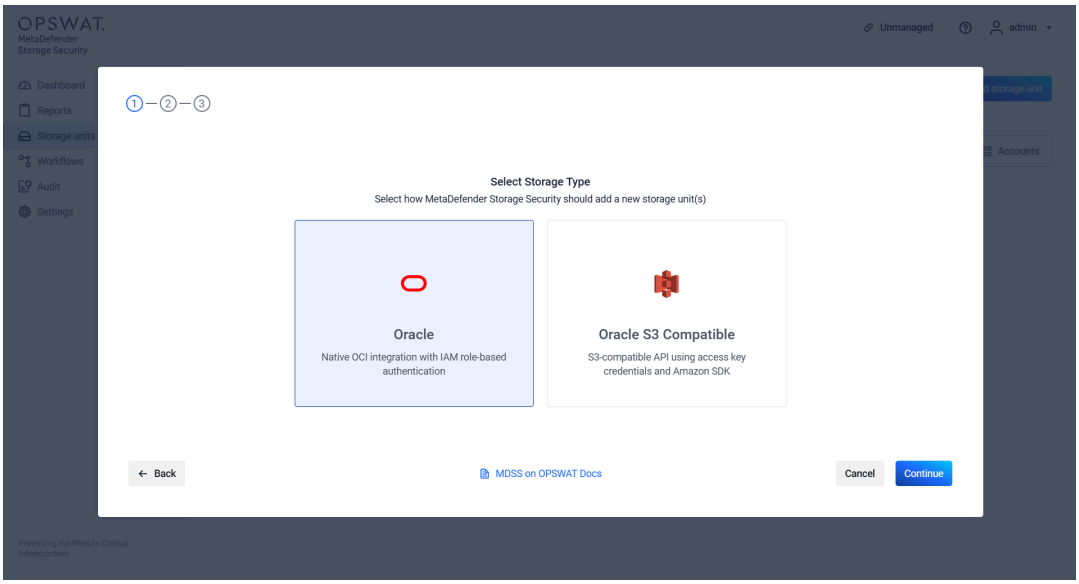
This setup allows your MDSS instance to securely access Oracle Object Storage using Oracle Cloud's native identity management system.

Add An Oracle SDK Storage

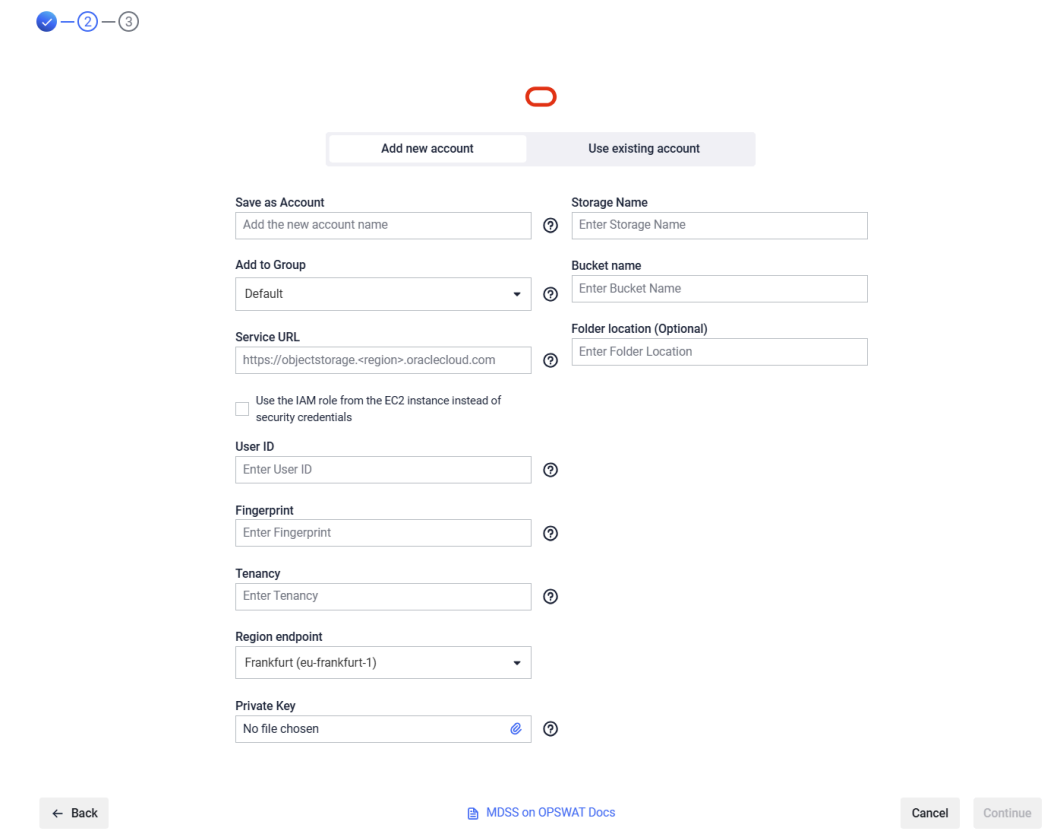
1. Prerequisites

In order to Add an Oracle SDK storage in MDSS, you will need an **Oracle API Key including UserId, Fingerprint, Tenancy, Region and Private Key, and the bucket name**

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **Oracle** and click **Next**



2. Choose the **Oracle** option



3. Give your storage a name so you can easily identify it later
4. There are two ways to connect:
   - **via the IAM role from the instance:** please skip to using IAM Role for authentification
   - **using security credentials**
5. Enter your **User Id**, **Fingerprint**, **Tenancy**, **Region** and **Private Key**
6. Enter the **name of the bucket** you wish to process with **MetaDefender Storage Security**.
7. If you wish to only process a particular folder enter the name or path of the folder in the **Folder location** field. Leave this field empty if you wish to process the entire bucket.
8. Select **Continue** in order to finish the process.

Where can I find the Credentials?

To generate the required credentials please follow these steps:

1. Log in to Oracle Console and go to **My Profile** in the top right
2. From there go to **Token and Keys** tab
3. Click **Add API Key**

4. Download the **Private Key** and Click **Add**
5. After that all the required credentials will be generated and can be copied into **MetaDefender Storage Security**

Using IAM Role for authentication

Allow your **MetaDefender Storage Security** instance to use the **Instance Principals** to access Object Storage (buckets and objects) securely without embedding credentials.

1. Create a **Dynamic Group** with at least the following matching rule `instance.id = <your instance id>`
2. Create an **IAM Policy** with the following statements
    1. Allow dynamic-group to manage buckets in tenancy
    2. Allow dynamic-group to manage objects in tenancy

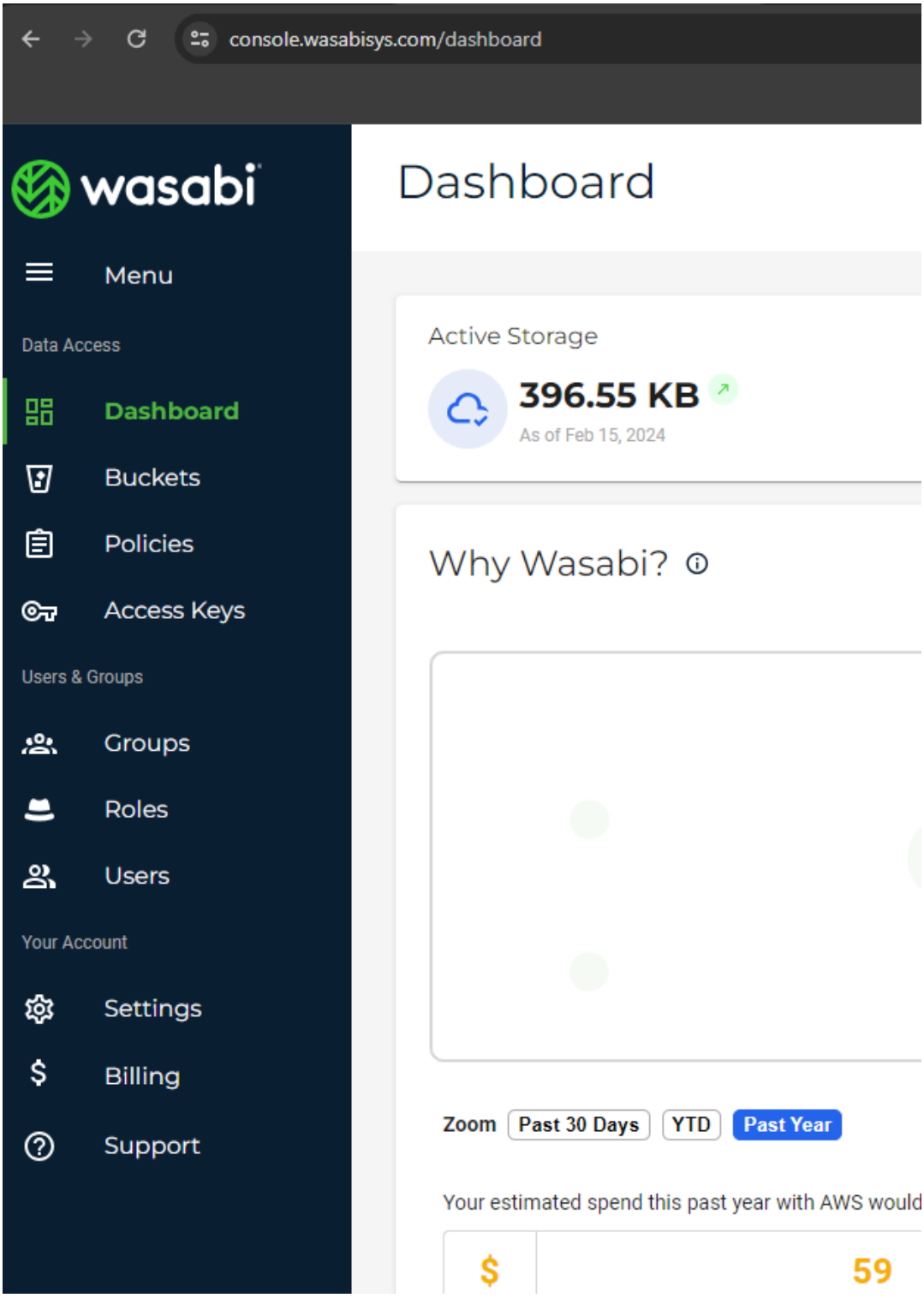
Add A Wasabi Storage

1. Prerequisites

In order to Add a **Wasabi** storage in MDSS, you will need the **Wasabi Service URL**, **Access Key Id / Secret Access Key** and the **bucket name**. If you want to enable **path style addressing** for the bucket, tick the checkbox.

2. Obtaining the Service URL

1. **Login:** Sign into your Wasabi account at https://console.wasabisys.com/
2. **Find Bucket Region:** In the bucket management section, note your bucket's region.
3. **Construct URL:** Format your service URL as `https://s3.<bucket-region>.wasabisys.com`, replacing `<bucket-region>` with your bucket's actual region.
4. Wasabi service URL docs



3. Obtaining Access Keys

1. **Access Console:** Go to the Wasabi console dashboard.
2. **Find Access Keys:** Look for “Access Keys” in the main menu [https://console.wasabisys.com/access_keys/]
3. **Generate Keys:** Create a new set of keys, following any provided guidelines.

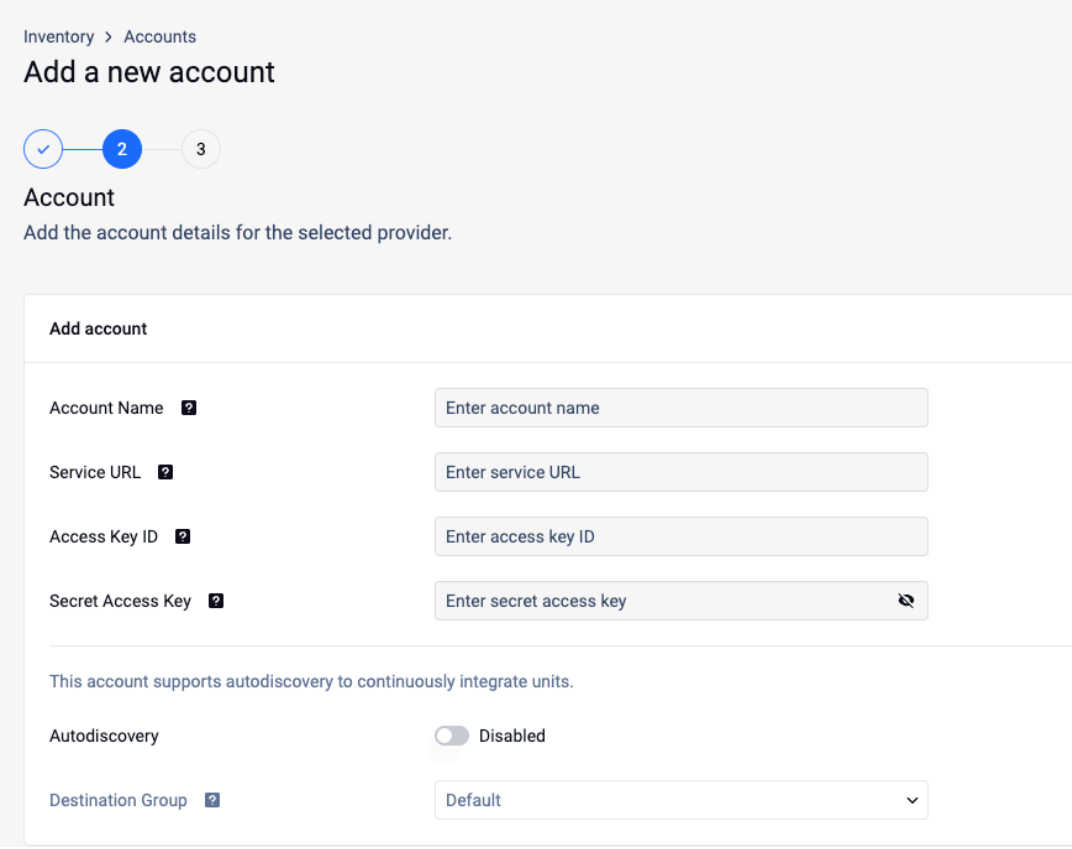
4. **Record Keys:** Securely note down your Access Key ID and Secret Access Key.

4. Obtaining the bucket name

1. **Login:** Sign into Wasabi and navigate to the dashboard.
2. **Go to Buckets:** Locate the “Buckets” section [https://console.wasabisys.com/file_manager/]
3. **Identify Bucket:** Find your bucket in the list and note its name.

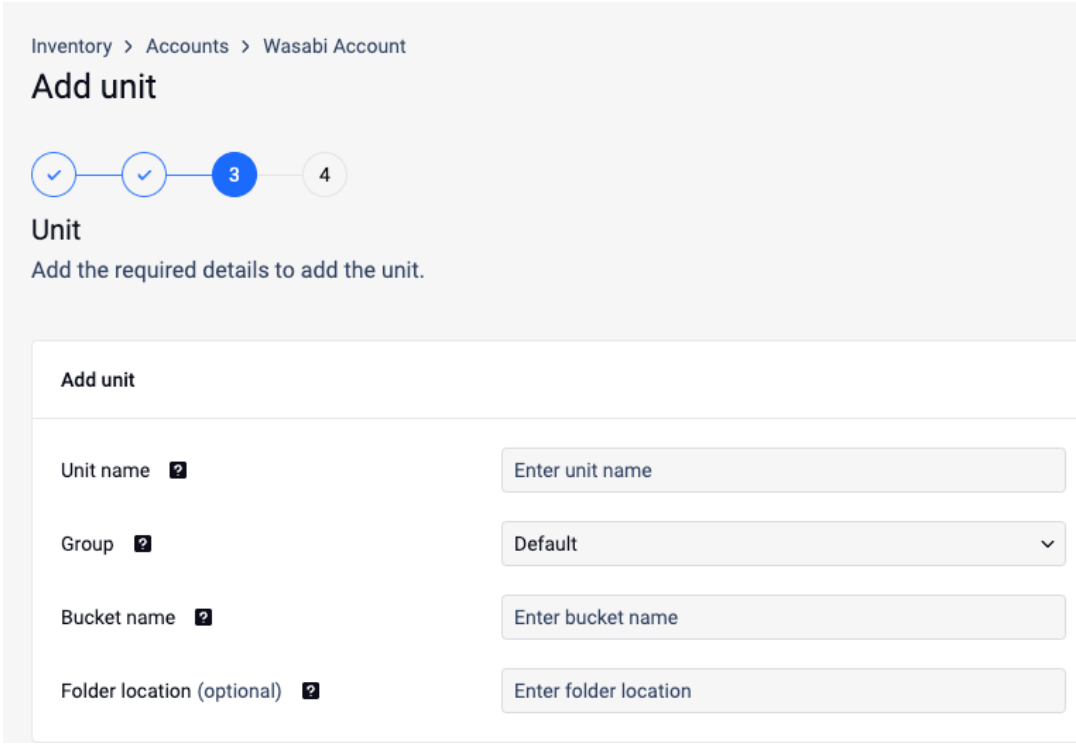
5. Adding Wasabi storage in MDSS

Once you have the details above, go to your MDSS web page, **Inventory** tab, click **Add New** > **Account**, choose **Wasabi** and fill the form.



Once the account is successfully added, please add the storage unit. Navigate to **Inventory** > **Add New** > **Unit** and choose the previously created Wasabi account:

1. Give your storage unit an unique name
2. Choose the group
3. Enter a valid bucket name
4. Enter folder location (optional)



Info

Supports both virtual-host and path-based style addressing

6. Limitations

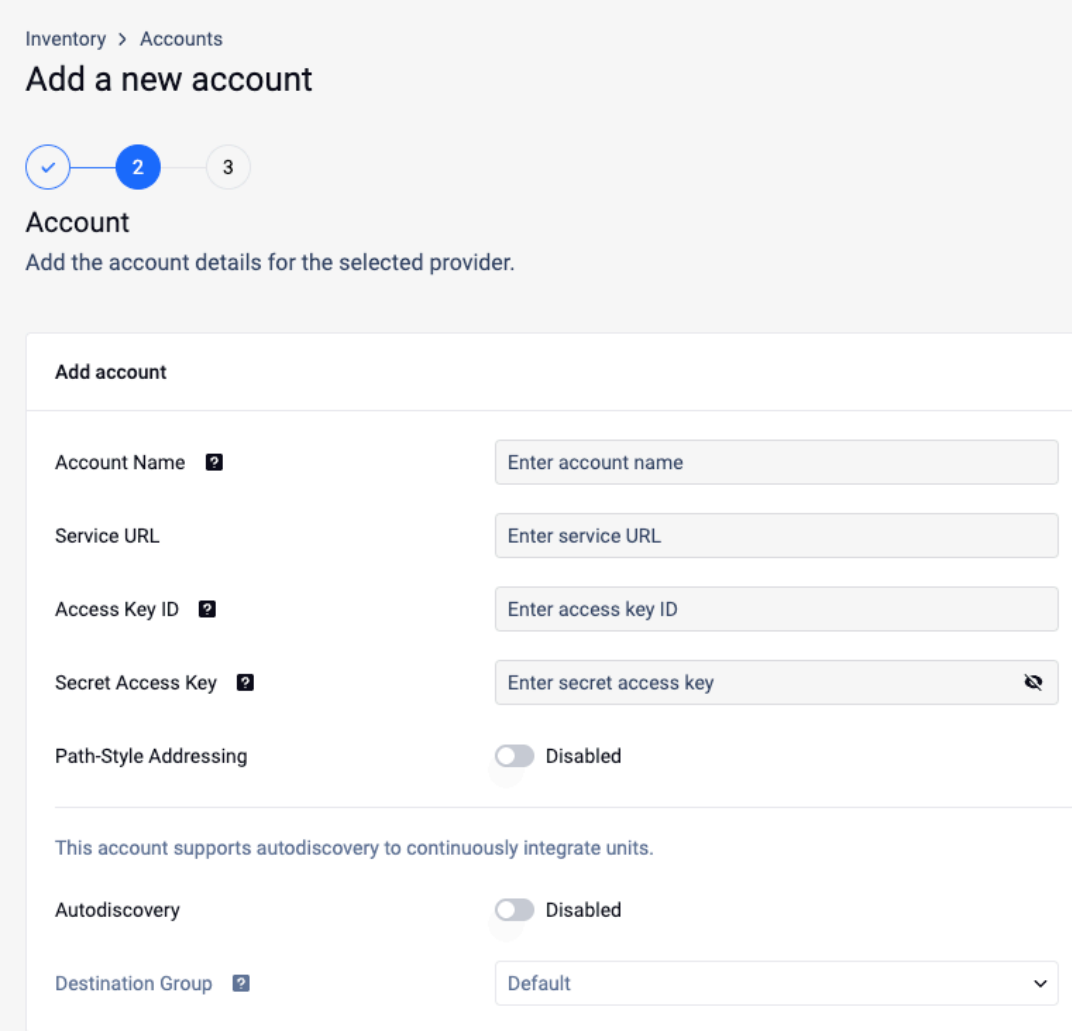
Real-Time Scans and File Renaming in Wasabi Storage

Due to a known issue with the Wasabi web interface, renaming a file does not change its “Last Modified” field. Consequently, MDSS cannot initiate real-time scans when files are renamed within Wasabi storage. This limitation stems from MDSS’s reliance on detecting changes to the “Last Modified” field to trigger scans, which does not occur in this specific scenario with Wasabi storage.

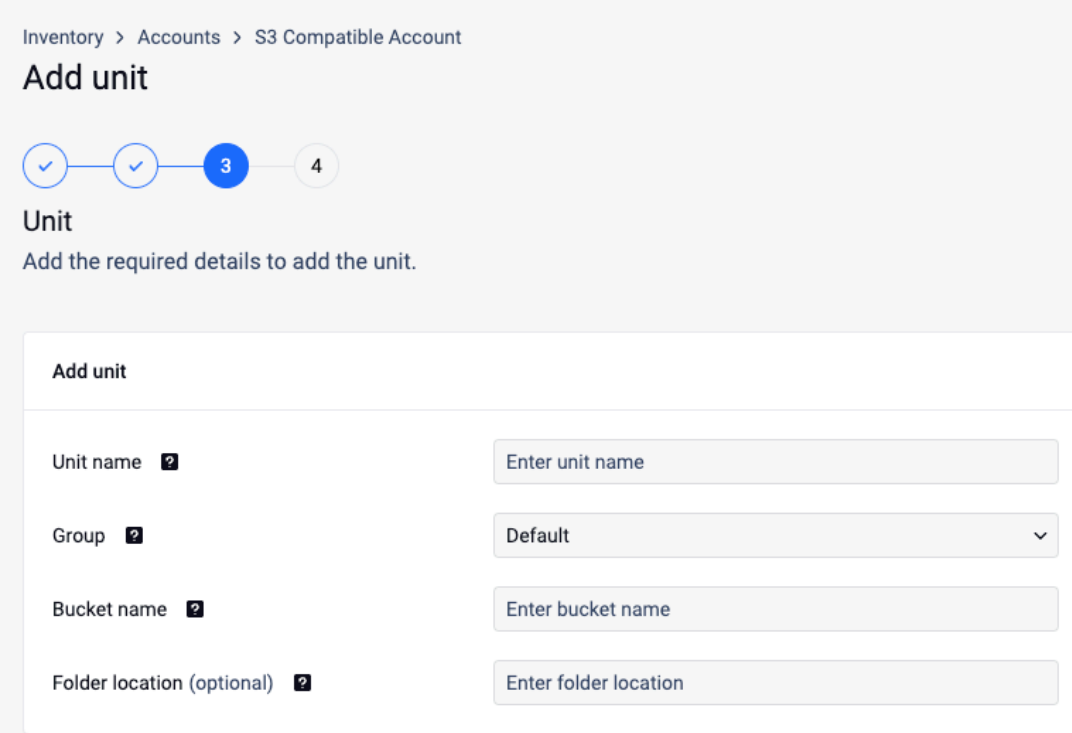
Add a Cloudian Storage

In order to integrate Cloudian On-Premises with MetaDefender Storage Security please follow these steps:

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **S3 Compatible** and click **Next**



2. Give your account a name so you can easily identify it later
3. Enter the Service URL for your Cloudian endpoint
4. Enter your **Access Key ID**. Scroll to the end of this page for more details.
5. Enter your **Secret Access Key.** Scroll to the end of this page for more details.
6. If you want to enable **path style addressing** for the bucket, tick the checkbox.
7. (Optional) Enable Autodiscovery
8. Once the account is successfully added, navigate to **Inventory**, click **Add New** > **Unit**, choose the account and click **Next**

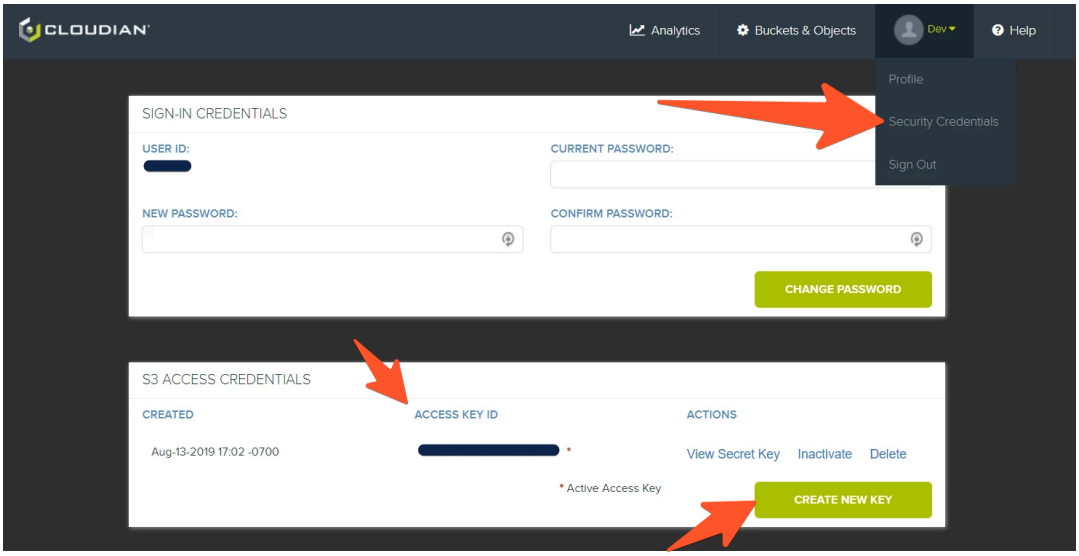


9. Give your unit an unique name
10. Choose Group
11. Enter a valid bucket name
12. Enter a Folder location (optional)

Where can I find the Access Key ID and Secret Access Key

To generate an Access Key ID / Secret Access Key pair please follow these steps:

1. Log in to **Cloudian Console** and go to **Security Credentials**
2. From the S3 Access Credentials zone, copy an existing Access Key ID and Secret Key or generate a new one by clicking **Create New Key.**

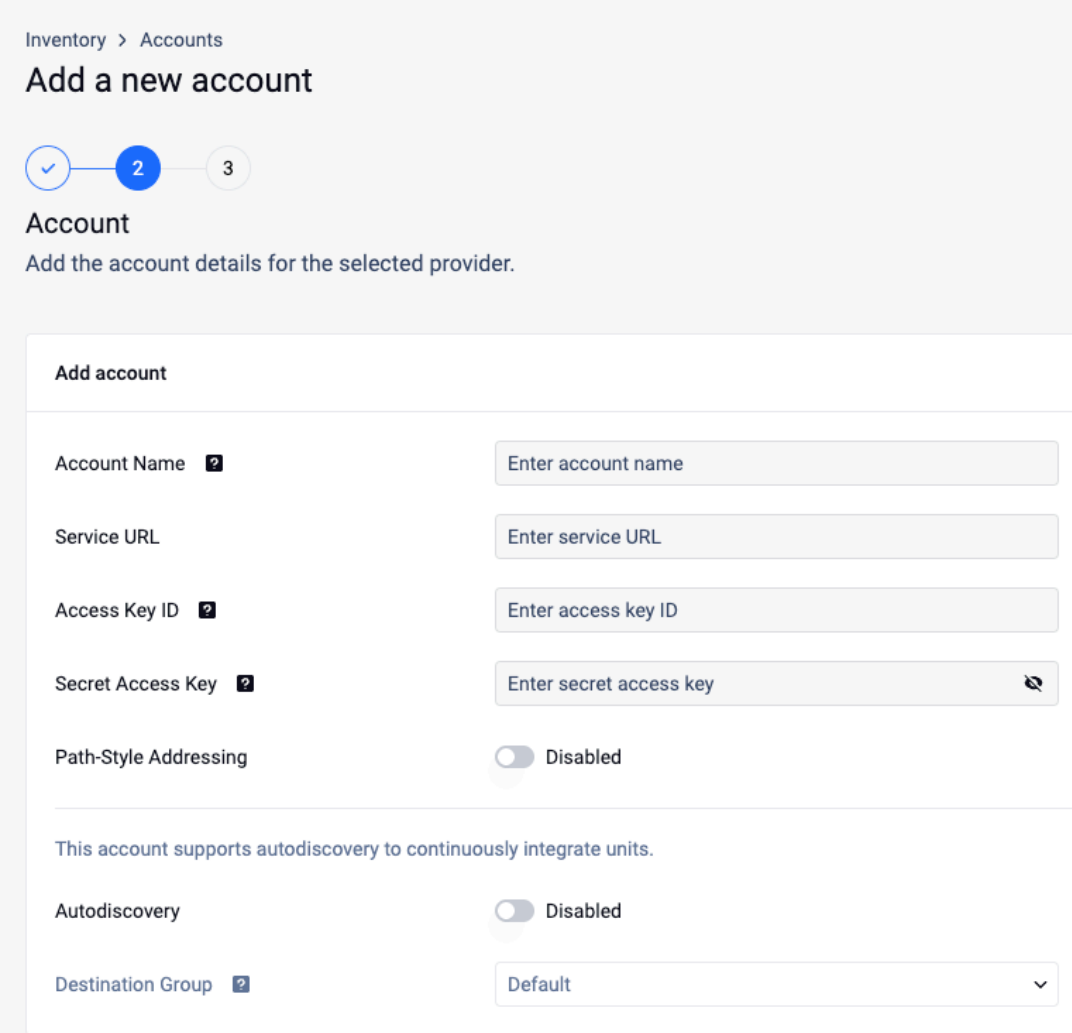

CLOUDIAN
Analytics
Buckets & Objects
Dev
Help
Profile
Security Credentials
Sign Out
SIGN-IN CREDENTIALS
USER ID:
CURRENT PASSWORD:
NEW PASSWORD:
CONFIRM PASSWORD:
CHANGE PASSWORD
S3 ACCESS CREDENTIALS
CREATED
ACCESS KEY ID
ACTIONS
Aug-13-2019 17:02 -0700
View Secret Key
Inactivate
Delete
* Active Access Key
CREATE NEW KEY

Add a MinIO Storage

> **Info**
>
> The minimum supported MinIO version is **RELEASE.2020-09-10T22-05-01Z**.

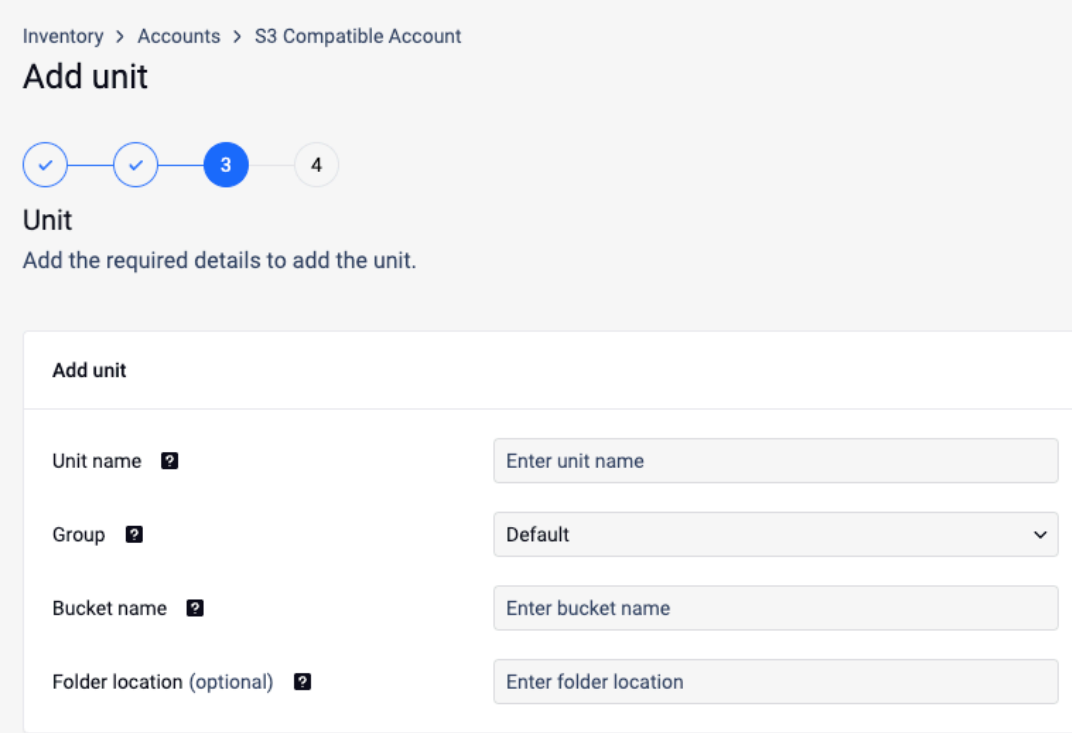
In order to integrate MinIO with MetaDefender Storage Security please follow these steps:

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **S3 Compatible** and click **Next**



2. Give your account a name so you can easily identify it later
3. Enter the Service URL for your MinIO endpoint
4. Enter your **Access Key ID**. Scroll to the end of this page for more details.
5. Enter your **Secret Access Key.** Scroll to the end of this page for more details.
6. If you want to enable **path style addressing** for the bucket, tick the checkbox.
7. (Optional) Enable Autodiscovery

8. Once the account is successfully added, navigate to **Inventory**, click **Add New** > **Unit**, choose the account and click **Next**

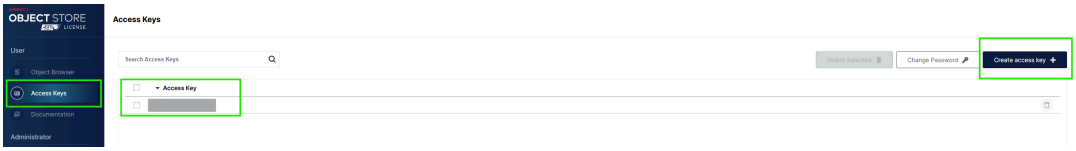


9. Give your unit an unique name
10. Choose Group
11. Enter a valid bucket name
12. Enter a Folder location (optional)

Where can I find the Access Key ID and Secret Access Key

To generate an Access Key ID / Secret Access Key pair please follow these steps:

1. Log in to **MinIO Console** and go to **Access Keys**
2. Copy an existing Access Key ID and Secret Key or generate a new one by clicking **Create New Key.**

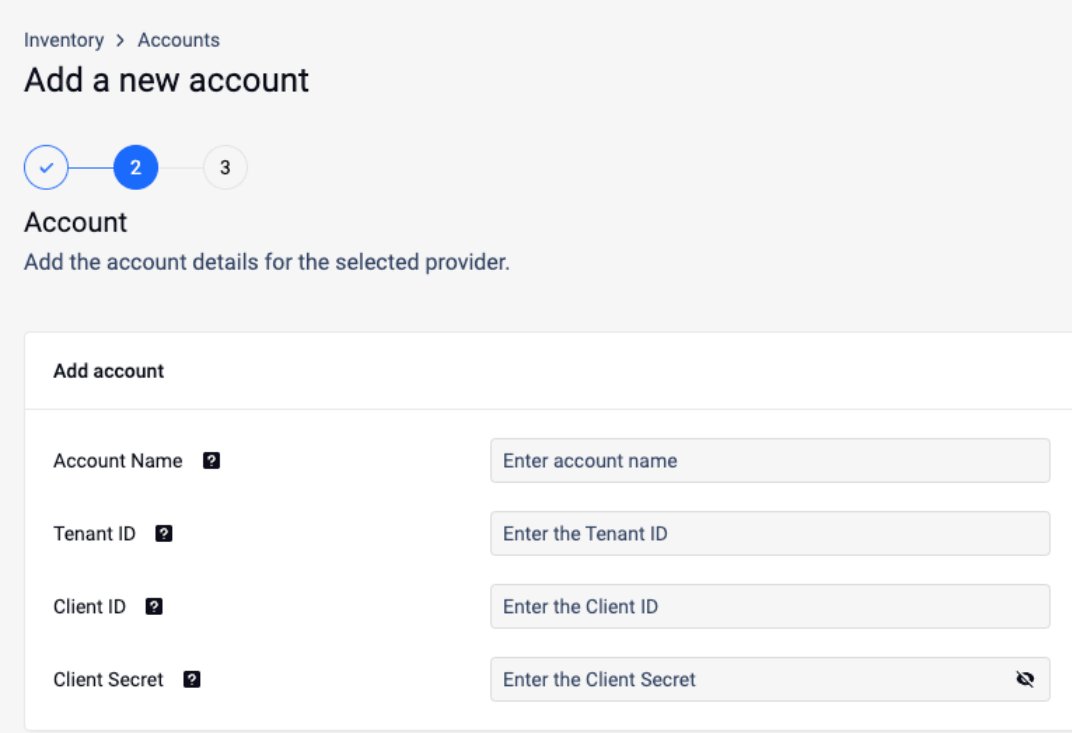


Add an OneDrive Storage

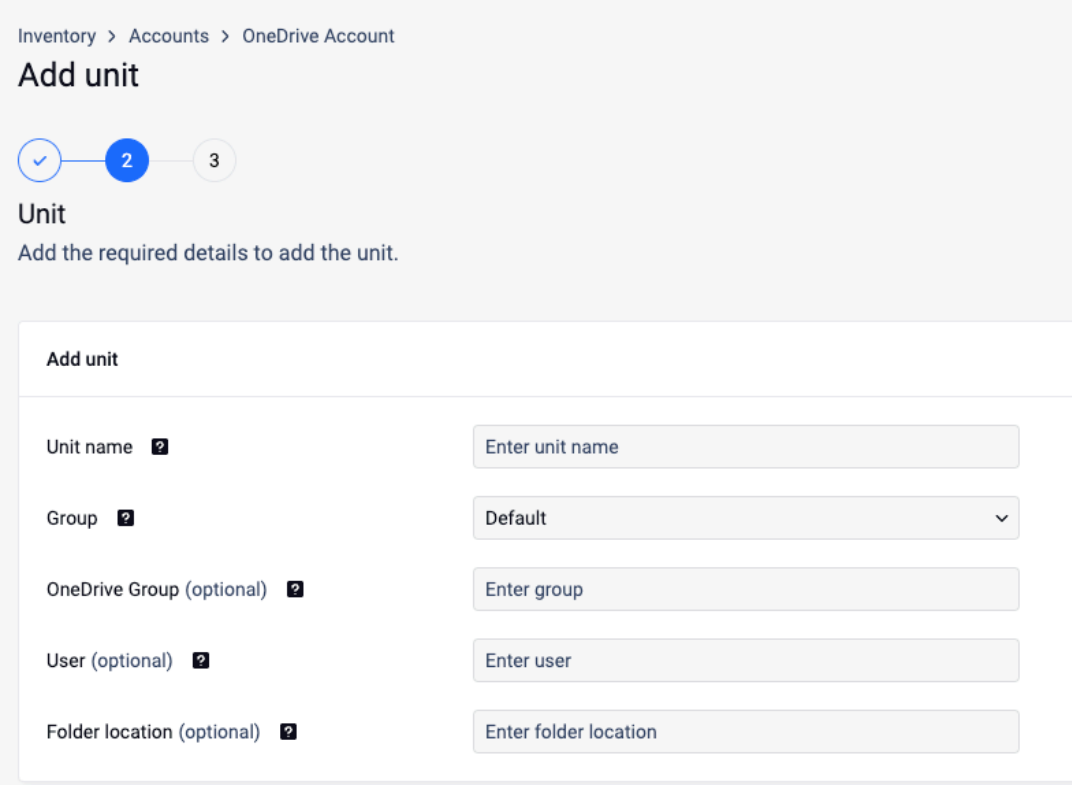
In order to integrate Microsoft OneDrive service with MetaDefender Storage Security you will be required to create an application registration, assign the necessary permissions and then generate a secret key. Please follow these steps:

> **Info**
>
> When the upload process takes more than a few seconds, OneDrive creates a temporary file having the same name as your uploading file and the size of 0 bytes. By default, we will discover that file and show it into the list of discovered files. Once the upload is complete, we will discover the original uploaded file and process it accordingly. To skip the discovery of the temporary empty file created, please follow these instructions: How to handle empty objects in Discovery?

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **OneDrive** and click **Next**



2. Give your account a name so you can easily identify it later
3. Enter your **Tenant ID**. Scroll to the end of this page for more details.
4. Enter your **Client ID.** Scroll to the end of this page for more details.
5. Enter your **Client Secret.** Scroll to the end of this page for more details.
6. Once the account is added, proceed with adding the unit

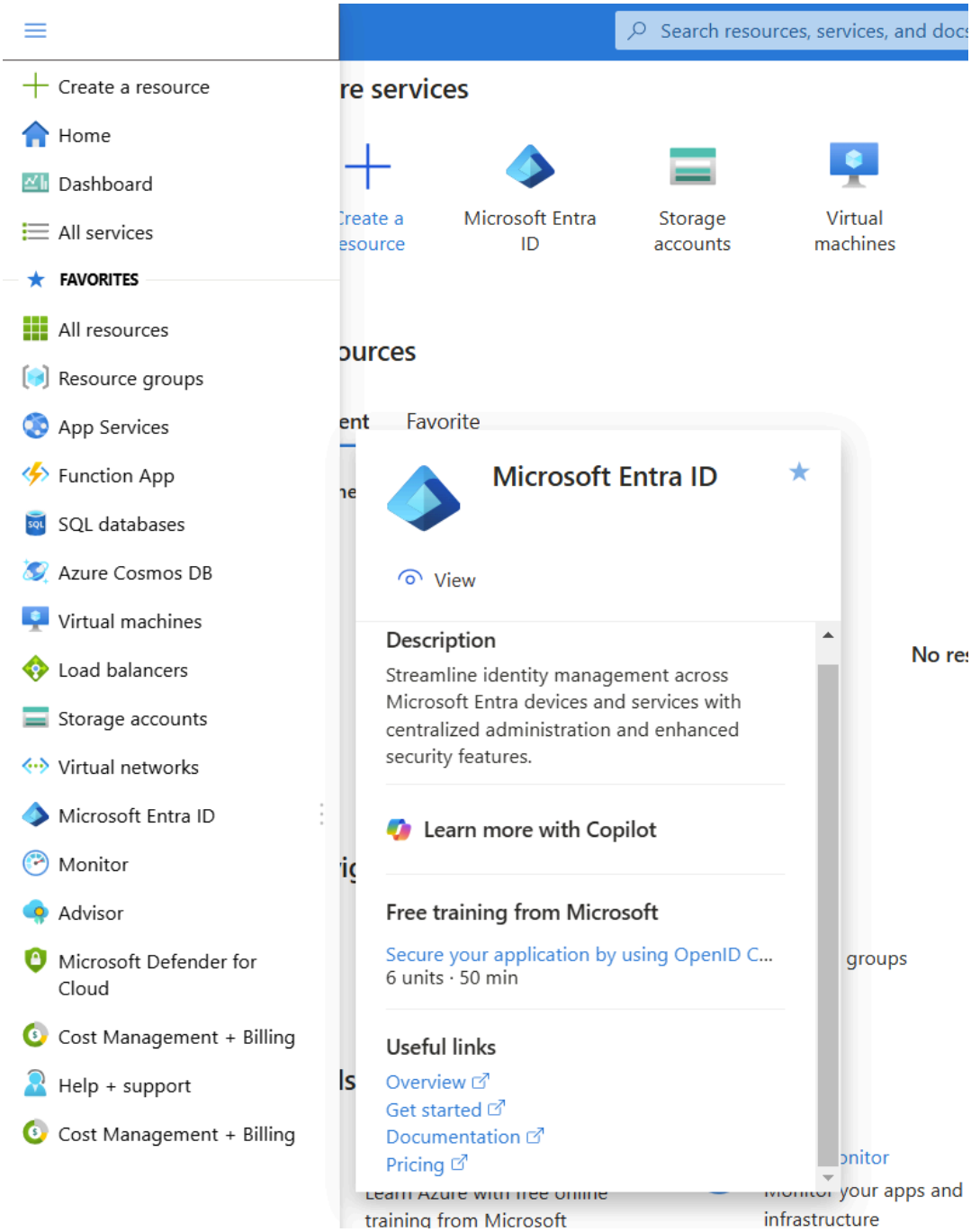


6. Enter an unique name for your unit
7. Assign the unit to a Group
8. Optionally, enter the name of a OneDrive Group that you want to protect
9. Optionally, enter the **UPN** of a user from the selected group if you wish to only process files from that specific user.
10. Optionally, enter the **name of a folder** from the user's files if you wish to only process files from that folder. This field will appear only if there is a value inputted in the **User** field.
11. Select **Continue** in order to finish the process.

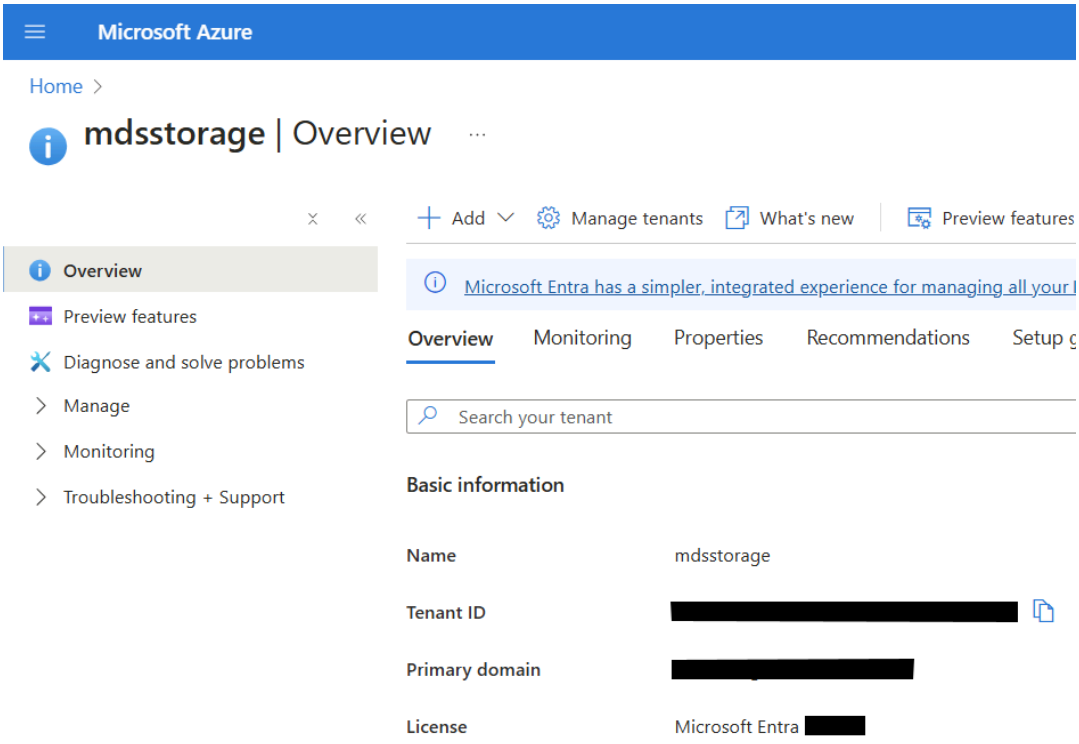
How to create a OneDrive application

In order to process your OneDrive files and folders with MetaDefender Storage Security you will be required to add a new app registration in your Microsoft Entra ID Azure Portal.

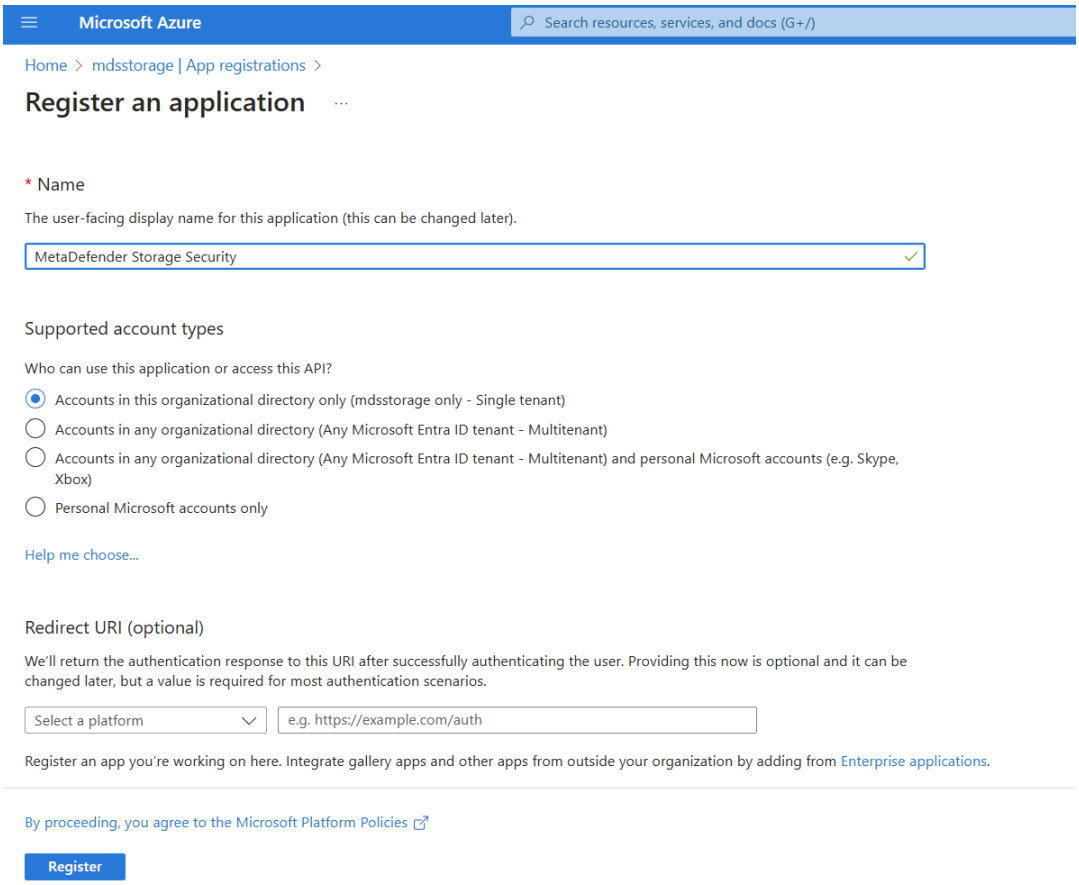
1. Log in to Azure Portal and from the left navigation menu choose **Microsoft Entra ID**



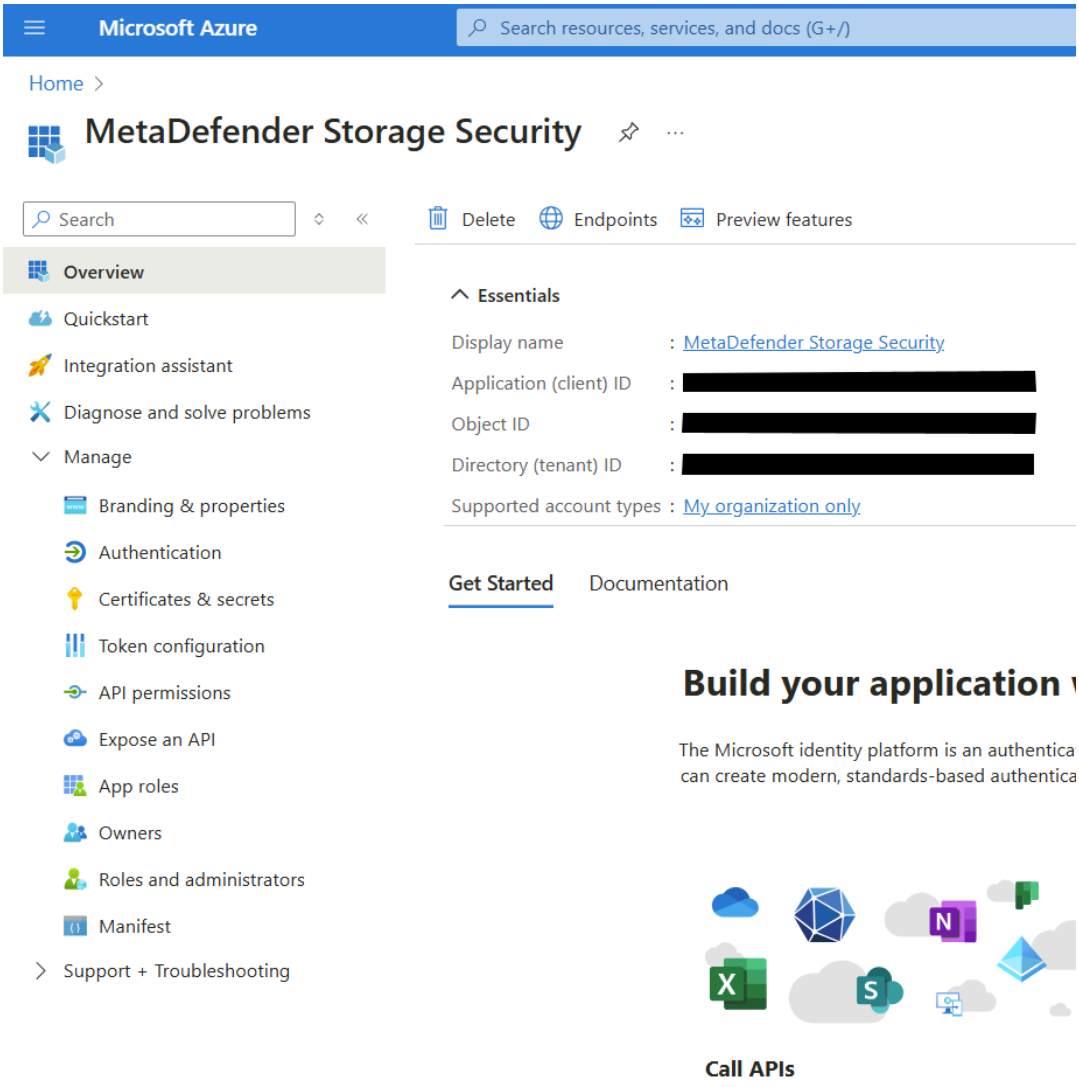
2. Make a copy of the **Tenant ID** from the overview page



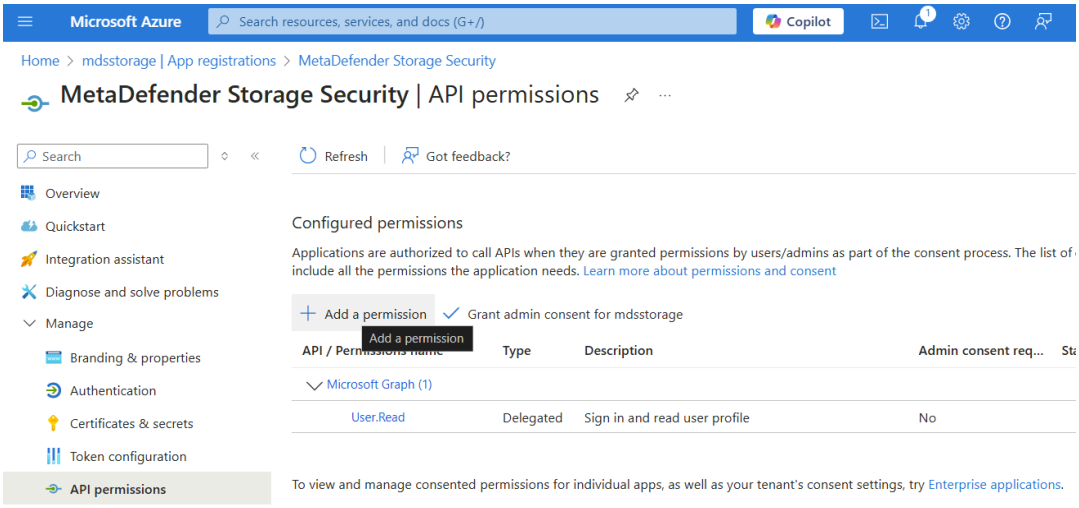
3. From the left side menu, choose **App registrations**
4. Click **New registration**
5. Give your app a name so you can easily identify it



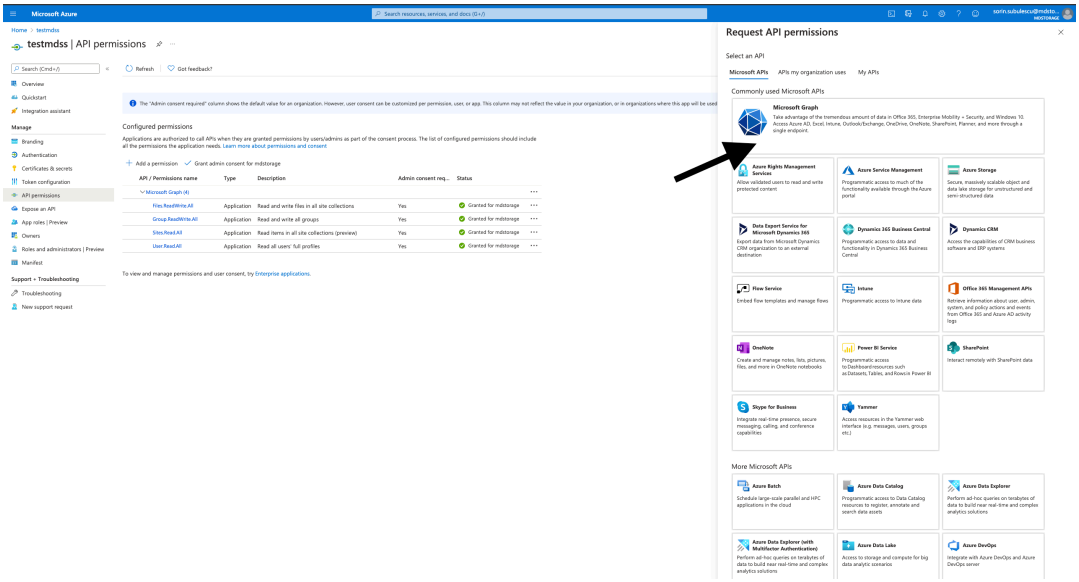
6. There is no need to modify the other properties. When ready, please click **Register**
7. From the **Overview** page of your newly created application, make a copy of **Application (client) ID**



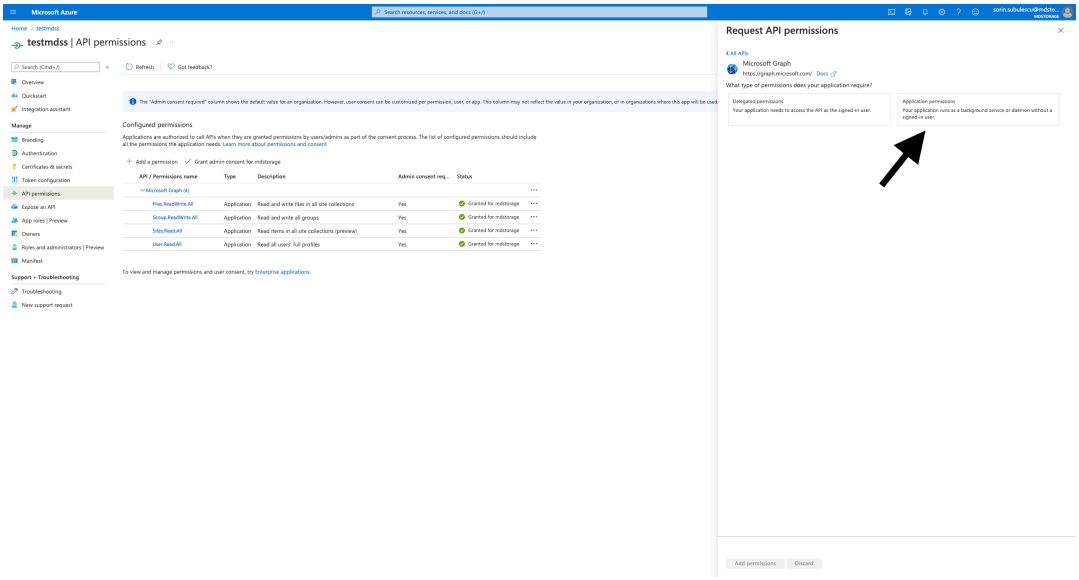
8. You now need to specify which permissions should the application have. To do this, please navigate to **API permissions** from the left-side menu
9. Click **Add a permission**



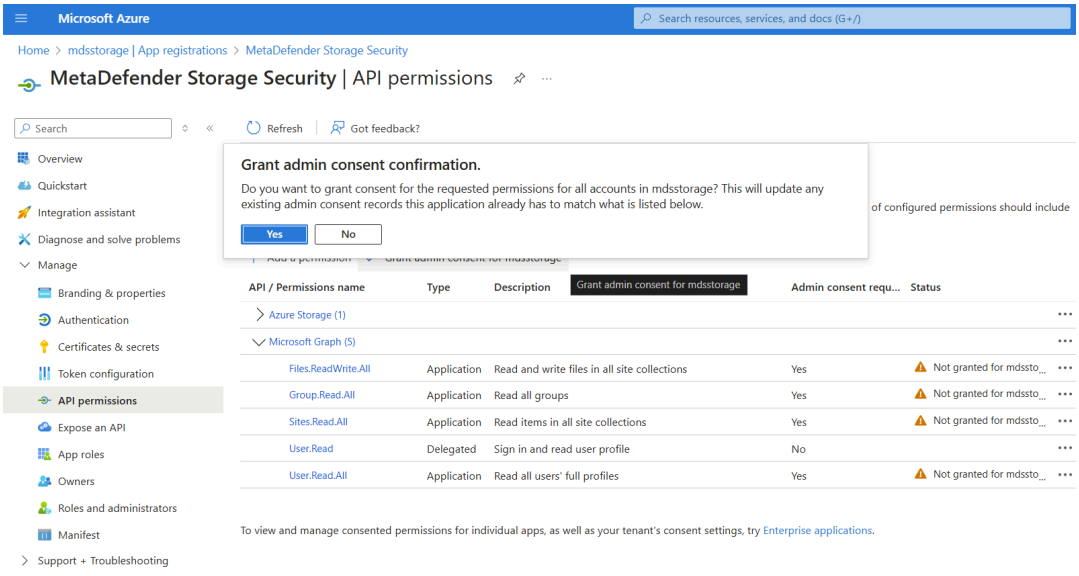
10. Select **Microsoft Graph** from the right-side menu



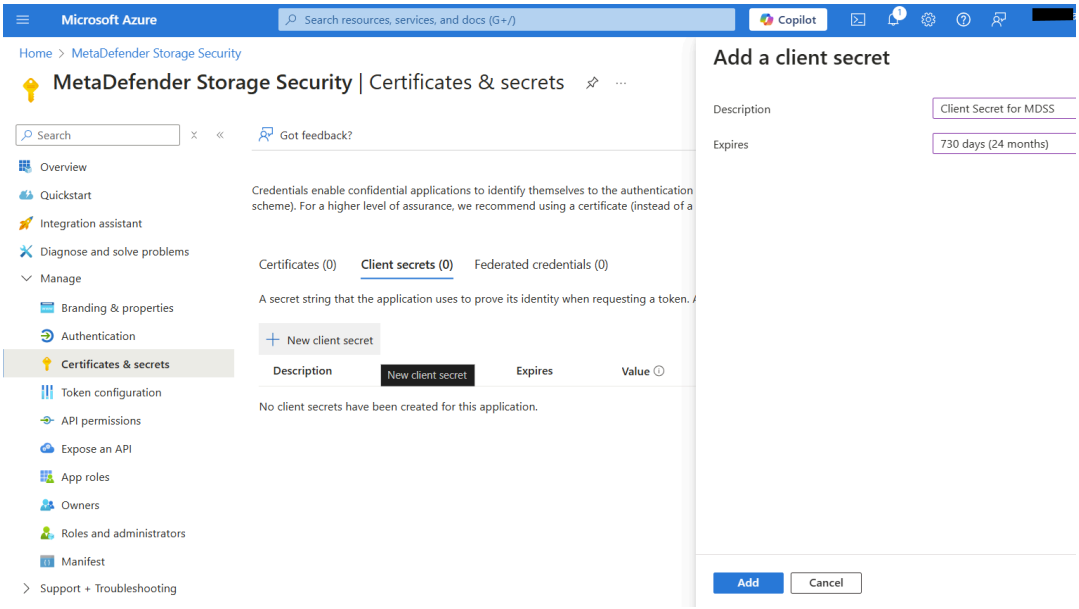
11. Select **Application permissions**



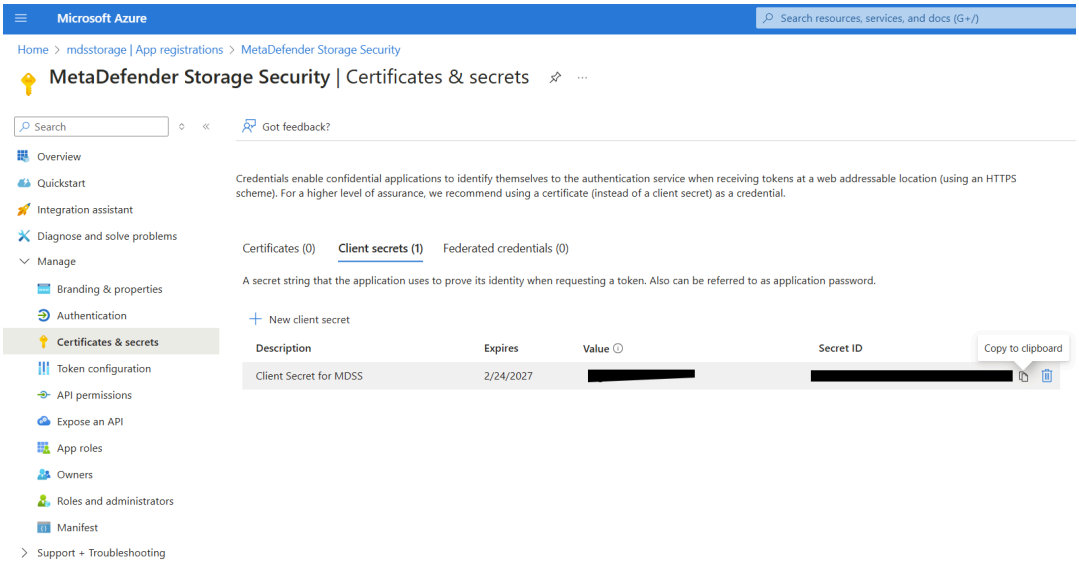
12. In order for MetaDefender Storage Security to correctly work please add the following permissions:
    1. Files.ReadWrite.All
    2. Group.Read.All
    3. Sites.Read.All
    4. User.Read.All
13. When ready, click **Add permissions**
14. We need to grant admin consent for the added permissions



15. Now that the required permissions are in place, you are ready for the final step. Please navigate to **Certificates & secrets** from the left-side menu
16. Click **New client secret** and choose Expires in 24 months



17. Click **Add** and then your will need to make a copy of the generated **secret key** because it will not be available later
18. Now that you have the **Tenant ID,** the **Client ID** and the **Client Secret Key** you can go back in **MetaDefender Storage Security** and finish the OneDrive integration. Congratulations!



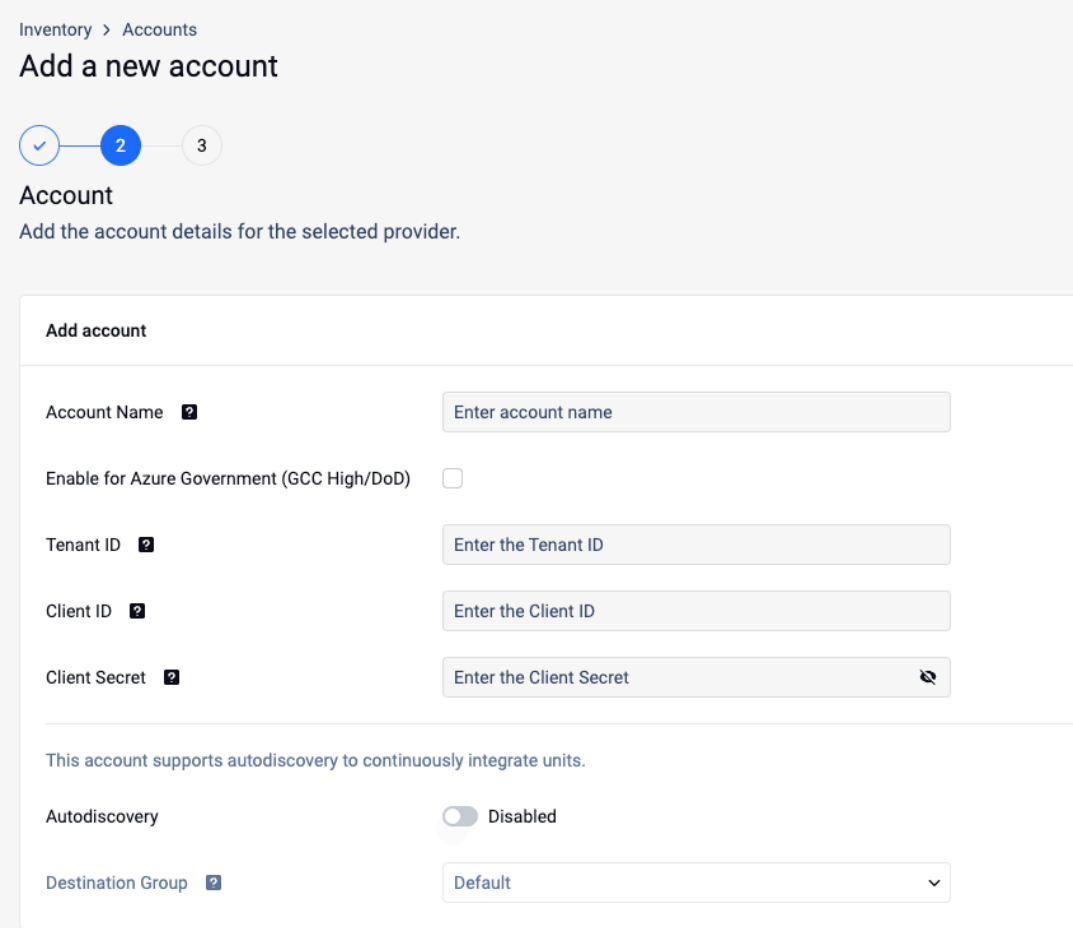
Add a SharePoint Online Storage

In order to integrate Microsoft SharePoint service with MetaDefender for Secure Storage you will be required to create an application registration, assign the necessary permissions and then generate a secret key. Please follow these steps:

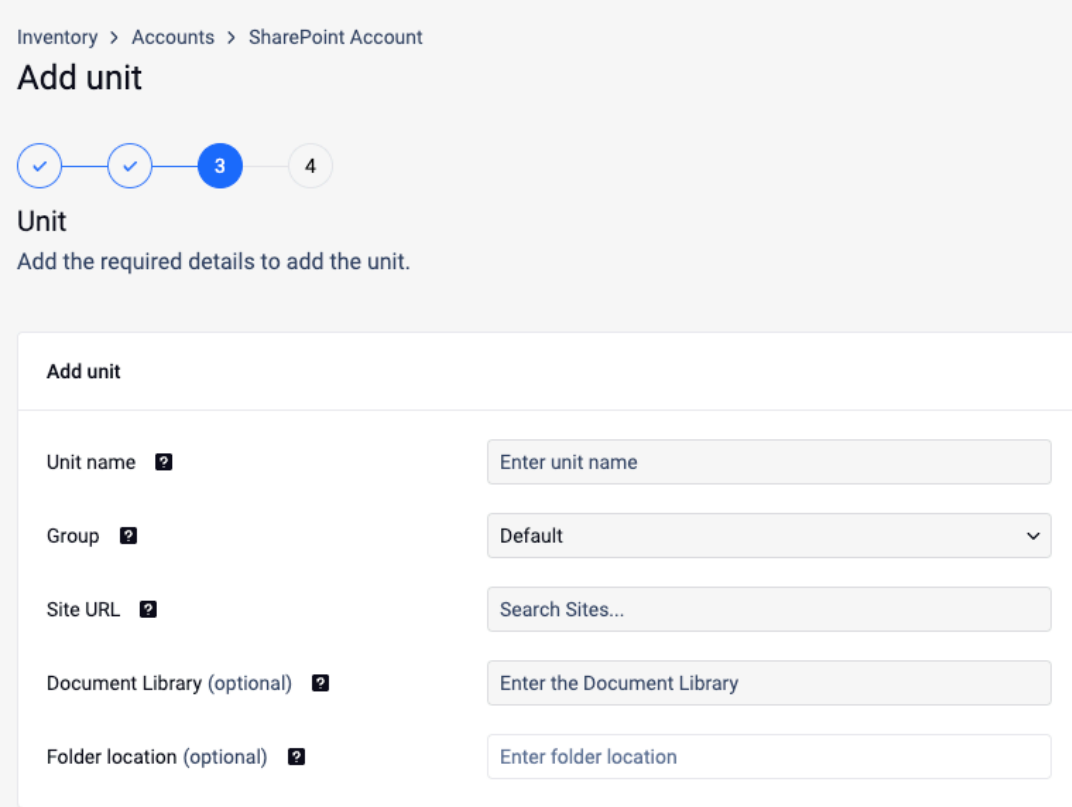
> **Info**
>
> Please note this page is **applicable only for SharePoint Online** and not applicable for SharePoint On Premise. For a guide to integrating a Sharepoint On-Premise deployment, consult this page: Add a SharePoint On-Premise Storage.

> **Info**
>
> When the upload process takes more than a few seconds, SharePoint creates a temporary file having the same name as your uploading file and the size of 0 bytes. By default, we will discover that file and show it into the list of discovered files. Once the upload is complete, we will discover the original uploaded file and process it accordingly. To skip the discovery of the temporary empty file created, please follow these instructions: How to handle empty objects in Discovery?

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **SharePoint** and click **Next**



2. Give your account a name so you can easily identify it later
3. Check **Enable for Azure Government (GCC High/DoD)** only if your organization uses Microsoft 365 GCC High or DoD environments. Leave unchecked for commercial or standard GCC environments.
4. Enter your **Tenant ID**. Scroll to the end of this page for more details.
5. Enter your **Client ID.** Scroll to the end of this page for more details.
6. Enter your **Client Secret.** Scroll to the end of this page for more details.
7. Click Next, finish adding the account and then add a new unit to this account

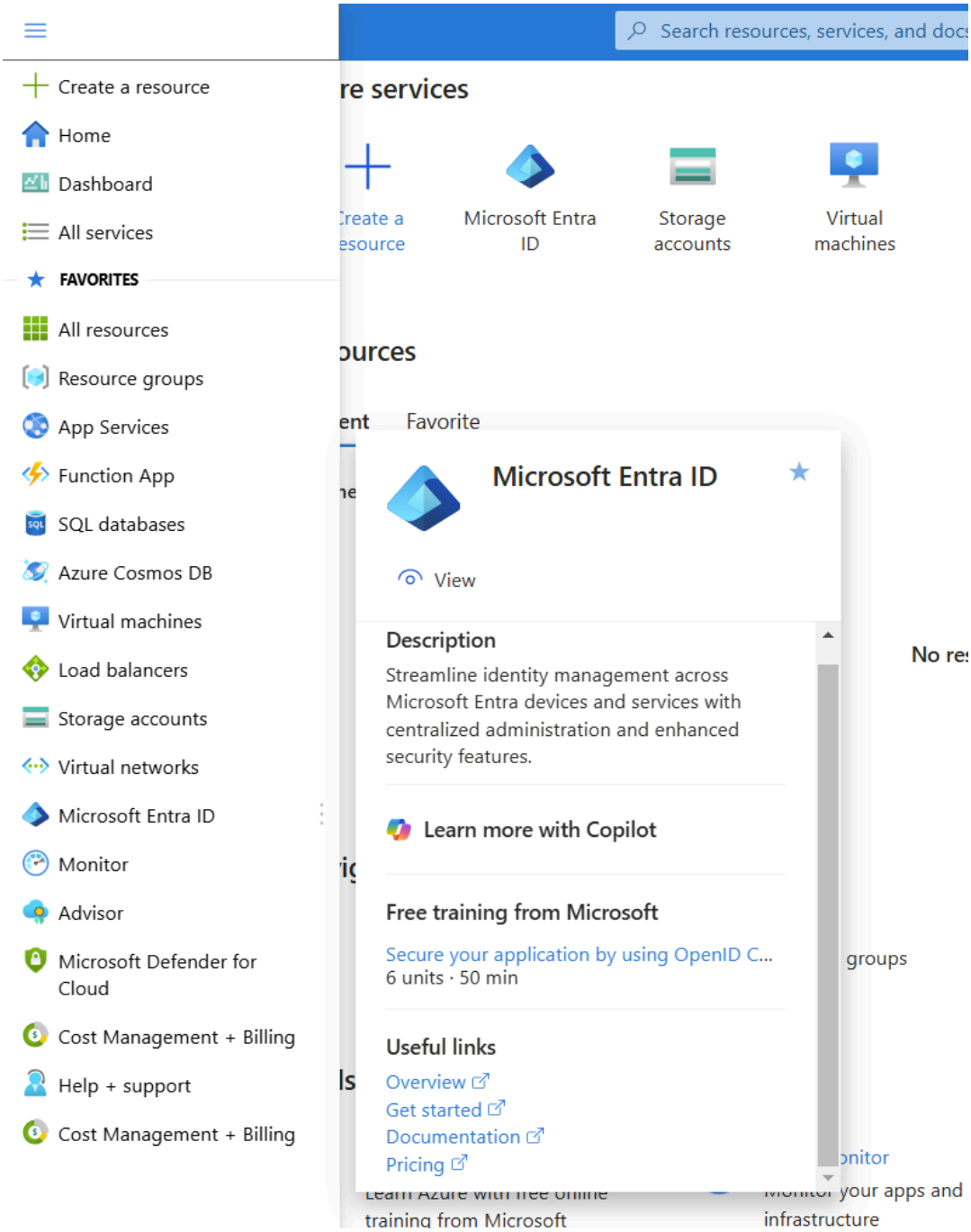


8. Give your unit an unique name
9. Assign your unit to a Group
10. Enter the **URL of a site** you wish to process files and folders.
11. Enter the **name of a Document Library** from the Site URL if you wish to process files and folders from a specific Document Library. This is optional. In case is not provided, we will process the files and folders from all the **Public Document Libraries** (we do not scan the private ones) that belong to the Site provided at step 10.
12. Enter the **name of a Folder** if you wish to process files and folders only from a specific folder, inside a specific Document Library. To be able to add a folder location, you need to provide a Document Library (11).
13. Select **Continue** in order to finish the process. If the data you entered is valid, a new SharePoint storage will be created. If is not valid, a specific error message will inform you.

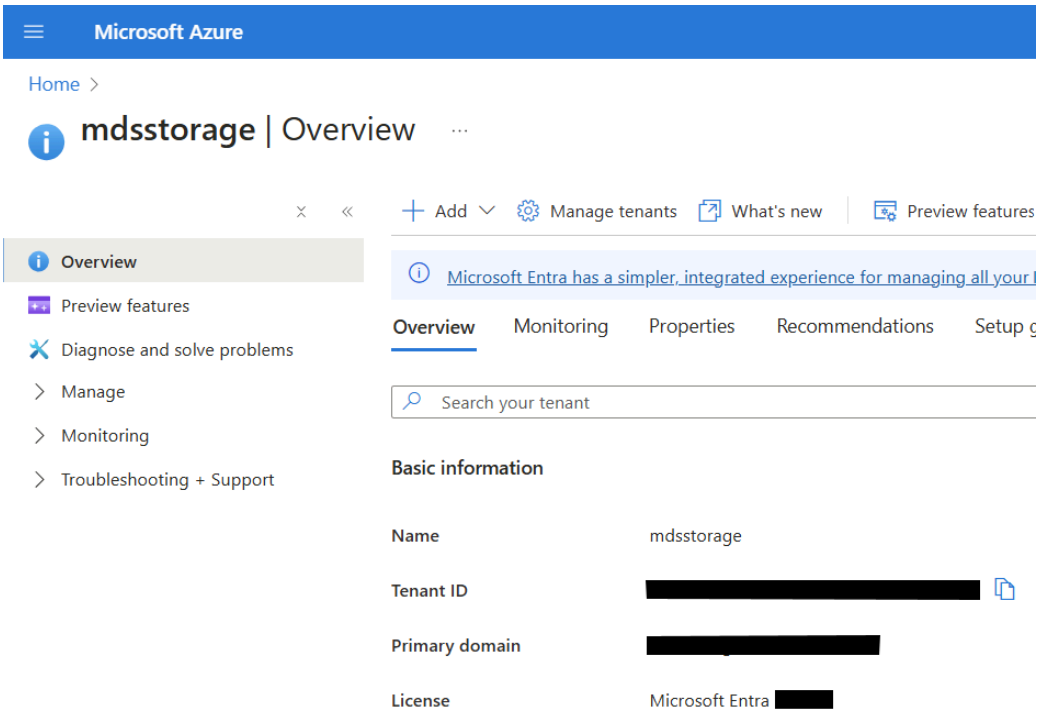
How to create a SharePoint application

In order to process our SharePoint files and folders with MetaDefender for Secure Storage you will be required to add a new app registration in your Microsoft Entra ID Azure Portal

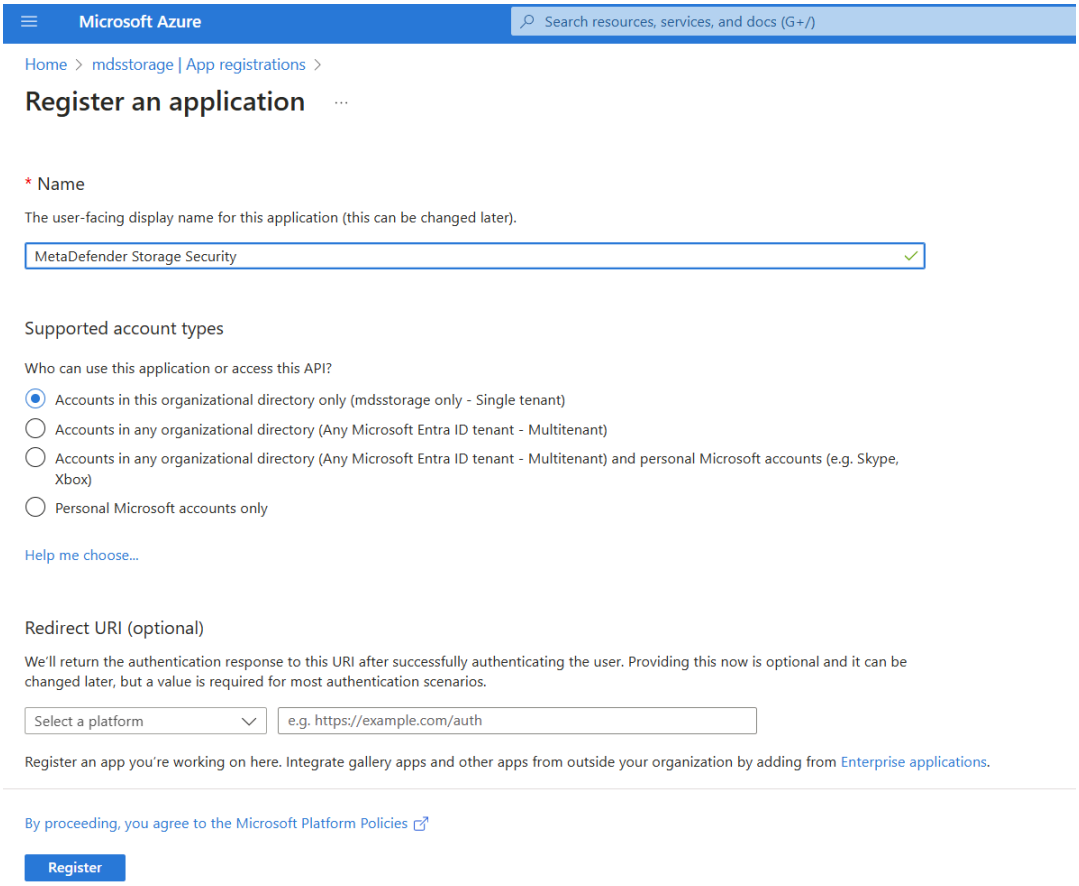
1. Log in to Azure Portal and from the left navigation menu choose **Microsoft Entra ID**


Search resources, services, and docs
Create a resource
Home
Dashboard
All services
FAVORITES
All resources
Resource groups
App Services
Function App
SQL databases
Azure Cosmos DB
Virtual machines
Load balancers
Storage accounts
Virtual networks
Microsoft Entra ID
Monitor
Advisor
Microsoft Defender for Cloud
Cost Management + Billing
Help + support
Cost Management + Billing
re services
Create a resource
Microsoft Entra ID
Storage accounts
Virtual machines
ources
ent
Favorite
Microsoft Entra ID
View
Description
Streamline identity management across Microsoft Entra devices and services with centralized administration and enhanced security features.
Learn more with Copilot
Free training from Microsoft
Secure your application by using OpenID C...
6 units · 50 min
Useful links
Overview
Get started
Documentation
Pricing
No res
groups
onitor
your apps and infrastructure
training from Microsoft

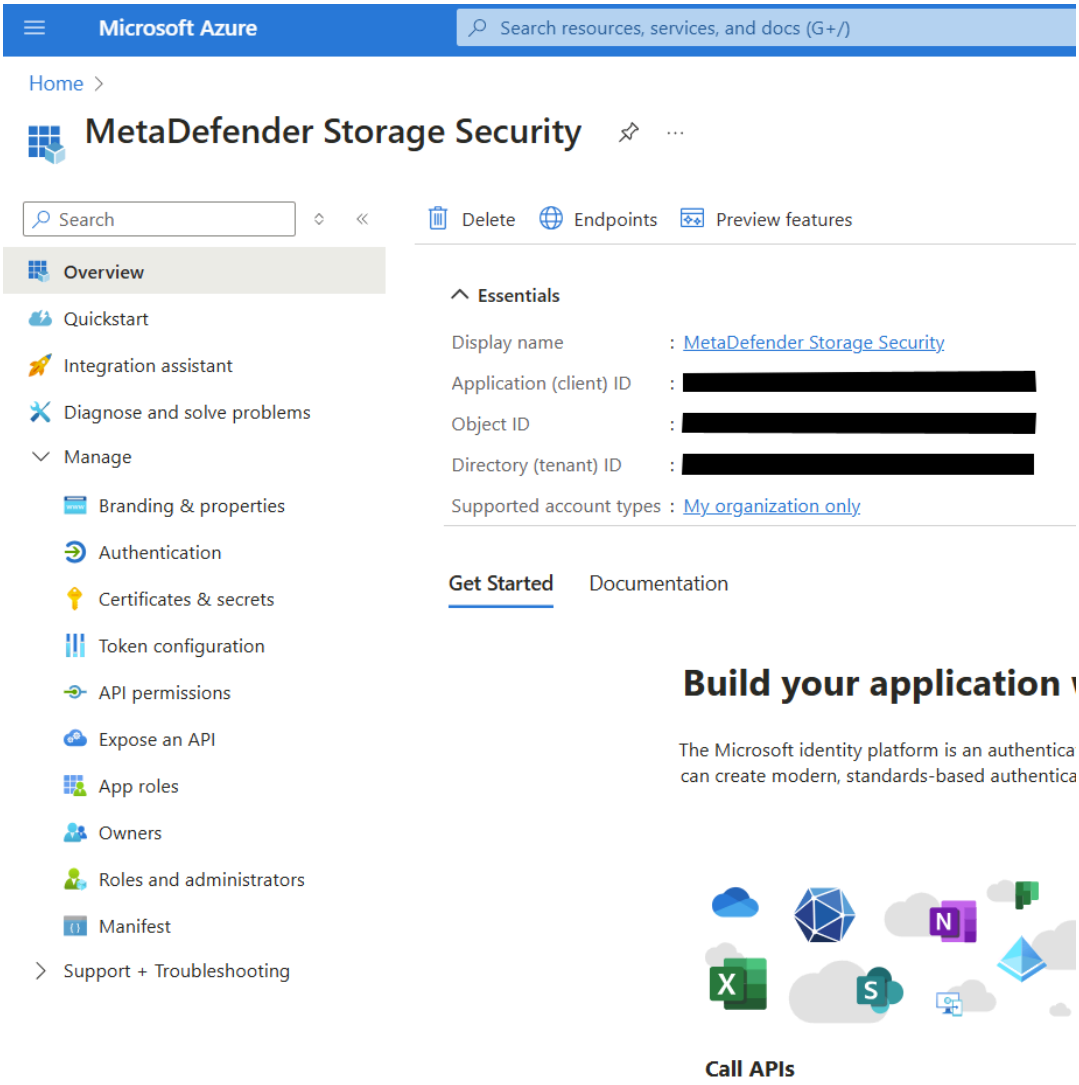
2. Make a copy of the **Tenant ID** from the overview page



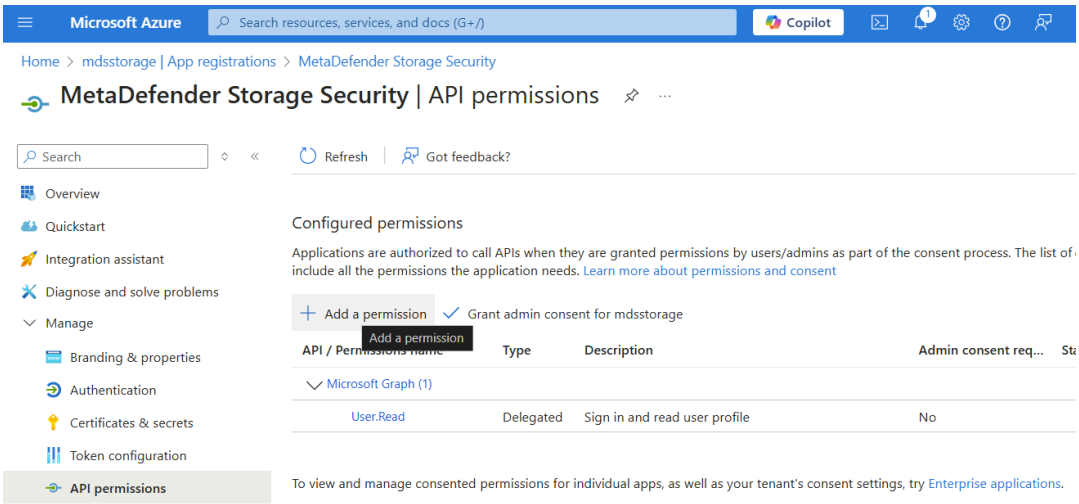
3. From the left side menu choose **App registrations**
4. Click **New registration**
5. Give your app a name so you can easily identify it



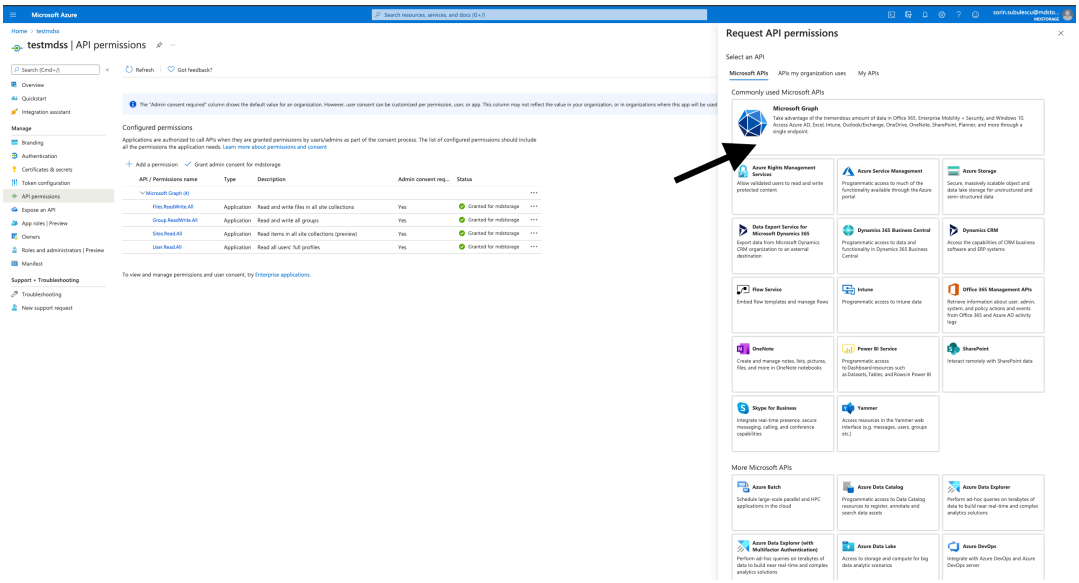
6. There is no need to modify the other properties. When ready, please click **Register**
7. From the **Overview** page of your newly created application, make a copy of **Application (client) ID **



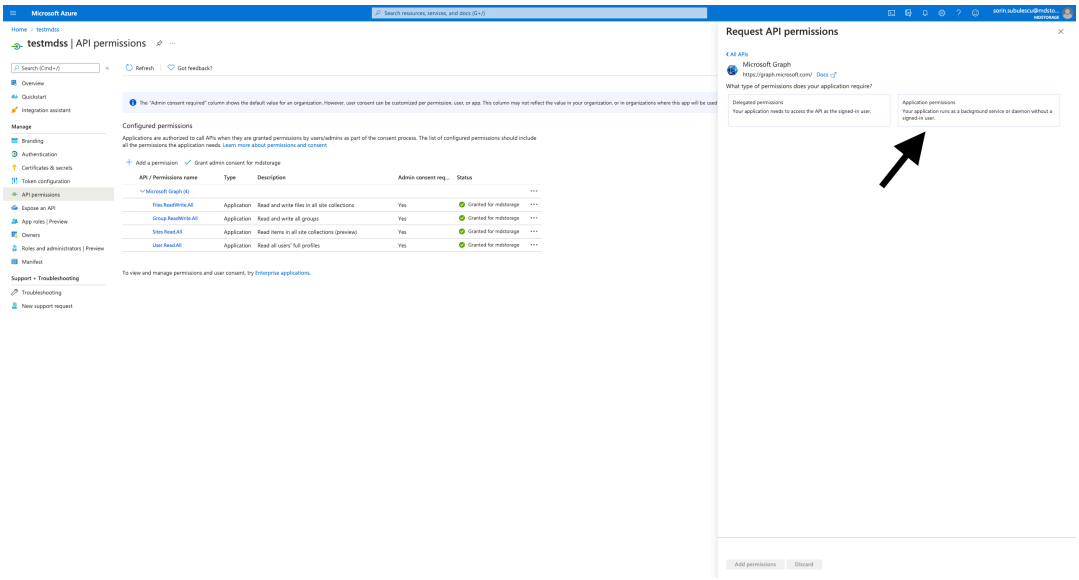
8. You now need to specify which permissions should the application have. To do this, please navigate to **API permissions** from the left-side menu
9. Click **Add a permission **



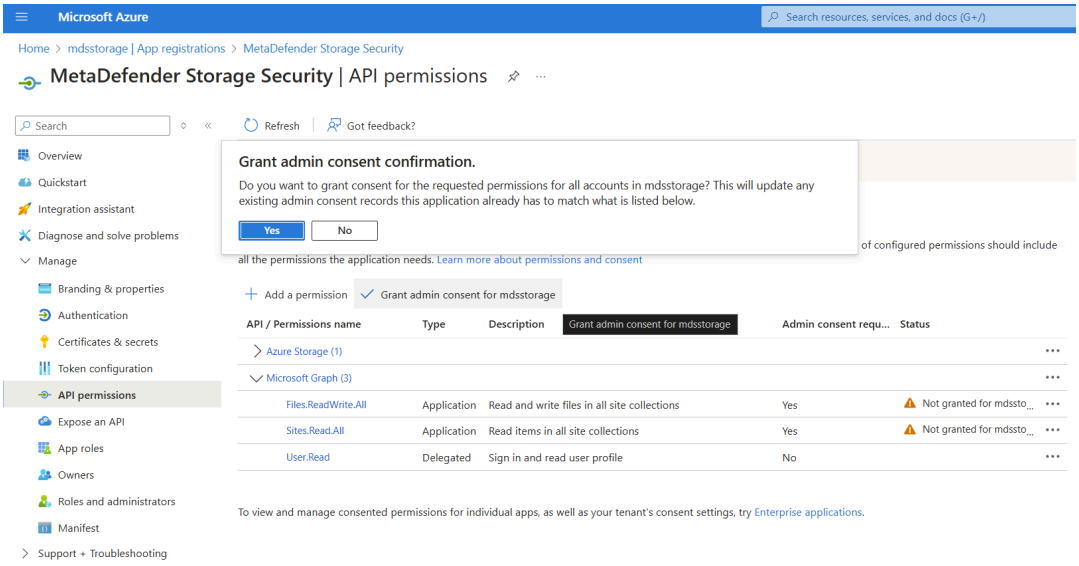
10. Select **Microsoft Graph** from the right-side menu



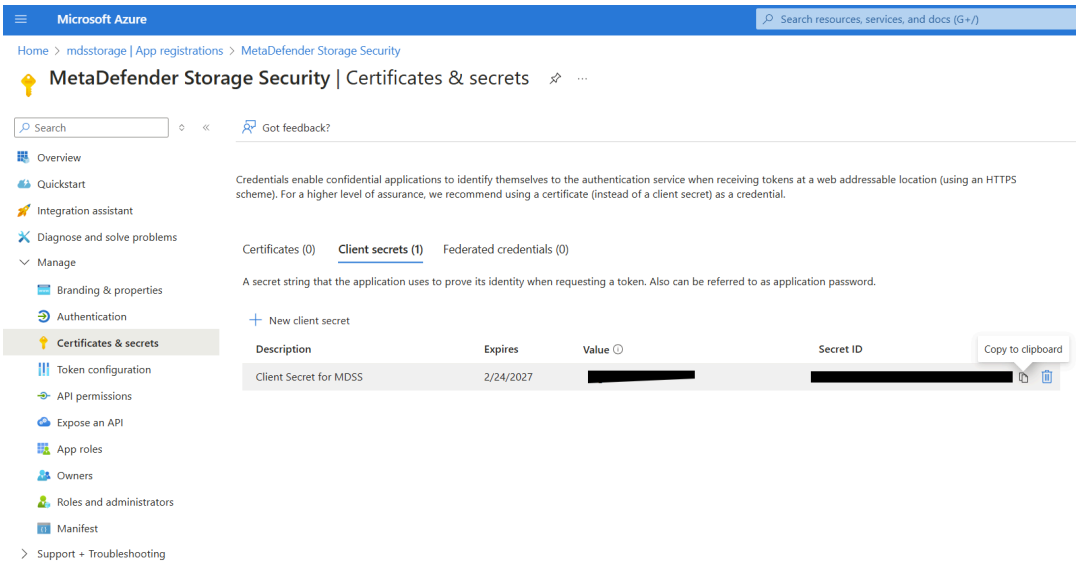
11. Select **Application permissions **



12. In order for MetaDefender Storage Security to correctly work please add the following permissions:
    1. Files.ReadWrite.All
    2. Sites.Read.All
13. When ready, click **Add permissions**
14. We need to grant admin consent for the added permissions



15. Now that the required permissions are in place, you are ready for the final step. Please navigate to **Certificates & secrets** from the left-side menu
16. Click **New client secret** and make sure that it **will not expire** as long as MetaDefender Storage Security is using it
17. Click **Add** and then you will need to make a copy of the generated **secret key** because it will not be available later



18. Now that you have the **Tenant ID,** the **Client ID** and the **Client Secret Key** you can go back in **MetaDefender Storage Security** and finish the SharePoint integration. Congratulations!

How to Enable Autodiscovery

This guide provides step-by-step instructions for enabling Autodiscovery for SharePoint Online in MetaDefender Storage Security.

Autodiscovery automatically detects and adds SharePoint Online sites as storage units, saving time and reducing configuration errors.

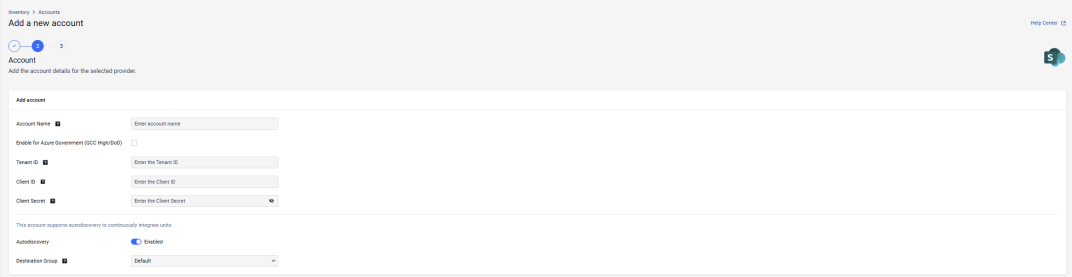
Prerequisites

Before enabling Autodiscovery, ensure you have:

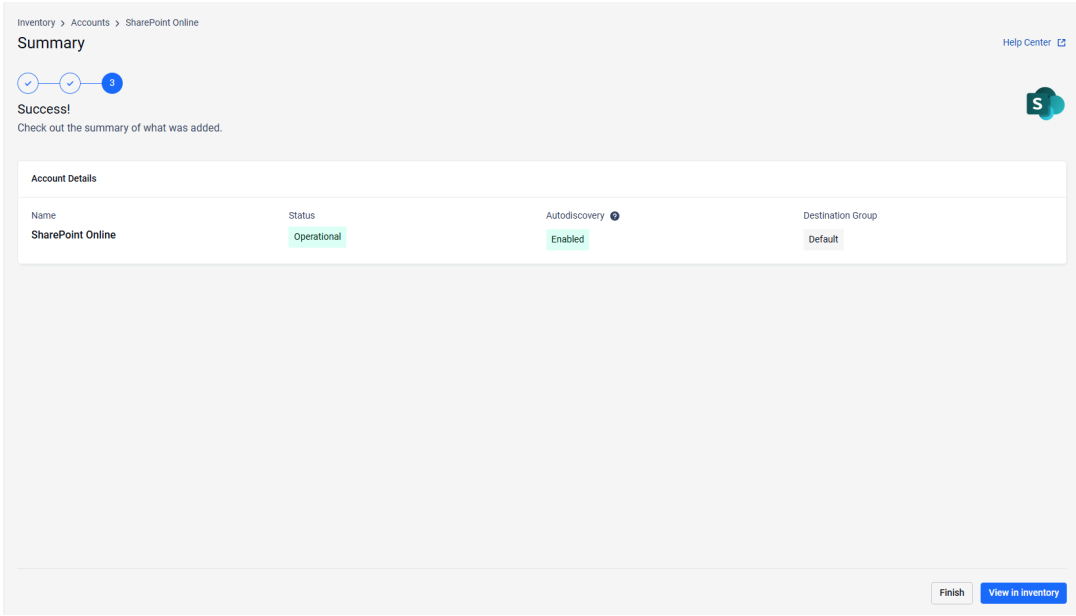
1. Valid SharePoint Online credentials with sufficient permissions
2. Permissions as stated in Add a SharePoint Online Storage

Method 1: Enable Autodiscovery When Adding an Account

- Navigate to **Inventory** › **Accounts** and click **Add account**
- Select the **SharePoint** provider
- Fill out the form with valid SharePoint Online credentials
- Turn on the **Autodiscovery** switch and select the **Destination Group** where the discovered sites will be added



- Continue to the **Summary** step. Autodiscovery starts in the background and the discovered sites are added as storage units to the destination group. Click **View in inventory** to open the account, or **Finish**

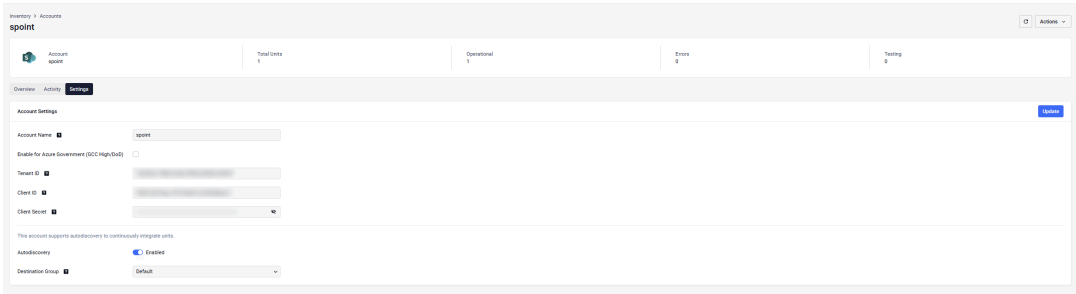


To learn more, please consult the Autodiscovery documentation.

> **ADDING SPECIFIC SITES ONLY**
>
> To add only specific sites, leave **Autodiscovery** disabled, click **Add unit** on the **Summary** step, and choose **Single unit** or **Multiple units**.

Method 2: Enable Autodiscovery for an Existing Account

- Navigate to **Inventory** › **Accounts** and open the account you want to start autodiscovery for
- Open the **Settings** tab
- Turn on the **Autodiscovery** switch and select the **Destination Group**
- Enter the **Client Secret**
- Click **Update**



Add a Microsoft Teams Storage

MetaDefender Storage Security supports scanning files shared through **Microsoft Teams** by integrating with the underlying Microsoft storage services used by Teams: **OneDrive** and **SharePoint**.

How Teams Stores Files

Teams Chat (1:1 and Group Chats → OneDrive)

When a user sends a file in a Teams chat, the file is uploaded to the sender's OneDrive

These files are stored in a folder named **Microsoft Teams Chat Files**

- Only files sent by that user appear in their OneDrive folder
- Other chat participants see read-only references to the file, but the actual file remains in the sender's OneDrive

Teams Channels (Channels → SharePoint)

- Files sent in a Teams channel are stored in the team's SharePoint site
- They appear under the Documents library, inside a folder named after the channel

Integrating OneDrive

In order to add a OneDrive storage in MetaDefender Storage Security, follow the documentation.

Recommended Configuration for Teams Chat

For scanning files sent in Teams chats, we recommend adding the user and the folder location **Microsoft Teams Chat Files.**

This configuration ensures that only files sent by that user in Teams chats are scanned.

Integrating SharePoint

In order to add a **SharePoint** storage in MetaDefender Storage Security, follow the documentation.

Recommended Configuration for Teams Channels

For scanning files shared in Teams channels, we recommend integrating with the **channel folder** inside the SharePoint site's *Documents* library

This configuration ensures that only files shared in the selected Teams channel are scanned.

Scanning with Deep CDR

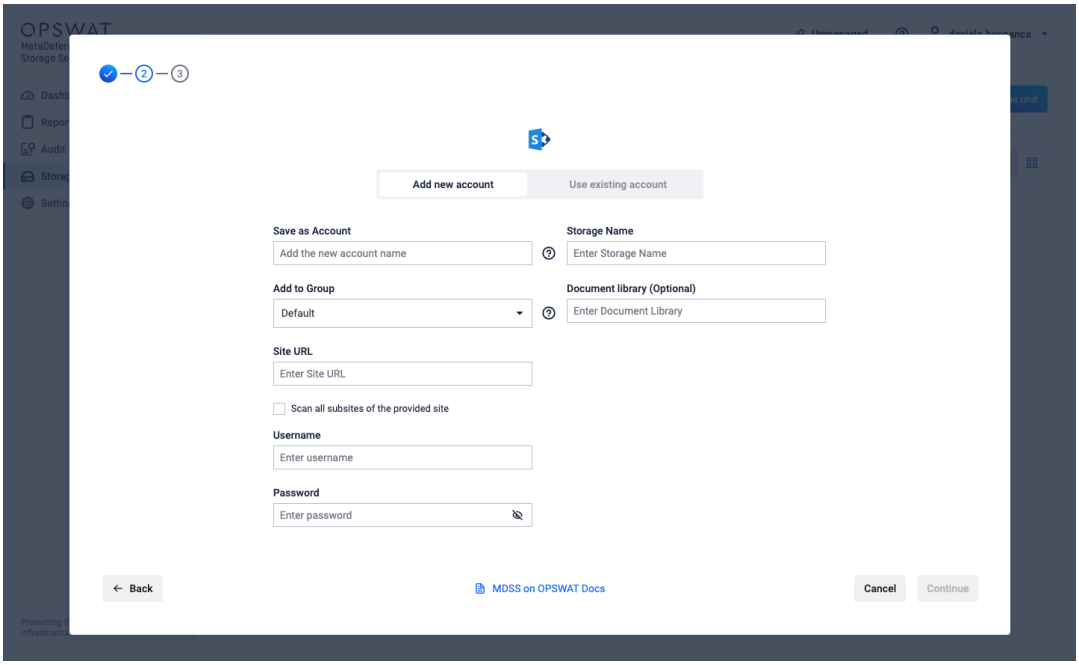
When using Deep CDR, the sanitized file will not be accessible in Teams by default.

To ensure Teams users can access the sanitized file, you must enable the **"Replace Original File"** feature, following the documentation.

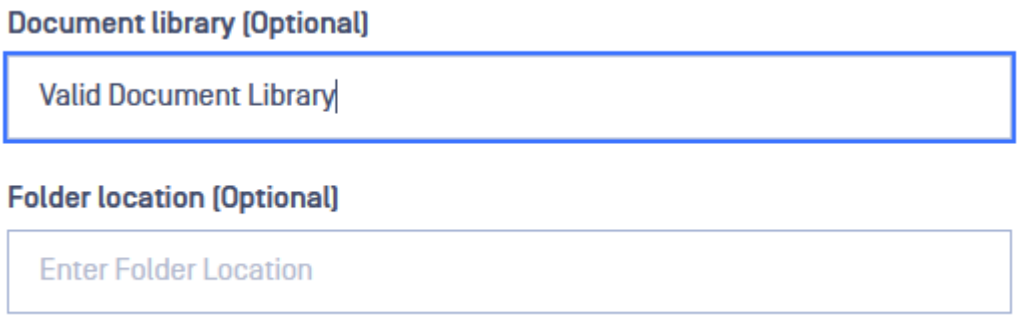
This replaces the original Teams file with the sanitized version, meaning Teams users will always access the sanitized file in their chat or channel.

Add a SharePoint On-Premise Storage

> **Info**
>
> Please note this page is **applicable only for SharePoint On-Premise** and not applicable for SharePoint Online. For a guide to integrating a Sharepoint Online deployment, consult this page Add a SharePoint Online Storage. Moreover, the SharePoint On-Premise integration is available **only on Windows MetaDefender Storage Security Deployments.**



1. Give your storage a name so you can easily identify it later.
2. Enter the **Site URL** associated with the Site you want to integrate from you Sharepoint Server.
3. Enter the **domain** and **username** of the user you want to log in as. This should be in the form **DOMAIN/USERNAME.** The user must have at least **Contribute** permissions on the site.
4. Enter the **password** associated with the user account you are logging in as.
5. Optionally, enter the name of a **Document Library** from the site URL. If this field contains a valid value, then only the files from this specific document library will be processed. If no value is entered here, files from all public document libraries belonging to the site will be processed.
6. If the **Document Library** field has a value, the **Folder Location** field will appear. Optionally, you can enter the name of a **Folder** that is part of the input document library. If this field contains a valid value, then only the files from this folder inside the document library will be processed.



7. The **Scan all subsites of the provided site** checkbox enables you to optionally explore the entire subsite hierarchy of the integrated site, allowing for the processing of files from all document libraries within the subsites. This option is applicable only when a **Site URL** is entered without specifying a **Document Library**. If you do select one, the integration will be restricted to that particular library within the site.

> **Warning**
>
> Known Limitation: there is a heavy limitation in regards to the number of concurrent requests a user can make to the SharePoint Server. Thus, when doing remediation actions with a large number of files, the heavy throttling will cause the operations to have to wait for the SharePoint Server to free up resources and will result in a considerably long scanning time. For more information, you can access this article:
>
> To customize the values used in the error handling and retry mechanism to better suit your deployment, please visit this page for more information.

Configuring Error Handling and Retry Mechanisms for SharePoint On-Premise

This guide is for administrators and users managing the MetaDefender Storage Security integration with SharePoint On-Prem. It details crucial environment variables that control error handling, request retries, connection management, and batch processing. Properly tuning these settings is essential for maximizing your deployment's performance and reliability. Fine-tuning helps prevent SharePoint server throttling, which can occur under heavy load, ensuring smoother operations and minimizing disruptions.

Key Configuration Settings (Environment Variables)

Below are the environment variables you can configure to optimize communication with your SharePoint On-Prem server.

SHAREPOINT_ON_PREM_MAX_CONNECTIONS sets the maximum number of simultaneous connections to SharePoint On-Prem that can be held in the connection pool. Requests

exceeding this limit will wait for an existing connection to become available.

Impact

- Higher Value
  - Can lead to faster processing of requests to the same site and better performance during high-demand periods by allowing more concurrent operations.
  - Increases the risk of hitting the SharePoint server's own simultaneous connection limits, potentially causing throttling, slower responses, and service disruptions.
- Lower Value
  - Enhances connection reusability, leading to more reliable and stable request management.
  - May reduce processing speed, as fewer simultaneous connections can lead to longer wait times for requests, especially during peak usage.
- Default is set to 5 maximum connections for every pool
- Units measured in number of connections

`SHAREPOINT_ON_PREM_MAX_QUERY_RETRIES` controls how many times a failed SharePoint On-Prem query will be automatically retried. Retries use an exponential backoff strategy (increasing delays between attempts) to avoid overwhelming a struggling server

Impact

- Higher Value
  - Increases the chance of a query succeeding during temporary server issues.
  - Can lead to longer overall operation times if failures are frequent or persistent.
- Lower Value
  - Reduces time spent retrying, leading to quicker finalization of operations (either success or failure)
  - May result in a higher rate of failed operations if transient issues are common, as the system gives up sooner.
- Default is set to 5 retries per query
- Configurable Range goes from 0 to 8 (MetaDefender Storage Security enforces a hard upper limit of 8 retries)
- Units measured in number of retries

SHAREPOINT_ON_PREM_CIRCUIT_BREAKER_THRESHOLD defines the number of consecutive connection failures to SharePoint that must occur before the circuit breaker "opens." Once open, connection attempts are temporarily halted to prevent overwhelming an unhealthy SharePoint server.

Impact

- Higher Value
  - Makes the system more tolerant of repeated transient issues before halting requests
  - Risks overwhelming an unhealthy SharePoint server for a longer duration, potentially worsening performance issues and recovery times.
- Lower Value
  - Reacts more quickly to potential SharePoint server issues, helping to prevent server overload and allowing for faster recovery.
  - May lead to more frequent (though temporary) interruptions in service if transient issues are common.
- Default is set to 5 consecutive failed requests
- Units measured in number of failures

SHAREPOINT_ON_PREM_CIRCUIT_BREAKER_DURATION_SECS specifies how long (in seconds) the circuit breaker remains open after tripping. During this "cooldown" period, all requests to SharePoint are automatically rejected. After the duration, the circuit enters a "half-open" state to test if the SharePoint server has recovered.

Impact

- Higher Value
  - Provides a longer recovery window for a significantly troubled SharePoint server before new connection attempts are made
  - Results in a longer period of unavailability for users trying to access SharePoint resources.
- Lower Value
  - Reduces downtime by attempting to reconnect to SharePoint sooner.
  - Risks repeatedly tripping the circuit breaker if the server hasn't fully recovered, potentially increasing load on an unstable server.
- Default is set to 30 seconds

SHAREPOINT_MAX_CONCURRENT_REQUESTS controls the maximum number of SharePoint requests that MetaDefender Storage Security will process simultaneously. This helps prevent overwhelming the SharePoint server.

Impact

- Higher Value
  - Allows more requests to be processed at once, potentially improving throughput and performance during high demand.
  - Increases the risk of overwhelming the SharePoint server, which can lead to throttling or degraded performance.

- Lower Value
  - Helps prevent server overload and ensures more stable performance, especially during peak usage.
  - May result in slower overall processing times and longer wait times for users, as requests are queued.
- Default is set to 5 concurrent requests
- Units: Number of concurrent requests

`SHAREPOINT_BATCH_SIZE` specifies the maximum number of SharePoint requests to be processed in a single batch operation. Batching optimizes performance by reducing round-trips but needs to be balanced against server resource capacity

Impact

- Higher Value
  - Can improve efficiency and performance by processing more requests in each interaction with SharePoint.
  - Larger batches consume more resources and increase the risk of overwhelming the SharePoint server, potentially leading to throttling.
- Lower Value
  - Helps maintain stability and prevent resource saturation on the SharePoint server.
  - May lead to increased overall processing times due to a higher number of smaller batch operations.
- Default is set to 10 requests per batch.

SHAREPOINT_BATCH_INTERVAL_MS defines the time interval in milliseconds between the processing of SharePoint request batches. This determines how frequently the system checks for and processes queued requests.

Impact

- Higher Value
  - Reduces the frequency of batch processing, which can lessen the load on the SharePoint server, especially during peak times.
  - May lead to increased wait times for users as requests are queued for longer before being processed.
- Lower Value
  - Improves responsiveness and user experience by processing queued requests more frequently.
  - Increases the load on the SharePoint server due to more frequent processing cycles, potentially leading to resource saturation or throttling if the interval is too short.

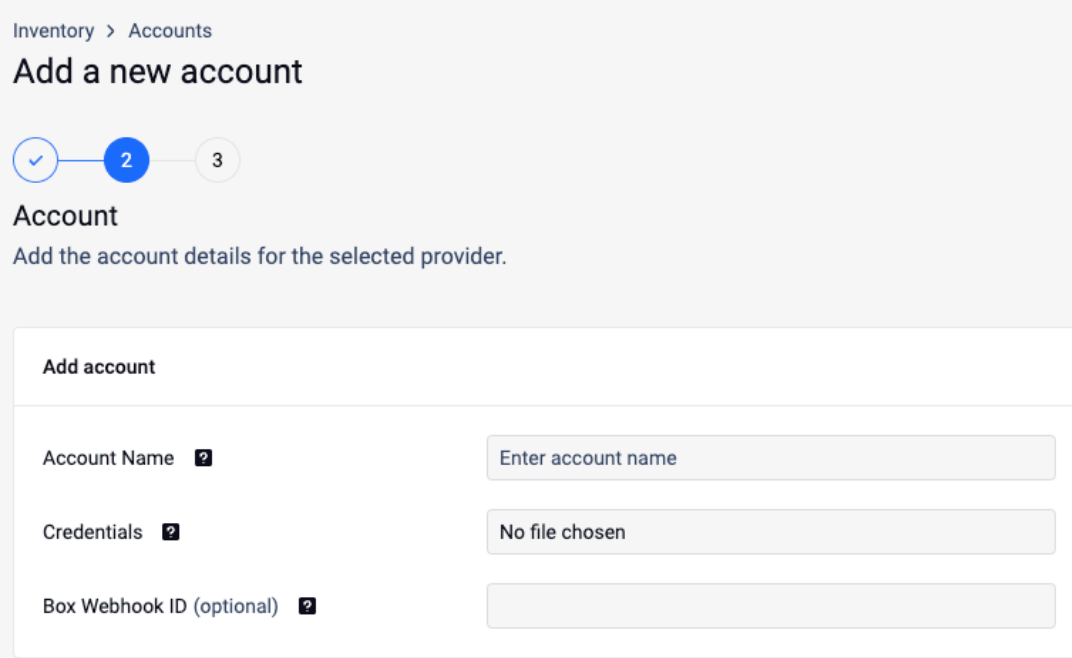
- Default is set to 1000 milliseconds (1 second)

Add a Box Storage

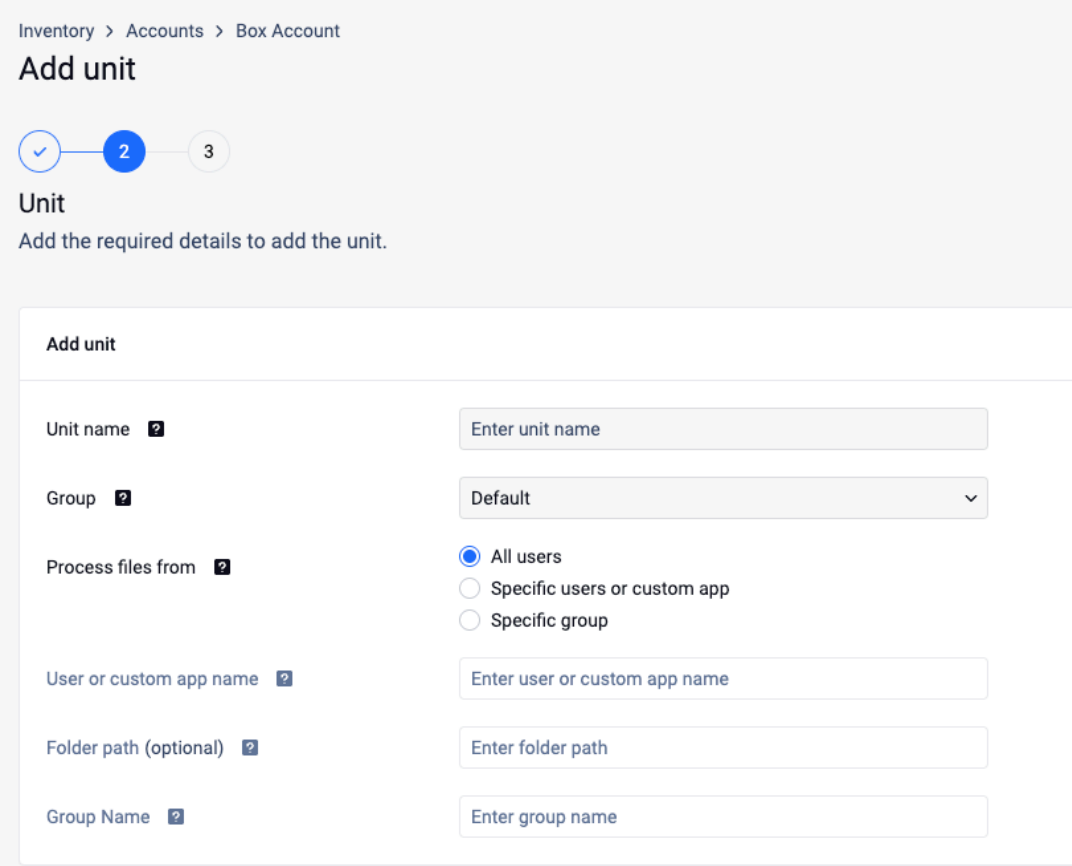
In order to integrate Box with MetaDefender Storage Security please follow these steps:

> **Info**
>
> Steps 4 and 5 are optional. Perform them in order to set up a specific location for file processing.

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **Box** and click **Next**



2. Give your account a name so you can easily identify it later
3. Upload your **Credentials** file. Scroll down for more details.
4. (Optional) Enter the Webhook ID, you can find more details in this page Event-based handling for Box
5. Click Next and after account is successfully created, please add a unit to it



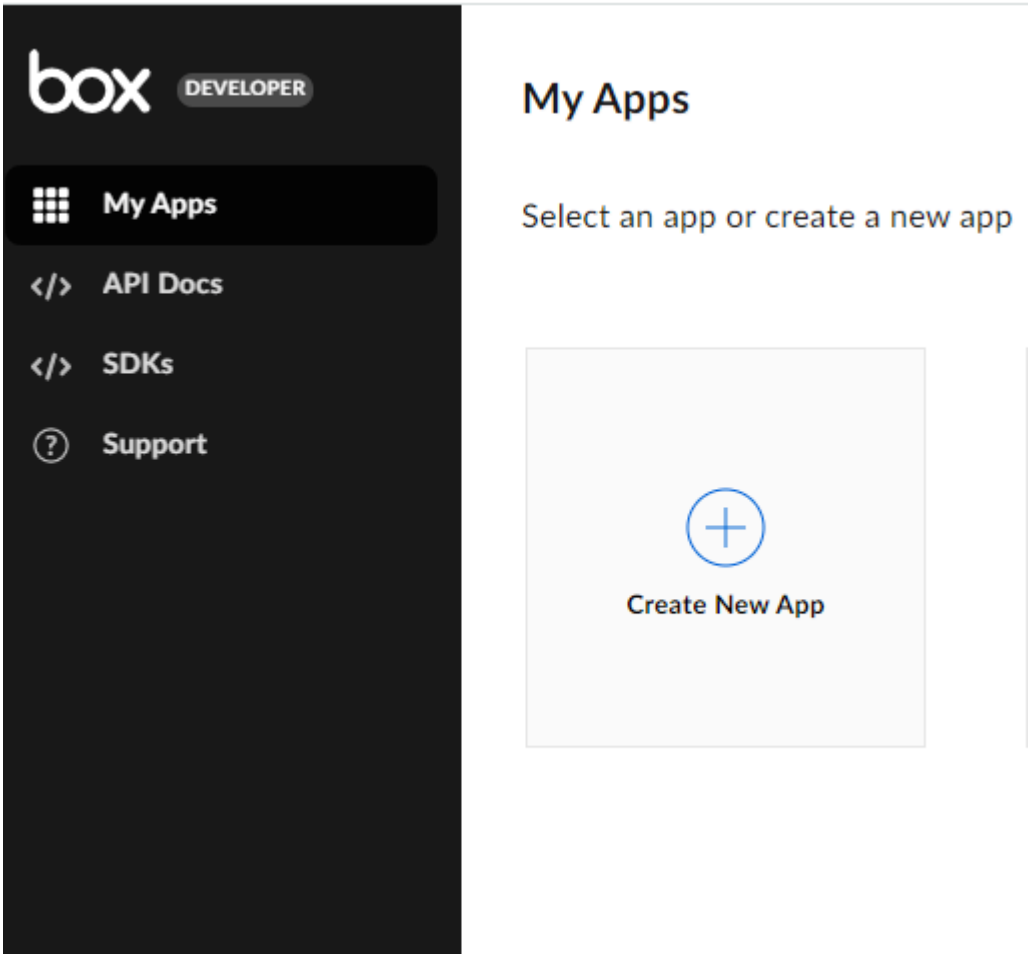
5. Enter an unique name for your unit.
6. Assign the unit to a Group.
7. Set up a specific location as a source of files that should be processed By MetaDefender Storage Security
    1. **All Users** - process files from all users created by the custom application, from the application itself and also from all the Enterprise users, depending on the Application Access provided
    2. **A specific user or custom app** - process only the files from a specific user/custom app. Additionally you can set up a location as a source of files, by filling in the full path of the folder. Make sure it's a valid path corresponding to the User or Custom App name you previously added. Note that it could take up to 10-15 minutes for newly added folders to be discovered.
    3. **A specific group***-* process only the files from a group of users
8. Select **Add** in order to finish the process.

> **Info**
>
> For most Box applications "All Files" is considered the root of the folder structure. To be able to scan entire structure of folders from a user or custom application, Folder Path field should remain empty.

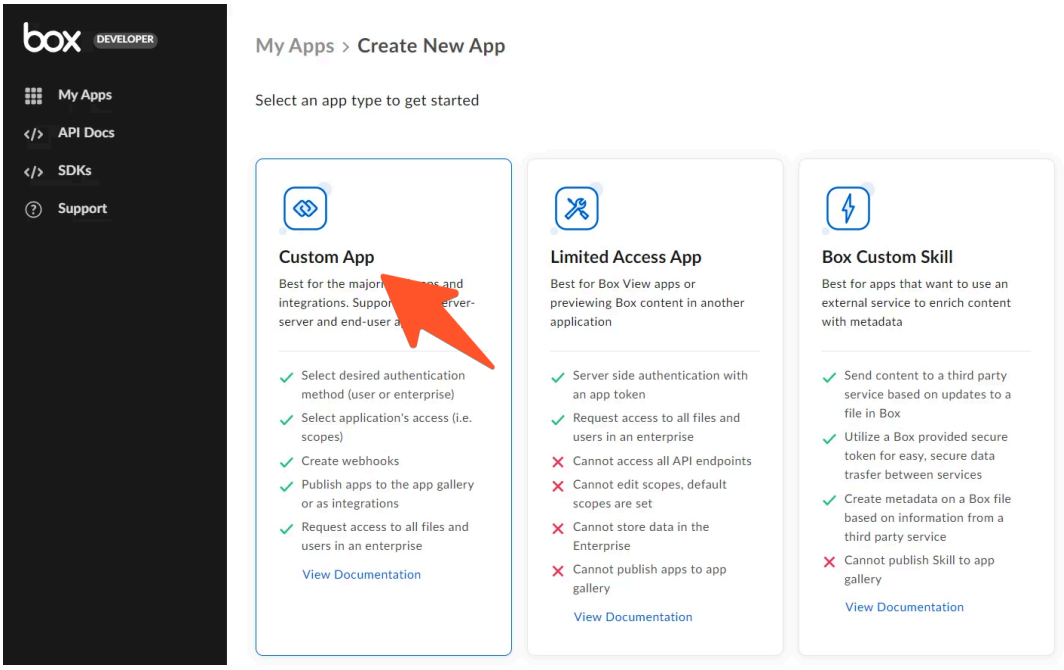
How can I generate a Public / Private key pair

To generate a file that contains all the necessary information to connect MetaDefender Storage Security to your Box storage please follow these steps:

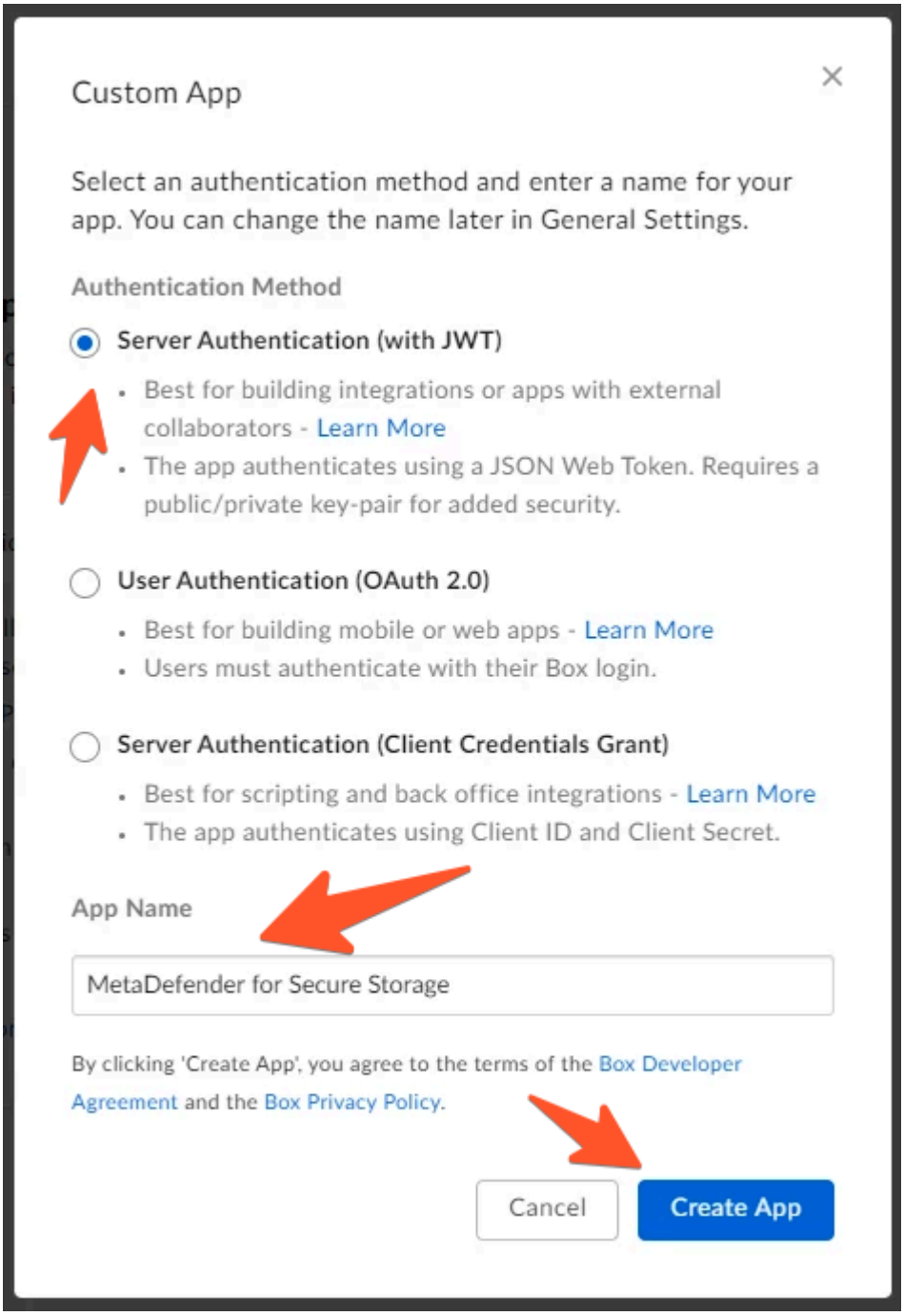
1. Log in to **Box** and then go to Dev Console
2. From the console, choose **Create New App**



3. Select **Custom App** and click **Next**

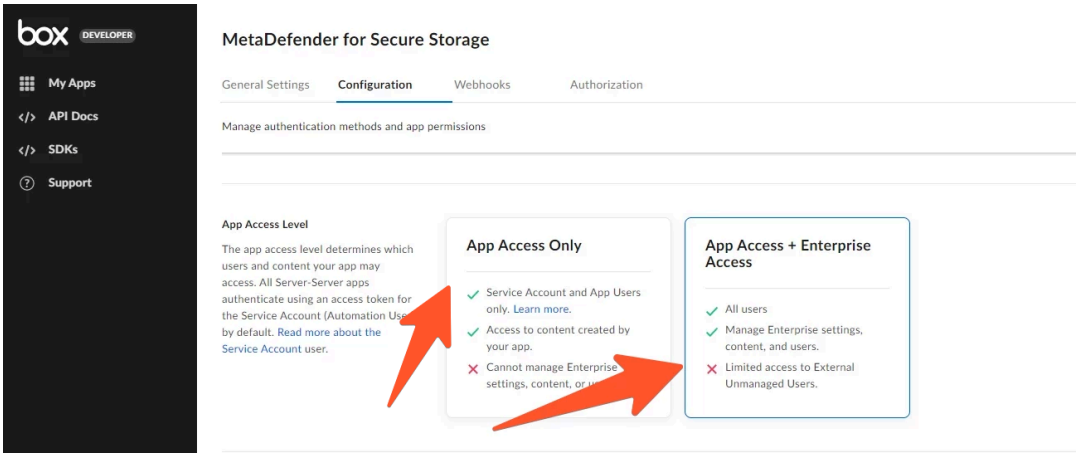


4. Choose **OAuth 2.0 with JWT (Server Authentication)** as the authentication method and give your app a name, for example **MetaDefender Storage Security** and then click **Create App**

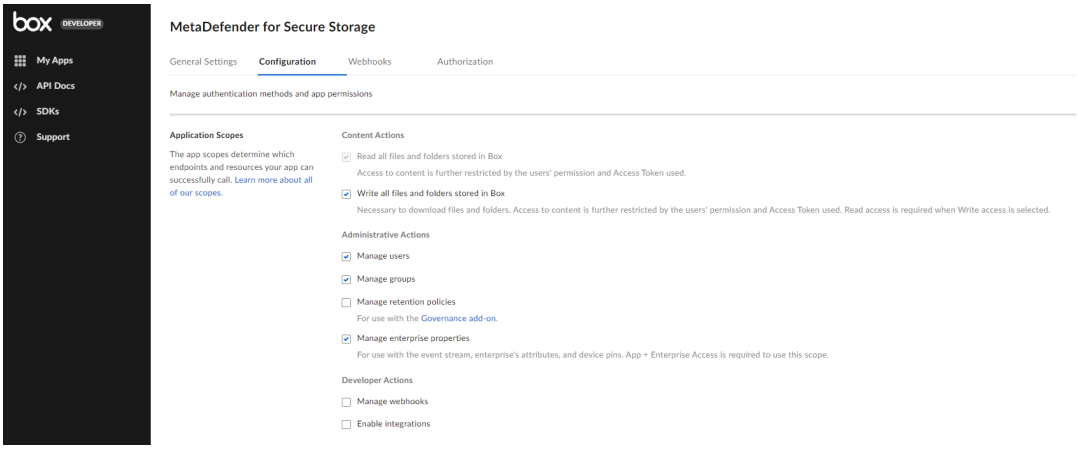


5. You will receive a message saying that your app has been successfully created. Click **View Your App**.
6. You will be redirected to the app **General Settings** page, switch to **Configuration**.
7. Scroll down to the **Application Access** section and choose **App Access Only** or **App Access + Enterprise Access**. Selecting **App Access only** will allow MetaDefender Storage Security to be able to process files from the users created by the custom application and also from the application itself. Selecting **App Access + Enterprise Access** will allow MetaDefender Storage Security to be able to process files from the

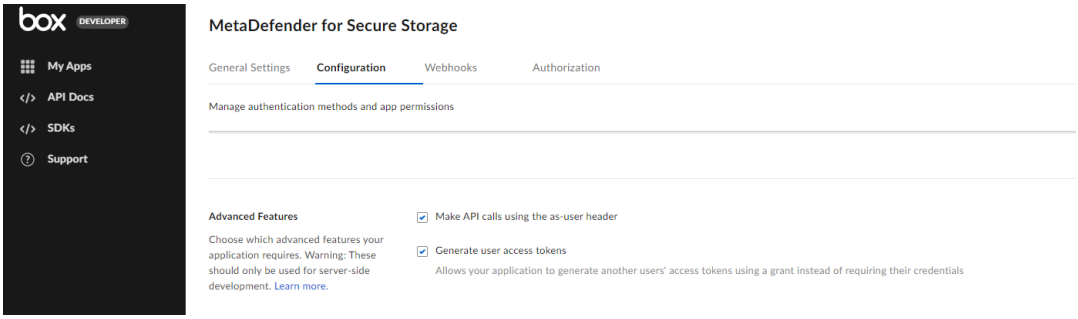
users created by the custom application, from the application itself and also from the Enterprise users.



8. Scroll down to the **Application Scopes** section and make sure that the following permissions are checked:
    1. Read all files and folders stored in Box
    2. Read and write all files and folders stored in Box
    3. Manage users
    4. Manage groups

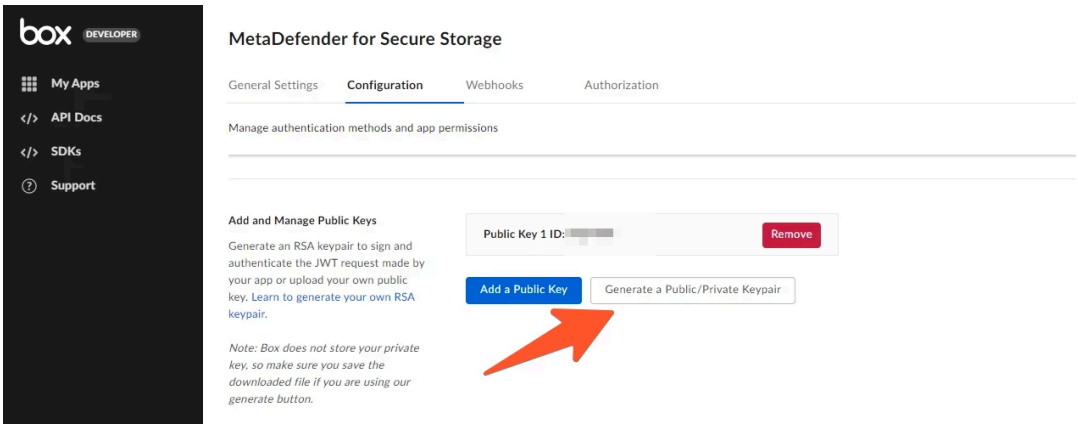


9. Next, please scroll down to the **Advanced Features** section and make sure that both **Perform Actions as Users** and **Generate Users Access Tokens** are enabled

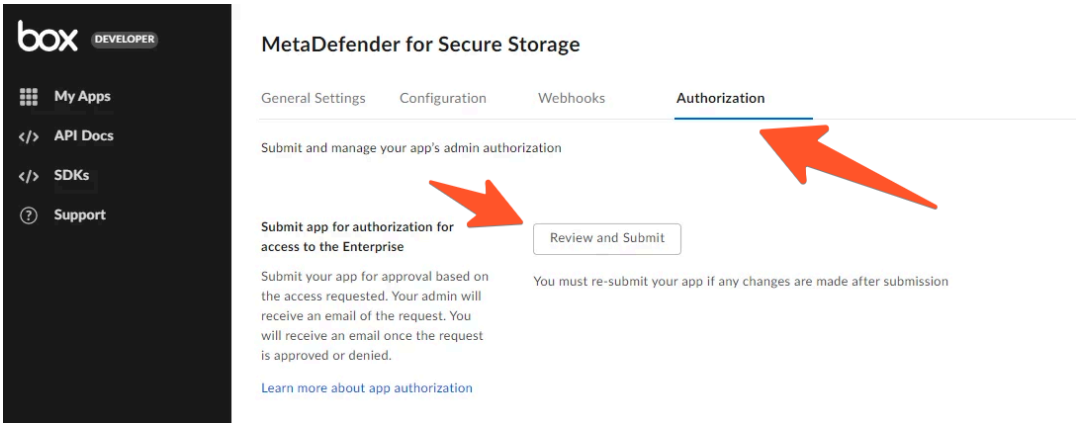


10. Make sure you **save your changes.**
11. The next step will be to generate a Public/Private Keypair

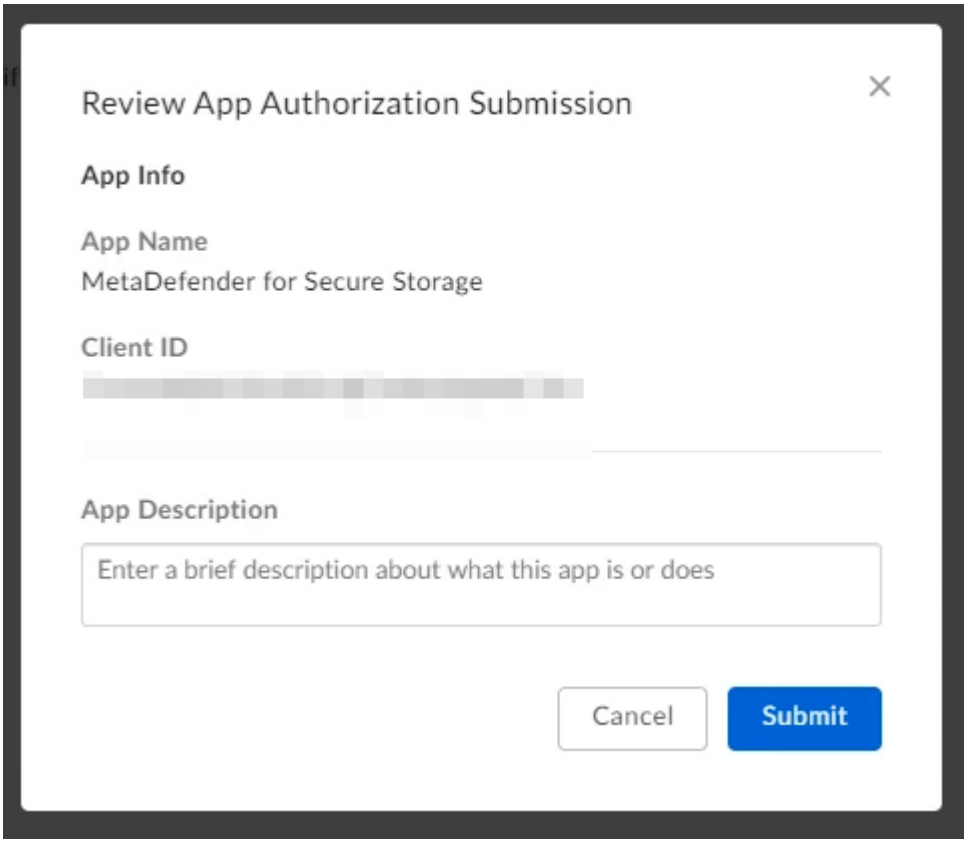
Note: When performing this action you may be required to re-authenticate using a two-factor authentication method. If this happens, after re-authenticating, please go back and click **Generate Public/Private Keypair** again



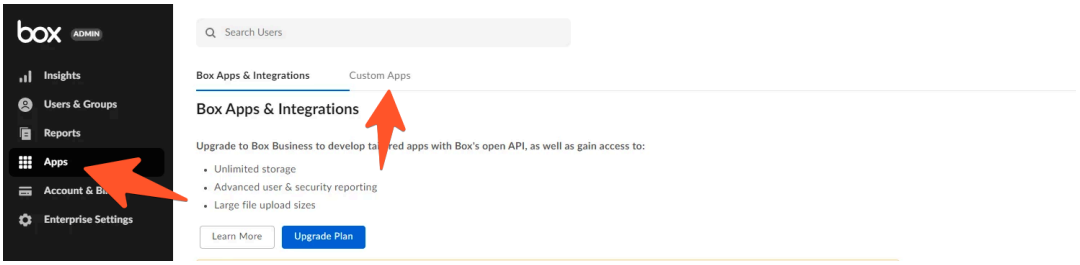
12. A configuration file with all of your app settings will be downloaded. Please store it securely as you will need it a bit later
13. Click to **Authorization** menu.



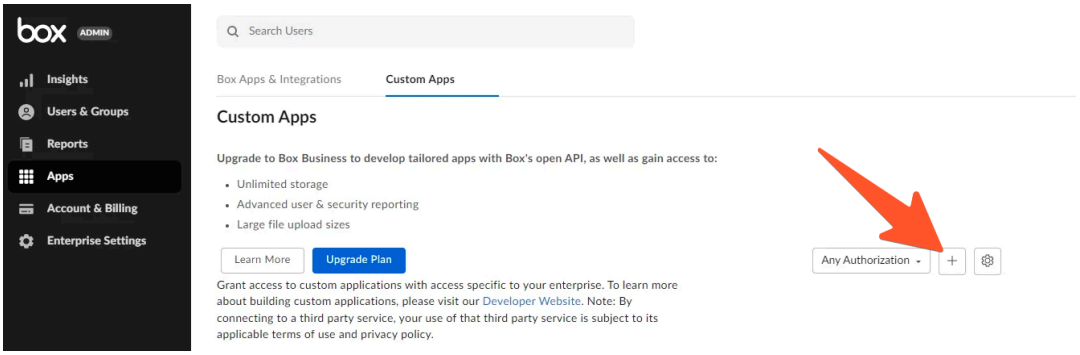
14. Click **Review and Submit**. From the popup that will appear, please make a copy of the **Client ID**

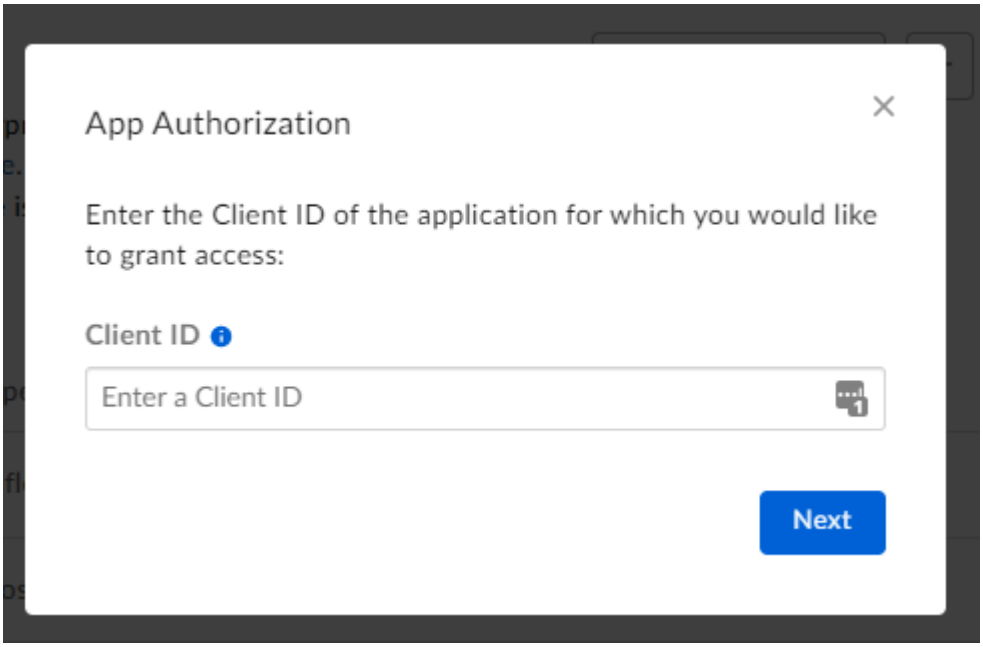


15. Click **Submit**
16. From the left-side menu, please press **Back to My Account** and switch to **Admin Console**.
17. Using the Admin Console left-side menu, navigate to **Apps** and then **Custom Apps**

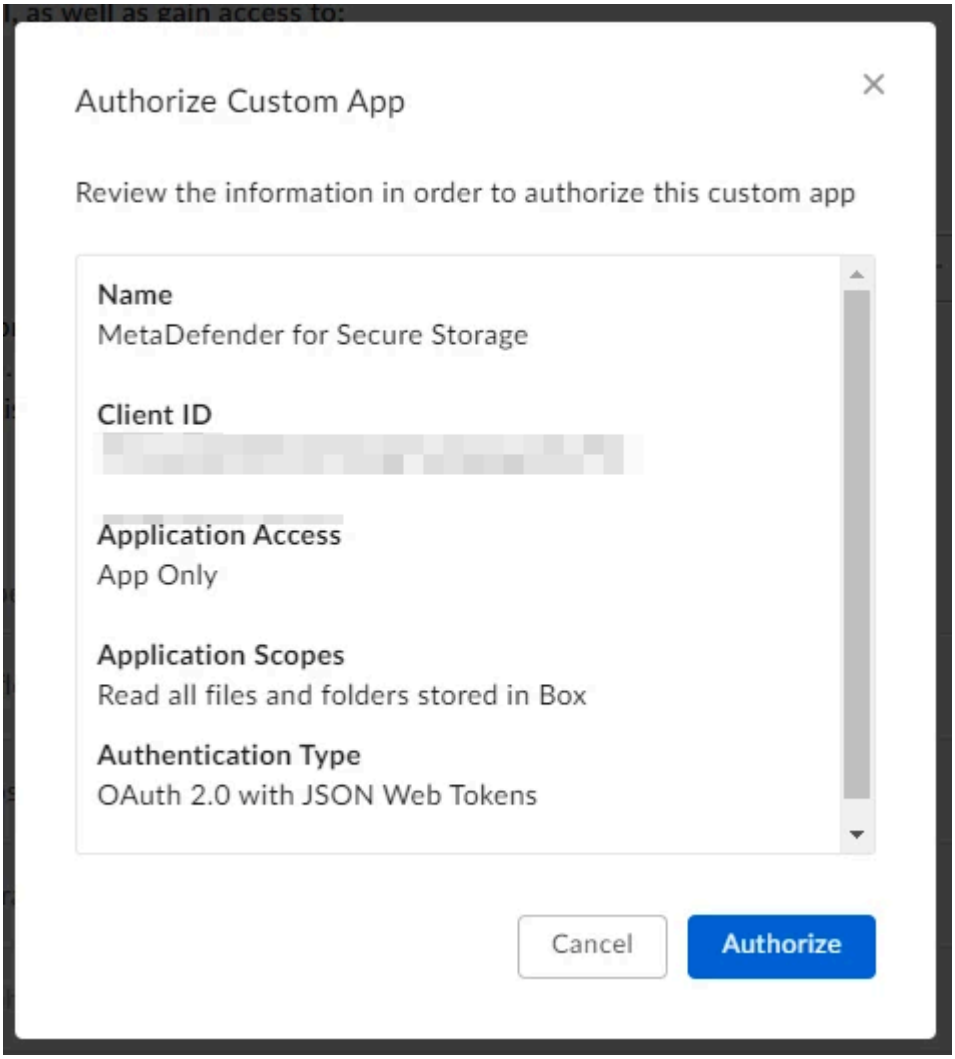


18. Click "**+**" and paste the previously copied **Client ID** and then click **Next**





19. Review the application permissions once again and then click **Authorize**



20. Almost done! All you need to do is go back to the MetaDefender Storage Security web interface and upload the **Public/Private Keypair** that you generated earlier in the **Credentials** field.

How my metadata is managed

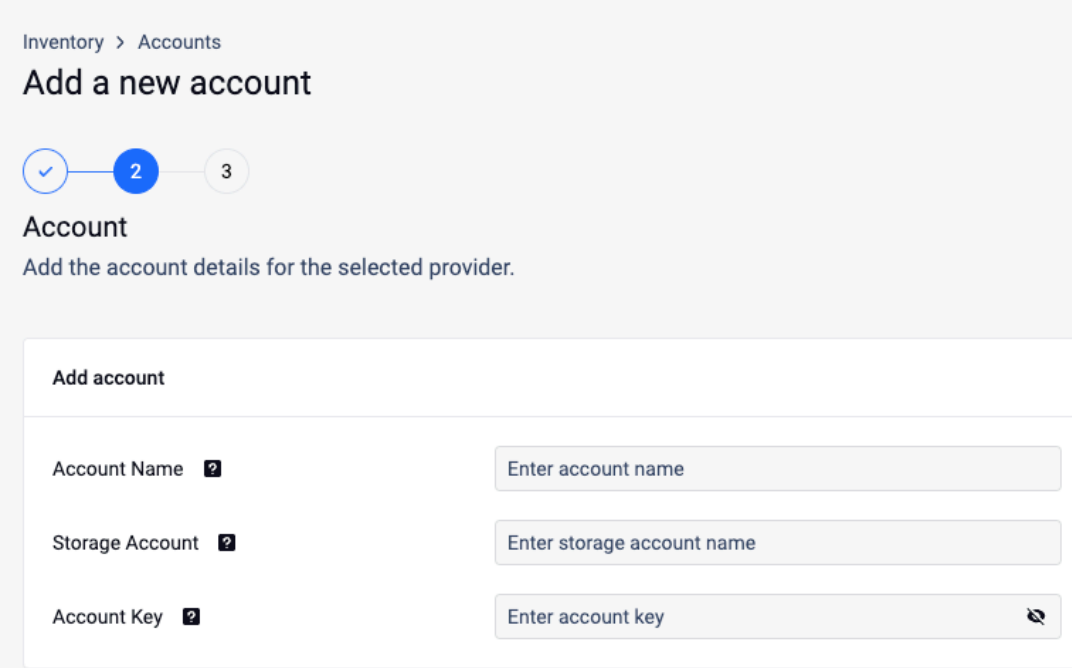
> **Warning**
>
> In Box, we support metadata, but with some limitations:
>
> 1. If the remediation actions are enabled and the file needs to be moved between accounts of the same enterprise, the metadata will be fully transferred.
> 2. If the remediation actions are enabled and the file needs to be moved between accounts of different enterprises, the metadata and its fields must be the same in these accounts in order to match. Otherwise, the metadata that does not match will be lost.

Add an Azure Files Storage

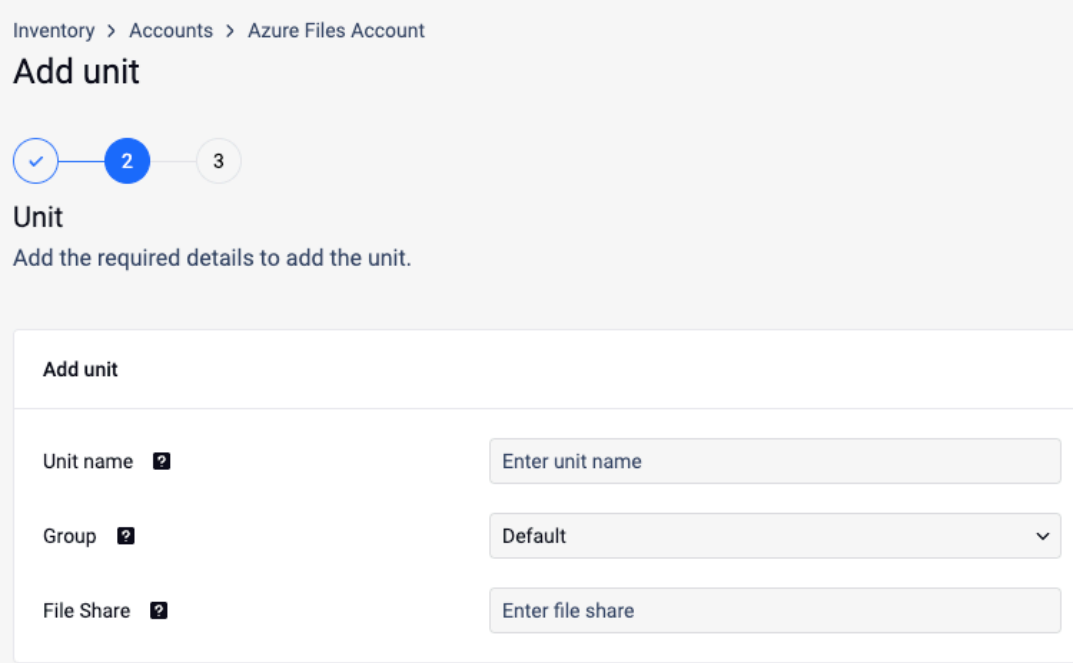
> **Warning**
>
> **Limitation :** Due to an Azure Files restriction, uploading 0-byte files is not supported. If a remediation action attempts to upload an empty file, the operation will fail.

In order to integrate Azure Files with MetaDefender Storage Security please follow these steps:

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **Azure Files** and click **Next**



2. In **Account Name**, enter a name that identifies this account in MetaDefender Storage Security. It is not the Azure storage account name.
3. In **Storage Account**, enter the name of your Azure storage account.
4. Enter your **Account Key**. Scroll to the end of this page for more details.
5. Click **Next** and finish the process.
6. Add a new unit for the previously created account (**Inventory** > **Add New** > **Unit**).



7. Give your unit an unique name
8. Assign your unit to a Group
9. Enter the **name of the share** you wish to process with MetaDefender Storage Security.
10. Click **Next** in order to finish the process.

Where can I find the Account Key?

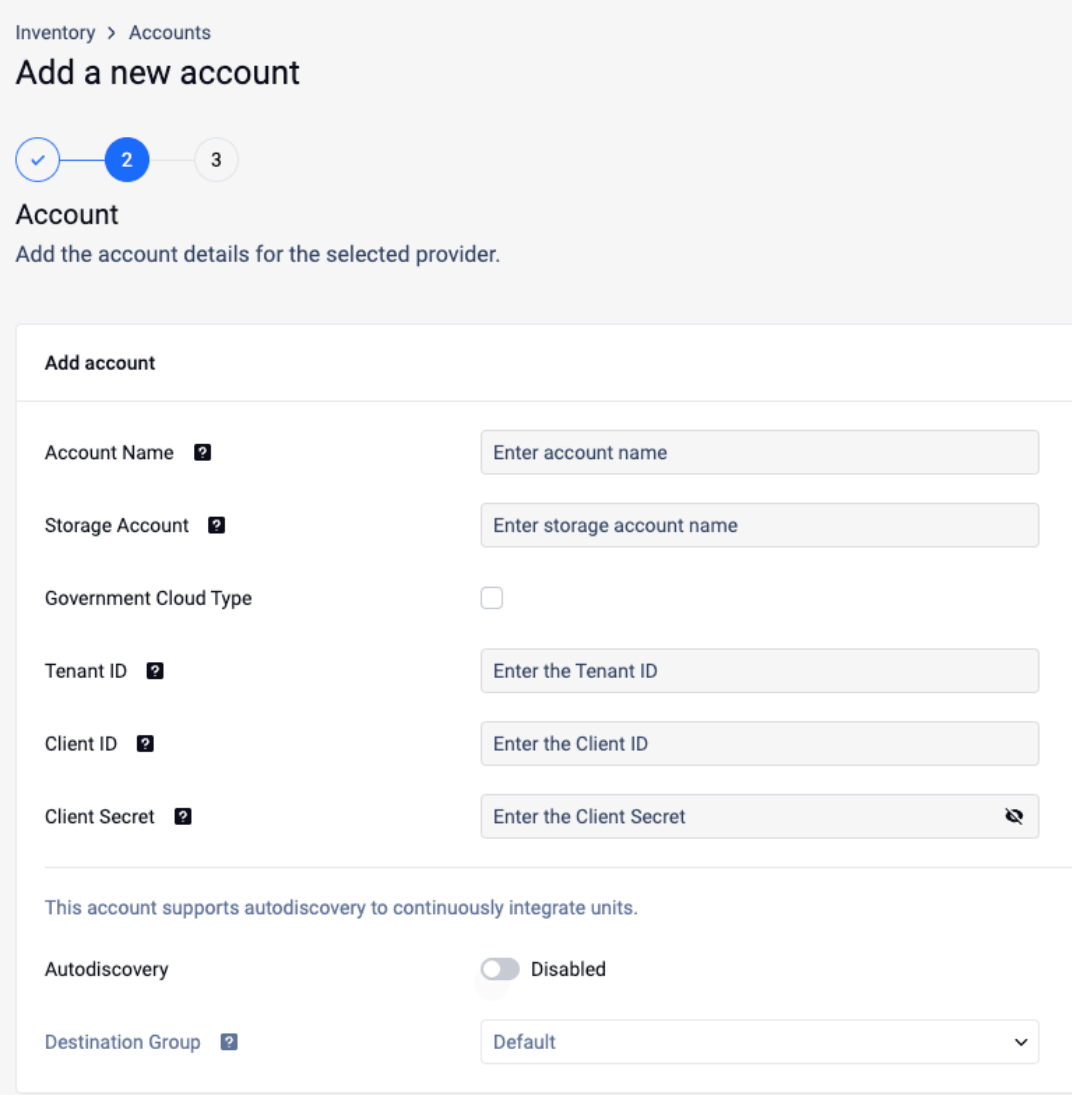
To find the Account Key, please follow these steps:

https://learn.microsoft.com/en-us/azure/storage/common/storage-account-keys-manage?tabs=azure-portal

Add an Azure Blob Storage

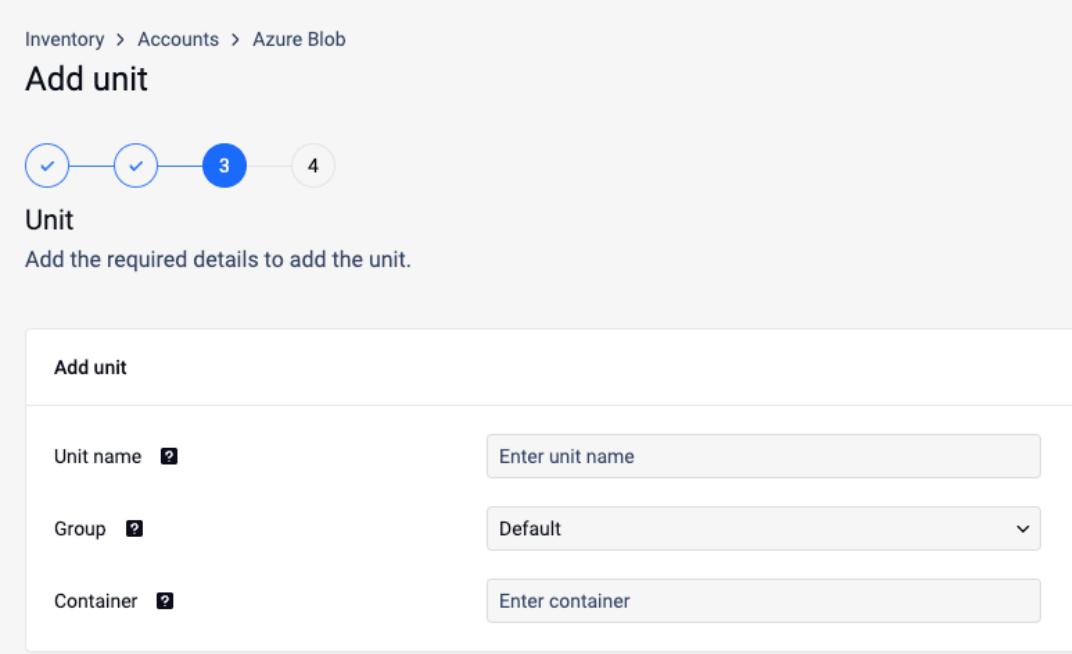
In order to integrate Azure Blob with MetaDefender Storage Security you will be required to create an application registration, assign the necessary permissions and roles, and then generate a secret key. Please follow these steps:

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **Azure Blob** and click **Next**



2. Give your account a name so you can easily identify it later
3. Enter the Azure **Storage Account** name.
4. Select **Government Cloud type** if the storage belongs to Azure US Government Cloud.
5. Enter your **Tenant ID**. Scroll to the end of this page for more details.
6. Enter your **Client ID.** Scroll to the end of this page for more details.
7. Enter your **Client Secret.** Scroll to the end of this page for more details.
8. Click Next to finish the process.

9. Add a new unit for the created account

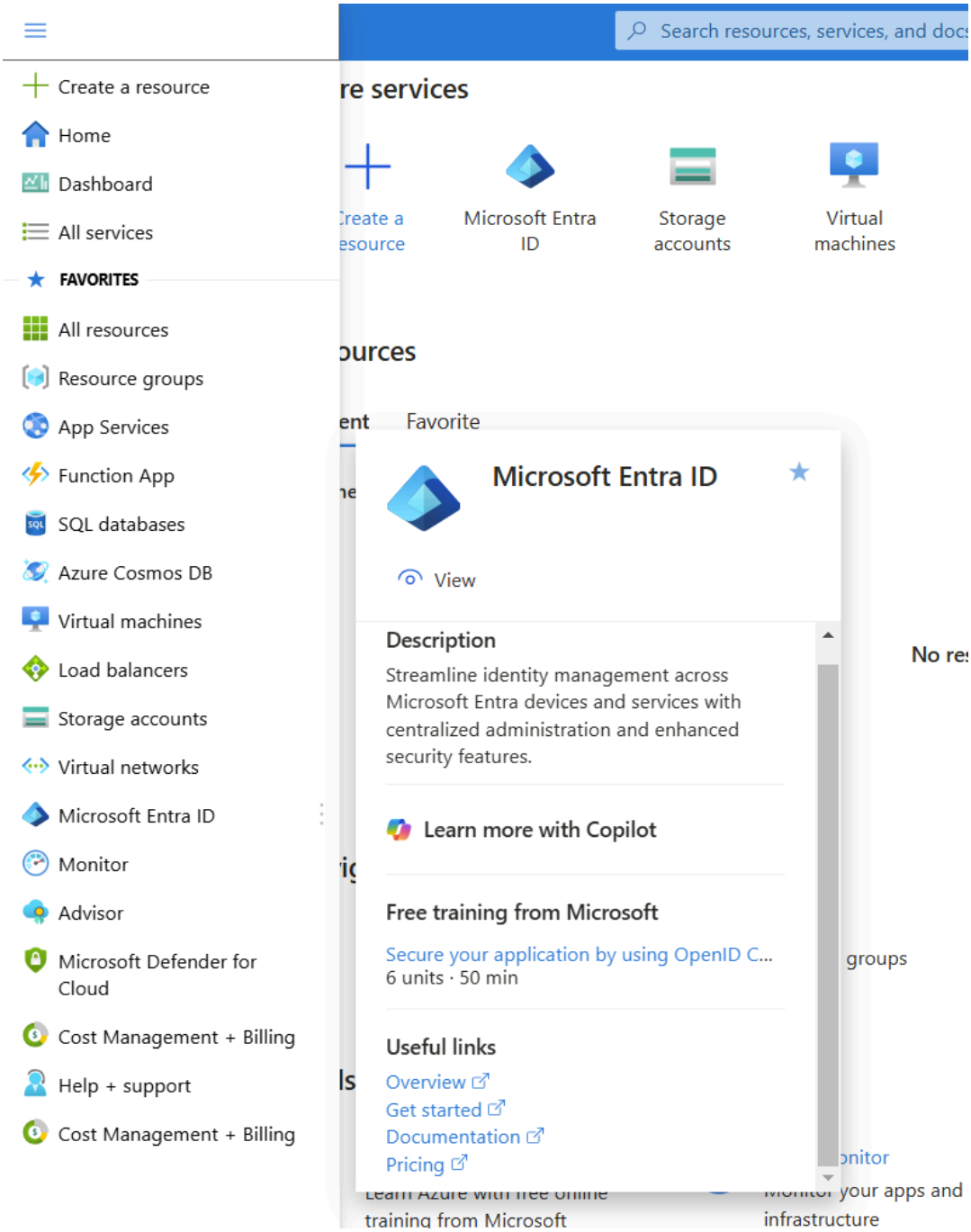


10. Enter an unique name for your unit.
11. Assing your unit to a Group
12. Enter the **name of a container** to process objects from a particular container.
13. Select **Next** in order to finish the process.

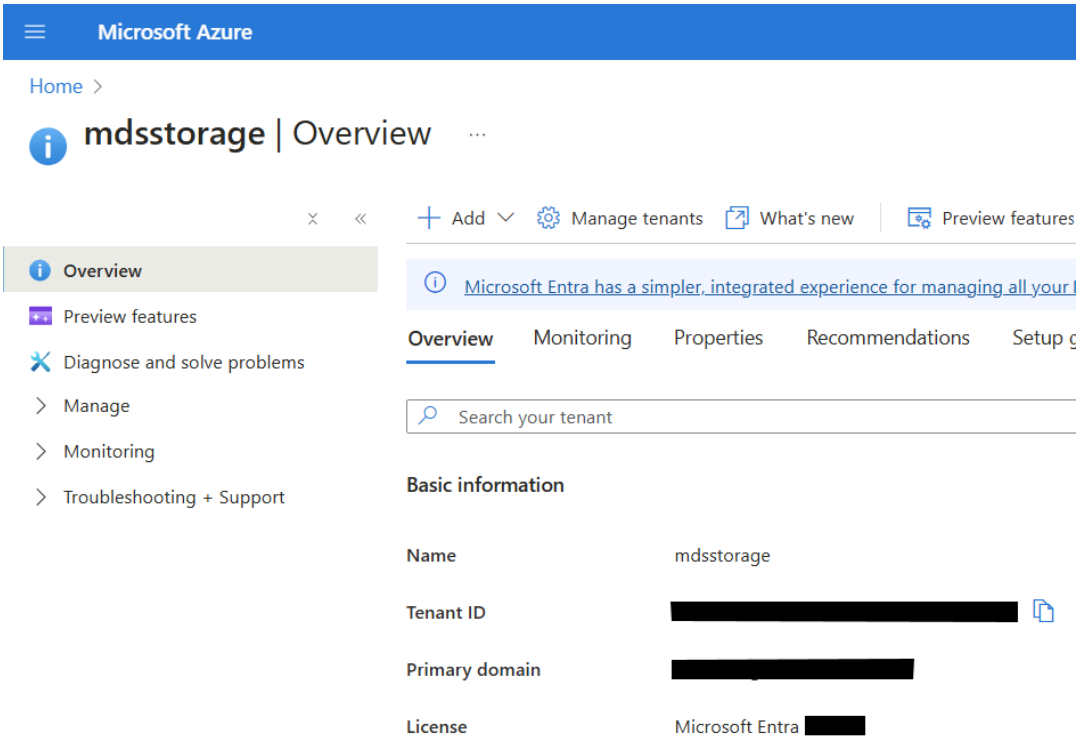
How to create an Azure Blob application?

In order to process your Azure Blob objects with MetaDefender Storage Security you will be required to add a new app registration in your Microsoft Entra ID Azure Portal.

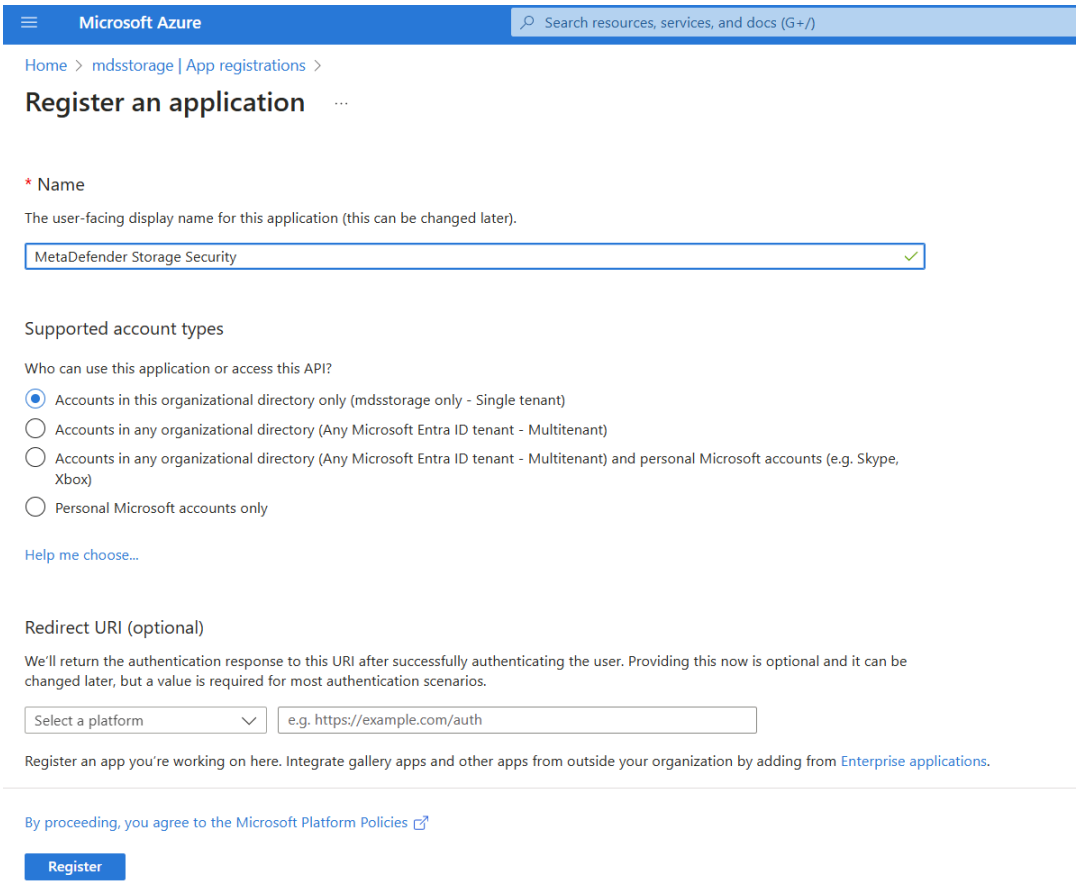
1. Log in to Azure Portal and from the left navigation menu choose **Microsoft Entra ID**



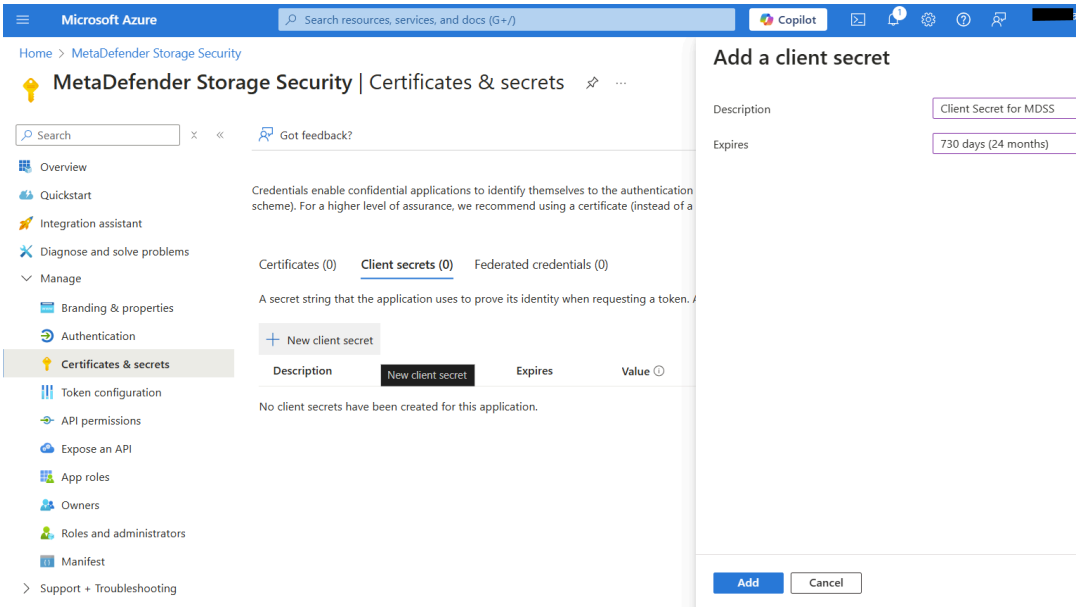
2. Make a copy of the **Tenant ID** from the overview page



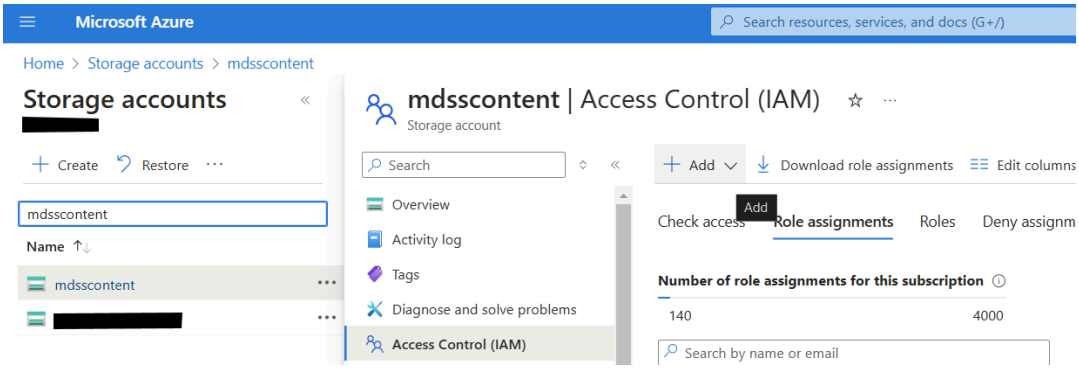
3. From the left side menu, choose **App registrations**
4. Click **New registration**
5. Give your app a name so you can easily identify it



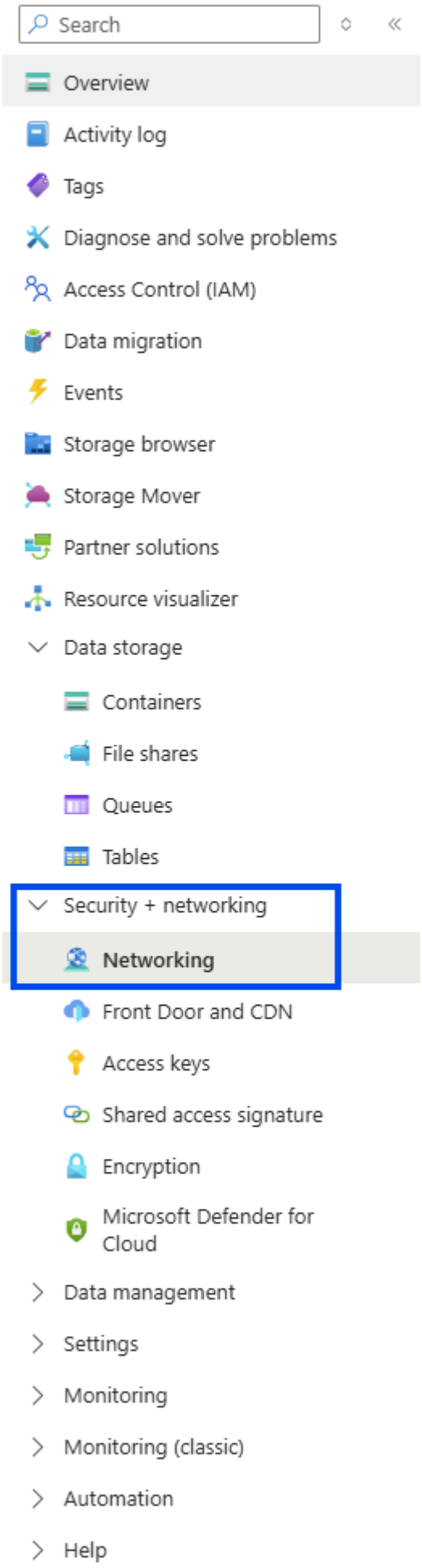
6. There is no need to modify the other properties. When ready, please click **Register**
7. From the **Overview** page of your newly created application, make a copy of **Application (client) ID**
8. Navigate to **Certificates & Secrets** from the left-side menu
9. Click **New client secret** and choose Expires in 24 months



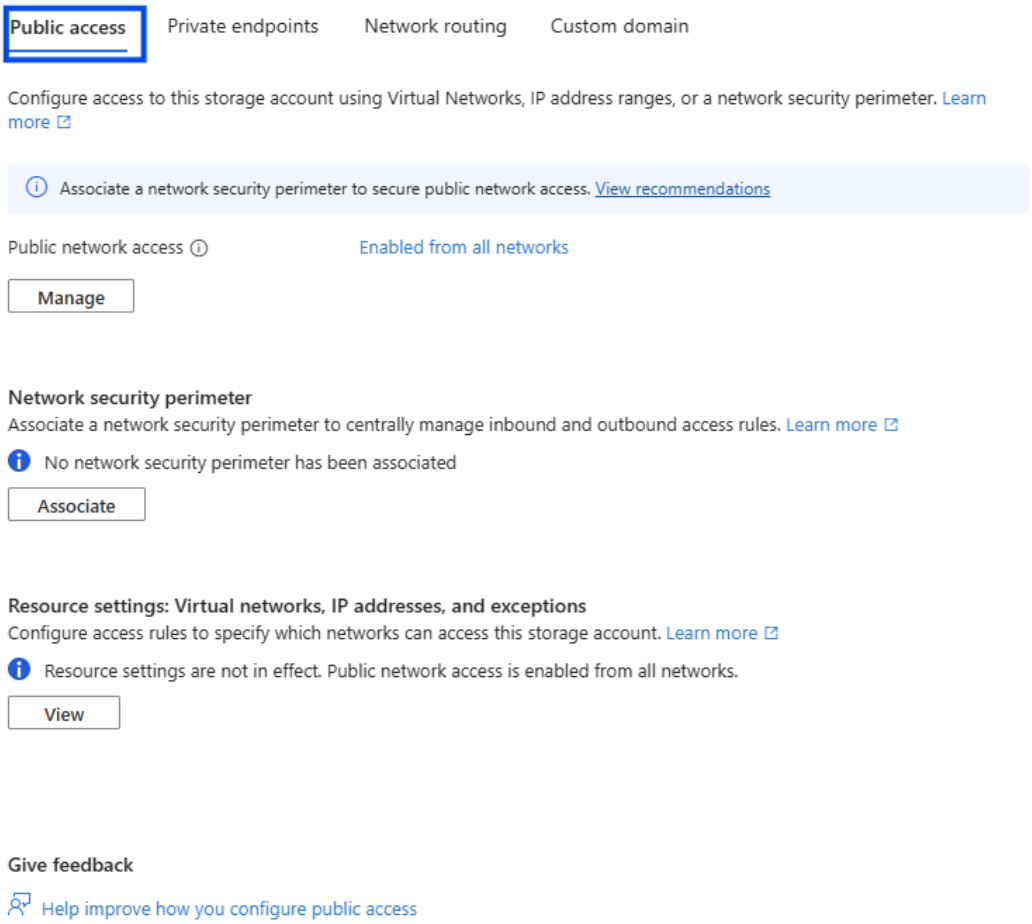
10. Click Add and then you will need to make a copy of the generated secret key because it will not be available later
11. Navigate to your **Storage Account** and select the **Access Control (IAM)** menu
12. Select the **Role assignments** tab, then select the **Add role assignment** option from the **Add** dropdown.



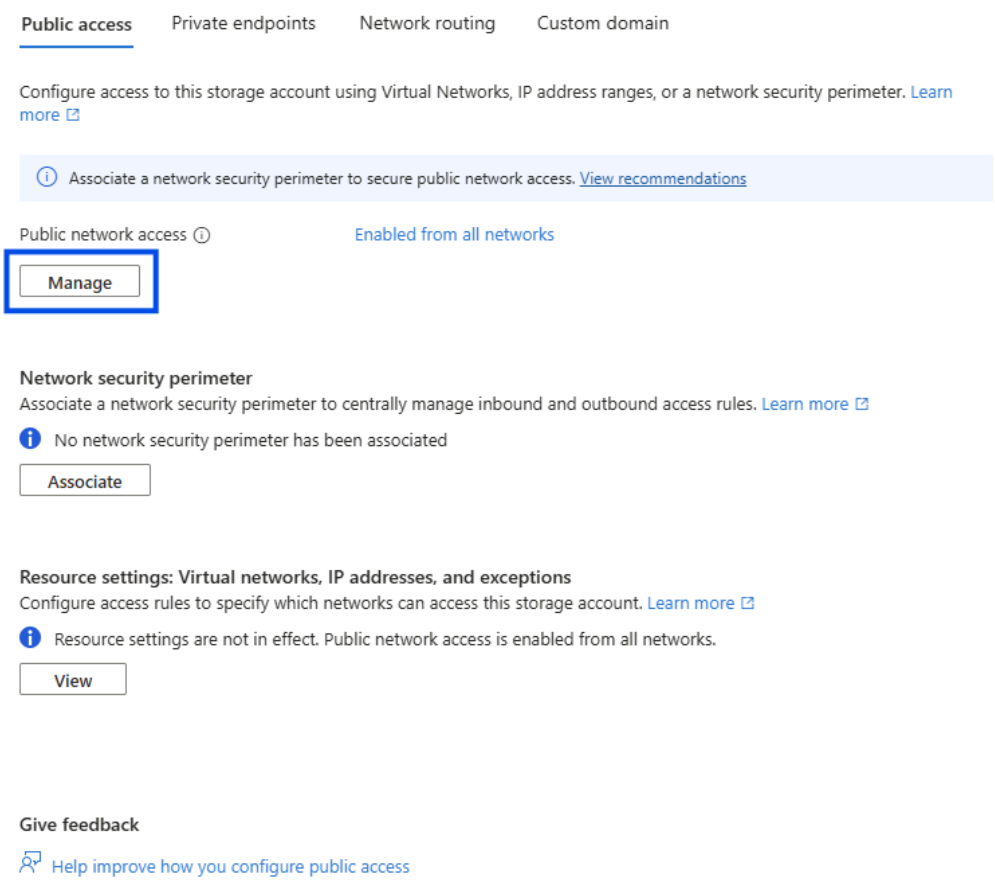
13. Assign the following roles to the **Client Application** you previously registered:
    1. Storage Blob Data Contributor
    2. Reader
14. Navigate back to your **Storage Account** and from **Security + networking** menu, select **Networking**


Search
Overview
Activity log
Tags
Diagnose and solve problems
Access Control (IAM)
Data migration
Events
Storage browser
Storage Mover
Partner solutions
Resource visualizer
Data storage
Containers
File shares
Queues
Tables
Security + networking
Networking
Front Door and CDN
Access keys
Shared access signature
Encryption
Microsoft Defender for Cloud
Data management
Settings
Monitoring
Monitoring (classic)
Automation
Help

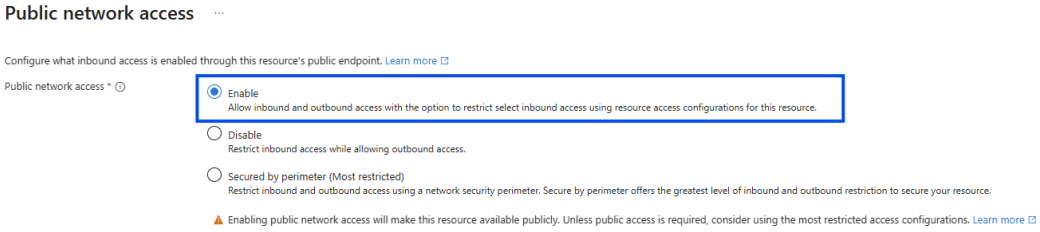
15. Navigate to **Public access** tab (the first and default one)



16. Click Manage on the Public network access section

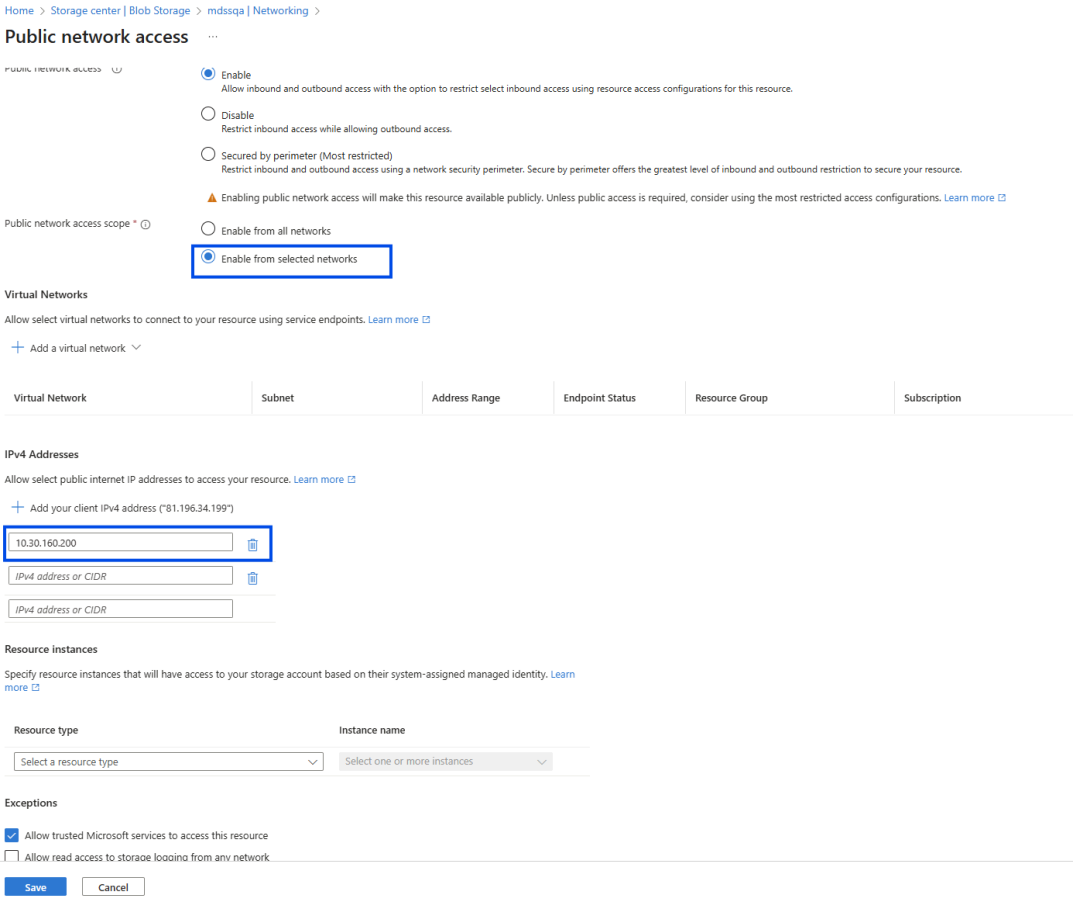


17. Enable **Public network access**



18. From the **Public network access scope**, you have two options:
    1. Enable from all networks: this is less restrictive and will allow any applications to try to establish a successful connection to your storage account.
    2. Enable from selected networks: this will restrict access only to specific IPv4 addresses. If you enable this option, add the public IP address from which MetaDefender Storage Security reaches Azure. For a Docker or Windows deployment this is the public address the host machine uses for outbound traffic: its own IP, or the address of the NAT or firewall in front of it. For a Kubernetes deployment it is the public address the cluster uses for outbound traffic, such as a NAT gateway or load balancer; if the nodes reach the internet through their own public IPs, add the address of every node that runs MDSS services. The Storages,

Discovery, Remediations and Scanning services all connect to the storage account from that address, including when a file is moved from Azure Blob to another type of storage unit.



19. Now that you have the **Tenant ID**, **Client ID** and the **Client Secret Key**, you can go back in **MetaDefender Storage Security** and finish the Azure Blob integration. Congratulations!

For additional configuration of how MetaDefender Storage Security handles blobs please check custom configuration.

How to Enable Autodiscovery

This guide provides step-by-step instructions for enabling Autodiscovery for Azure Blob storage in MetaDefender Storage Security.

Autodiscovery automatically detects and adds Azure Blob containers as storage units, saving time and reducing configuration errors.

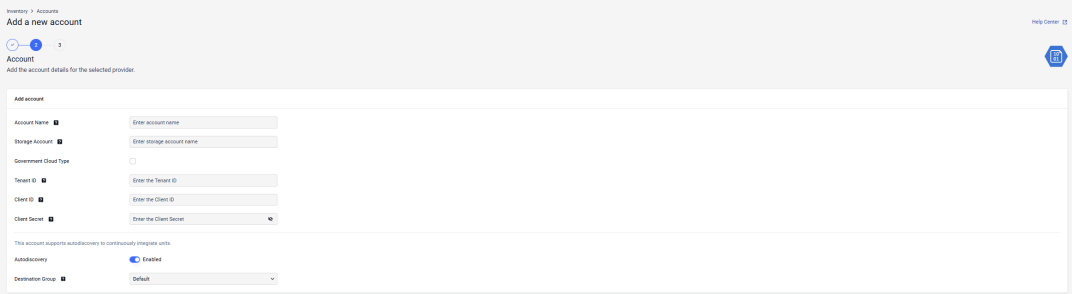
Prerequisites

Before enabling Autodiscovery, ensure you have:

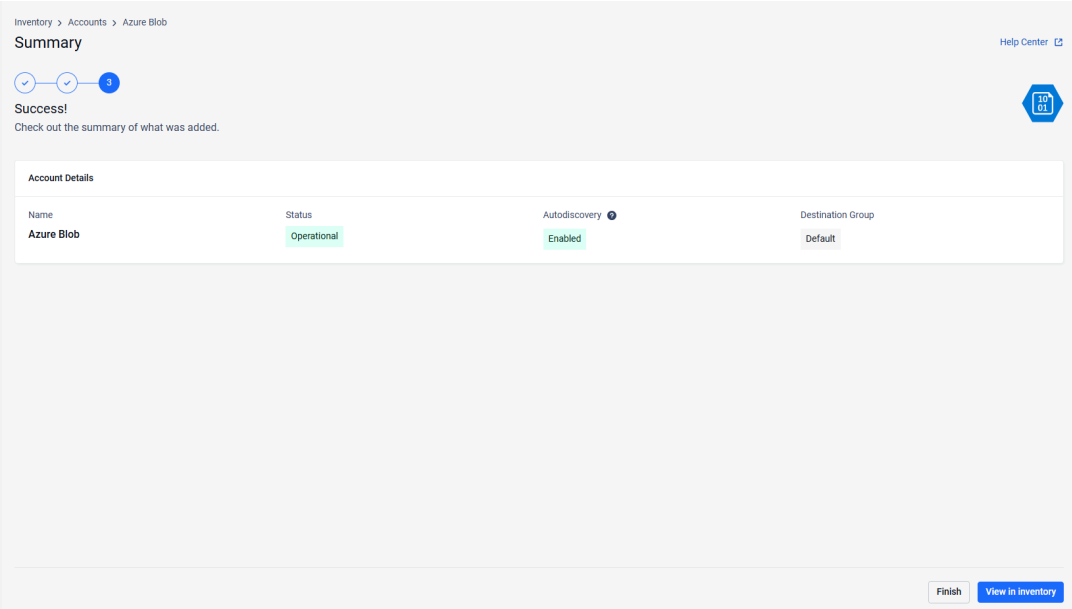
1. Valid Azure Blob credentials with sufficient permissions
2. Permissions as stated in Add an Azure Blob storage

Method 1: Enable Autodiscovery When Adding an Account

- Navigate to **Inventory** › **Accounts** and click **Add account**
- Select the **Azure Blob** provider
- Fill out the form with valid Azure Blob credentials
- Turn on the **Autodiscovery** switch and select the **Destination Group** where the discovered containers will be added



- Continue to the **Summary** step. Autodiscovery starts in the background and the discovered containers are added as storage units to the destination group. Click **View in inventory** to open the account, or **Finish**



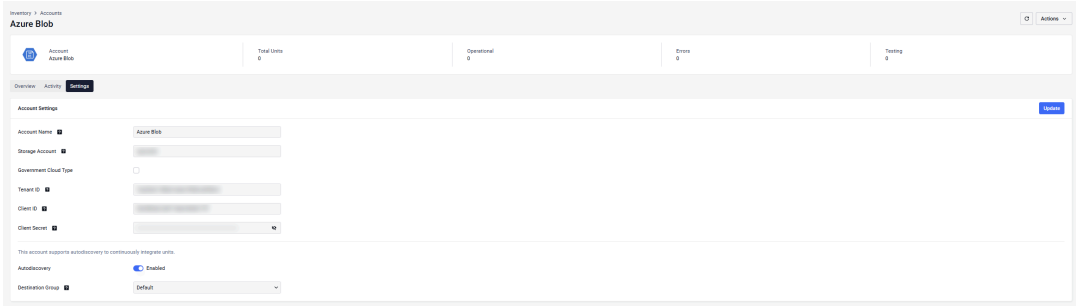
To learn more, please consult the Autodiscovery documentation.

> **ADDING SPECIFIC CONTAINERS ONLY**

> To add only specific containers, leave **Autodiscovery** disabled, click **Add unit** on the **Summary** step, and choose **Single unit** or **Multiple units**.

Method 2: Enable Autodiscovery for an Existing Account

- Navigate to **Inventory** › **Accounts** and open the account you want to start autodiscovery for
- Open the **Settings** tab
- Turn on the **Autodiscovery** switch and select the **Destination Group**
- Enter the **Client Secret**
- Click **Update**



Add an Alibaba Cloud Storage

In order to integrate Alibaba Cloud with MetaDefender Storage Security please follow these steps:

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **Alibaba Cloud** and click **Next**
2. Give your account a name so you can easily identify it later
3. Enter the **Endpoint** URL for *Access Over Internet*. Scroll to the end of this page for more details.
4. There are two ways to connect:
   - **via the RAM role from the ECS instance**: select **Use RAM Role** (If you choose this option, skip the following two steps and continue from step 7)
   - **using security credentials**
5. Enter your **Access Key ID**. Scroll to the end of this page for more details.
6. Enter your **Access Key Secret .** Scroll to the end of this page for more details.
7. Click **Next** to create the account
8. On the **Summary** step, click **Add unit** (or navigate to **Inventory**, click **Add New** > **Unit**, choose the previously created account and click **Next**)
9. Give your unit a name so you can easily identify it later
10. Assign your unit to a Group
11. Enter the **name of the bucket** you wish to process with MetaDefender Storage Security.
12. If you wish to only process a particular folder enter the name or path of the folder in the **Folder location** field. Leave this field empty if you wish to process the entire bucket.
13. Select **Next** in order to finish the process.

Where can I find the Endpoint URL?

The endpoint URL can be found in the following list of the official documentation of Alibaba Cloud.

https://www.alibabacloud.com/help/en/object-storage-service/latest/regions-and-endpoints

Where can I find the Access Key ID and Access Key Secret?

To generate an Access Key ID / Access Key Secret pair please follow these steps, explained in the official documentation of Alibaba Cloud.

https://www.alibabacloud.com/help/en/basics-for-beginners/latest/obtain-an-accesskey-pair

Using RAM role for Alibaba Cloud integration

On Alibaba Cloud integration with RAM, the role configured on the ECS instance will be used to authenticate.

Please use the following links to configure and attach a RAM role to the ESC instance:

https://www.alibabacloud.com/help/en/ram/product-overview/best-practices-for-identity-and-access-control

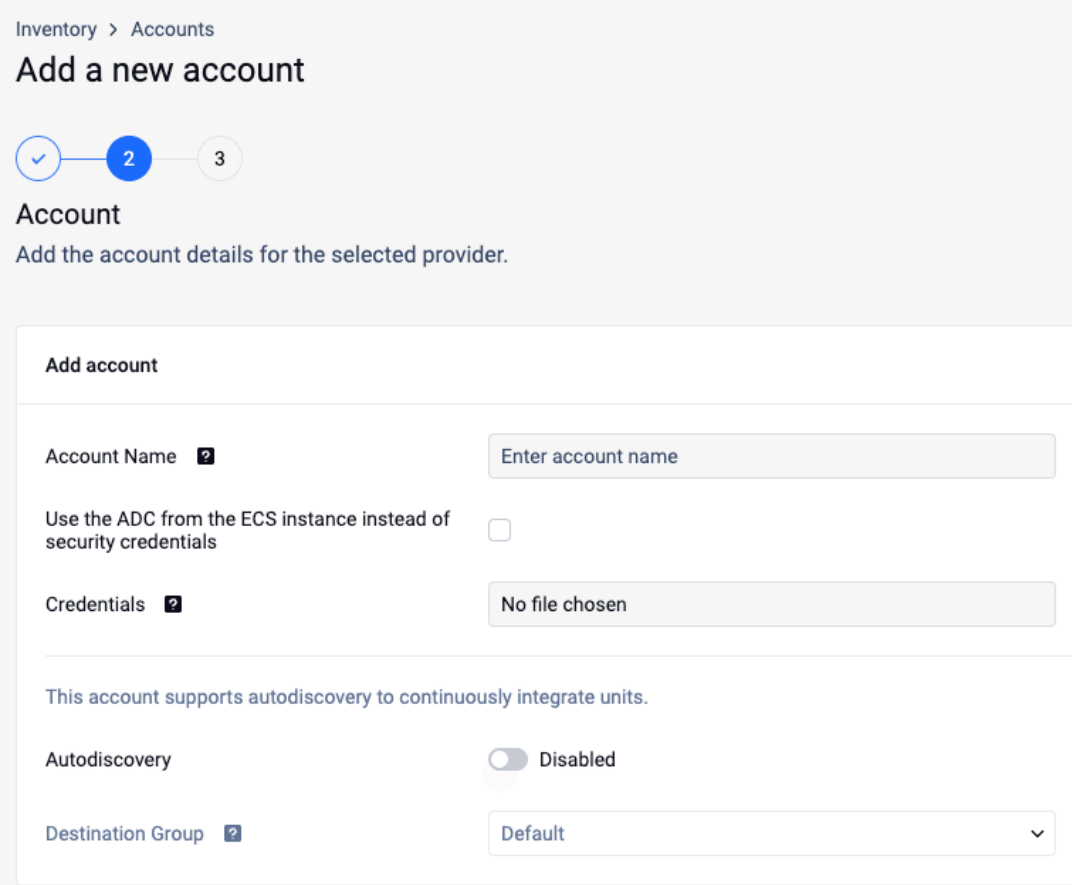
https://www.alibabacloud.com/help/en/elastic-compute-service/latest/use-ram-roles-to-access-other-alibaba-cloud-services

https://www.alibabacloud.com/help/en/elastic-compute-service/latest/attach-an-instance-ram-role

Add a Google Cloud Storage

In order to integrate Google Cloud with MetaDefender Storage Security please follow these steps:

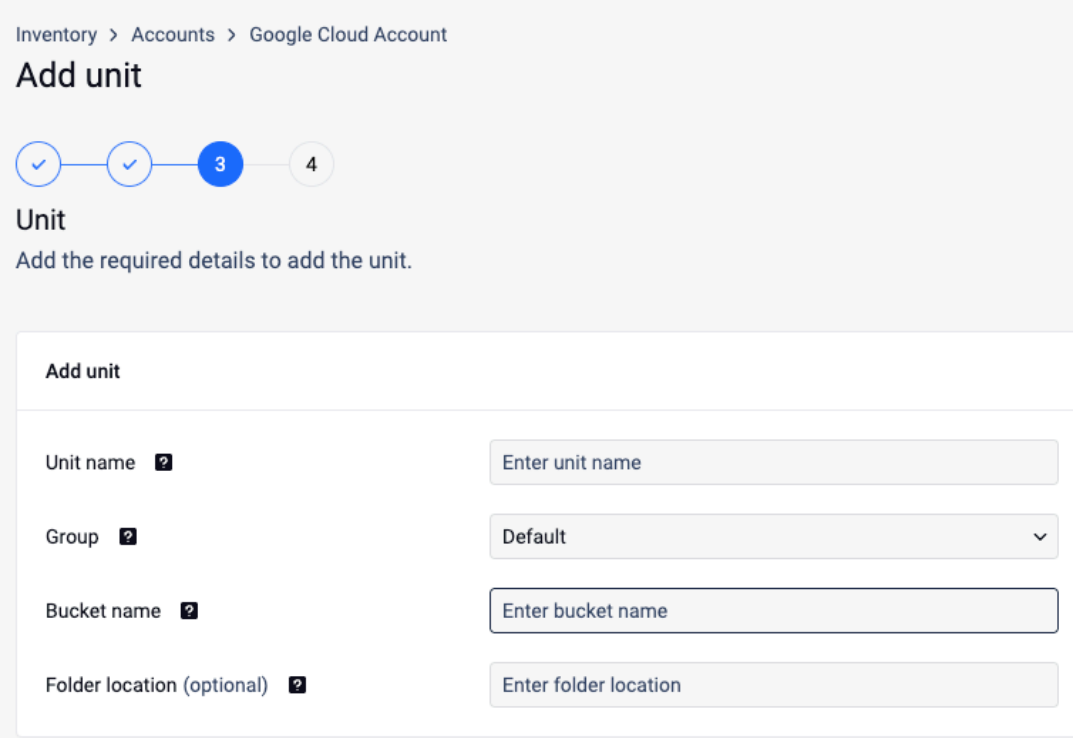
1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **Google Cloud** and click **Next**



2. Give your account a name so you can easily identify it later
3. There are two ways to connect:

- **via the ADC from the ECS instance** (If you choose this option, skip the following step and continue from step 5)
- **using security credentials**

4. Upload your **Credentials** file. Scroll down for more details.
5. Click Next in order to finish the process
6. Add an unit for the created account



7. Give your unit an unique name
8. Assign your unit to a Group
9. Enter the **name of the bucket** you wish to process with MetaDefender Storage Security.
10. If you wish to only process a particular folder enter the name or path of the folder in the **Folder location** field. Leave this field empty if you wish to process the entire bucket.
11. Select **Next** in order to finish the process.

> **Info**
>
> To be able to use **autodiscovery**, the credentials need the **storage.buckets.list** permission on the project, because autodiscovery lists every bucket in the project. The **Storage Legacy Bucket Reader** role does not include it. Grant it through a custom role or a predefined role that contains it, such as **Storage Admin** (**roles/storage.admin**).

Where can I generate the Credentials file?

To generate a credentials file, please follow these steps, explained in the official documentation of Google Cloud.

https://developers.google.com/workspace/guides/create-credentials#service-account

Using ADC for Google Cloud integration

On Google Cloud integration with ADC, the default credentials configured on the ECS instance will be used to authenticate.

Please use the following links to attach application default credentials to the ESC instance:

https://cloud.google.com/docs/authentication/application-default-credentials

https://cloud.google.com/iam/docs/impersonating-service-accounts#attaching-to-resources

> **Warning**
>
> If the google credentials are **not** set in the **GOOGLE_APPLICATION_CREDENTIALS** variable, then the service account assigned to the VM instance requires the **iam.serviceAccounts.signBlob** permission (included in the **Service Account Token Creator** role) and the `https://www.googleapis.com/auth/iam` OAuth access scope (or the broader `https://www.googleapis.com/auth/cloud-platform` scope).

Using GCP Buckets With Hierarchical Namespace (HNS)

We recommend enabling **Hierarchical Namespace (HNS)** on your Google Cloud Storage buckets. When HNS is enabled, folder paths behave as true directories, allowing MetaDefender Storage Security to correctly validate and process **empty folders** and **folders containing only subfolders**.

With HNS enabled:

- Empty folders can be added as destinations
- Folder existence is reliably detected

Without HNS:

- Empty folders may be rejected as “non-existent”
- Hidden system objects may affect folder validation

Learn more in Google’s official documentation:
https://docs.cloud.google.com/storage/docs/folders-overview

Google Cloud: Handling Rate Limiting

For transient errors on Google Cloud requests a retry with exponential backoff can be configured to adjust the required needs. The HTTP error codes covered are the `TooManyRequests`, `RequestTimeout`and `ServiceUnavailable`.

The retry policy can be customized using the following environment variables:

Parameters	Value
GCP_MAX_RETRY_ATTEMPTS	10
GCP_MAX RETRY_DELAY	120 (seconds)

The retry policy implements exponential backoff, starting with a 200-millisecond delay between retries and capping at a maximum delay of 120 seconds after 10 retries.

How to Enable Autodiscovery

This guide provides step-by-step instructions for enabling Autodiscovery for Google Cloud storage in MetaDefender Storage Security.

Autodiscovery automatically detects and adds GCP buckets as storage units, saving time and reducing configuration errors.

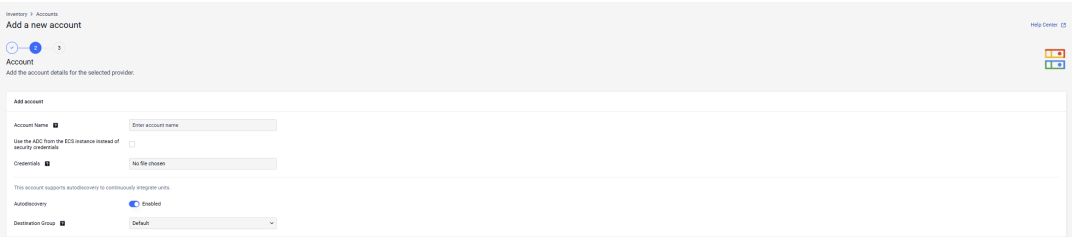
Prerequisites

Before enabling Autodiscovery, ensure you have:

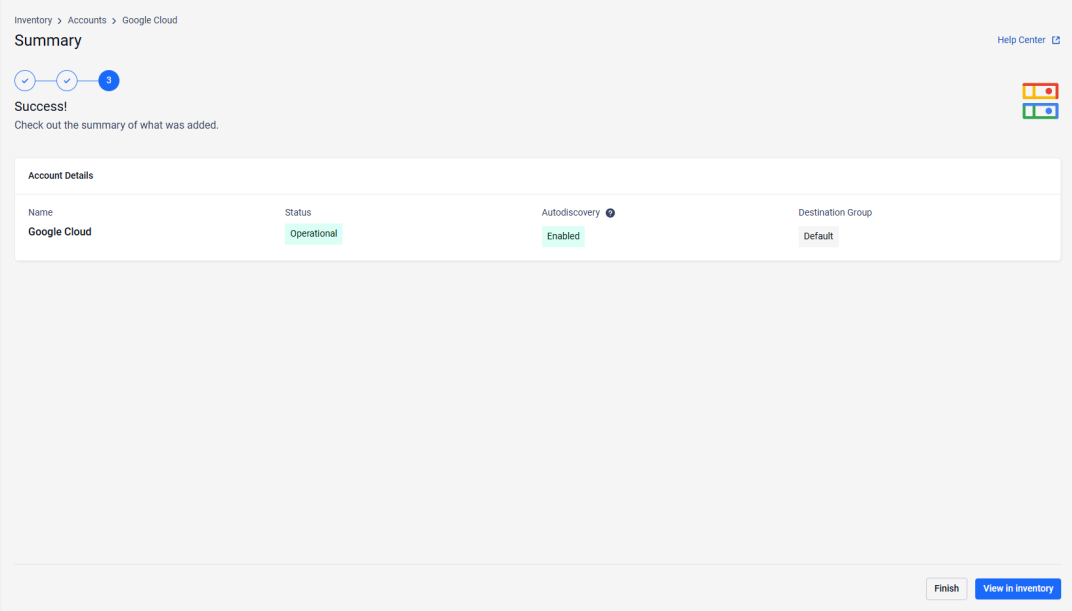
1. Valid GCP credentials file with sufficient permissions
2. Permissions as stated in Add a Google Cloud Storage

Method 1: Enable Autodiscovery When Adding an Account

- Navigate to **Inventory** › **Accounts** and click **Add account**
- Select the **Google Cloud** provider
- Fill out the form with valid GCP credentials
- Turn on the **Autodiscovery** switch and select the **Destination Group** where the discovered buckets will be added



- Continue to the **Summary** step. Autodiscovery starts in the background and the discovered buckets are added as storage units to the destination group. Click **View in inventory** to open the account, or **Finish**

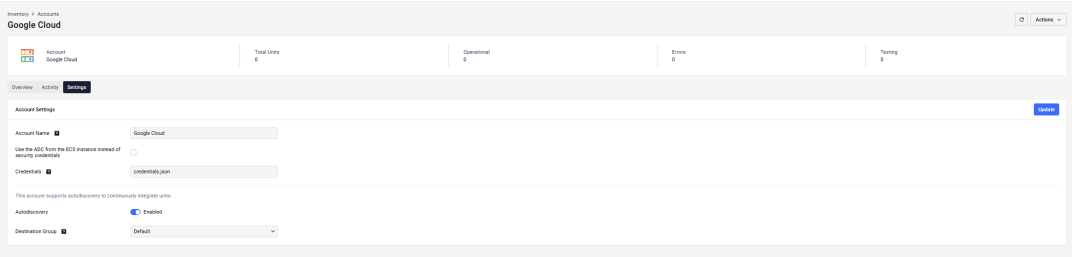


To learn more, please consult the Autodiscovery documentation.

> **ADDING SPECIFIC BUCKETS ONLY**
>
> To add only specific buckets, leave **Autodiscovery** disabled, click **Add unit** on the **Summary** step, and choose **Single unit** or **Multiple units**.

Method 2: Enable Autodiscovery for an Existing Account

- Navigate to **Inventory** › **Accounts** and open the account you want to start autodiscovery for
- Open the **Settings** tab
- Turn on the **Autodiscovery** switch and select the **Destination Group**
- Upload the GCP credentials file in the **Credentials** field
- Click **Update**



Add an OPSWAT MFT Storage

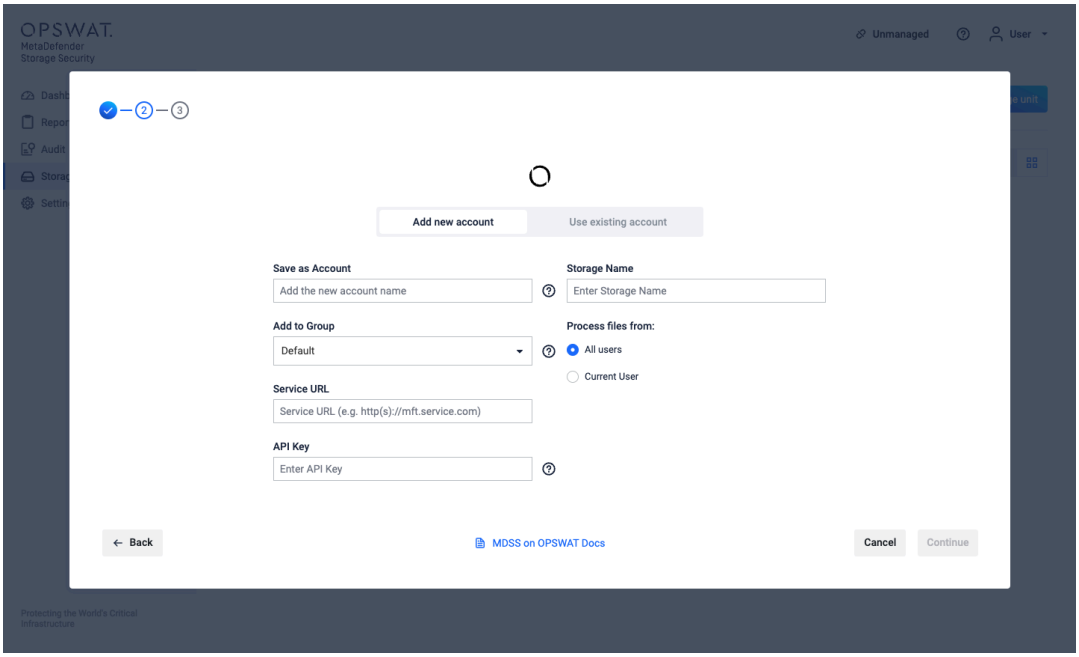
This guide outlines the necessary steps to integrate an OPSWAT Managed File Transfer (MFT) storage solution with your OPSWAT MetaDefender Storage Security (MDSS) instance.

> **LIMITATION**
>
> The MFT integration does not currently support the “Shared Space” feature within Custom Groups. Consequently, while files in a Custom Group’s Shared Space will be discovered during a scan using the “All users” storage option, any attempts to apply remediation actions to these files will fail.

1. Prerequisites

Before you begin adding an MFT storage unit in MDSS, make sure you have the following:

- an **MFT Service URL** - this is the base web address used to access your OPSWAT MFT provider
- an **API Key** required for MDSS to authenticate and interact with your MFT service
- You will need **administrator privileges** for both your MFT dashboard and your MDSS web page.



2. Obtaining the Service URL

The MFT Service URL is the primary address for your MFT provider.

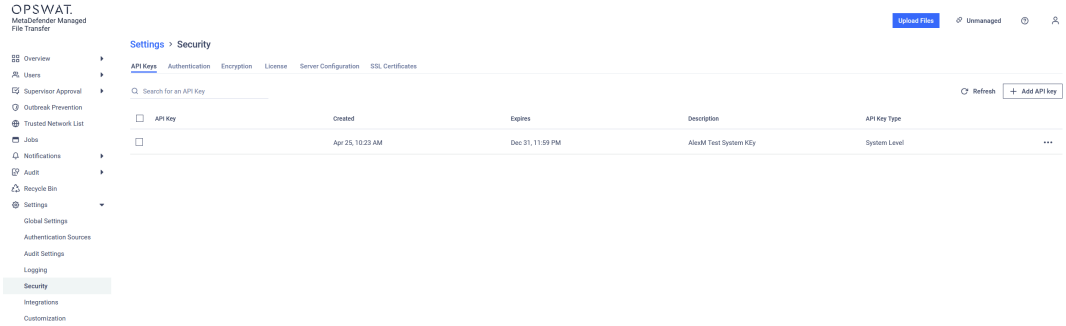
- Determine the complete base URL where your MFT service is hosted.

- Make sure the URL includes the correct protocol, either `http://` or `https://`.
  - Example: `https://mft.yourcompany.com` or `http://mft.internal.net`

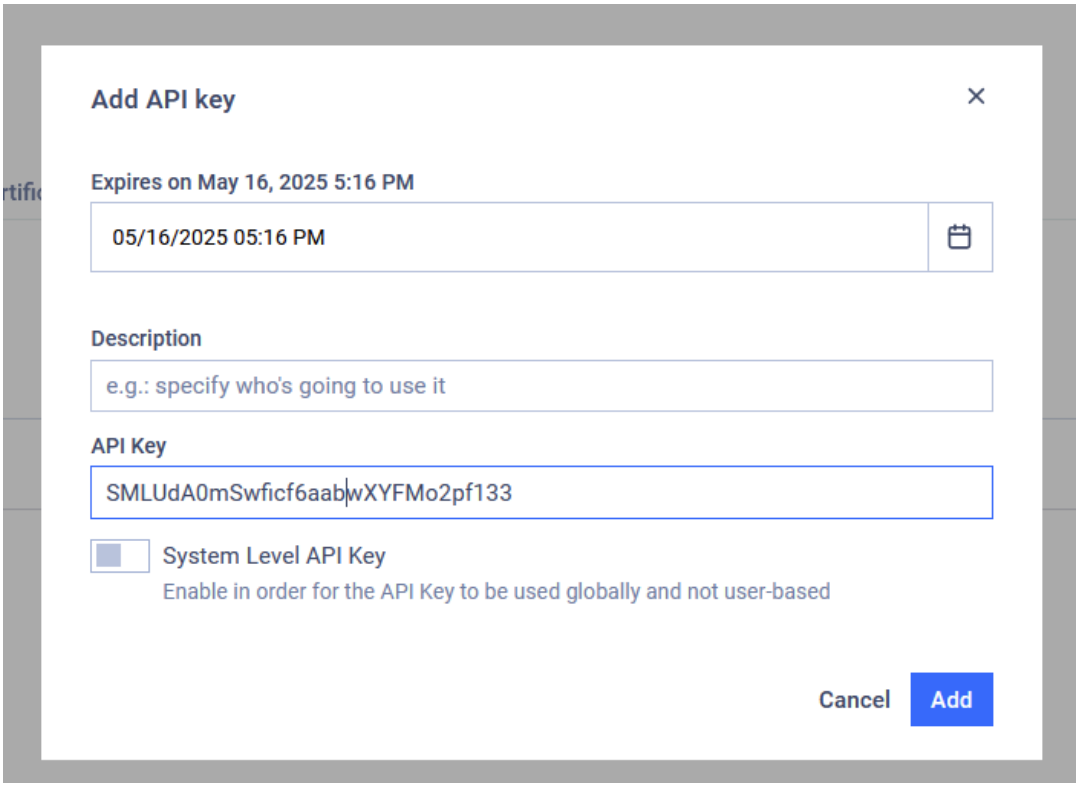
3. Generating Your MFT API Key

An API key is essential for secure communication between MDSS and your MFT service.

- Log in to your MFT dashboard with administrator credentials and navigate to the **Settings** page.
- Within the Settings menu, find and select the **Security** section. This is often located in a side navigation panel.
- Within the Security section, select the **API Keys** tab.



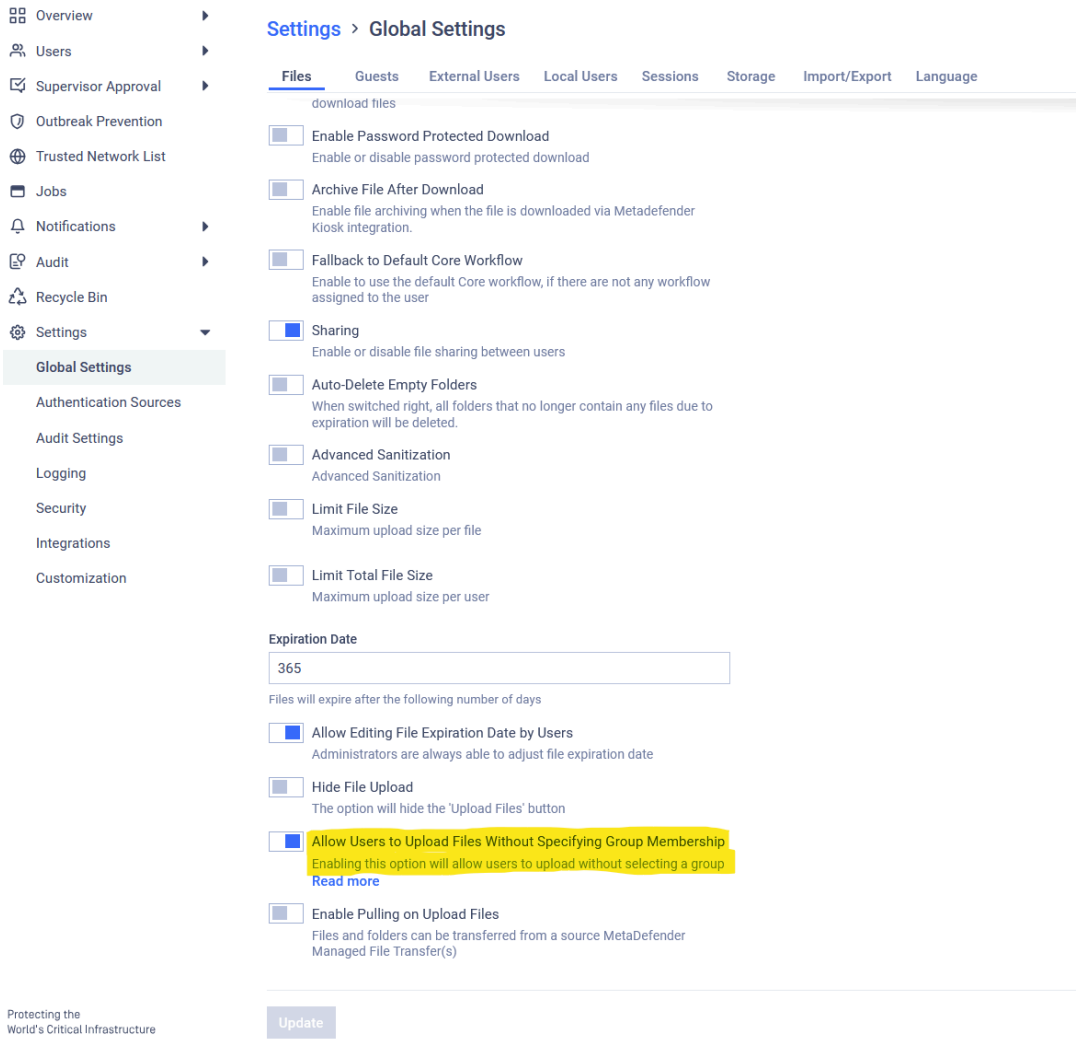
- Click on the **Add API key** button (or a similarly named option like “Generate New Key”).
- Set an appropriate expiration date for the API key to enhance security. Consider your organization’s security policies.
- Once generated, the API key will be displayed. **Immediately and securely copy and store this key.** For security reasons, treat this key like a password.



4. Configuring MFT File Upload Settings

To ensure MDSS can successfully upload files, a specific MFT setting needs to be enabled. This allows MDSS to upload files on behalf of users, even if those users are part of a group and the group membership isn't explicitly specified during the upload.

- As an administrator, navigate to the **Settings** page on your MFT dashboard
- Under the **Global Settings** section, locate and select the **Files** tab
- Make sure that the option labeled **"Allow Users to Upload Files Without Specifying Group Membership"** is **checked (enabled)**
- If you've made any modifications to this setting, ensure you save them



5. Adding the MFT Storage Unit in MDSS

With the MFT Service URL and API key obtained, and the necessary MFT setting configured, you can now add the MFT storage to MDSS. [**Note**: The latest supported version of MFT Storage is **3.9.4**]

- Log in to your MDSS web interface.
- From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **OPSWAT MFT** and click **Next**
- Fill in the account fields:
  - **Account name** - a name so you can easily identify the account later.
  - **Service URL** - enter the URL obtained in Step 2.
  - **API Key** - enter the API key generated and saved in Step 3.
- Click **Next** to create the account.

- On the **Summary** step, click **Add unit** (or navigate to **Inventory**, click **Add New** > **Unit**, choose the MFT account and click **Next**)
- Fill in the unit fields:
  - **Unit name** - a name so you can easily identify the unit later.
  - **Group** - the group the unit is assigned to.
  - **Process files from** - choose **All users** or **Current user**.
- Click **Next** to finish. It is highly recommended to test the integration by uploading or scanning a file to ensure the MFT storage is working correctly with MDSS.

> **IMPORTANT**
>
> MDSS will not discover or process files that OPSWAT MFT has already blocked and made unavailable for download.

Add an NFS Storage

Network File System (NFS) is a popular distributed file system protocol that allows a user on a client computer to access files over a network in a manner similar to how local storage is accessed. This document guides you through the process of integrating NFS storage with MetaDefender Storage Security.

1. Understanding NFS Server and Client

In a NFS setup, there are two primary components:

1. **NFS Server**: This is the machine where your files are stored. It must be configured to share files using the NFS protocol.
2. **NFS Client**: This is the machine where MetaDefender Storage Security is installed. It accesses the files stored on the NFS Server.

2. Setting up the NFS Server machine

Please make sure your files are shared via NFS protocol.

To enable remediations capabilities of MetaDefender Storage Security (sanitization, file moving / deleting), we need to have write access to your shared folder. To allow write access, please choose the appropriate configuration for your server. More details in this article.

The next steps are tailored to your deployment type, please choose yours:

- Windows deployment
- Linux deployment

> **Info**
>
> **Symlink Support:** MetaDefender Storage Security is able to follow all types of symbolic links (symlinks) and process the files they point to when integrating with NFS storage on both Linux and Windows deployments. This includes:
>
> - **Soft links (symbolic links)**
> - **Hard links**
> - **Junction points (on Windows)**
>
> For more details on this functionality, follow this link.

[Linux] How to add an NFS storage unit

> **Info**
>
> For steps 1 and 2, please see this page

3. Setting up the NFS Client machine

Before integrating NFS storage with MetaDefender Storage Security, a mount needs to be created from the NFS Client machine to the NFS Server. Follow these steps **on the NFS Client machine**:

1. Install NFS:
   - For Debian-based distributions, install `nfs-common`.
   - For Fedora-based distributions, install `nfs-utils`.
   - Our product is not bound to a specific NFS version. It should work as expected with the version provided by the above package.
2. Prepare the Mount Directory:
   - The default directory for all mounts is `/mnt/mdss/nfs`. Within this directory, create a new folder, for example, `company-files`, where the NFS mount will be performed

> **Info**
>
> /mnt/mdss/nfs is the default mount directory. This can be changed a custom location by setting *NFS_MOUNT_LOCATION=/path/to/your/location* in customer.env. Information on how to modify customer.env: Custom configuration

All new files and folders created by MetaDefender Storage Security (during Sanitization, Remediations) will adopt the same read, write and execute permissions as the parent directory. This means you need to set the appropriate permissions to the `company-files` folder in order new items to be accessible for the desired audience.

1. Perform the NFS Mount:
   - Execute the following command: **sudo mount -t nfs -o rw,hard,intr,nolock {nfs-server-ip}:/srv/company-shared-files /mnt/mdss/nfs/company-shared-files/**
   - This command mounts the shared files from the NFS Server to the `company-files` directory on the NFS Client.
   - It is required the `mount`command to have the `nolock` flag as some older NFS servers are locking the file when opened by another application. Apart from `nolock` , we do not require any specific flags. As long as the mount command is working as expected and we have permissions to the mount directory, should be enough. READ

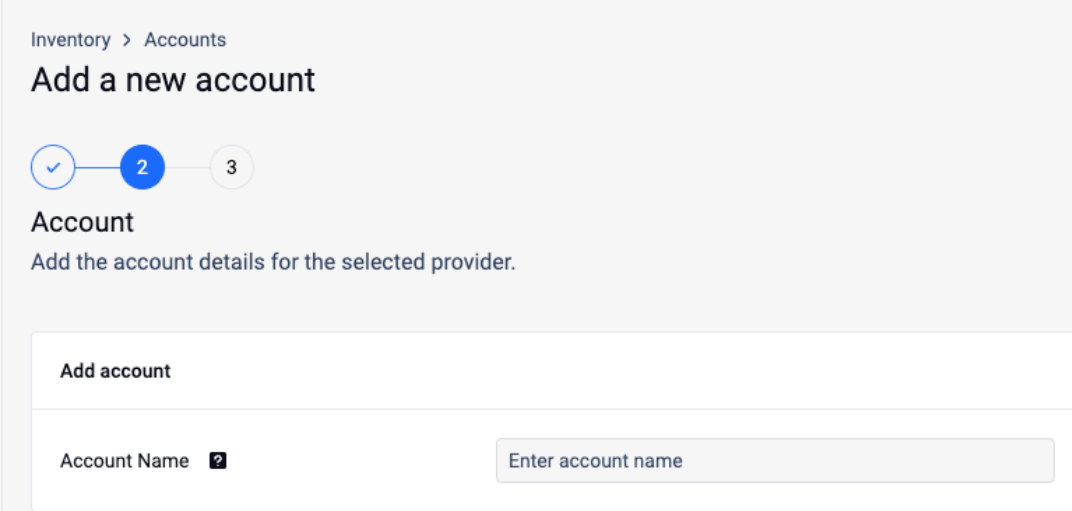
permissions are required for scanning, WRITE permissions are required for remediation operations that implies writing to the storage.

2. Ensure Persistent Mounting:
   - To maintain the mount across reboots, add the following entry to the `/etc/fstab` file: `{nfs_server_ip}:/srv/company-shared-files /mnt/mdss/nfs/company-files nfs rw,hard,intr,nolock 0 0`
   - Without this step, the mount will be lost when the NFS Client machine is restarted.

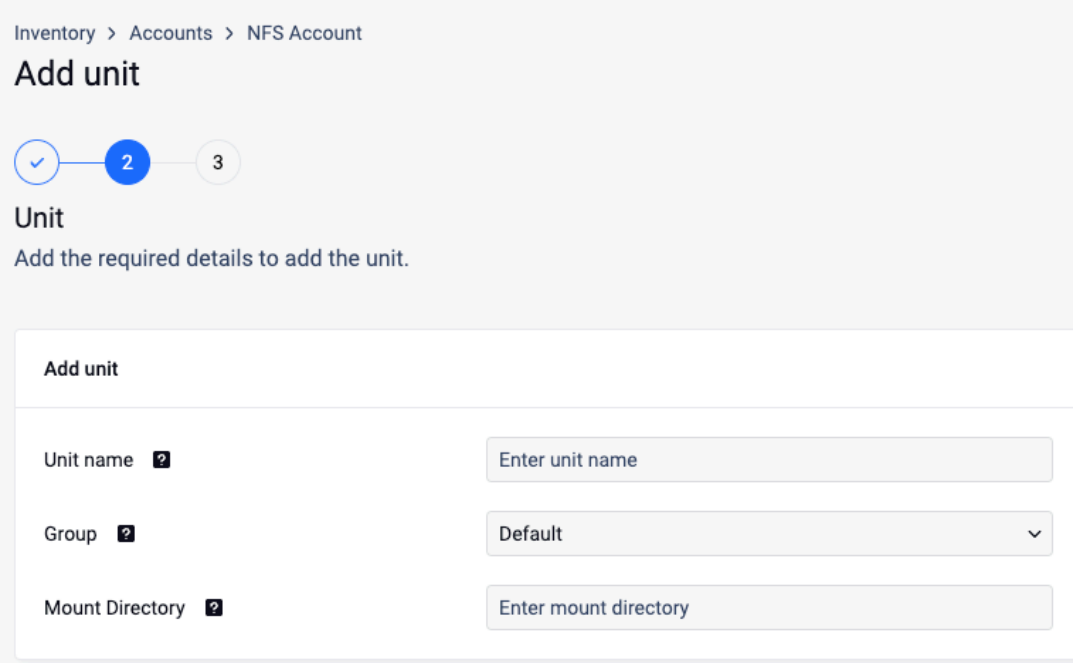
4. Adding an NFS storage in MDSS

In order to integrate NFS with MetaDefender Storage Security, please follow these steps:

1. As NFS is not using any credentials, you can have only one NFS account in MDSS. If you already have one, please go to step 4
2. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **NFS (Network File System)** and click **Next**.
3. Give your account a name so you can easily identify it later



4. From the left side menu, navigate to **Inventory**, click **Add New** > **Unit**, choose the previously created account and click **Next**.



5. Give your unit an unique name so you can easily identify it later
6. Assign your unit to a Group
7. Input the mount directory where your NFS shared files are mounted on the MetaDefender Storage Security machine. For instance, if your files are mounted at `/mnt/mdss/nfs/company-files`, simply enter `company-files` in the text box.
8. Click **Next** in order to finish the process.

> **Warning**
>
> Absolute paths (e.g. `/home`) are not functional in versions prior to 4.4.0. Paths configured this way must be manually reconfigured using relative paths — there is no automatic migration.
>
> Starting in 4.4.0, path normalization (`/home` → `home`) is supported for newly created storage.

5. Additional NFS Configuration

NFS offers a wide range of configurations to suit different needs. For more detailed information on NFS configurations, please refer to the following resources based on your operating system:

- NFS Server and NFS Client configuration on Fedora
- NFS Server and NFS Client configuration on Ubuntu

[Windows] How to add an NFS storage unit

> **Info**
>
> For steps 1 and 2, please see this page

> **Limitation**
>
> NFS shares originating from **Dell OneFS PowerScale** are incompatible with MDSS deployments on Windows. A Linux-based MDSS deployment is required to scan these shares.
>
> To add NFS share on Linux deployment, please follow this article.

3. Setting up the NFS Client machine

On the same machine where MetaDefender Storage Security is installed, run this powershell command from an admin console:

```
Install-WindowsFeature NFS-Client
```

With the default options, a user has read permissions when mounting an NFS share using the anonymous user. We can give the anonymous user write permissions by changing the UID and GID that it uses to mount the share.

To change the UID and GID we must make a simple change to the Windows registry by performing the following steps:

- Open regedit by typing it in the search box end pressing Enter.
- Browse to `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ClientForNFS\CurrentVersion\Default`.
- Create a new New DWORD (32-bit) Value inside the Default folder named `AnonymousUid` and assign the UID 0
- Create a new New DWORD (32-bit) Value inside the Default folder named `AnonymousGid` and assign the GID 0
- Reboot the machine to apply the changes.

4. Mount the NFS share

The below command needs to be run as NTSYSTEM user because NFS service is running as NTSYSTEM user.

To run terminal as NTSYSTEM, please do the following:

- download and extract PsTools
- Open an Administrator console
- cd to PsTools folder
- run `.\PsExec.exe -i -s cmd.exe`. It will open another terminal as a NTSYSTEM user

In the above-opened terminal, run the following command:

```
mount -o "nolock,sec=sys" {nfs-server-ip}:{nfs-share-location}
{mounted-drive}
```

For example, the NFS share will be mounted in the Y drive:

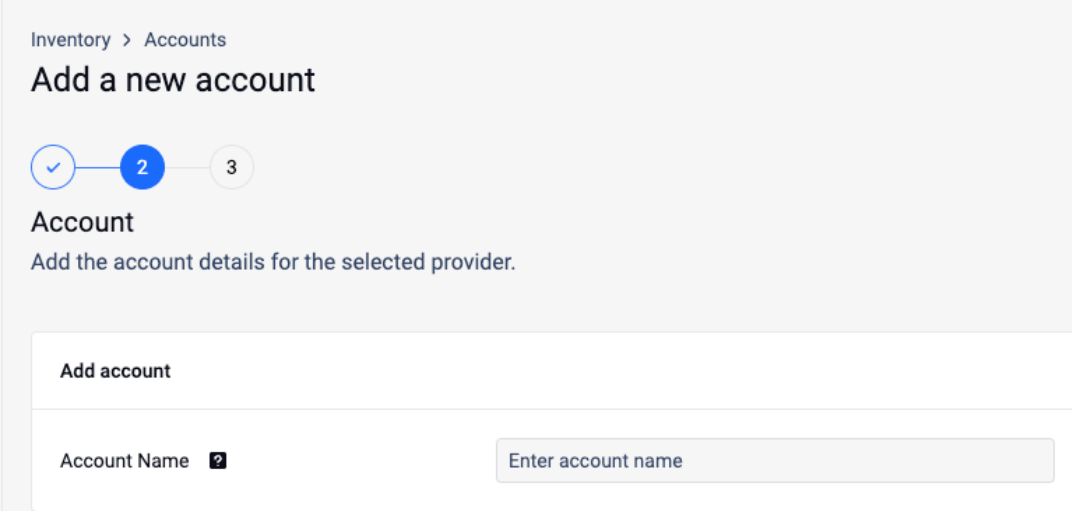
```
mount -o "nolock,sec=sys" 192.168.207.139:/srv/company-public-files/ Y:
```

To make the mount persistent, please follow this article.

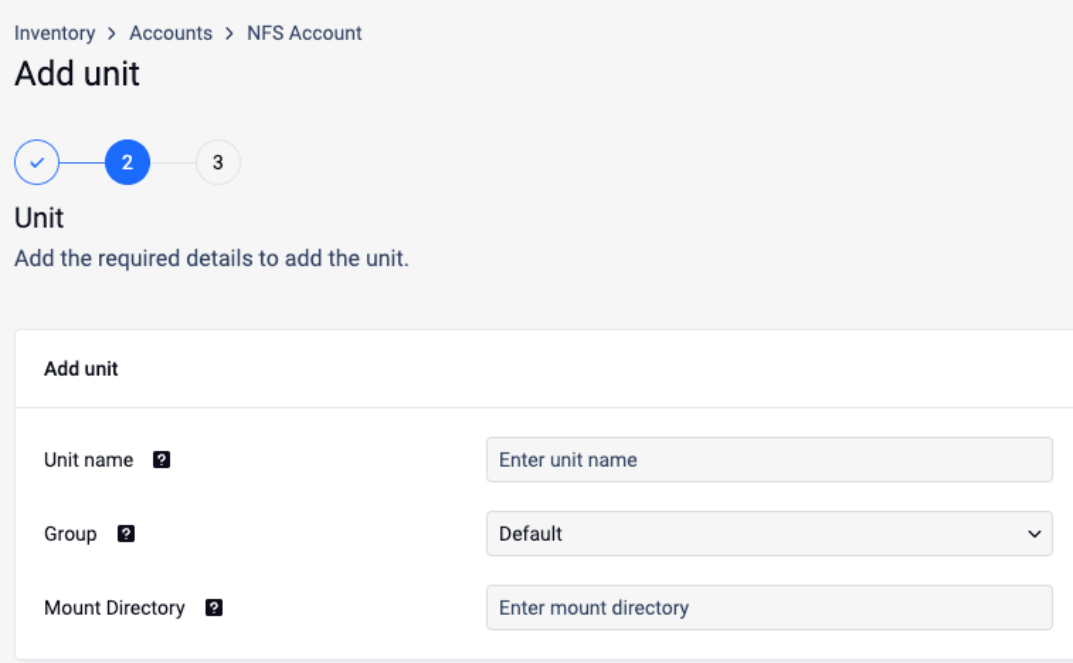
5. Add the NFS share in MDSS UI

In order to integrate NFS with MetaDefender Storage Security, please follow these steps:

1. As NFS is not using any credentials, you can have only one NFS account in MDSS. If you already have one, please go to step 4
2. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **NFS [Network File System]** and click **Next**.
3. Give your account a name so you can easily identify it later



4. From the left side menu, navigate to **Inventory**, click **Add New** > **Unit**, choose the NFS account created above and click **Next**.



5. Give your unit an unique name so you can easily identify it later
6. Assign your unit to a Group
7. In **Mount Directory**, enter the drive the share was mounted on in *4. Mount the NFS share*. In this example, `Y:\`
8. Click **Next** in order to finish the process.

> **Info**
>
> Please follow the Microsoft recommendations for file and directory names.

How to add an Amazon EFS as an NFS storage unit

Prerequisites:

1. an Amazon EFS file storage.
2. a Linux machine where MDSS is installed. This machine needs to be inside AWS and to have access to file storage from 1

Step 1:

On MDSS machine, please install the **cifs-utils** package

Step 2:

On machine where MetaDefender Storage Security is installed, create a new folder at the following path **/mnt/mdss/nfs/{my-new-folder}**

Step 3:

On machine where MetaDefender Storage Security is installed, mount the Amazon EFS file storage in the folder created at step 1. In our testing, we have used the following command, provided by Amazon:

```
sudo mount -t nfs4 -o
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresv
port fs-42f1fa845a1b23ece.efs.eu-central-1.amazonaws.com:/
/mnt/mdss/nfs/{my-new-folder}
```

Please note the mount command has multiple parameters that can be configured in many ways. Feel free to adapt it to your needs. Any configuration works with our product, as long as the Amazon EFS file storage is mounted successfully and we have, at least, read access to it (to benefit from our remediation capabilities, we will need write access).

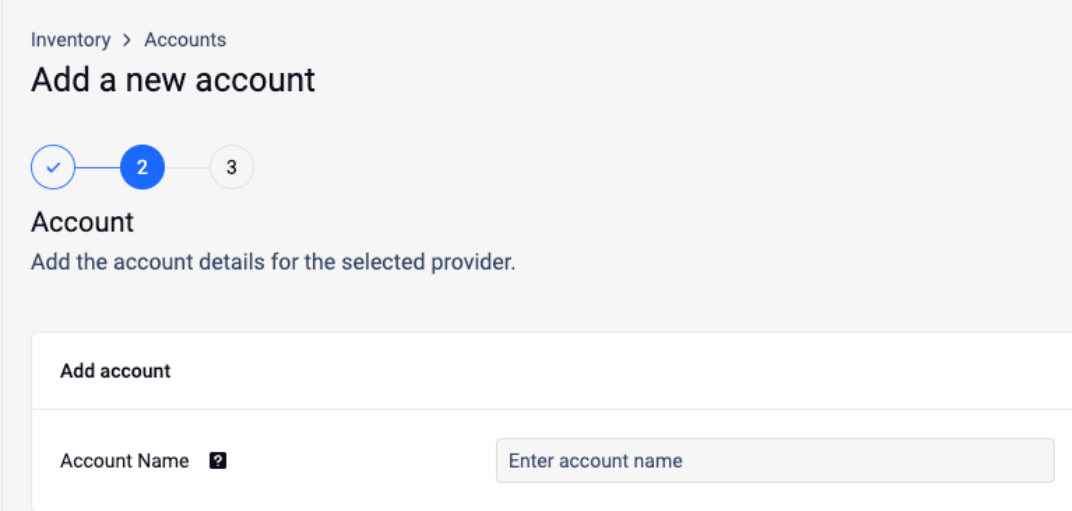
Step 4:

To ensure the mount is persistent, please add it in **/etc/fstab .** Otherwise, the mount will be lost when machine restarts.

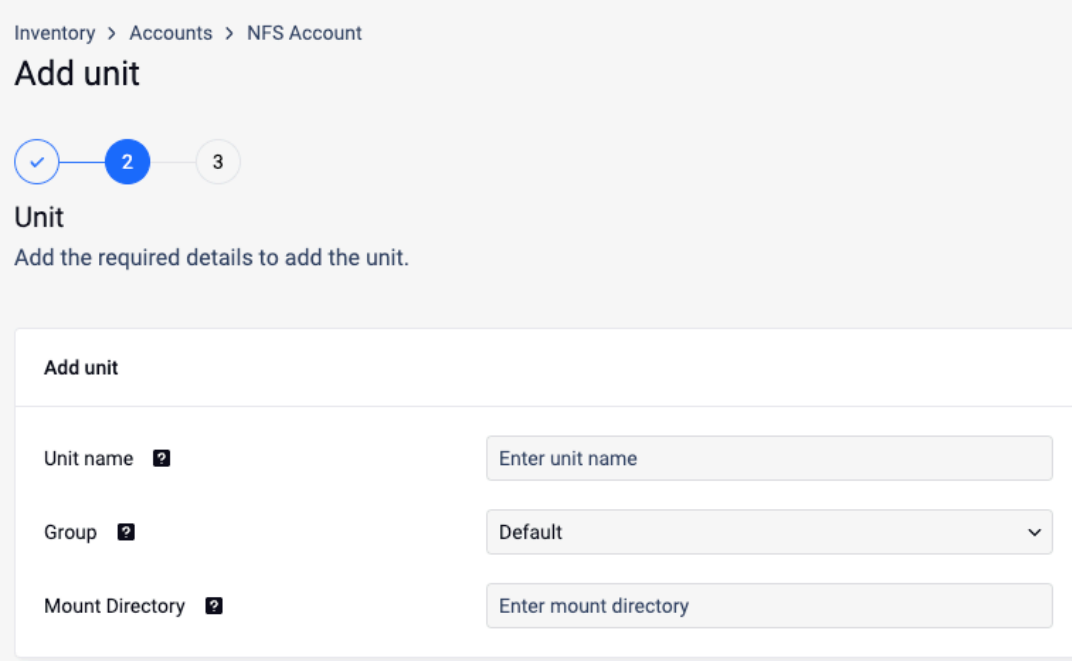
Step 5:

In order to integrate Amazon EFS with MetaDefender Storage Security, please follow these steps:

1. As NFS is not using any credentials, you can have only one NFS account in MDSS. If you already have one, please go to step 4
2. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **NFS (Network File System)** and click **Next**.
3. Give your account a name so you can easily identify it later



4. From the left side menu, navigate to **Inventory**, click **Add New** > **Unit**, choose the previously created account and click **Next**.



5. Give your unit an unique name so you can easily identify it later
6. Assign your unit to a Group
7. Input the mount directory where your Amazon EFS shared files are mounted on the MetaDefender Storage Security machine. For instance, if your files are mounted at `/mnt/mdss/nfs/efs`, simply enter `efs` in the text box.
8. Click **Next** in order to finish the process.

[Windows] How to Expose an SMB Share as an NFS Storage Unit

> This guide uses the **SMB/CIFS protocol** exclusively for all storage operations. Despite references to "NFS storage" in MDSS terminology, no NFS (Network File System) protocol is involved. The term "NFS storage" refers to MDSS's storage subsystem name that is using the local file system API, not the NFS network protocol. **All connections use SMB/CIFS (Common Internet File System).**

1. Map SMB share as a drive

The below command needs to be run as NTSYSTEM user because our internal NFS service is running as NTSYSTEM user.

To run terminal as NTSYSTEM, please do the following (on the same machine where MetaDefender Storage Security is installed):

- download and extract PsTools
- Open an Administrator console
- cd to PsTools folder
- run `.\PsExec.exe -i -s cmd.exe`. It will open another terminal as a NTSYSTEM user

In the above-opened terminal, run the following command:

```
net use {drive-letter} {smb-share-path} /user:{DOMAIN\username}
{password} /persistent:yes
```

Alternatively, if supported, you can map the drive using QUIC transport:

```
net use {drive-letter} {smb-share-path} /user:{DOMAIN\username}
{password} /persistent:yes /TRANSPORT:QUIC
```

For example, this SMB share is mapped to drive G:

```
net use G: \\152.168.212.101\smb-share /user:OPSWAT\smbuser
smbpassword1234 /persistent:yes
```

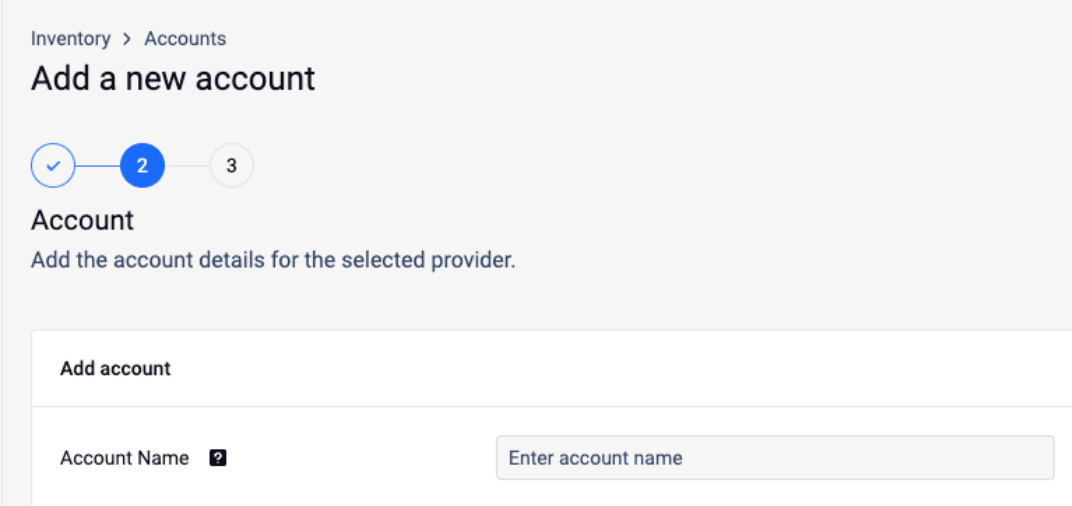
> **Warning**
>
> The SMB share mapping configured using this method is temporary and will not survive system reboots. **The mapping will be automatically removed when the machine is restarted.**

> **To create a persistent mapping** please refer to this article..

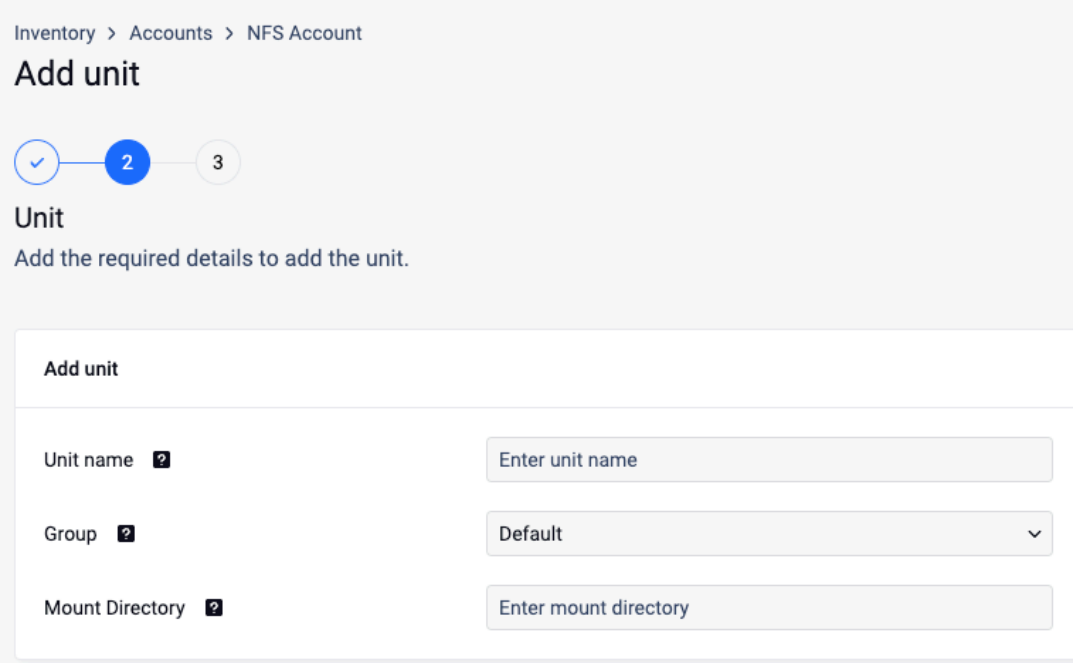
2. Add the SMB share as NFS storage in MDSS UI

In order to integrate SMB share as NFS unit with MetaDefender Storage Security, please follow these steps:

1. As NFS is not using any credentials, you can have only one NFS account in MDSS. If you already have one, please go to step 4
2. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **NFS [Network File System]** and click **Next**.
3. Give your account a name so you can easily identify it later



4. From the left side menu, navigate to **Inventory**, click **Add New** > **Unit**, choose the previously created account and click **Next**.



5. Give your unit an unique name so you can easily identify it later
6. Assign your unit to a Group
7. In **Mount Directory**, enter the drive the share was mapped to in *1. Map SMB share as a drive*. In this example, `G:\`
8. Click **Next** in order to finish the process.

> **Info**
>
> Please follow the Microsoft recommendation for file and directory names.

How to Enable NFS scanning on a Kubernetes Cluster

In order to scan an NFS share when having MDSS deployed in a Kubernetes cluster, the share location has to be mounted in the `nfsservice` container as a directory in the following path: `/opt/mdss/nfs/` . For example, the following values file can be applied when deploying MDSS using the helm chart.

```
mdss_components:
  nfsservice:
    extraVolumes:
      nfsTest:
        name: nfs-storage-unit
        nfs:
          server: <NFS_SERVER_IP>
          path: <PATH_EXPORTED_ON_SERVER>
          readOnly: no
    extraVolumeMounts:
      - name: nfs-storage-unit
        mountPath: /opt/mdss/nfs/<MOUNT_DIRECTORY>
        subPath: <SUBPATH_TO_SCAN>
```

This example file can also be found on our GitHub repo for k8s deployments here

After replacing the placeholders, these values can be applied when deploying MDSS:

```
helm install mdss <PATH_TO_HELM_CHART> -f mdss-nfs-example.yml
```

Notes:

- Any storage unit that can be mounted as a Persistent Volume in the container can also have it's contents scanned using the NFS module
- Make sure that the share is mounted with the appropriate permissions, for example, whenever remediations or CDR are configured write permissions are also required

What is User Squashing for Network File System (NFS)

In NFS (Network File System), **user squashing** is a security mechanism that controls how user IDs (UIDs) are mapped between the client and server machines. It helps protect the server from potential security risks when users connect with elevated privileges.

Here's a breakdown of different concepts, containing an example of how the `/etc/exports` file look like:

1. Root Squashing:

- By default, root squashing is enabled. This means when a user with root privileges (UID 0) on the client attempts to access the NFS share, their UID is mapped to a specific user on the server, typically "nobody" (UID 65534). This user typically has limited permissions, preventing the client root user from performing actions that could compromise the server.

2. All Squashing:

- This option maps all client user IDs to a single user on the server, usually “nobody.” This provides minimal permissions to all users accessing the share, regardless of their privileges on the client machine. Use this cautiously as it can severely limit functionality.

```
/shared/data *(rw,sync,nfsvers=4,all_squash)
```

- There is also the option to map all client user IDs to a specific user. You will need to create the dedicated user, set permissions and assign it to a group. Here is an example of how `/etc/exports` file looks

```
/path/to/shared/folder *(rw,sync,root_squash,usermapping=
(to="dedicated_user", squash))
```

3. No_root_squash:

- This option disables user squashing altogether. It allows client users, including root, to retain their original UIDs on the server. This can be risky as it grants potentially harmful permissions to users, especially root. Only use this in trusted environments and with explicit security measures.

```
/shared/data *(rw,sync,nfsvers=4,no_root_squash)
```

You can find more details on the official documentation of the NFS protocol: https://docs.kernel.org/admin-guide/nfs/index.html

Add an NFS with Data Wrapping Storage

NFS with Data Wrapping works like a standard NFS storage, except that the files on the share are kept in a wrapped form. Each file is unwrapped as it gets read, so scans always run on the original content, and wrapped as the file gets written to the share.

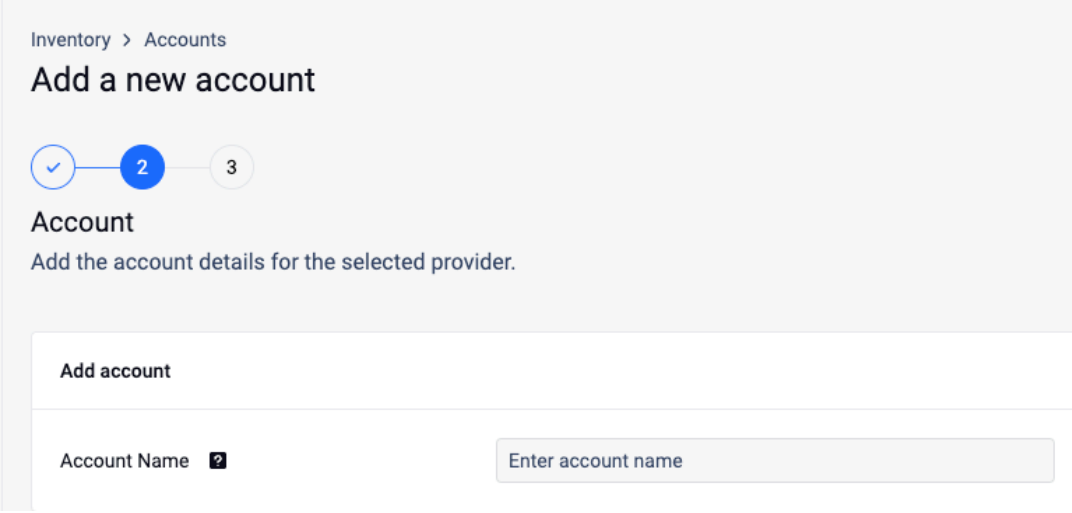
Step 1:

To scan a storage that contains wrapped files, you have to integrate it as an NFS storage unit. Please see this page.

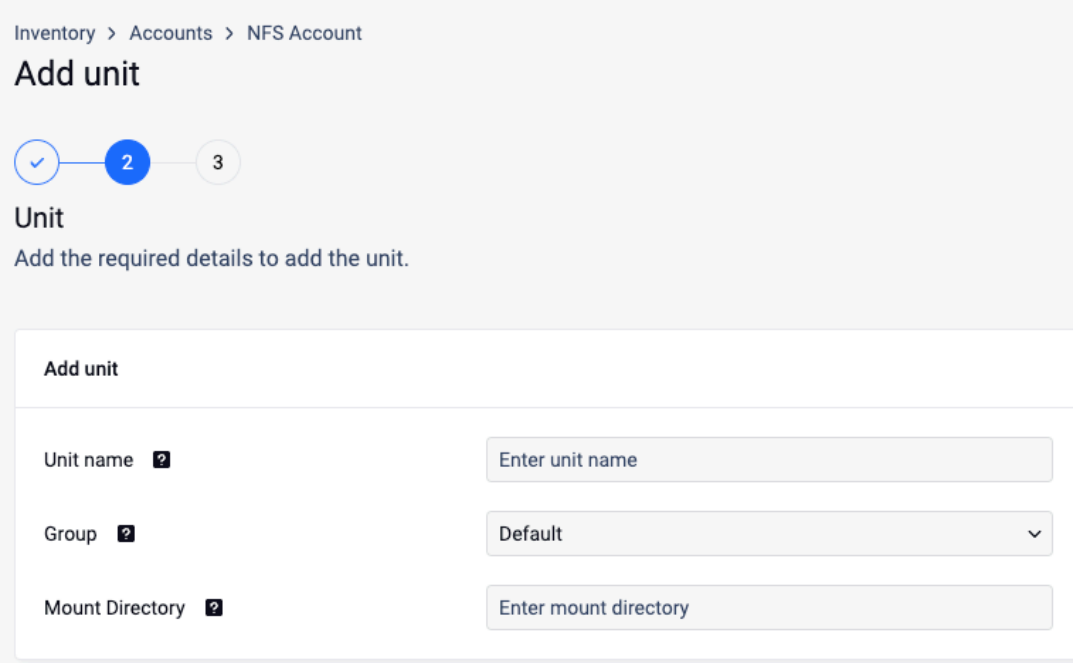
Step 2:

In order to integrate NFS with Data Wrapping with MetaDefender Storage Security, please follow these steps:

1. As NFS with Data Wrapping is not using any credentials, you can have only one NFS with Data Wrapping account in MDSS. If you already have one, please go to step 4
2. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **NFS with Data Wrapping** and click **Next**.
3. Give your account a name so you can easily identify it later



4. From the left side menu, navigate to **Inventory**, click **Add New** > **Unit**, choose the previously created account and click **Next**.



5. Give your unit an unique name so you can easily identify it later
6. Assign your unit to a Group
7. Input the mount directory to the folder created at Step 1.
8. Click **Next** in order to finish the process.

Info

Remediation behavior is as follows:

- Files moved between NFS with Data Wrapping storage units remain unchanged
- Files moved from NFS with Data Wrapping to another storage type (such as Amazon S3) are unwrapped during the move
- Files moved from another storage type (such as Amazon S3) to NFS with Data Wrapping are wrapped during the move

Warning

The wrap/unwrap action is not configurable.

Add an SMB-Compatible Storage

Product Specification Requirements	Version
Service Message Block (SMB) protocol	Windows deployments: SMB 2.0.2 to SMB 3.1.1. Linux deployments: SMB 3.1.1, 3.0 or 2.1.

> **Info**
>
> Integration with SMB Compatible Storage is accomplished via the Service Message Block (SMB) protocol over TCP. Before you begin, make sure SMB is enabled on the file server and that the MDSS machine can reach it on TCP port 445. Linux deployments also accept a custom port; Windows deployments use port 445 only.

Please choose your deployment type:

- Add an SMB-Compatible unit in Windows deployments
- Add an SMB-Compatible unit in Linux deployments

How to scan Windows client SMB shares?

> **Info**
>
> Windows client shares are **not officially supported**. This document is intended solely for demonstration, testing, and proof-of-concept (PoC) purposes to help you learn how to scan them. It is not suitable for use in production environments.

> **Check Your Version:**
>
> This article applies to the legacy SMB service only: Linux deployments that use the `smb_legacy` module, and Windows deployments with `USE_OPSWAT_SMB_SERVICE=false`. The default SMB service on Windows (4.5.0 and later) and on Linux (4.6.0 and later) does not use `SMBSERVICE_SESSIONS_ON_STORAGE`.

Symptoms

When the connection limit is reached, the following error will appear in SMB service logs:

```
No more connections can be made to this remote computer at this time
because the computer has already accepted the maximum number of
connections. (3221225680) STATUS_REQUEST_NOT_ACCEPTED: 0xc00000d0
```

Solution

When SMB shares are configured on a Windows client (Windows 10, Windows 11) there is a limit of connections that the SMB server allows. By default the limit is a total of 20 connections.

With its default configuration, the legacy SMB service can open up to 50 connections to the SMB server. However, it is possible to adjust certain configurations on the MDSS side to accommodate scenarios where fewer connections are necessary.

First, check how many connections are used by clients other than MDSS. You can check this using the following command that should be executed on the SMB Server with Administrative privileges

```
Get-SMBSession
```

It will return a list of open sessions connections. Based on the number of connections you can adjust the custom configuration using an environment variable called **SMBSERVICE_SESSIONS_ON_STORAGE** not have more than 20 connections used on the SMB Server.

> **Info**
>
> The legacy SMB service runs 5 processes, and each can open up to **SMBSERVICE_SESSIONS_ON_STORAGE** connections (default 10), which results in 50 connections.
>
> For Windows clients SMB shares, the recommendation is to configure **SMBSERVICE_SESSIONS_ON_STORAGE** to 2 or 3, but it depends of the number of connections already present.

Recommended approach when adding SMB units on Linux deployments

> This article is applicable only for MetaDefender Storage Security deployed on Linux. If your deployment is on Windows, please follow Add an SMB-Compatible unit in Windows deployments.

Starting with version 4.6.0, the recommended way to add an SMB share on a Linux deployment is the **SMB Compatible** storage type. The SMB service mounts each share on demand with the SMB/CIFS client built into the Linux kernel, so the deployment gets the stability and performance of an operating-system mount without any manual mount, `fstab` entry or `cifs-utils` package on the MDSS machine. Follow Add an SMB-Compatible unit in Linux deployments.

Releases before 4.6.0 relied on an application-level SMB client, and for production environments the recommendation was to mount the share on the MDSS machine and add it as an NFS storage unit instead. Units configured that way keep working after the upgrade. To move them to the SMB service, add the share as an SMB Compatible storage and remove the NFS unit and the host mount.

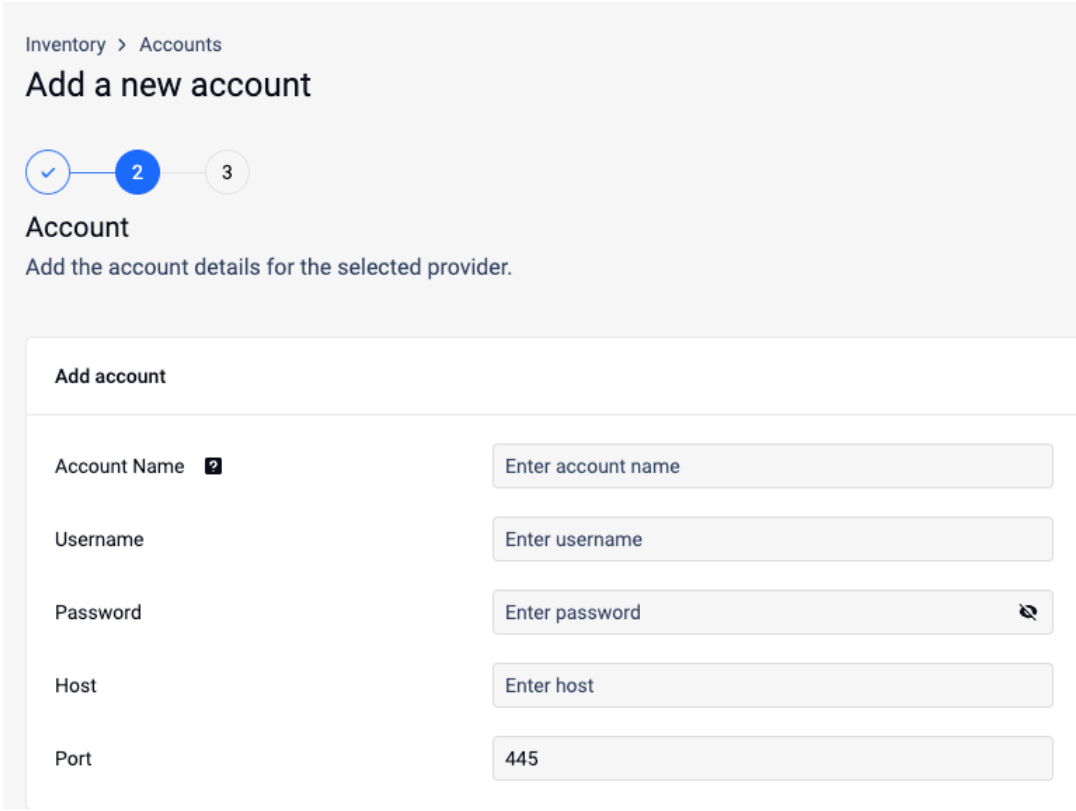
Hosts that cannot run the new SMB service (no `cifs` kernel module, a rootless container runtime, or a policy that refuses its container privileges) can use the previous SMB service through the `smb_legacy` module. See *Fallback: the legacy SMB service* in Add an SMB-Compatible unit in Linux deployments.

Add an SMB-Compatible unit in Windows deployments

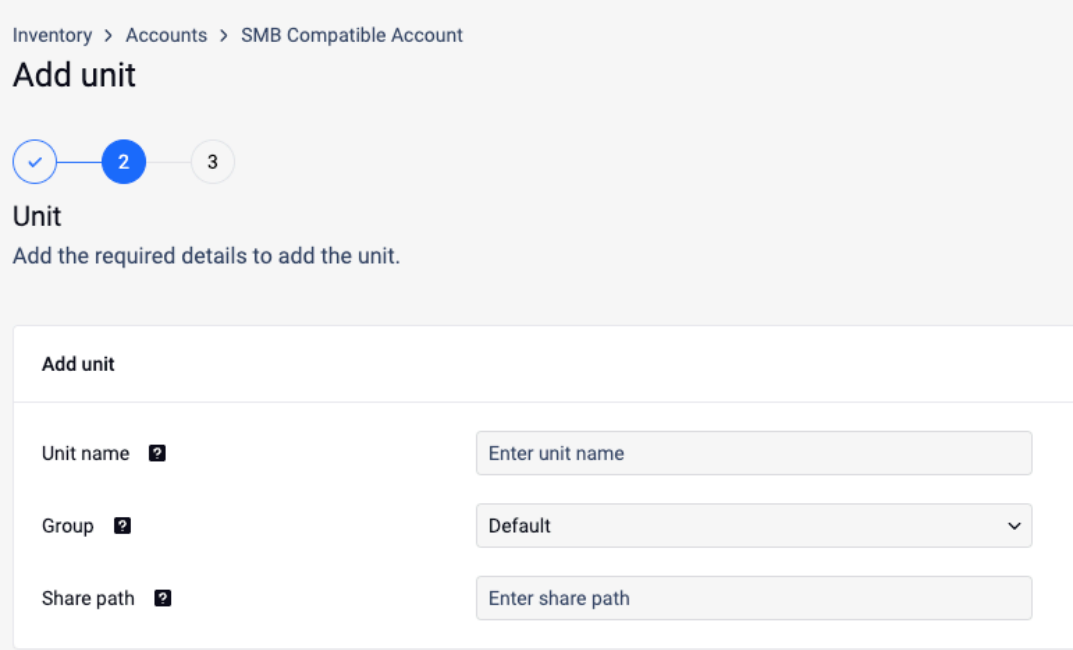
> This article is **applicable only for MetaDefender Storage Security** deployed on **Windows Server**.
>
> If your MetaDefender Storage Security is deployed on **Linux** based distributions, please follow this article.

In order to integrate SMB Compatible Storage with MetaDefender Storage Security please follow these steps:

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **SMB Compatible** and click **Next**



2. Give your account a name so you can easily identify it later
3. Enter the name of the user that will be used to access the SMB share
4. Enter the password of the above user.
5. Enter the IP (or hostname) of the SMB server you wish to connect to.
6. Specify the port number used for the SMB connection, **445** being the default SMB port
7. Click on Next to finish the process.
8. From the left side menu, navigate to **Inventory**, click **Add New** > **Unit**, choose the previously created account and click **Next**



9. Give your unit a unique name so you can identify it later.
10. Assign your unit to a Group
11. Indicate the specific share path on the SMB server from which you want to access files. The Share path is the share name as published on the SMB server, optionally followed by a subfolder path inside that share. It must not contain the server. The server comes from the account, and MDSS builds the UNC path `\\<server>\<share path>` itself.

    Format: `<share name>[/<subfolder>/<subfolder>...]`

    - Forward slashes and backslashes are both accepted.
    - Leading and trailing slashes are ignored.
    - The value is used exactly as typed, so avoid leading or trailing spaces. **Examples** (server 10.0.0.5 configured on the account):

Share path	Connects to
finance	\10.0.0.5\finance
finance/reports	\10.0.0.5\finance\reports
finance\reports\2026	\10.0.0.5\finance\reports\2026
Backups$	\10.0.0.5\Backups$ (hidden share)

12. Click **Next** in order to finish the process.

> **Info**

To avoid authentication issues the Username should have one of the following formats:

- local users should be .\username (for example: .\Administrator)
- domain users should be username@domain or domain\username

Info

A Samba file server needs to be enabled in order to connect to a Unix-based system **SharePath.** Since Samba doesn't use the system account password, it needs to be set up using the *sudo smbpasswd -a username* command.

Info

If you use a hostname in the host portion of IPv4/v6 socket address, the program may show a non-deterministic behavior. The socket address will be resolved differently into an actual IPv4/v6 address, depending on the results from DNS resolution and/or the host configuration. For deterministic behavior use a numeric address in host portion.

Info

Windows client shares have a default configuration of 20 connections in the same time. Our default configuration will require 50 connections. To adjust this SMBSERVICE_SESSIONS_ON_STORAGE should be set to a value lower than 4

Info

Please note that SMB over QUIC is not supported at this time - only traditional SMB over TCP connections are compatible with this integration. See [Windows] How to Expose an SMB Share as an NFS Storage Unit for more details.

Add an SMB-Compatible unit in Linux deployments

This article is applicable only for MetaDefender Storage Security deployed on Linux distributions. For Windows deployments, see Add an SMB-Compatible unit in Windows deployments.

Starting with version 4.6.0, MetaDefender Storage Security (MDSS) connects to SMB shares on Linux through its **SMB service**, which mounts each share on demand using the

SMB/CIFS client built into the Linux kernel. The share is added in MDSS as an **SMB Compatible** storage. No manual mount, `fstab` entry or `cifs-utils` package is needed on the MDSS machine.

> Before 4.6.0, SMB Compatible units on Linux were served by the previous SMB service, which connects to shares over the network without mounting them. That service is still shipped; see *Fallback: the legacy SMB service* below. Mounting the share on the MDSS machine and adding it as an NFS storage unit was the alternative recommended for production; see the last section of this article.

Prerequisites

- A Linux machine with MetaDefender Storage Security 4.6.0 or later installed.
- An SMB share hosted on a Windows machine, a NAS, or a Linux machine running Samba, reachable from the MDSS machine on TCP port 445 or on the custom port the server uses.
- The `cifs` kernel module available on the MDSS machine (see Step 1).

> SMB over QUIC is not supported at this time - only traditional SMB over TCP connections are compatible.

Step-by-Step Guide

Step 1: Make sure the cifs kernel module is available

The SMB service mounts shares with the kernel's CIFS driver, so the module has to be available on the MDSS machine itself; a container cannot load kernel modules. On most distributions the module ships with the kernel and is loaded automatically the first time a share is mounted.

To check that the module is available, run:

```
grep cifs /proc/filesystems
```

If the command prints nothing, load the module:

```
sudo modprobe cifs
```

To keep it loaded across reboots on machines where automatic module loading is disabled:

```
echo cifs | sudo tee /etc/modules-load.d/cifs.conf
```

> `sudo mdss -c start` checks this prerequisite before it pulls the images and stops with instructions when the module is missing. If a storage is added while the module is missing, MDSS reports *The CIFS kernel module is not installed or not enabled on the machine running MetaDefender Storage Security. Install or enable the module, then try again.*

Step 2: Container privileges

Mounting a filesystem requires the `SYS_ADMIN` capability, and Docker's default AppArmor profile denies mount operations. The `docker-compose.smb.yml` file shipped with MDSS therefore starts the SMB service container with:

```
cap_add:
  - SYS_ADMIN
security_opt:
  - apparmor:unconfined
```

No change is needed for a standard installation. `sudo mdss -c start` verifies after the start that the SMB service container is running with these privileges and reports the exact problem when it is not.

Keep the following in mind if your organization applies a hardening policy to containers:

- `SYS_ADMIN` is granted to the SMB service container only. No other MDSS container receives it.
- `apparmor:unconfined` has no effect on machines without AppArmor. On machines with AppArmor, `sudo mdss -c start` accepts only `unconfined`: with any other profile it stops with the fallback instructions, unless `SKIP_SMB_CHECKS=yes` is set in `/etc/mdss/customer.env`.
- Shares are mounted inside the container's own mount namespace. They never appear on the MDSS machine's filesystem and are released when the container stops.
- Every share is mounted with the `nosuid`, `nodev` and `noexec` options, so nothing stored on the share can be executed inside the container. On Linux kernel 5.10 or later, shares are also mounted with `nosymfollow`, so symbolic links found on the share are not followed.

> If the container runtime cannot grant these privileges (a hardened Docker daemon, a policy engine, or a rootless runtime), the SMB service cannot mount shares. Use the legacy SMB service instead. See *Fallback: the legacy SMB service* below.

Step 3: Add the SMB account in MDSS

1. Navigate to **Inventory** > **Add New** > **Account**, choose **SMB Compatible** and click **Next**.
2. Give your account a name so you can easily identify it later.
3. Enter the **Username** and **Password** of the account used to connect to the share.
4. Enter the **Host** (host name or IP address of the SMB server) and the **Port**. Leave the port at 445 unless the server listens on a different port.
5. Click **Next**. MDSS verifies the credentials against the server before the account is saved.

> To avoid authentication issues the Username should have one of the following formats: domain users should be `domain\username`, `domain/username` or `username@domain`; local users on the server should be the username alone. Passwords containing `,` `;` `=` or spaces are supported.

> A Samba file server needs to be enabled in order to connect to a Unix-based system share.

Step 4: Add the SMB unit in MDSS

1. Navigate to **Inventory** > **Add New** > **Unit**, select the account created in Step 3 and click **Next**.
2. Give your unit a name and assign it to a group.
3. Enter the **Share path**: the name of the share, optionally followed by a folder inside it, for example `share` or `share/folder/subfolder`. When a folder is given, only that folder and its subfolders are discovered and scanned.
4. Click **Next** in order to finish the process. MDSS mounts the share once to verify that the path exists and that the account can access it.

> The share path must start with the share name. A path such as `folder/subfolder` without the share name is reported as *The object was not found on the server*. The same rule applies to the partition of a scan on an SMB unit: it replaces the share path, so it must include the share name as well.

Custom configuration

The SMB service reads its settings from environment variables. To change one, add it to `/etc/mdss/customer.env` in the format `CONFIGURATION_PROPERTY_NAME=VALUE`, save the

file and restart MetaDefender Storage Security with `sudo mdss -c restart`. See Custom configuration. The defaults suit most deployments.

Variable	Default	Description
SMBSERVICE_MOUNT_DIRECTORY	/var/opswat/mdss/smb-mounts	Directory inside the SMB service co shares are mounted.
SMBSERVICE_MOUNT_IDLE_TTL_SECONDS	300	A share that has not been used for unmounted. It is mounted again th needed.
SMBSERVICE_MOUNT_TIMEOUT_SECONDS	60	Maximum time to wait for a mount
SMBSERVICE_SMB_SEAL	0	Set to 1 to require SMB 3 encryptio Servers that do not support encryp mount.
SMBSERVICE_MOUNT_UID / SMBSERVICE_MOUNT_GID	Not set	Owner reported for the files on the gid mount options). The permissio server still decide what MDSS can r
SMBSERVICE_MOUNT_OPTIONS	Not set	Additional CIFS mount options, cor example cache=strict, actimeo=30, e Options that the SMB service mana server address, port, SMB version, uid, gid) are not accepted here: t start and logs the reason.
SMBSERVICE_DISCOVERY_DEBOUNCE_TIME	5	A file modified less than this many considered still being written and i next discovery. Set it to 0 to proces are found. On Linux, MDSS cannot r another client still has a file open o window is the main protection aga partially copied file. Increase it whe the share by slow clients.
SMBSERVICE_DOWNLOAD_CHUNK_TIMEOUT_SECONDS	60	Maximum wait for each block of da (30 to 300). On Linux, a read the ke not interrupted by this timeout: it e gives up on the session, about thre default settings (see the note belo
SMBSERVICE_UPLOAD_CHUNK_TIMEOUT_SECONDS	60	Maximum wait for each block of da (30 to 300).

> When an SMB server stops responding in the middle of a transfer, the kernel keeps the session open for about three minutes before it reports the failure. The kernel reconnects on its own once the server answers again. Add `echo_interval=30` to `SMBSERVICE_MOUNT_OPTIONS` to shorten that window.

Fallback: the legacy SMB service

The SMB service from earlier releases, which reads shares over the network without mounting them, is still shipped as the `smb_legacy` module. It needs no extra container privileges and no kernel CIFS support. The settings above do not apply to it by default: it skips a file that another client still holds open, and uses `SMBSERVICE_DISCOVERY_DEBOUNCE_TIME` only when `SMBSERVICE_FILE_READY_CHECK_MODE=0` is set.

1. Open `/etc/mdss/customer.env` and set `ENABLED_MODULES` to the complete list of modules you want enabled, keeping `smb` in the list and adding `smb_legacy` at the end:

```
ENABLED_MODULES=azureblob,amazonsdk,googlecloud,alibabacloud,azurefiles
,smb,box,graph,sftp,nfs,mft,ftp,oraclesdk,smb_legacy
```

2. Restart MetaDefender Storage Security:

```
sudo mdss -c restart
```

> `smb` has to stay in the list: it is the module that enables the SMB integration in the product. `smb_legacy` only selects which service implements it. `customer.env` replaces the value from `.env`, so the line must list every module you want enabled.

> When `sudo mdss -c start` finds that the MDSS machine cannot run the SMB service (missing cifs module, rootless container runtime, privileges refused), it stops and prints this exact line to set. If a check reports a problem that does not apply to your machine, for example a custom AppArmor profile that you know permits `mount`, set `SKIP_SMB_CHECKS=yes` in `/etc/mdss/customer.env`.

Troubleshooting

The message shown in MDSS when an account or unit is added identifies the cause. The SMB service log, in `services_logs/mdintegrations_smbservice` under the installation

directory, carries the details: the server address, the share, the SMB version and the error code returned by the kernel.

When an account is added, every failure to reach the server (a name that cannot be resolved, an address with no server behind it, or a port with nothing listening on it) is currently reported as *Failed to connect to the server. The name could not be resolved.* The connection and timeout messages in the table below appear when a unit is added.

Message shown in MDSS	Cause and what to do
The CIFS kernel module is not installed or not enabled on the machine running MetaDefender Storage Security.	The `cifs` module is not available on the MDSS machine. Follow Step 1.
Authentication failed. Please verify your credentials.	The server rejected the username or password, or the account has no access to the share. Check the credentials and the account's permissions on the share. Domain accounts may need the domain prefix.
The object was not found on the server. Please verify your path and permissions.	The share, or a folder in the share path, does not exist on the server. Make sure the share path starts with the share name and that every folder in it exists.
Failed to connect to the server. The name could not be resolved. Please verify that the path or hostname is correct and reachable.	The server name cannot be resolved, or nothing is listening on the configured port. Verify the server address and the port.
Connection to the unit share could not be established.	The kernel reported the server or its network as unreachable, or MDSS could not reach the SMB service container. Check the route and the firewalls between the MDSS machine and the server on the SMB port, and check that the `mdintegrations_smbservice` container is running.
IO timeout. Please verify your configuration and try again.	The kernel reported a timeout while connecting to the server. Check the route and the firewalls between the MDSS machine and the server on the SMB port.
Communication with the unit share failed. Please check that the username and password are correct.	The server did not answer (it is down, or a firewall drops the traffic), the mount did not complete within `SMBSERVICE_MOUNT_TIMEOUT_SECONDS`, the server rejected the mount options or the SMB version negotiation failed, the account can mount the share but cannot list the folder, or the SMB service container is running without the privileges described in Step 2. Check that the server is up, reachable on the SMB port and supports SMB 2.1 or later, and that the account can list the folder. Run `sudo mdss -c restart` and read its output, which reports missing container privileges, and check the kernel log below.

For any mount failure, the kernel log on the MDSS machine shows the CIFS client's own diagnostics:

```
sudo dmesg | grep 'CIFS: VFS'
```

A mount that fails does not leave a session open on the server, so a corrected configuration can be retried immediately.

Notes

- The SMB service negotiates SMB 3.1.1 first, then SMB 3.0, then SMB 2.1, and stops at the first version the server accepts. SMB 1 is never used.
- A share is mounted on first use and unmounted after 5 minutes without activity. A scan that starts later mounts it again.

Units added as NFS storage in earlier releases

Before 4.6.0, the setup recommended for production was to mount the share on the MDSS machine with `cifs-utils` and add it in MDSS as an NFS storage unit pointing at the mount directory. Units configured that way keep working after the upgrade. To move a unit to the SMB service, add the share as an SMB Compatible storage as described above, then remove the NFS unit and the mount from the MDSS machine.

Add a NetApp ONTAP Storage

NetApp ONTAP is a powerful data management software that provides unified storage for various protocols, including NFS, SMB/CIFS, iSCSI, and Fibre Channel.

It offers features like high availability, data protection, and storage efficiency, making it a popular choice for enterprise storage needs.

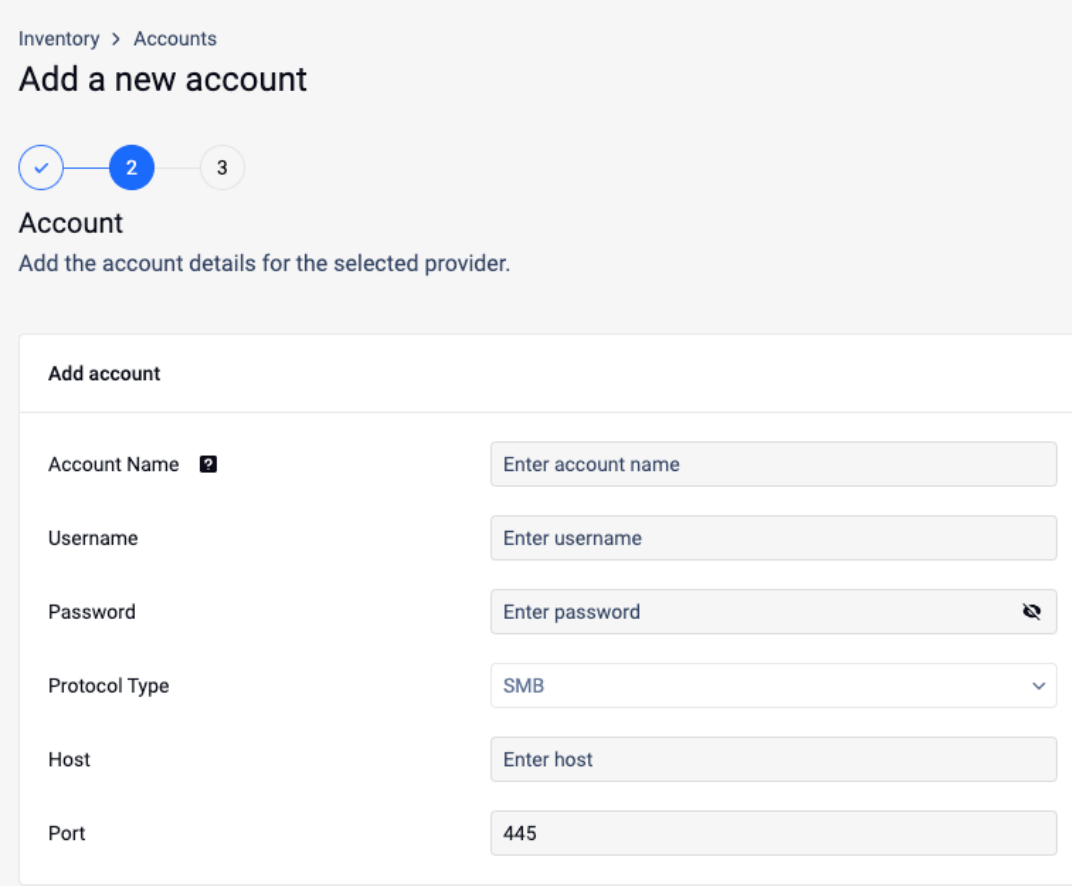
> **Key information:**
>
> To connect with your NetApp ONTAP storage, we rely on the Service Message Block (SMB) protocol. **Please enable SMB on your storage system beforehand for seamless integration.**

Product Specification Requirements	Version
Service Message Block (SMB) protocol	SMB 3.1.1 (or SMB3)
NetApp ONTAP	9.15.1 or later

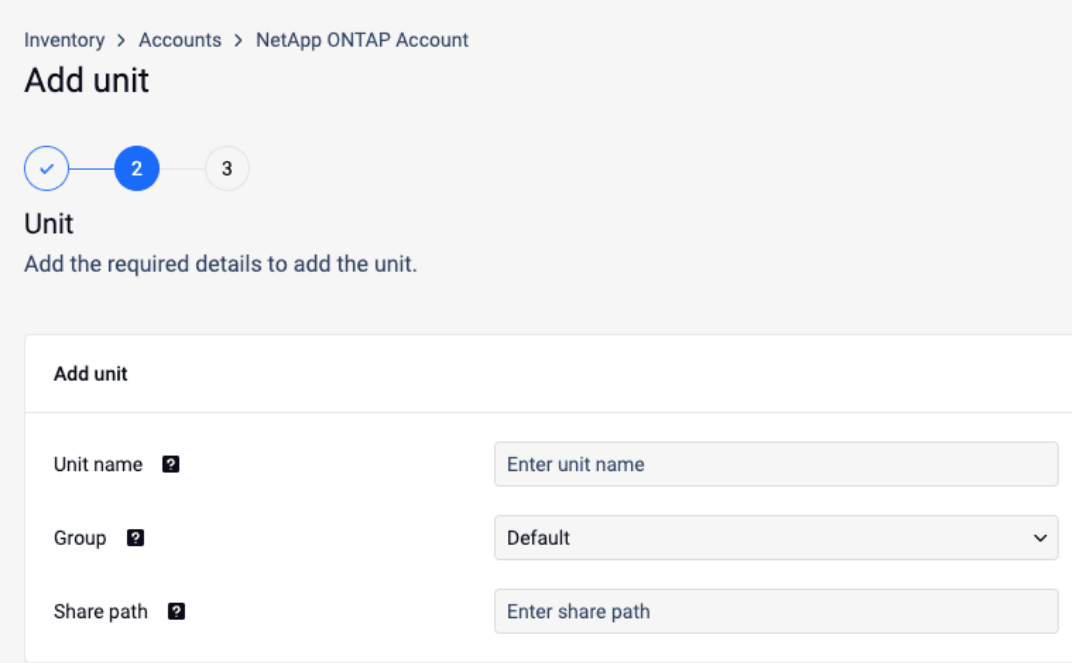
> **Key Information:**
>
> To use the virus scanning features (VSCAN) of your NetApp ONTAP storage with MetaDefender Storage Security **on your Windows deployment**, check out this article for detailed instructions.

To integrate your NetApp ONTAP storage with this system, follow these simple steps:

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **NetApp** and click **Next**



3. **Provide an Account Name:** Give your account a name so you can easily identify it later
4. **Enter User Credentials**:
    1. Provide the username of an account with at least read and write permissions to the desired share.
    2. For Deep CDR & Remediations capabilities, the user needs modify permissions as well.
    3. Enter the password for the specified user.
5. Enter the IP address of the SMB server you wish to connect to.
6. Specify the port number used for the SMB connection, **445** being the default SMB port
7. Click on **Next** to finish the process.
8. From the left side menu, navigate to **Inventory**, click **Add New** > **Unit**, choose the previously created account and click **Next**



8. Give your unit an unique name so you can easily identify it later
9. Assign your unit to a Group
10. **Specify the Share Path.** Enter the path to your network share using one of the supported formats:
    1. file://servername/pathToShare
    2. \servername\pathToShare
    3. //servername/pathToShare
    4. smb://servername/pathToShare
11. **Complete the Process:** Click **Next** to finish integrating your NetApp ONTAP storage

Once added, you can use this storage within the system for various functionalities, leveraging the capabilities of NetApp ONTAP for data management and access.

Vscan | How to enable

Step 1: Understanding Vscan

What is Vscan?

Vscan is an antivirus scanning solution developed by NetApp that allows customers to protect their data from being compromised by viruses or other malicious code. It combines partner-provided antivirus software with ONTAP features to give customers the flexibility they need to manage file scanning.

Major components

- **Vscanner** or **Vscan Server**) - This is a Microsoft Windows server used for antivirus (AV) scanning.
- **Vscan engine** - This is the partner-supplied antivirus (AV) software that runs on Vscanner(s).
- **ONTAP Antivirus Connector** - This is NetApp-provided software that runs on Vscan Server
- **LIF (logical interface)** - This is the IP address used to access a cluster
  - To send files for virus scanning, you must configure ONTAP Antivirus Connector to connect to one or more SVMs by entering the Data ONTAP management LIF, the poll information, and the account credentials.
  - The management LIF is polled to retrieve the list of data LIFs
- **Storage Virtual Machine (SVM)** or **virtual storage server (Vserver)** - this is the logical storage object in NetApp ONTAP software
  - Vscan configuration is defined at the SVM level. The scope of the configuration is the SVM, and it only operates on SVM resources.

Vscan configurable components

- **Scanner pool** - This component defines the Vscanners and privileged users that can connect to SVMs. It also defines a scan request timeout period, after which the scan request is sent to an alternative Vscanner, if one is available.
- **Privileged user** - This component is a domain (recommended) user account that a Vscanner uses to connect to the SVM. The account must exist in the list of privileged users in the scanner pool. To have a local user as the privileged user, please follow this NetApp article.
- **Scanner policy** - This component determines whether a scanner pool is active. Scanner policies are system-defined, so you cannot create custom scanner policies. Only these three policies are available
  - **Primary** specifies that the scanner pool is active
  - **Secondary** specifies that the scanner pool is active, only when none of the Vscanners in the primary scanner pool are connected
  - **Idle** specifies that the scanner pool is inactive
- **On-Access Policy** - This component defines the scope of an on-access scan
  - You can specify the maximum file size to scan, file extensions and paths to scan, and file extensions and paths to exclude from the scan

  - By default, **only read-write** volumes are scanned. You can specify filters that enable scanning of read-only volumes or that restrict scanning to files opened with execute access:
    - `scan-ro-volume` enables scanning of read-only volumes
    - `scan-execute-access` restricts scanning to files opened with execute access. Note: “Execute access” is different from “execute permission.” A given client will have “execute access” on an executable file only if the file was opened with “execute intent.”
  - Within on-access mode you can choose from these two mutually-exclusive options
    - **Mandatory**: With this option, Vscan tries to deliver the scan request to the server until the timeout period expires. If the scan request is not accepted by the server, then the client access request is denied
    - **Non-Mandatory**: With this option, Vscan always allows client access, whether or not a Vscanner was available for virus scanning
- **On-demand task** - This component defines the scope of an on-demand scan
  - You can specify the maximum file size to scan, file extensions and paths to scan, and file extensions and paths to exclude from the scan. Files in subdirectories are scanned by default.
  - You use a cron schedule to specify when a task runs
- **Vscan file-operations profile (on-access scanning only)** - The vscan-fileop-profile parameter for the `vserver cifs share create` command defines which SMB file operations trigger virus scanning. By default, the parameter is set to **standard**. File operations are suspended until the external server reports the scan status of the file. If the file has already been scanned, ONTAP allows the file operation. Otherwise, it requests a scan from the server. You can adjust this parameter as necessary when you create or modify an SMB share:
  - **no-scan** specifies that virus scans are never triggered for the share
  - **standard** specifies that virus scans are triggered by open, close, and rename operations
  - **strict** specifies that virus scans are triggered by open, read, close, and rename operations
    - The strict profile provides enhanced security for situations in which multiple clients access a file simultaneously. If one client closes a file after writing a virus to it, and the same file remains open on a second client, strict ensures that a read operation on the second client triggers a scan before the file is closed
    - You should be careful to restrict the strict profile to shares containing files that you anticipate will be accessed simultaneously. Since this profile generates more scan requests, it may impact performance

- **writes-only** specifies that virus scans are triggered only when modified (or newly added) files are closed
  - Because writes-only generates fewer scan requests, it typically improves performance.
  - If you use this profile, the scanner must be configured to delete or quarantine unrepairable infected files, so they cannot be accessed. If, for example, a client closes a file after writing a virus to it, and the file is not repaired, deleted, or quarantined, any client that accesses the file without writing to it will be infected
- **On-access scanning is not supported for NFS**

Important information

MetaDefender Storage Security supports all of the parameters (**writes-only, standard**, **strict** or **no-scan**) file-operations profiles, but **writes-only** is the recommended configuration on the ONTAP side. With **writes-only**, only newly created or modified files are sent by ONTAP to MetaDefender Storage Security for scanning.

Because **writes-only** triggers the scan when the file is closed, and not when it is opened, two things follow:

- File access is not blocked while or after the file is scanned, even if **scan-mandatory** is enabled on the on-access policy.
- The scan request timeout configured from the ONTAP CLI (the **-request-timeout** parameter of the scanner pool, 30 seconds by default) **has no effect** on file access. That timeout defines how long the ONTAP node waits for a verdict from the Vscan server while it is still holding the client's pending file operation, and when it expires ONTAP allows or denies that operation according to scan-mandatory and the remaining lifetime of the client request. With **writes-only**, the scan is requested after the file has already been written and closed, so there is no pending client operation for the timeout to release or deny. The scan completes on the MetaDefender Storage Security side however long it takes, and the verdict is acted on by the configured remediation actions rather than by ONTAP access control.

Protection with **writes-only** therefore depends on post-scan remediation: configure the remediation actions you want (for example, quarantine or delete infected files) for the SVM in the MetaDefender Storage Security UI. If no remediation action is configured, a file found to be infected remains accessible on the share.

For more details about file-operations profile types, see t**his article** from NetApp.

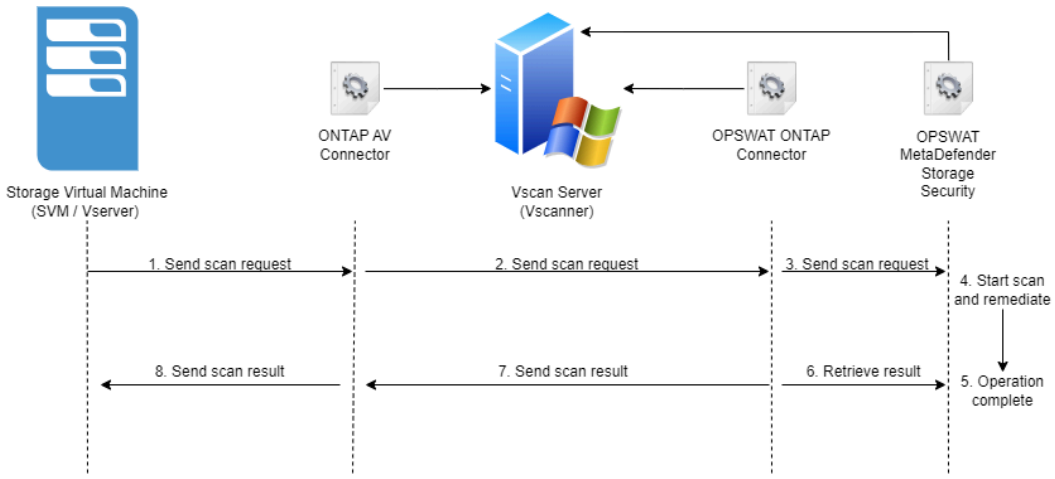
> On-Access scanning is not supported for NFS shares. If you want to protect NFS shares, please use our MDSS UI to do so.

How virus scanning works

Storage systems offload scanning operations to external servers hosting antivirus software from third-party vendors.

Based on the active scanning mode (Vscan file-operations profile), ONTAP sends scan requests when clients access files over SMB (on-access).

The ONTAP Antivirus Connector, provided by NetApp and installed on the external server, handles communication between the storage system and the antivirus software.



Step 2: Prerequisites

1. **Windows Server machine (minimum required version is 2019)**. This will be the Vscan Server where the following components will be installed:
    1. NetApp ONTAP Antivirus Connector
    2. OPSWAT ONTAP Antivirus Connector
2. Windows or Linux machine where MDSS is installed. **For best performances, we recommend installing MetaDefender Storage Security on the same machine from step 1**
3. One or multiple SVMs created from NetApp ONTAP that contain ore or multiple SMB shares.
4. An AD (recommended) user that will be logged in on the Vscan Server. This user will become the privileged user for your SVMs / SMB shares and will be used later to add the SVMs / SMB shares to MetaDefender Storage Security. To have a local user as the privileged user, please follow this NetApp article.

5. **Read the best practices recommended by NetApp team here.**

Step 3: Enable Vscan from ONTAP

Assuming you have already created an SVM / SMB share and a Vserver, please execute the below commands from ONTAP CLI.

Step 3.1: Create scanner pool

```
vserver vscan scanner-pool create -vserver {your-vserver} -scanner-pool
{scanner-pool-name} -hostnames {IP-of-Vscan-Server} -privileged-users
{AD-Privileged-User}
```

> **Attention**
>
> If MetaDefender Storage Security is **running on a different machine than Vscan server**, the IP address of that machine needs to be added to this scanner pool. Example below:

```
vserver vscan scanner-pool servers add -vserver {your-vserver} -
scanner-pool {scanner-pool-name} -hostnames {IP-of-MDSS-machine}
```

For more details about creating a scanner pool, please read this article.

Step 3.2: Apply policy to your scanner pool

```
vserver vscan scanner-pool apply-policy -vserver {your-vserver} -
scanner-pool {scanner-pool-name} -scanner-policy primary
```

For more details about applying a policy to your scanner pool, please read this article.

Step 3.3: Create on-access policy to receive scan requests

```
vserver vscan on-access-policy create -vserver {vserver-name} -policy-
name {policy-name} -protocol CIFS
```

For more details about creating an on-access-policy, please read this article.

Step 3.4: Modify file operations profile to writes_only(will trigger a file scanning, when the file will be Closed, only for newly created or modified files)

```
vserver cifs share modify -share-name {cifs-share-name} -vscan-fileop-
profile writes-only
```

For more details about modifying a CIFS share, please read this article.

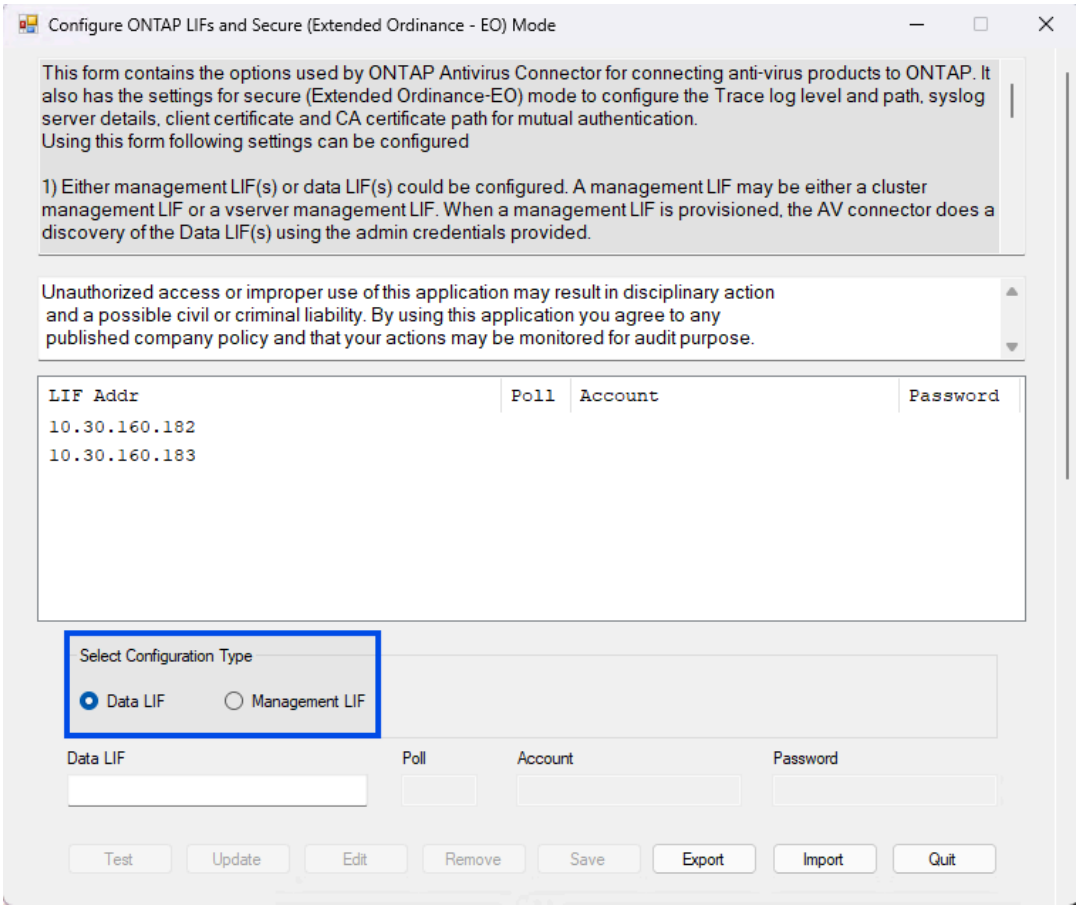
Step 3.5: Enable Vscan

```
vserver vscan enable -vserver {vserver-name}
```

Step 4: Install NetApp ONTAP AV Connector on the Vscan Server machine

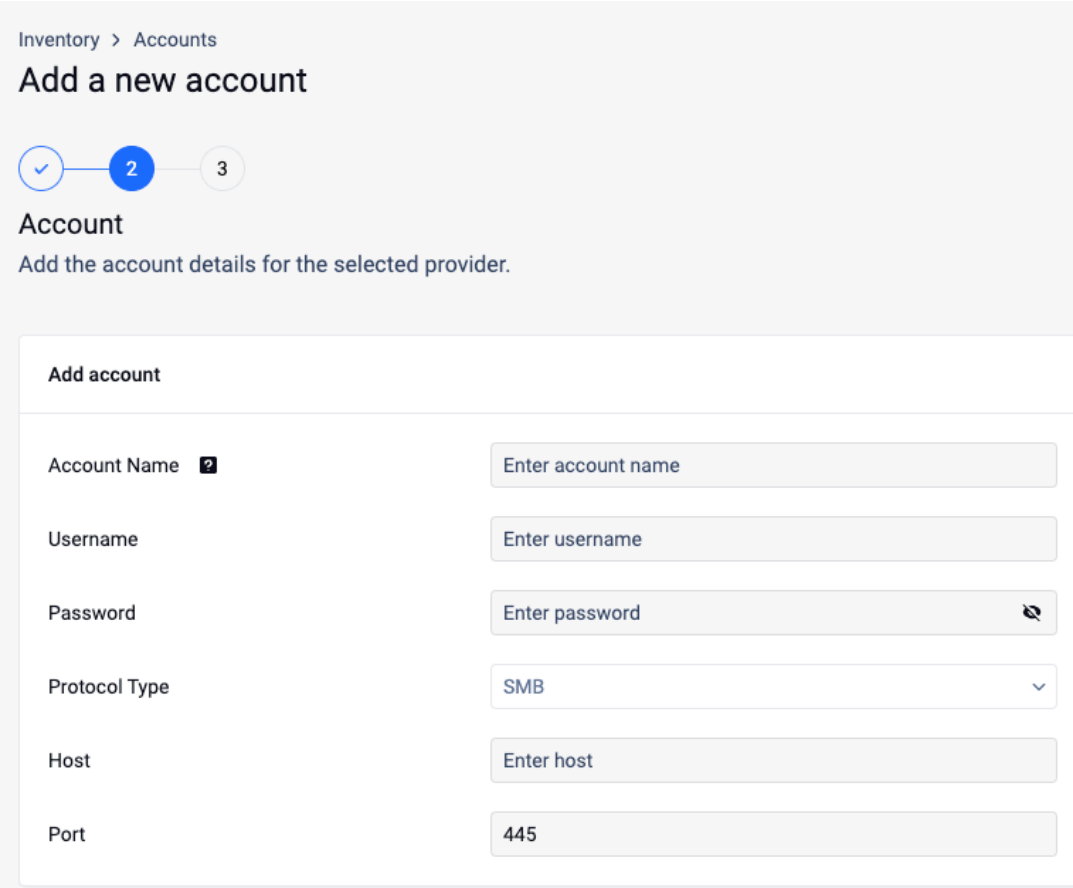
Please download the installer from here and follow the guidelines from NetApp team. The ONTAP AV Connector should be installed under the privileged user configured at step 3.1.

Both configuration types are supported from our side: **Data LIF** and **Management LIF**

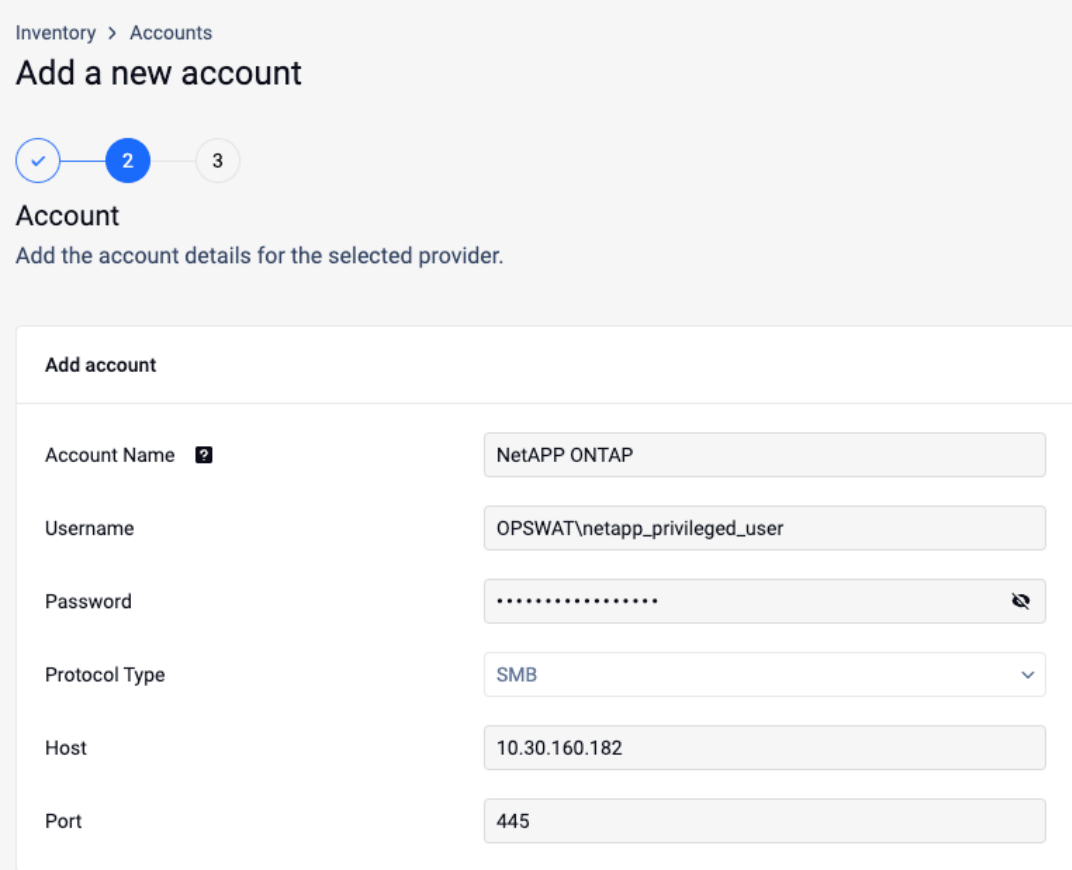


Step 5: Add the SVM in MetaDefender Storage Security

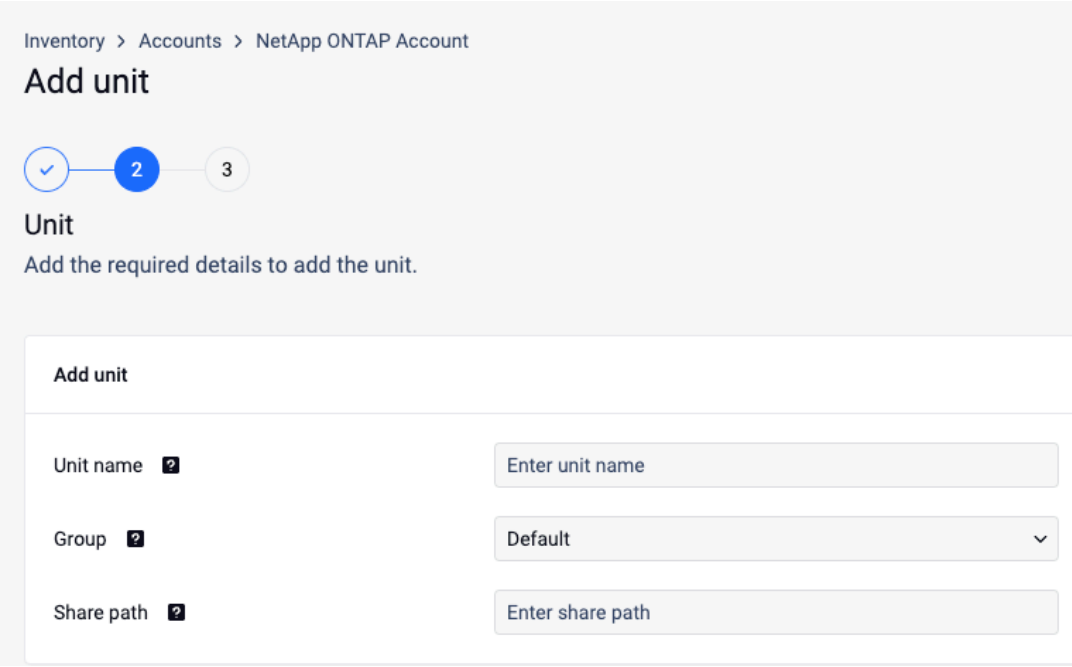
1. From Vscan server machine navigate to MDSS UI (localhost or remote machine)
2. Navigate to **Inventory**, click **Add New** > **Account**, choose **NetApp** and click **Next**



3. Give your account an unique name so you can easily identify it later
4. **Username**: Enter the privileged user configured at step 3.1. The following format is required: DOMAIN\Username
5. **Password**: Enter the password of the privileged user.
6. **Host**: Enter the IP address of the SVM
7. **Port**: Enter the port where the SMB server can be reached (default is 445)
8. Here is an example of how this should look like:

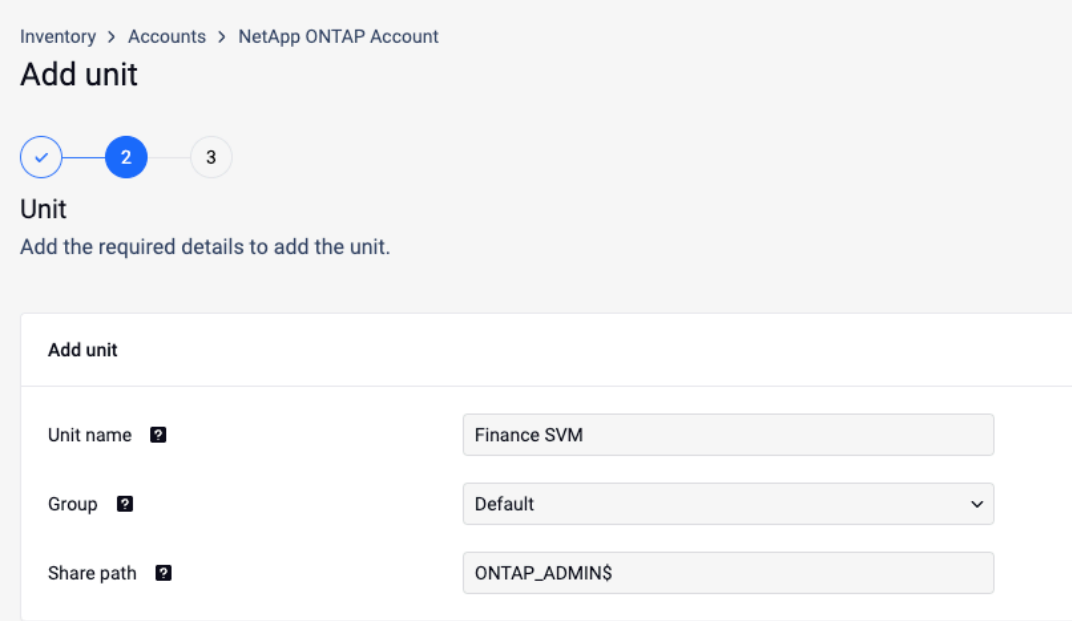


9. Click **Next** in order to finish the process of adding the account.
10. From the left side pane, navigate to **Inventory**, click **Add New** > **Unit**, choose the previously created account and click **Next**



11. Give your unit an unique name so you can easily identify it later.

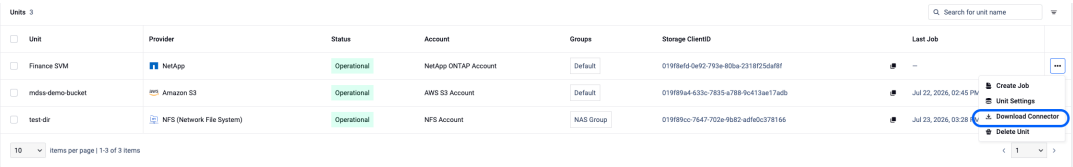
12. Assign your unit to a Group
13. **SharePath** should be **ONTAP_ADMIN$** to protect all volumes on the SVM. ONTAP_ADMIN$ is a privileged path created by ONTAP when Vscan is enabled. This path is available only for the privileged user and from the privileged machine configure on the SVM' scanner pool from ONTAP.
14. Here is an example of how adding the unit should look like



7. Click **Next** in order to finish the process.

Step 6: Install OPSWAT ONTAP Connector on the Vscan Server machine

1. From MetaDefender Storage Security, navigate to Inventory -> Units -> Click on the three dots from your NetApp ONTAP Storage -> Click **Download Connector**



2. Unzip the downloaded archive
3. Open a console with admin privileges at the same path where the archive was unzipped
4. Run the following command line instruction

```
.\OPSWAT-ONTAP-Connector.exe --install
```

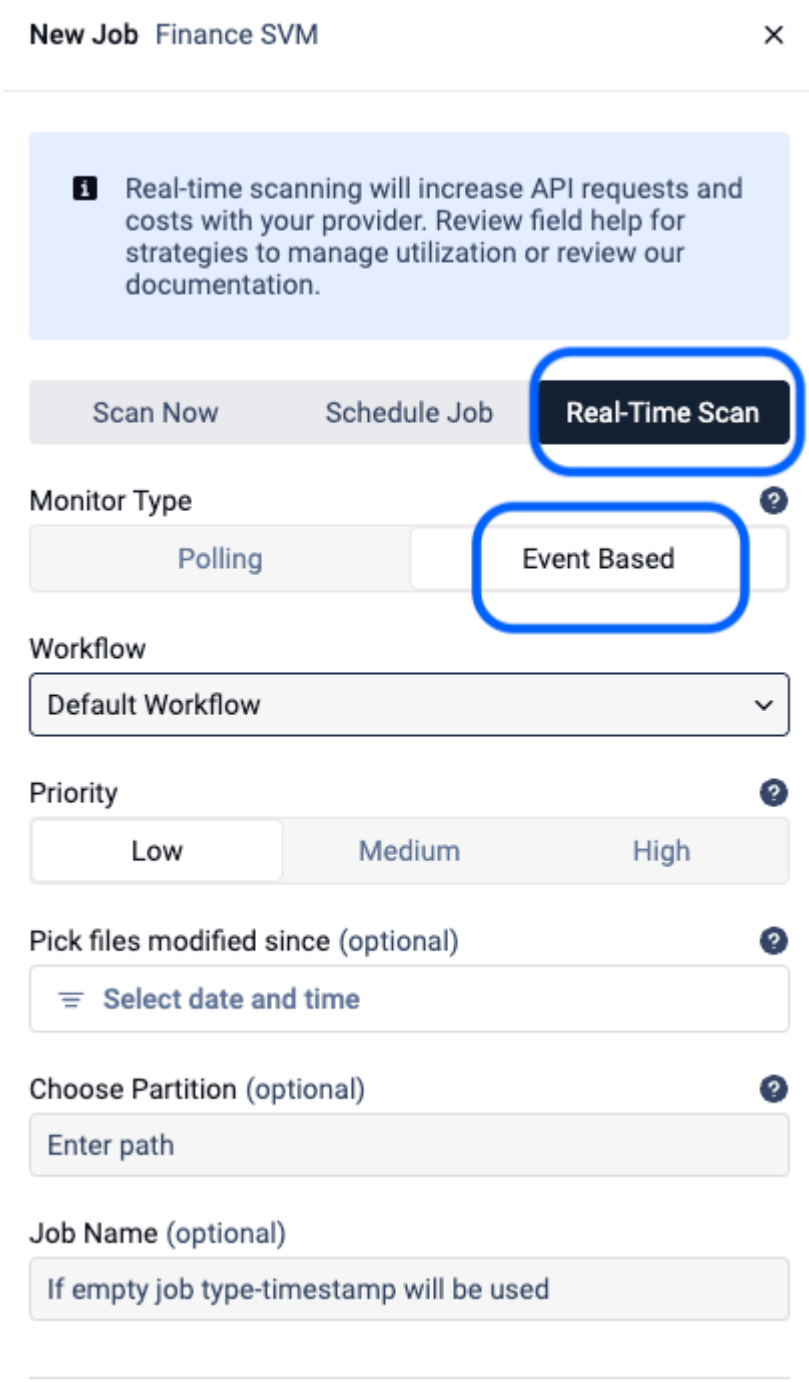
5. You will be asked if you want to install the OPSWAT ONTAP Connector under the Local System Account. In our testing, we have used this user and everything worked as expected. However, NetApp team recommends to install it under the same privileged user configured at step 3.1. To do that, close the current installation and run again with following parameters:

```
.\OPSWAT-ONTAP-Connector.exe --install /user DOMAIN\privilegeduser
/password your-password
```

6. A confirmation message should be received once the installation is complete.

Step 7: Test the integration

1. From MetaDefender Storage Security UI, navigate to Inventory -> Units -> Click on the three dots from the end of the row -> Create Job
2. Select Real-Time Scan and Event-Based and start the job



2. Copy a new file to one of the SVM volumes
3. From MetaDefender Storage Security UI, navigate to Dashboard -> Choose your NetApp ONTAP Storage -> Click on **View running jobs** -> Click on the previously started RTP scan
4. The newly uploaded file should be detected and scanned by MetaDefender Storage Security
5. Open the file and modify its content, then save it.
6. The modified file should be detected and scanned by MetaDefender Storage Security. If it's not the case, please see the below section named `Troubleshooting` .

> **Info**
>
> For different ways to enhance your scanning experience, please read this article.

Troubleshooting

Step 1 - Check if the file is received by NetApp ONTAP AV Connector

1. Enable verbose logging on ONTAP AV Connector by following the recommendations from this article.
2. Put a new file in the SMB share and then check the log file configured at previous step. If you see similar logs like this, it means the file is received by the ONTAP AV Connector and sent to our ONTAP Connector

```
2025-01-10T07:45:39.000564Z [pipe: 10.40.160.140]Server: Received 1152 bytes, ofsPartReq: [0]
2025-01-10T07:45:39.000564Z
[Pipe: 10.40.160.140]
       magic_num    : [4e74417041760003]
       session_id   : [cf99361b6bb7374d]
       len          : [1152]
       reqId        : [5]
       type         : [15, req_SCAN_EXTENDED]

[dump_mem] len: 66
 5c  0 6d  0 79  0 76  0   6f  0 6c  0 5c  0 75  0    [\ m y v o l \ u ]
 70  0 6c  0 6f  0 61  0   64  0 2d  0 62  0 79  0    [p l o a d - b y ]
 2d  0 67  0 61  0 62  0   72  0 69  0 65  0 6c  0    [- g a b r i e l ]
 2d  0 76  0 31  0 33  0   2e  0 74  0 78  0 74  0    [- v 1 3 . t x t ]
  0  0                                               [   ]
2025-01-10T07:45:39.000567Z Sending id 1 (rsrv-id: 0) for \\?\UNC\10.40.160.140\ontap_admin$\myvol\upload-by-gabriel-v13.txt
2025-01-10T07:45:39.000567Z Sent!
```

3. If the file is not received by NetApp ONTAP AV Connector, it means there is a misconfiguration in your side and needs to be fixed. You can start from section `Step 3: Enable Vscan from ONTAP`, located above in this page.

Step 2 - Check if the file is received by OPSWAT ONTAP Connector

1. Navigate to `C:\ProgramData\OPSWAT-ONTAP-Connector\logs`
2. Open the log file from today and search for your file name. You should see a message like this if your file was sent by ONTAP AV Connector and received by OPSWAT ONTAP Connector:

```
[{Time}] [Information] VS_ScanRequest received. Id: {sca-id}. File path
: \\?\UNC\{IP-of-smb-share}\ontap_admin$\{volume-name}\{file-name-and-
extension}
```

```
[2024-10-21 15:33:40] [Information] VS_ScanRequest received. Id: 3.
File path : \\?\UNC\160.160.160.160\ontap_admin$\myvolume\test-file.txt
```

3. If file is not received by our ONTAP Connector, there is a misconfiguration somewhere. Please check:
    1. privileged user from ONTAP to be the same as NetApp ONTAP user from storage unit from MDSS UI
    2. ONTAP AV Connector connected to management LIF that has the cluster IP address
4. You can also enable logging for ONTAP AV Connector by following this guideline

Step 3 - Check if the file is sent by OPSWAT ONTAP Connector to MetaDefender Storage Security

1. In the logs from step 1.2, search for your file name. You should see a log containing the following text: `File successfully sent for scanning`
2. If file is not sent to MDSS for scanning, please check the following:
    1. Real Time Processing with Event-Based to be enabled for your NetApp ONTAP storage unit
    2. `Apikey`and `StorageClientId` from config file `C:\ProgramData\OPSWAT-ONTAP-Connector\config.ini` to be the same as their values from MDSS UI.

Step 3 - Check if file is discovered by MetaDefender Storage Security

1. In MDSS UI, navigate to `Dashboard -> NetApp ONTAP -> Choose your storage unit -> Real-Time Processing [Event Based] -> View Report`
2. Check if the file appears in the table
3. If not, something happened with our SMB service. You can check the logs from `C:\Program Files\OPSWAT\MetaDefender Storage Security\services\opswat.mdcs.smbservice\logs\600x` to see if you spot anything. Please check in each of the 600x folders. If you need guidance from us, please generate a support package and send it to our support team.

Uninstall

To uninstall OPSWAT-ONTAP-Connector, run the following command line in an admin console

```
.\OPSWAT-ONTAP-Connector.exe --uninstall
```

A confirmation message will be received.

If you run the command with the executable from `C:\ProgramData\OPSWAT-ONTAP-Connector\OPSWAT-ONTAP-Connector.exe`, the service will be uninstalled but the

executable file will not be deleted because is in use. Finish the uninstallation process and then remove the folder manually.

Vscan | Best practices

Privileged User

- Restrict privileged users to virus-scanning operations. Normal users should be discouraged from using privileged user credentials. This restriction can be achieved by turning off login rights for privileged users on Active Directory.
- Privileged users are not required to be part of any user group that has a large number of rights in the domain, such as the administrators group or the backup operators group
- Privileged users must be validated only by the storage system so that they are allowed to create Vscanner connections and access files for virus scanning.

Vscanner

- Use the computers running Vscanners only for virus-scanning purposes. To discourage general use, disable the Windows terminal services and other remote access provisions on these machines and grant the right to install new software on these machines only to administrators.
- Dedicate Vscanners to virus scanning and do not use them for other operations, such as backups. You might decide to run the Vscanner as a virtual machine (VM). If this is the case, ensure that the resources allocated to the VM are not shared and are sufficient to perform virus scanning.
- Provide adequate CPU, memory, and disk capacity to the Vscanner to avoid resource bottlenecks. Most Vscanners are designed to use multiple CPU core servers and to distribute the load across the CPUs.
- For remote sites and branch offices, NetApp recommends using a local Vscanner rather than a remote Vscanner because the former is a perfect candidate for high latency. If cost is a factor, use a laptop or PC (as a Vscanner) for moderate virus protection.

Vscanner Networking

- Connect the NetApp storage system and the Vscanner by using at least a 1GbE network.
- For an environment with multiple Vscanners, connect all servers that have similar high-performing network connections. Connecting the Vscanners improves performance by allowing load sharing.

- NetApp recommends using a dedicated network with a private VLAN for the connection from the SVM to the Vscanner so that the scan traffic is not affected by other client network traffic. Create a separate NIC that is dedicated to the antivirus VLAN on the Vscanner and to the data LIF on the SVM. This step simplifies administration and troubleshooting if network issues arise.
- The AV traffic should be segregated using a private network. The AV server should be configured to communicate with domain controller (DC) and clustered Data ONTAP in one of the following ways:
  - The DC should communicate to the AV servers through the private network that is used to segregate the traffic.
  - The DC and AV server should communicate through a different network (not the private network mentioned previously), which is not the same as the CIFS client network. Private Network is optional.
- For Kerberos authentication to work for the AV communication, create a DNS entry for the private LIFs and a service principal name on the DC corresponding to the DNS entry created for the private LIF. Use this name when adding a LIF to the Antivirus Connector. The DNS should be able to return a unique name for each private LIF.

Node Failover

If the LIF for Vscan traffic is configured on a different port than the LIF for client traffic, the Vscan LIF might fail over to another node in case of a port failure. The change will make the Vscanner not reachable from the new node and the scan notifications for file operations on the node will fail.

> **Info**
>
> For more details about this type of storage please read the official ONTAP Documentation

Vscan | Scalability

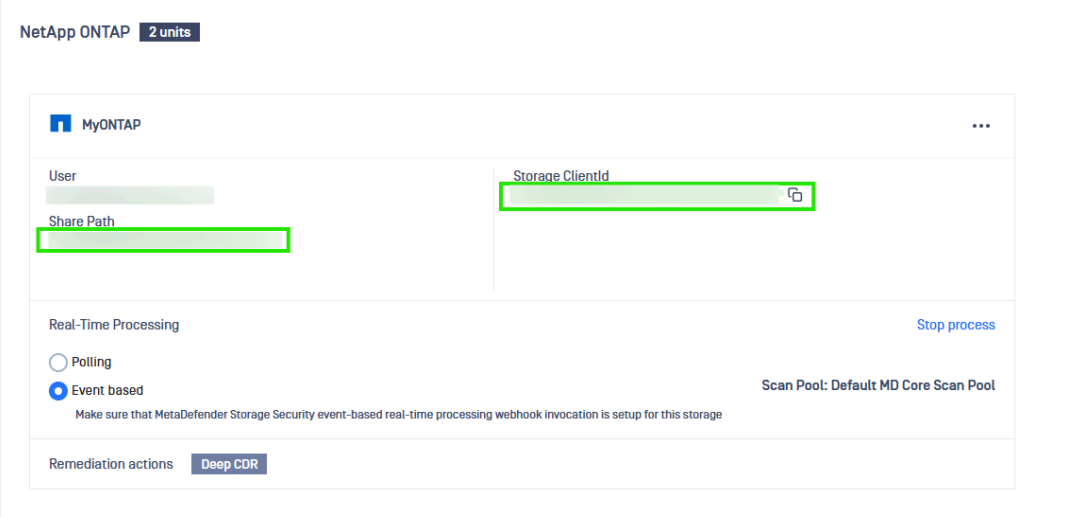
Our recommendation is to start with the basic use case, monitor the performance and, depending your business needs, scale the system accordingly.

Description	Use case	Solution
Small scale deployments with one SMB share	0 - 20,000 files per hour* 1 x SMB share	1 x Vscan Server
Small scale deployments with multiple SMB shares	0 - 20,000 files per hour* 2 or more SMB shares	1. Connect multiple SMB shares to the same OPSWAT-ONTAP-Connector 2. Connect each SMB share to a separate MDSS instance. You will need a Vscan Server for each MDSS instance
Medium to Large scale deployments	20,000 + files per hour *	1. Configure multiple Vscan Servers to support your SMB share. 2. Split your data in multiple SMB shares each one being connected to a separate Vscan Server having separate MDSS instances.

*** The data presented in the table was derived from our sizing guide. These results are intended as guidelines rather than precise performance guarantees due to the multitude of variables that can impact performance, such as file types, network configurations, and hardware specifications. For critical throughput requirements, OPSWAT recommends conducting site-specific benchmarks prior to deploying a production solution.

How to connect multiple SMB shares to the same OPSWAT-ONTAP-Connector

1. Add your SMB share to MetaDefender Storage Security following this article
2. Copy your `StorageClientId`and `SharePath`for your new storage



3. Navigate to `C:\ProgramData\OPSWAT-ONTAP-Connector\config.ini`
4. Open the file in text editor (requires admin privileges) and modify the number of storage units: `StorageUnits=2`
5. Add details about the second storage unit:

```
[Storage Unit 2]
ClientId={StorageClientId}
SharePath={SharePath}
```

6. Restart the service with name`opswat-ontap-connector` in order changes to take effect

How to configure multiple Vscan Servers

1. You need to have a new Windows Server machine and another AD user (privileged user). For each scan server you will need to have a separate Windows Server machine because **only one ONTAP Connector can be installed on the same Windows Server machine**
2. From ONTAP CLI, add the new IP of the Vscan Server and the new privileged user to scanner pool

```
vserver vscan scanner-pool modify -vserver {your-vserver} -scanner-pool
{scanner-pool-name} -hostnames {list-of-host-names} -privileged-users
{list-of-privileged-users}
```

```
vserver vscan scanner-pool modify -vserver opswat-vserver -scanner-pool
opswat-scanner-pool -hostnames 10.30.160.180,10.30.160.181 -privileged-
users TestAD\privilegeduser1,TESTAD\privilegeduser2
```

For more information about `vserver vscan scanner-pool modify` command, please read this article

3. Install ONTAP AV Connector and OPSWAT ONTAP Connector on the new Windows Server machine. Follow this guidance. To install OPSWAT ONTAP Connector, please copy the zip from first Vscan Server and install from there.
4. Navigate to `C:\ProgramData\OPSWAT-ONTAP-Connector\config.ini` , open it in a text editor with admin privileges
5. Modify the IP of the MetaDefender Storage Security from `127.0.0.1` with the IP of the first Vscan Server, where your MDSS instance is installed (**Note**: Please be aware that the IP displayed or assigned may differ if a proxy or VPN is being used. To ensure accurate IP information, verify if any proxy or VPN is active)
6. Restart service with name `opswat-ontap-connector` in order changes to take effect
7. Now you should have two scanning servers connected to the same SMB share and same MDSS instance.
8. If you want to add more scan servers, please repeat the process.

> **Info**

> When multiple scan servers are utilized, files are distributed to the scan servers using a round-robin method.

AWS FSx for NetApp ONTAP

Metadefender Storage Security is fully compatible with AWS FSx for NetApp ONTAP, leveraging NetApp's Vscan feature for seamless integration. This allows MDSS to scan files stored on your FSxN volumes for threats, helping to ensure the integrity and security of your data.

Before you begin, make sure you have:

- An operational AWS FSx for NetApp ONTAP file system.
- A Windows Server machine designated to run Metadefender Storage Security.
- Appropriate licenses for both AWS FSx for NetApp ONTAP and Metadefender Storage Security.
- Administrative access to both your AWS environment and your NetApp ONTAP configuration.

Setup and Configuration

Follow these steps to integrate MDSS with your FSxN instance:

1. Set Up AWS FSx for NetApp ONTAP (if not already deployed)
    - Follow the official AWS documentation to create and configure your FSx for NetApp ONTAP file system:
        - AWS FSx for ONTAP Setup Guide: Amazon FSx for NetApp ONTAP User Guide
2. Configure Vscan for MDSS Integration
    - Vscan enables communication between your NetApp ONTAP system and external virus scanning servers, such as your MDSS instance.
    - Detailed instructions for enabling and configuring Vscan can be found in the NetApp and AWS documentation. Refer to the following resources, starting with **step 3** of the OPSWAT Vscan guide which typically outlines the ONTAP-side configuration:
        - OPSWAT Vscan Integration Guide (refer to ONTAP configuration steps): MetaDefender Storage Security - Add A NetApp ONTAP Storage (This page also contains instructions for adding NetApp ONTAP as a storage unit in MDSS).
        - AWS Guide for Vscan with FSx for ONTAP: Use NetApp ONTAP Vscan with FSx for ONTAP
        - NetApp ONTAP Vscan Enablement (CLI reference): vserver vscan enable - NetApp ONTAP CLI

3. Domain Membership - CRITICAL for proper authentication and Vscan communication
    - The Windows Server machine running Metadefender Storage Security **must** be a member of the same Active Directory domain as the AWS FSx for NetApp ONTAP SVM (Storage Virtual Machine) that you intend to scan.
4. Install and Configure Metadefender Storage Security
    - Install MDSS on your designated Windows Server.
    - During the MDSS configuration, add your FSxN SVM as a NetApp ONTAP storage unit, providing the necessary credentials and connection details as prompted by the MDSS interface. Refer to the MetaDefender Storage Security documentation for specific instructions on adding storage units.

Troubleshooting

If you encounter issues during or after the integration, use the following NetApp ONTAP CLI commands on your FSxN instance for diagnostics:

- `event log show` this command displays the event log, which may contain error messages related to Vscan or authentication. Filter for Vscan-related events for more targeted information.
    - Example: `event log show -message-name *vscan*`
- `vserver vscan connection-status show-all` this command shows the status of Vscan connections from the SVM to the scanning servers (your MDSS instances). This helps verify if the MDSS server is successfully connected.
- `vserver vscan show` this command displays the Vscan configuration for a specified SVM, allowing you to verify settings like active scanners and policies.

Additional Troubleshooting Tips

- Make sure there are no firewall rules (either on the MDSS server, AWS Security Groups, or network ACLs) blocking the necessary Vscan communication ports between the FSxN SVM and the MDSS server. (Default Vscan port is 139/445 for SMB).
- Verify that the service account used by MDSS for Vscan has the appropriate permissions on the NetApp SVM and the shares being scanned.
- Confirm that the MDSS server and the FSxN SVM can resolve each other's hostnames via DNS.
- Make sure that the MDSS server and the FSxN instance (and domain controllers) have their time synchronized. Significant time skew can cause authentication issues.
- Check the Metadefender Storage Security logs on the Windows server for any specific error messages related to the NetApp integration.

Performance Testing

The validation was performed using the basic deployment infrastructure model:

- **Windows Server**: 1 x MDSS instance and 1 x OPSWAT ONTAP Connector instance
- **Linux**: 1 MetaDefender Core instance

The performance metrics obtained from this configuration aligned with the specifications outlined in our sizing guide. The test successfully validated the expected performance parameters for this particular deployment architecture.

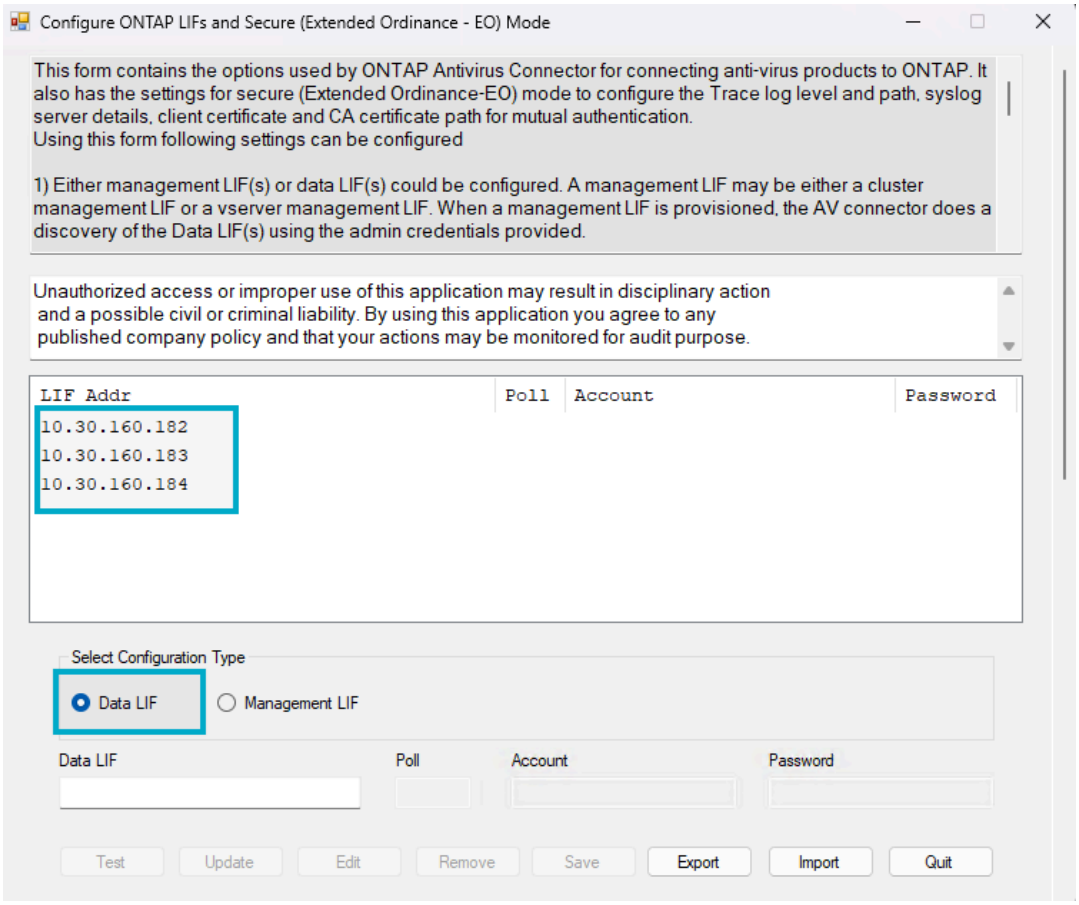
> **Note:** For detailed performance metrics and specific hardware recommendations, please refer to the official sizing guide.

Vscan | How to connect multiple SVMs

When Vscan is enabled, **multiple SVMs** can be connected to the same MDSS instance.

Step 1: Add the SVM on ONTAP AV Connector

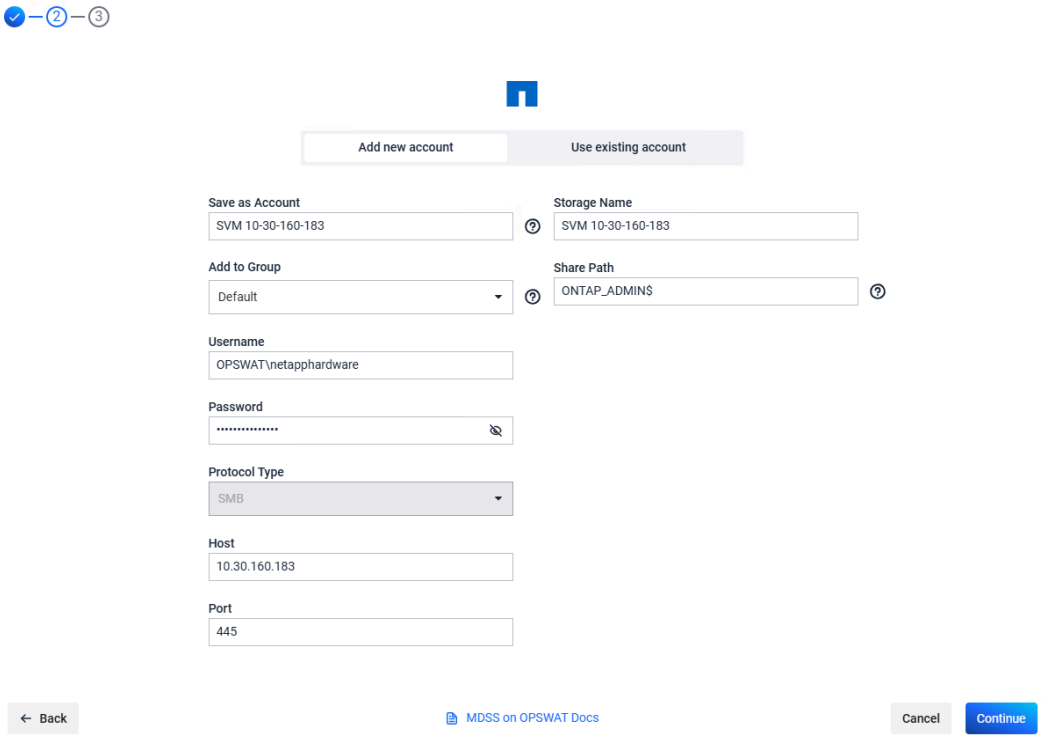
1. From MDSS machine, open the **Configure ONTAP LIFs** app provided by NetApp and available with the ONTAP AV Connector installation.
2. Add the additional SVMs (can be either data LIF or management LIF). In our example, we have used Data LIF as does not require authentication



3. Once all the SVMs are added, save and quit.

Step 2: Add each SVM as a different storage unit in MDSS

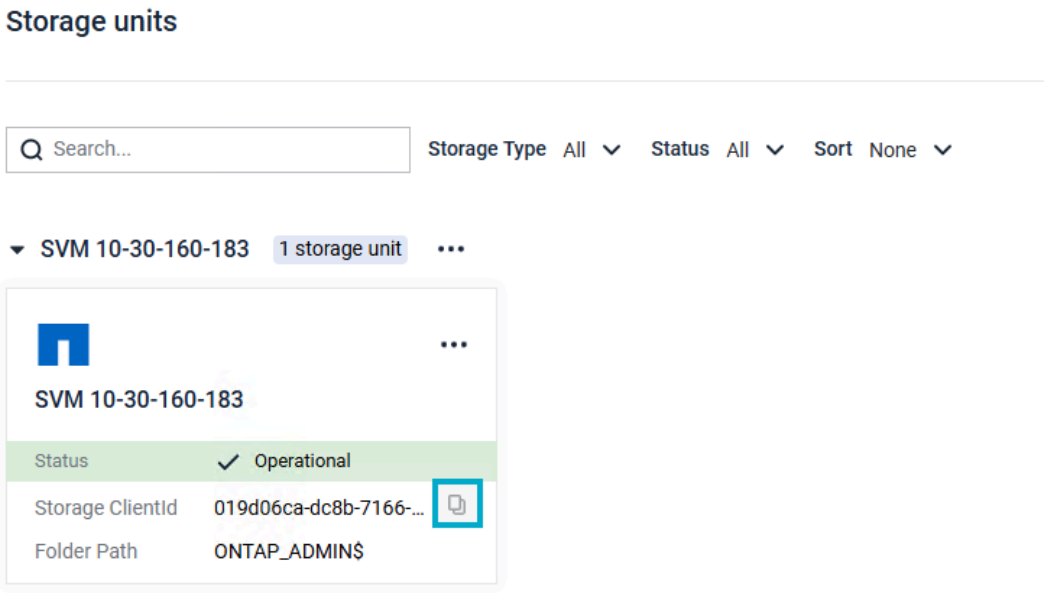
1. Navigate to MetaDefender Storage Security UI
2. Go to **Inventory** > **Add New** > **Account** and choose **NetApp**
3. For the Account, input:
    1. name
    2. username and password of the privileged user configured on ONTAP
    3. Host (IP) of the SVM
4. For the Storage, input:
    1. name
    2. Share Path. In case you want to add all the SMB shares created under the SVM, input **ONTAP_ADMIN$** (privileged path created by ONTAP) that is available only for the privileged user.
5. Here is an example of how the form should look like



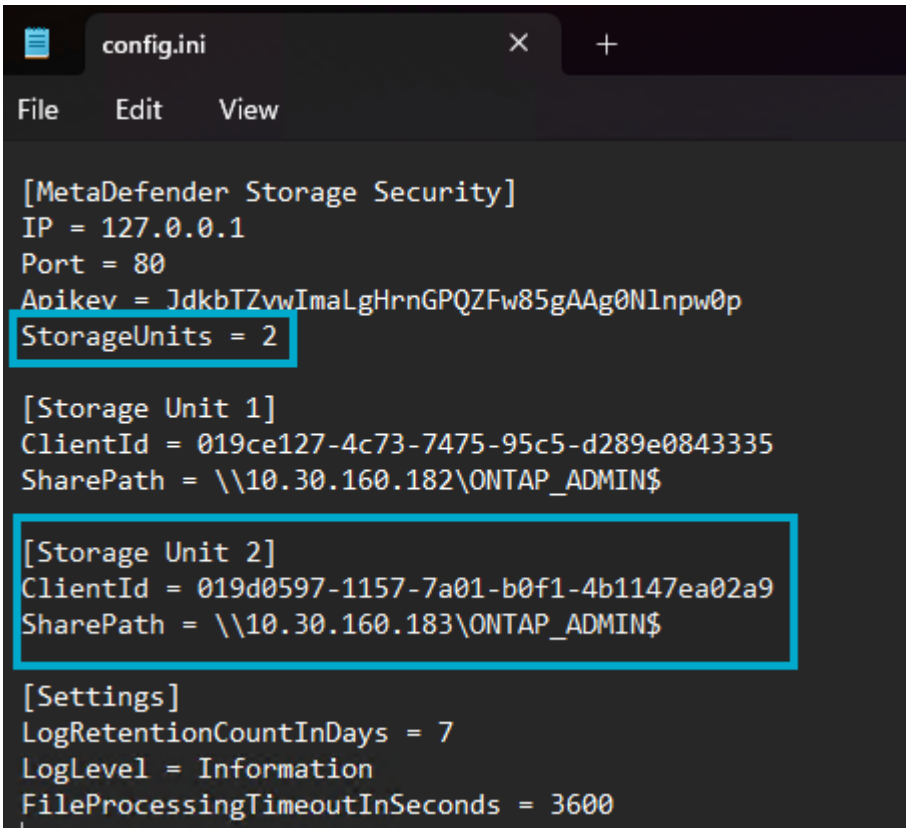
6. Click Continue and wait for the storage to be added.
7. If **OPSWAT ONTAP Connector** is installed, do not download or install it.
8. Repeat the steps with all the SVMs you want to add

Step 3: Add each SVM in the OPSWAT ONTAP Connector config file

1. From MDSS machine, navigate to **C:\ProgramData\OPSWAT-ONTAP-Connector**
2. Open **config.ini** file
3. Set **Storage Units** to the number of SVMs you have added in MDSS
4. Add each Storage Unit and its corresponding details.
    1. **ClientId** can be copied from MDSS UI by Navigating to Storage Units -> Your storage unit card and click the copy button

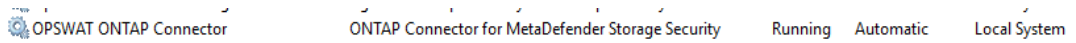


5. Here is an example of how the config file should look like:



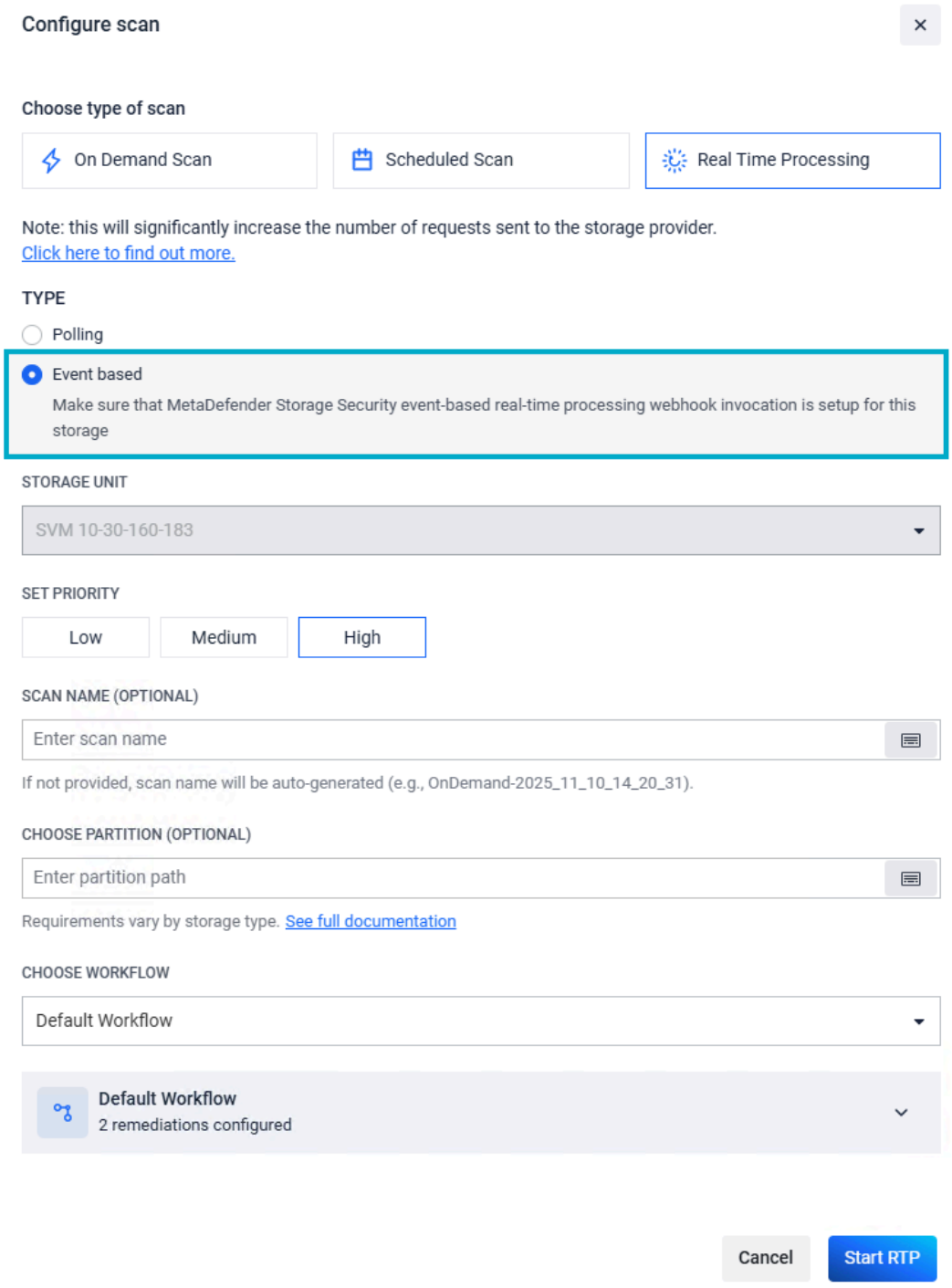
Step 4: Restart OPSWAT ONTAP Connector service

This service needs to be restarted for the changes to take effect



Step 5: Enable RTP with Event-Based

One last step is to enable Real Time scan with Event-Based for all the storage units corresponding to your SVMs, so that MDSS can processed the new files



Vscan | Configuration

The OPSWAT connector for NetApp ONTAP is configured through a **config.ini** file. This page documents the settings available in that file, grouped by section. **IP**, **Port**, **Apikey**, **StorageUnits** and the keys of every **[Storage Unit N]** section are required: if one of them is missing, empty or invalid, the connector does not start and reports the setting. Every other setting is optional: if it is left empty or omitted, or its value is outside the allowed range or of the wrong type, the connector uses the default value and logs a warning.

For the full behaviour of every key, including validation messages and troubleshooting, see ONTAP Connector Configuration file.

Example

The template the connector package is built from, with every optional key at its default. A connector downloaded from MetaDefender Storage Security has **Apikey**, **Port** and **PortSsl** filled in, **StorageUnits=1**, and a **[Storage Unit 1]** section for the storage it was downloaded for.

```
[MetaDefender Storage Security]
IP=127.0.0.1
Port=80
PortSsl=443
IgnoreCertErrors=0
Apikey=
StorageUnits=0

[Settings]
LogRetentionCountInFiles=7
LogMaxFileSizeInMegabytes=10
LogPath=
LogLevel=Information
FileProcessingTimeoutInSeconds=3600
RpcLocalPipeHost=127.0.0.1
ZoneIdentifierFiles=0
TimeoutResponse=4
StorageSyncIntervalInSeconds=15
```

[MetaDefender Storage Security] section

Defines how the connector reaches the MetaDefender Storage Security instance.

Setting	Type	Default	Description
IP	Text (IPv4 address)		Required. IP address of the MetaDefender Storage Security instance. Hostnames are not accepted. The template uses `127.0.0.1`.
Port	Integer (1-65535)		Required. HTTP port of the MetaDefender Storage Security instance. The template uses `80`, but there is no default: an empty or invalid value stops the connector.
PortSsl	Integer (1-65535)	443	HTTPS port of the MetaDefender Storage Security instance. The connector switches between **Port** and **PortSsl** automatically, and the switching cannot be turned off. A value outside the range, including `0`, is replaced with 443.
IgnoreCertErrors	Integer (0-1)	0	1 ignores TLS certificate errors on HTTPS requests, for example with the default self-signed MDSS certificate.
Apikey	Text		Required. API key used to authenticate with MetaDefender Storage Security. It also needs read permission on accounts for the storage synchronisation; without it, every synchronisation cycle is skipped.
StorageUnits	Integer (0 or greater)		Required. Number of **[Storage Unit N]** sections. Maintained by the connector; `0` means no NetApp ONTAP storage currently has event-based real-time protection running. The template uses `0`; a connector downloaded from MetaDefender Storage Security starts at 1.

[Storage Unit N] sections

One section per configured storage unit, numbered sequentially (**[Storage Unit 1]**, **[Storage Unit 2]**, and so on). The number of these sections must match the **StorageUnits** value above. If **StorageUnits** is less than the number of **Storage Unit** sections in the file, only the first N sections are read and the rest are ignored. If **StorageUnits** is greater, the connector looks for a section that does not exist and fails to start. The same happens when a section in the sequence is missing (e.g. **StorageUnits=3**, **[Storage Unit 1]** and **[Storage Unit 3]** are present but **[Storage Unit 2]** is missing). Both **ClientId** and **SharePath** are required in every section that is read.

Every **StorageSyncIntervalInSeconds** the connector reads the NetApp ONTAP accounts from MetaDefender Storage Security, keeps the storages whose event-based real-time protection is running, and rewrites the **[Storage Unit N]** sections, the **StorageUnits** count and the ONTAP AV Connector `data-lifs` registry values to match. Changes apply without

a service restart. Set **StorageSyncIntervalInSeconds=0** to keep hand-written sections untouched.

Setting	Type	Default	Description
ClientId	Text		Identifier of the storage unit generated by MetaDefender Storage Security
SharePath	Text		Path of the share to protect as it is integrated in MetaDefender Storage Security

[Settings] section

General connector behavior.

Setting	Type	Default	Description
LogRetentionCountInFiles	Integer (1-200)	7	Number of log files kept; older files are deleted. Replaces **LogRetentionCountInDays**, which is still read when the new key is absent.
LogMaxFileSizeInMegabytes	Integer (1-1024)	10	Size at which the current log file is rolled over to a new timestamped file (`YYYYMMDD_HHMMSS.log`). An identical warning or error is written once per log file.
LogPath	Text (absolute or relative path)		Directory the log files are written to. Empty keeps the default `<install dir>\logs`. A relative path is resolved against the installation directory. An unusable path falls back to the default and the reason is logged.
LogLevel	Text	Information	Logging verbosity. One of: Verbose, Debug, Information, Warning, Error, Fatal. The value is case-sensitive.
FileProcessingTimeoutInSeconds	Integer (1 or greater)	3600	Maximum time allowed to process a single file before it times out
RpcLocalPipeHost	Text	127.0.0.1	Local host address used for the connector's internal communication
ZoneIdentifierFiles	Integer (0-3)	0	Handling mode for Windows *Zone.Identifier* files (see below).
TimeoutResponse	Integer (0-5)	4	Verdict sent to ONTAP when a file hits **FileProcessingTimeoutInSeconds**. 4 blocks only where scanning is mandatory.
StorageSyncIntervalInSeconds	Integer (0-86400)	15	How often the storage units are synchronised with MetaDefender Storage Security. `0` disables the synchronisation and leaves `config.ini` and the registry as configured by hand.

ZoneIdentifierFiles values

Zone.Identifier files are small metadata streams Windows attaches to files downloaded from the internet. They hold no user content, so scanning them is usually unnecessary. This setting selects how the connector handles them; all other files always go through the normal scanning pipeline regardless of this value.

Value	Mode	Behavior
0	Skip (default)	Zone.Identifier files are ignored, not scanned and no verdict is returned
1	Scan	Zone.Identifier files are sent to MetaDefender Storage Security and scanned like any other file
2	Allow	Zone.Identifier files are immediately allowed without scanning
3	Block	Zone.Identifier files are immediately blocked without scanning

ONTAP Connector Configuration file

The OPSWAT ONTAP Connector is the Windows service (`opswat-ontap-connector`) that sits between a NetApp ONTAP Vscan (AV) connector and a MetaDefender Storage Security (MDSS) instance. It receives scan requests from ONTAP over MSRPC named pipes, forwards each file to MDSS for real-time processing, and returns the verdict back to ONTAP.

All of its behaviour is driven by a single `config.ini` file. This page documents every section, key, accepted value and default.

File location and load behaviour

File name	`config.ini`
Location	The folder containing `OPSWAT-ONTAP-Connector.exe`
Installed location	`C:\ProgramData\OPSWAT-ONTAP-Connector\config.ini`
When it is read	Once, at process start, before any other initialization
Applying a change	**Restart the opswat-ontap-connector service.** The file is never re-read at runtime. The only exception is the `[Storage Unit N]` sections and the `StorageUnits` count, which the connector itself rewrites while running (see `StorageSyncIntervalInSeconds`).

`config.ini` is copied from the executable folder into the installation folder by `--install` and `--upgrade`, so edit the *installed* copy when tuning a deployed connector.

Only a few values can stop the connector from starting: `IP`, `Port`, `Apikey`, `StorageUnits`, and the keys inside the `[Storage Unit N]` sections. If the file cannot be read, or one of these values is missing, empty or fails validation, the connector logs `Error when reading config.ini file` at `Fatal` level and exits without starting. Every other key falls back to its default when its value cannot be used, and the connector logs a warning and starts.

Format rules

Standard INI semantics apply:

- Sections are declared as `[Section Name]`; section and key names are matched case-insensitively by Windows.
- Entries are `Key=Value`. Whitespace around the `=` is trimmed.
- Lines beginning with `;` are comments.
- Do **not** quote values — quotes become part of the value.
- A single value is limited to 254 characters; anything longer is truncated.
- A `[Settings]` key the connector does not recognise is ignored silently. A misspelled key therefore shows up as the setting having no effect, not as a startup error — check the effective values in the startup log line described under What the connector logs at startup.

Section `[MetaDefender Storage Security]`

Identifies the MDSS instance the connector talks to.

Key	Required	Type	Default	Accepted values
`IP`	Yes	IPv4 address	—	Dotted-quad IPv4 only, e.g. `10.0.0.15`
`Port`	Yes	Integer	—	MDSS HTTP port, 1 – 65535, e.g. `80`, `8008`
`PortSsl`	No	Integer	`443`	MDSS HTTPS port, 1 – 65535
`IgnoreCertErrors`	No	Integer	`0`	`0` or 1
`Apikey`	Yes	String	—	MDSS API key
`StorageUnits`	Yes	Integer	—	`0` or greater

`IP`

The address of the MDSS instance. **Validated against an IPv4 regular expression — hostnames. FQDNs and IPv6 addresses are rejected** and cause a fatal configuration error at startup.

`Port` and `PortSsl`

`Port` has no default: an empty or non-numeric value, or a value outside `1` – `65535`, is a fatal configuration error. The `80` in the template is a starting value, not a default. `PortSsl` falls back to `443` when its value is non-numeric or outside `1` – `65535`, and the connector logs a warning.

The connector does not require you to declare whether MDSS is served over HTTP or HTTPS. It starts every session on plain HTTP against `Port`, and switches transport automatically:

1. A request fails with a transport-level WinHTTP error: connection reset (`12030`), an invalid server response (`12152`), the port refusing connections (`12029`), or, on a plain-HTTP request only, a TLS handshake failure (`12175`).
2. The connector flips its transport mode (HTTP ⟷ HTTPS) and retries the request once on the other port. The switch is logged at `Warning` level (`MDSS request failed with a transport error. Switched transport to …`).
3. If the retry succeeds, the new mode is kept for subsequent requests. If the retry fails as well, MDSS is treated as unreachable rather than moved, and the previous mode is restored (`The retry failed as well, so MDSS is unreachable rather than moved. Transport reverted to …`).

The switch is bidirectional and self-healing, so an MDSS instance that moves from HTTP to HTTPS (or back) is picked up without restarting the connector. Automatic switching is always on. A `PortSsl` value of `0` does not turn it off: `0` is outside the valid range, so the connector uses `443`.

A certificate failure on an HTTPS request never makes the connector switch to plain HTTP, because the retry would send the API key and the file content unencrypted. Install a certificate the connector's host trusts, or see `IgnoreCertErrors`.

The connector negotiates TLS 1.2 and, where the build environment supports it, TLS 1.3.

IgnoreCertErrors

Value	Behaviour
`0` (default)	Full TLS certificate validation. A certificate MDSS cannot prove is trusted fails the request.
`1`	Certificate errors are ignored on HTTPS requests: unknown CA, wrong certificate usage, host-name mismatch, and expired/not-yet-valid dates.

Set this to `1` when MDSS is serving its default self-signed certificate. Any other value (for example `2`) falls back to `0`, full validation, and the connector logs a warning. The setting has no effect on plain HTTP requests.

> Security note: `IgnoreCertErrors=1` disables certificate validation entirely and leaves the channel open to an on-path attacker. Use it only on trusted management networks, or install a certificate the connector's host trusts.

Apikey

The MDSS API key, sent as the `Apikey` HTTP header on every call the connector makes (`POST /api/webhook/realtime`, `GET /api/file/{correlationId}`, `GET /api/health`, `GET /api/account`).

This will be automatically added when you download and install the OPSWAT ONTAP Connector from MDSS. `Apikey` is required: an empty value stops the connector from starting.

The storage synchronisation reads the NetApp ONTAP accounts through `GET /api/account`, so the key must have read permission on accounts:

- Without that permission, MDSS refuses the request and every synchronisation cycle is skipped (`StorageSyncMonitor: could not enumerate storages from MDSS, skipping this cycle`). Nothing is added or removed.
- With that permission, MDSS leaves out of its answer every storage the key has no access to. Such a storage is not added, and a storage unit already written for it is removed after three cycles.

StorageUnits

The number of `[Storage Unit N]` sections that follow. Must be an integer of `0` or greater; a negative number or a non-numeric value is a fatal configuration error.

`StorageUnits=0` is a valid state: it means no NetApp ONTAP storage currently has event-based real-time protection running. A connector downloaded from MDSS starts at `1`, with a `[Storage Unit 1]` section for the storage it was downloaded for. The connector then raises or lowers the count itself as storages are synchronised from MDSS.

The connector reads exactly the sections `[Storage Unit 1]` … `[Storage Unit <StorageUnits>]`. **Numbering must be consecutive and start at 1.** A gap in the numbering, or a count higher than the number of sections, makes the connector look for a section that does not exist, and the connector does not start. Sections beyond the declared count are ignored.

Section `[Storage Unit N]`

One section per storage unit, numbered `1` to `StorageUnits`. Both keys are required in every section, and an empty value stops the connector from starting.

Key	Required	Type	Default	Description
`ClientId`	Yes	String	—	The MDSS storage client id
`SharePath`	Yes	String	—	ONTAP-side path prefix for this storage unit

Storage units are maintained automatically

The connector keeps the configuration file in step with MDSS: every `StorageSyncIntervalInSeconds` it reads the NetApp ONTAP accounts and their storages from MDSS, keeps the storages whose event-based real-time protection is running, and rewrites the `[Storage Unit N]` sections and the `StorageUnits` count to match. It also adds the SVM host of each protected storage as a `data-lif::<host>` value under `HKLM\SOFTWARE\WOW6432Node\Data ONTAP\Clustered Data ONTAP Antivirus Connector\v1.0\data-lifs`, and removes the value again when no protected storage uses that host any more.

- A storage that starts event-based real-time protection in MDSS is added on the next cycle. `SharePath` is built from the account's SVM host and the storage's share name.
- An edited share path or a changed account SVM host is applied on the next cycle.
- A storage that stops event-based real-time protection, or is deleted, is removed once it has been missing from three consecutive cycles. A single short answer from MDSS is not treated as a deletion.
- If MDSS cannot be read (unreachable, non-200 response, unparseable payload), the cycle is skipped and nothing is removed.
- Changes apply to the running configuration immediately. No service restart is needed.

Storages protected by polling real-time protection are handled by MDSS directly and are left out of `config.ini`.

Hand-written `[Storage Unit N]` sections are still accepted. They are kept as long as the synchronisation is disabled (`StorageSyncIntervalInSeconds=0`). With the synchronisation enabled, a hand-written section whose storage is not reported by MDSS as event-based-RTP active is removed after three cycles.

`ClientId`

The client id of the storage configured in MDSS. It is sent as `storageClientId` in the real-time webhook payload and tells MDSS which storage the file belongs to. Copy it from the storage's configuration in MDSS.

`SharePath`

The ONTAP path prefix that identifies files belonging to this storage unit, for example `svm1\share1`. The value is used twice on each incoming scan request:

1. **Storage-unit selection.** The connector strips the `\\?\UNC\` prefix from the path ONTAP sent and looks for the storage unit whose `SharePath` matches the start of what remains. The match is case-insensitive and must end at a path separator: `svm1\share` does not claim a file under `svm1\share_2`. When several units match (nested share paths), the longest `SharePath` wins. If no unit matches, the file is dropped with `Failed to find a MDSS storage unit for file …` at `Error` level and ONTAP never receives a verdict for it.
2. **Computing the MDSS-relative path.** The connector strips `SharePath` plus one separator from the front of the path that remains after the `\\?\UNC\` prefix. The result is the relative path sent to MDSS as the file's `name`.

Because step 2 removes a fixed number of characters, `SharePath` must be **the exact leading portion of the ONTAP path after \?\UNC** — not merely a substring that happens to appear somewhere in it. A value that is too short or too long produces a wrong relative path and MDSS will not find the file.

Leading backslashes are stripped when the value is read, so `\svm1\share1` and `svm1\share1` are equivalent. Do not add a trailing separator. Share paths written by the synchronisation are normalised the same way.

Section `[Settings]`

Every key in this section is optional, and no value in it can stop the connector from starting:

- A missing or empty key uses its default.
- A non-numeric value, or a value outside the key's range, also uses the default. The connector logs a warning that names the key, the valid range and the default it used, for example:

```
Value out of range for key ZoneIdentifierFiles. Valid range: [0,3].
Using the default: 0
Invalid value (must be an integer) for key ZoneIdentifierFiles. Valid
range: [0,3]. Using the default: 0
```

The values in effect are shown in the configuration dump described under What the connector logs at startup.

Key	Type	Default	Range / accepted values
`LogRetentionCountInFiles`	Integer	`7`	`1` – `200`
`LogMaxFileSizeInMegabytes`	Integer	`10`	`1` – `1024`
`LogPath`	String	empty (`<install dir>\logs`)	An absolute or relative directory path, or empty
`LogLevel`	String	`Information`	`Verbose`, `Debug`, `Information`, `Warning`, `Error`, `Fatal`
`FileProcessingTimeoutInSeconds`	Integer	`3600`	1 or greater
`RpcLocalPipeHost`	String	`127.0.0.1`	An address, or the literal `NULL`
`ZoneIdentifierFiles`	Integer	`0`	`0` – `3`
`TimeoutResponse`	Integer	`4`	`0` – `5`
`StorageSyncIntervalInSeconds`	Integer	`15`	`0` – `86400`; `0` disables the synchronisation

`LogRetentionCountInFiles`

How many log files to keep. A rotation thread runs every 10 seconds and deletes the oldest of the connector's own log files until only `LogRetentionCountInFiles` remain. Each deletion is logged as `LogRotation - File deleted. File: <name>`.

Retention only ever counts and deletes files the connector itself named — `YYYYMMDD.log`, `YYYYMMDD_HHMMSS.log` and `YYYYMMDD_HHMMSS_<n>.log` — so a `LogPath` directory that also holds other content is safe.

A value outside `1` – `200` falls back to `7`, with the warning `Value out of range for key LogRetentionCountInFiles. Valid range: [1,200]. Using the default: 7.`

> **Upgrading from a version earlier than 4.6.0**
>
> `LogRetentionCountInDays` is the previous name of this key. Retention has always counted files, not days; the key was renamed to say so. A `config.ini` that still uses `LogRetentionCountInDays` keeps working: when `LogRetentionCountInFiles` is absent or empty, the value of `LogRetentionCountInDays` is applied to the file count, and a fallback warning names the key that was actually read. When both keys are present, `LogRetentionCountInFiles` wins.

`LogMaxFileSizeInMegabytes`

The size at which the current log file is rolled over. The connector writes one file per day, named `YYYYMMDD.log`. When that file reaches `LogMaxFileSizeInMegabytes`, logging continues in a new file named `YYYYMMDD_HHMMSS.log` (`YYYYMMDD_HHMMSS_<n>.log` when several rolls happen in the same second). The previous file is kept and is not appended to again; `LogRetentionCountInFiles` then bounds how many files remain.

An identical warning or error is written **once per log file**. A scanned file that is retried until its processing timeout produces one line rather than one per attempt. The suppression resets each time logging moves to a new log file, at the size limit and at the change of day, so every log file still carries the error at least once. It also resets after 5,000 distinct warnings and errors, so a repeated line can appear again in the same log file.

A value outside `1` – `1024` falls back to `10`, with the warning `Value out of range for key LogMaxFileSizeInMegabytes. Valid range: [1,1024]. Using the default: 10.`

`LogPath`

The directory the log files are written to. Leave it empty to keep the default `logs` folder next to the executable — `C:\ProgramData\OPSWAT-ONTAP-Connector\logs` for an installed connector. A directory that does not exist yet is created.

A relative path is resolved against the installation directory, not against the service's working directory. For example, `LogPath=connector-logs` on an installed connector writes to `C:\ProgramData\OPSWAT-ONTAP-Connector\connector-logs`. The resolved path is logged once at `Information` level: `Configured LogPath is relative, resolved against the installation directory: <path>`.

If the configured path cannot be used — for example a drive that does not exist, or a directory the service account cannot write to — the service still starts. Logging falls back to the default directory and one line at `Error` level records the configured path and an error code: `Configured LogPath cannot be used, logging to <default> instead. Configured path: <value>`. The code is:

- `5`, the Windows access-denied code, when the service account cannot create the directory.
- `-6` when the directory cannot be created for another reason, for example a drive that does not exist.
- `-11` when the directory exists but the log file cannot be opened in it.

`LogLevel`

The minimum severity written to the log. Messages below the configured level are discarded.

Value	Use
`Verbose`	Everything, including per-file ONTAP verdict mapping
`Debug`	Full HTTP request and response bodies exchanged with MDSS — the level to use when troubleshooting MDSS connectivity
`Information` (default)	Lifecycle events, per-file progress, configuration dump at startup
`Warning`	Transport switches, ONTAP service not yet available, non-fatal failures
`Error`	Failed operations
`Fatal`	Startup-blocking failures only

The value is compared case-sensitively, so it must be spelled exactly as listed. Any other value, including `information`, falls back to `Information`, with the warning `Invalid LogLevel value: information. Available options: Verbose, Debug, Information, Warning, Error, Fatal. Using the default: Information.`

Note that the level only takes effect once the service main thread starts; configuration parsing itself is logged at `Verbose`, which is why the startup banner and the configuration dump always appear.

When the executable is run interactively (`--install`, `--uninstall`, `--upgrade`) output goes to the console instead of a file. File logging is used for `--service` and `--debug`.

FileProcessingTimeoutInSeconds

How long a single file may stay in the connector's pipeline, measured from the moment ONTAP's scan request was received. When the timeout is exceeded and MDSS has still not returned a final verdict — or the connector cannot reach MDSS or ONTAP at all — the connector gives up on the file: it sends the verdict configured in `TimeoutResponse` to ONTAP and removes the file from its tracking list. Exactly one timeout verdict is sent per file.

The default `3600` (one hour) is deliberately generous: it must accommodate the slowest case MDSS can produce, including large archives and Deep CDR. Lowering it makes the connector give up sooner on files MDSS is still working on. The minimum is `1` and there is no upper bound. A lower or non-numeric value falls back to `3600`, with the warning `Value out of range for key FileProcessingTimeoutInSeconds. Minimum: 1. Using the default: 3600`.

Files still in flight are persisted to `data.json` next to the executable every 3 seconds, so a connector restart does not lose pending work; the timeout continues to be measured from the original request time.

RpcLocalPipeHost

The network address the connector uses when binding to the NetApp registration pipe (`\pipe\ntapvscp`) exposed by the ONTAP AV connector service (`ontapavc`) on the same machine.

Value	Behaviour
`127.0.0.1` (default)	Binds the named pipe using the loopback address
Any other address	Binds using that address as the network address
`NULL` (the literal four-character string)	Binds with a null network address — the local machine, with no explicit host

Use `NULL` when the loopback binding is refused by the local RPC configuration. The connector retries the binding and the ONTAP registration every 3 seconds until both succeed, so a wrong value shows up as a repeating `RpcConnectionThread: Failed to create connection` or `Failed to register to RPC server` in the log.

ZoneIdentifierFiles

Controls how `:Zone.Identifier` alternate-data-stream files are handled. Windows creates these tiny metadata streams alongside downloaded files, and ONTAP submits

them for scanning like any other file — which can double the request volume for no security benefit.

Value	Mode	Behaviour
`0`	Skip (default)	The request is logged and dropped. No scan, and **no verdict is returned to ONTAP** — ONTAP applies its own scan-request timeout policy.
1	Scan	Treated as a normal file and sent to MDSS through the usual pipeline.
2	Allow	An immediate allow verdict (`VSENGINE_NO_ERROR`) is returned without scanning.
3	Block	An immediate block verdict (`VSENGINE_INFECTED`) is returned without scanning.

A value outside `0` – `3` falls back to `0` (Skip), and the connector logs a warning.

Prefer `2` (Allow) over `0` (Skip) if ONTAP clients are seeing access delays on downloaded files: it settles the request immediately instead of leaving ONTAP waiting for its own timeout.

`TimeoutResponse`

The verdict sent to ONTAP when a file hits `FileProcessingTimeoutInSeconds`. Each value maps to a NetApp Vscan engine status, and it is ONTAP — not the connector — that decides what the client sees based on that status and the share's `scan-mandatory` setting.

Value	Mode	ONTAP status	Effect
0	Allow	VSENGINE_NO_ERROR	Access is allowed. The file is treated as clean.
1	TimeoutNoBlock	VSENGINE_INTERNAL_TIMEOUT_NOBLOCK	Access is allowed, but recorded as a scanner timeout.
2	InfectedBlock	VSENGINE_INFECTED	Access is always blocked and the file is reported as infected.
3	TimeoutBlock	VSENGINE_INTERNAL_TIMEOUT_BLOCK	Access is always blocked, with a timeout reason.
4	Timeout (default)	VSENGINE_INTERNAL_TIMEOUT	Blocked only where scanning is mandatory; allowed otherwise.
5	EngineBusyRetryLater	VSENGINE_BUSY_TRY_AGAIN_LATER	Reported as scanner busy so ONTAP can retry or hand the request to another scanner.

A value outside `0` – `5` falls back to `4` (Timeout), and the connector logs a warning. Use `0` or `1` for availability-first deployments, `2` or `3` for security-first, and `5` when several scanners serve the same ONTAP cluster and another one should get a chance at the file.

`StorageSyncIntervalInSeconds`

How often, in seconds, the connector synchronises its storage units with MDSS. Each cycle reads the NetApp ONTAP accounts and their storages from MDSS (`GET /api/account`), keeps the storages whose event-based real-time protection is running, and reconciles two places against that answer: the `[Storage Unit N]` sections in `config.ini`, and the `data-lif::<host>` values in the ONTAP AV Connector registry. The full behaviour is described under Storage units are maintained automatically.

`0` turns the synchronisation off. No cycle runs, `config.ini` and the `data-lifs` registry values are left exactly as configured by hand, and the startup log states `Storage synchronisation is disabled (StorageSyncIntervalInSeconds=0)`. Use this when the storage units are managed by another tool, or as a kill switch.

A value outside `0` – `86400` falls back to `15`, with the warning `Value out of range for key StorageSyncIntervalInSeconds. Valid range: [0,86400]. Using the default: 15`. Only the exact value `0` turns the synchronisation off; a negative value leaves it running every 15 seconds.

Complete example

```
[MetaDefender Storage Security]
IP=10.0.0.15
Port=8008
PortSsl=8443
IgnoreCertErrors=1
Apikey=b1946ac92492d2347c6235b4d2611184
StorageUnits=2

[Storage Unit 1]
ClientId=6f1c2f7a-6c1e-4f9a-9c0e-3ac0d1f2b111
SharePath=svm1\finance_share

[Storage Unit 2]
ClientId=9d4b8e01-11aa-42cc-8e77-5b6c9d0e2222
SharePath=svm1\hr_share

[Settings]
LogRetentionCountInFiles=14
LogMaxFileSizeInMegabytes=50
LogPath=D:\mdss-connector-logs
LogLevel=Debug
FileProcessingTimeoutInSeconds=1800
RpcLocalPipeHost=127.0.0.1
ZoneIdentifierFiles=2
TimeoutResponse=4
StorageSyncIntervalInSeconds=15
```

The template the connector package is built from, with every optional key at its default. It declares no storage units: the synchronisation adds them once storages with event-based real-time protection exist in MDSS. When you download the connector from MDSS, MDSS fills in `Apikey`, `Port` and `PortSsl`, adds a `[Storage Unit 1]` section for the storage the connector was downloaded for, and sets `StorageUnits=1`.

```
[MetaDefender Storage Security]
IP=127.0.0.1
Port=80
PortSsl=443
IgnoreCertErrors=0
Apikey=
StorageUnits=0

[Settings]
LogRetentionCountInFiles=7
LogMaxFileSizeInMegabytes=10
LogPath=
LogLevel=Information
FileProcessingTimeoutInSeconds=3600
RpcLocalPipeHost=127.0.0.1
ZoneIdentifierFiles=0
TimeoutResponse=4
StorageSyncIntervalInSeconds=15
```

Validation and troubleshooting

What the connector logs at startup

After parsing succeeds, the connector dumps the effective configuration at `Information` level — the resolved values, with defaults already applied. One line is written per section, plus one per storage unit:

```
MetaDefender Storage Security <IP: 10.0.0.15. Port: 8008. PortSsl:
8443. IgnoreCertErrors: 1. Apikey: …>
Settings <LogRetentionCountInFiles: 14. LogMaxFileSizeInMegabytes: 50.
LogPath: D:\mdss-connector-logs. LogLevel: Debug.
FileProcessingTimeoutInSeconds: 1800. RpcLocalPipeHost: 127.0.0.1.
ZoneIdentifierFiles: 2. TimeoutResponse: 4.
StorageSyncIntervalInSeconds: 15>
Storage Unit <ClientId: 6f1c2f7a-…. SharePath: svm1\finance_share>
```

This is the fastest way to confirm the connector is reading the file you think it is, and that your edits took effect. An unrecognised or misspelled `[Settings]` key does not appear here; the default it failed to override does. A value that fell back to its default also shows the default here, next to the warning that explains why.

Error messages and their causes

These stop the connector from starting:

Log message	Cause
`Failed to read from .ini file. Section: …. Key: …`	Names the required key that is absent or empty (`IP`, `Port`, `Apikey`, `StorageUnits`, `ClientId`, `SharePath`), or the `[Storage Unit N]` section that does not exist
`Failed to read from .ini file at location …`	Summary line after the line above, or the file is missing from the executable's folder, or `StorageUnits` is not a valid count
`Invalid format for key IP`	`IP` is a hostname or not a valid IPv4 address
`Invalid value (must be integer). Key: Port`	`Port` contains a non-numeric value
`Value not in range [1,65535] for key Port`	`Port` is not a valid TCP port
`Invalid value for key (integer expected). Key: StorageUnits`	`StorageUnits` contains a non-numeric value
`Value must not be negative. Key: StorageUnits`	`StorageUnits` is negative
`Invalid key for storage unit. Key: …`	A `[Storage Unit N]` section contains a key other than `ClientId` or `SharePath`
`Invalid configuration inside config.ini file`	Summary line written after an `IP`, `Port` or `[Storage Unit N]` error
`Error when reading config.ini file`	Final line written at `Fatal` level after any of the above; the service does not start

These do not stop the connector:

Log message	Cause
`Invalid value (must be an integer) for key …. … Using the default: …`	A `[Settings]` key, `PortSsl` or `IgnoreCertErrors` contains a non-numeric value. The default is used
`Value out of range for key …. … Using the default: …`	A `[Settings]` key, `PortSsl` or `IgnoreCertErrors` is outside its range. The default is used. For retention, the message names `LogRetentionCountInDays` when the legacy key was the one read
`Invalid LogLevel value: …`	Misspelled or wrongly-cased log level. `Information` is used
`Configured LogPath cannot be used, logging to … instead. Configured path: …`	`LogPath` points to a directory that does not exist and cannot be created, or that the service account cannot write to. Logs go to the default directory
`Configured LogPath is relative, resolved against the installation directory: …`	Informational. Shows where a relative `LogPath` points
`Storage synchronisation is disabled (StorageSyncIntervalInSeconds=0)`	Informational. The synchronisation was turned off on purpose
`StorageSyncMonitor: could not enumerate storages from MDSS, skipping this cycle`	MDSS is unreachable, refused the request (for example, the API key has no read permission on accounts), or returned an unusable answer. Nothing is added or removed in this cycle
`Failed to find a MDSS storage unit for file …`	No `SharePath` matches the path ONTAP submitted

Add an SFTP Storage

Product Specification Requirements	Version
Secure File Transfer Protocol (SFTP)	OpenSSH_for Windows_8.1p1 OpenSSH_7.9p1 Debian-10+deb10u2 OpenSSL 1.1.1(for Linux)

IMPORTANT

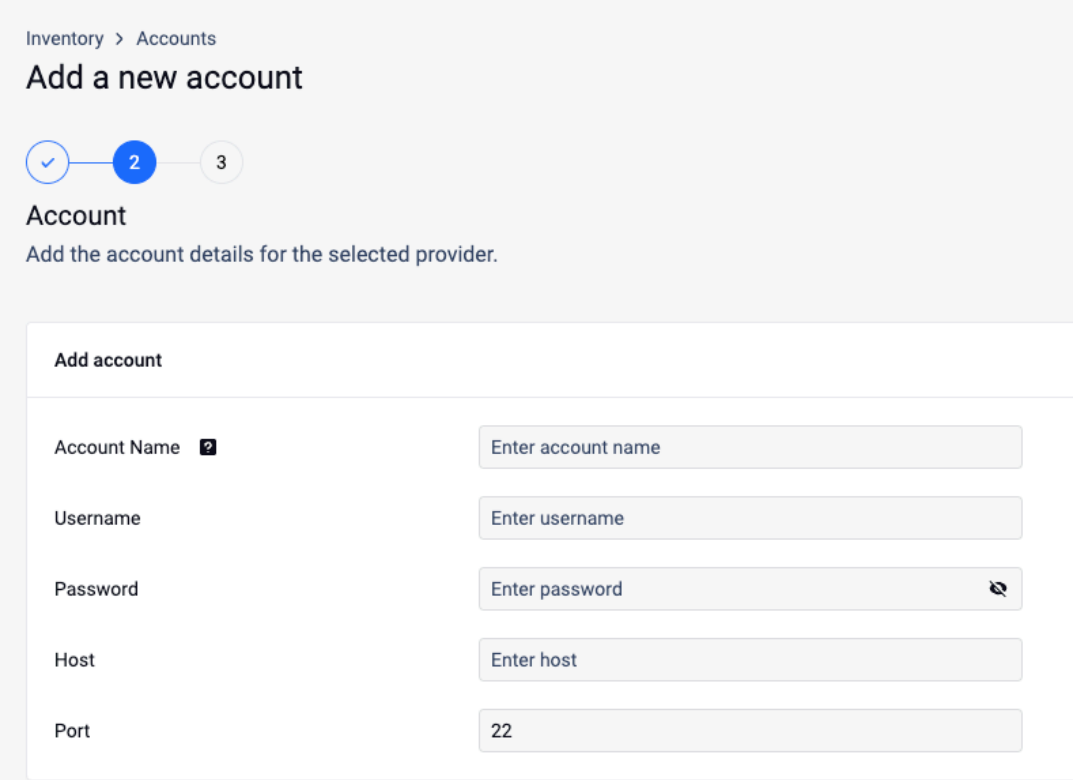
Integration with SFTP storage is achieved through the SSH File Transfer Protocol (SFTP). Before proceeding, please ensure that this protocol is enabled on your system. Our application is compatible with SFTP servers that adhere to the standard SSH File Transfer Protocol, including Cerberus SFTP for Windows and the OpenSSH SFTP subsystem, which is typically included with Linux and Unix SSH servers.

Info

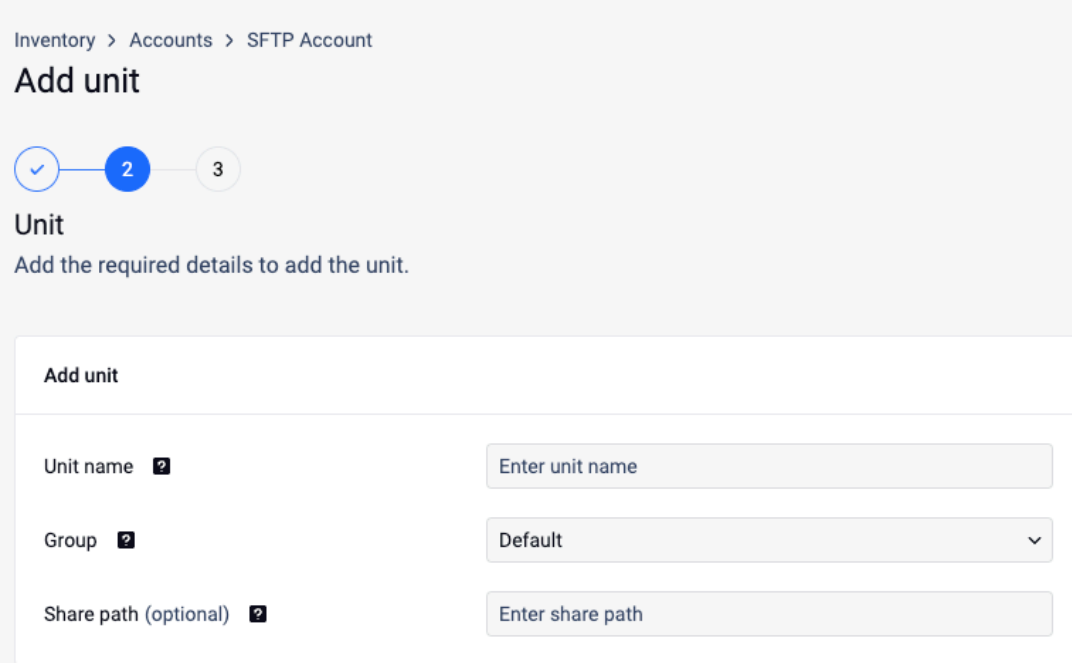
We recommend increasing the maximum number of concurrent connections allowed by your SFTP server from its default setting. This helps improve performance and prevent connection bottlenecks during high-load operations. After that, make sure to set the **SFTP_CONNECTION_POOL_MAX** environment variable in MDSS, which is at 100 by default.

In order to integrate SFTP with MetaDefender Storage Security please follow these steps:

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **SFTP (SSH File Transfer Protocol)** and click **Next**



2. Give your account a name so you can easily identify it later
3. Input the **Username** required to authenticate your access to the SFTP server. (The user should have at least read and write security permissions)
4. Provide the **Password** associated with the username provided at step 3.
5. **Host:** Enter the IP address of the SFTP server you wish to connect to.
6. **Port**: Specify the port number used for the SFTP connection, typically **22** for standard SFTP
7. Click **Next** in order to finish the process of adding the account
8. From the left side menu, navigate to **Inventory**, click **Add New** > **Unit**, choose the previously created account and click **Next**



9. Give your unit an unique name so you can easily identify it later.
10. Assign your unit to a Group
11. (Optional) Indicate the specific directory path on the SFTP server from which you want to access files.

> **Info**
>
> The folder path should start from the root of the shared directory, not the absolute path of the server. For example, if the shared directory is `/shared_files` and you want to access a folder named `documents` within it, you would enter `documents` as the folder path, rather than the full path like `/home/user/shared_files/documents`

9. Click **Next** in order to finish the process.

> **Info**
>
> A SSH connection is required to be enabled in order to connect to a Unix-based system **SharePath.** If the SSH is not installed please run the following commands:
>
> ```
> sudo apt-get update
> ```
>
> ```
> sudo apt-get upgrade
> ```
>
> ```
> sudo apt-get install openssh-client
> ```

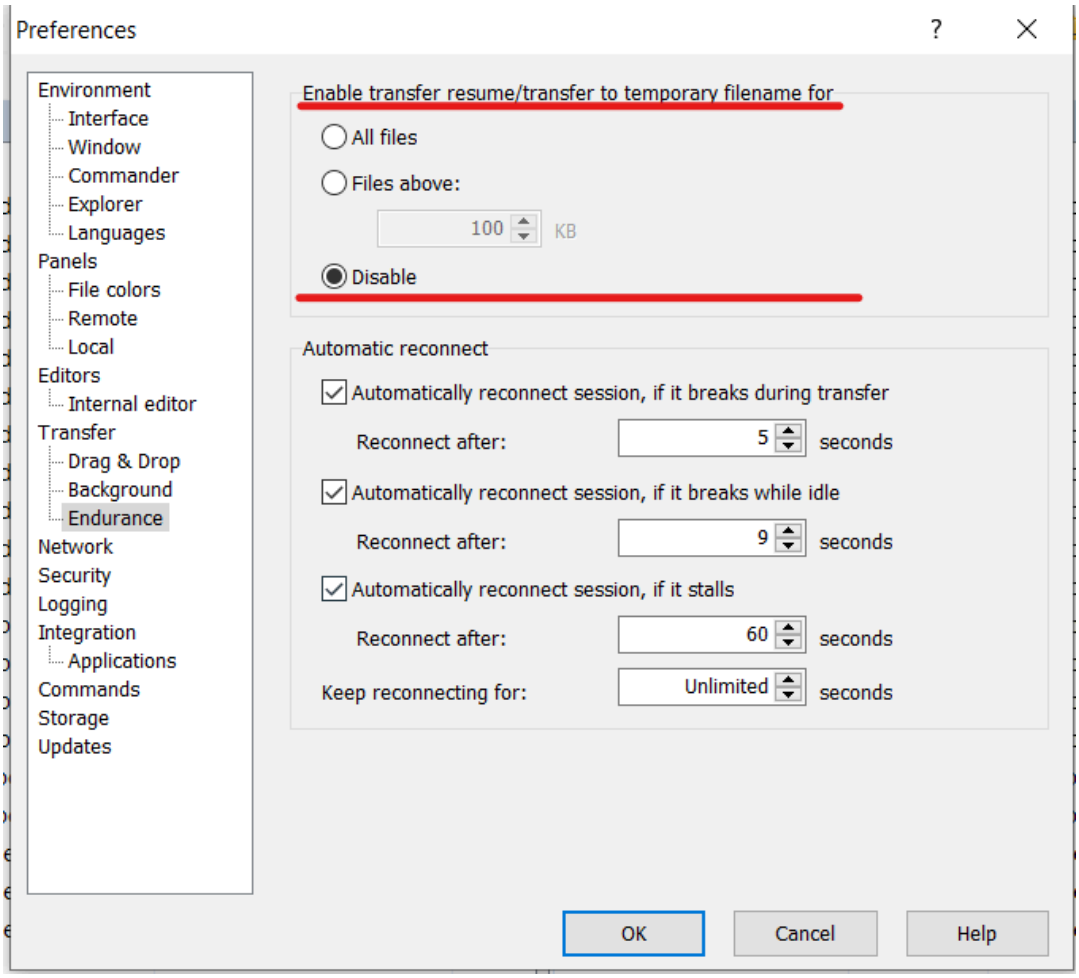
> **Info**

If you use a hostname in the host portion of IPv4/v6 socket address, the program may show a non-deterministic behavior. The socket address will be resolved differently into an actual IPv4/v6 address, depending on the results from DNS resolution and/or the host configuration. For deterministic behavior use a numeric address in host portion.

Warning

Please make sure that **Enable transfer resume/transfer to temporary filename** option is set on **Disable** if **using 3rd party SFTP clients** (if it allows this feature).

See the image below for **WinSCP client**.



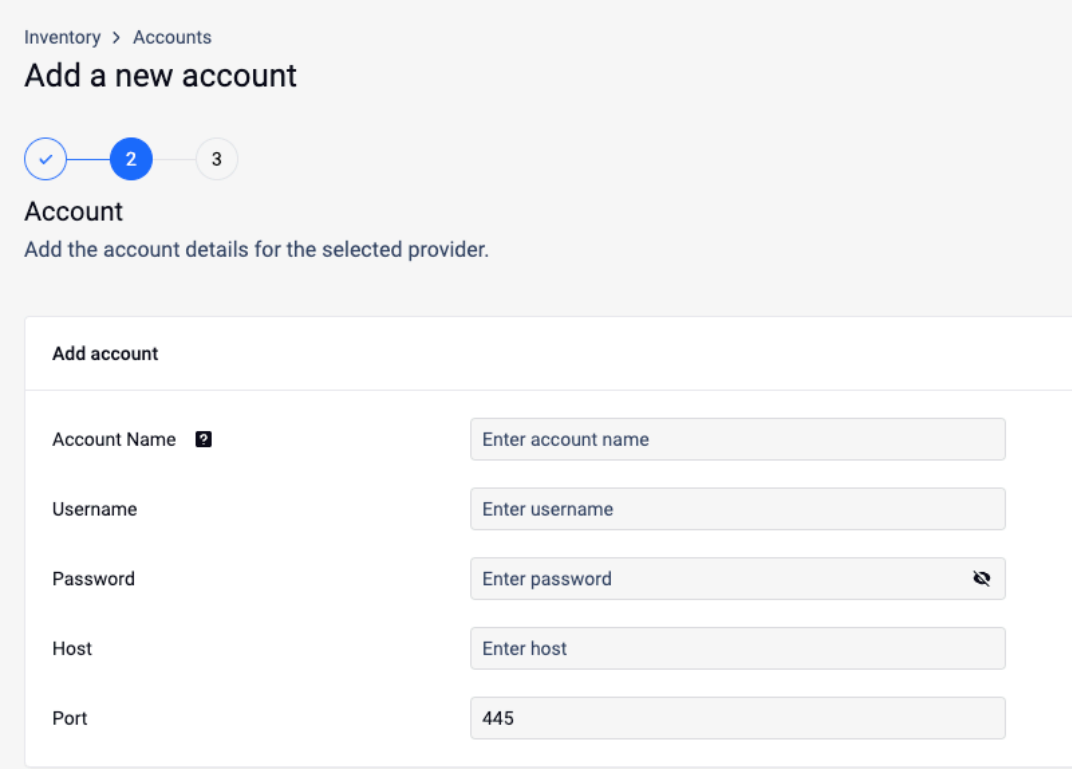
Add a Dell EMC Isilon Storage

Product Specification Requirements	Version
Dell EMC OneFS	Version 8.x
Service Message Block (SMB) protocol	SMB 3.1.1 (or SMB3)

> **Info**
>
> Integration with Dell Isilon is currently accomplished via the Service Message Block (SMB) protocol. Before you begin, you should ensure that this protocol is enabled in your Dell Isilon configuration.

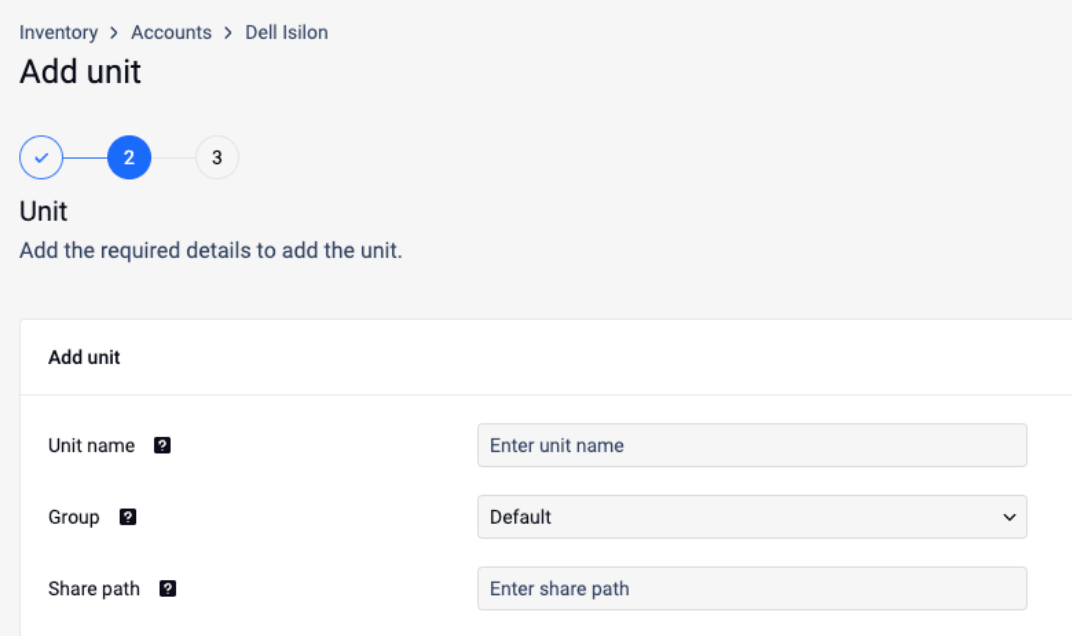
In order to integrate Dell Isilon with MetaDefender Storage Security please follow these steps:

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **Dell Isilon** and click **Next**



2. Give your account a name so you can easily identify it later
3. Input the **Username** required to authenticate your access to the server. (The user should have at least read and write security permissions)
4. Provide the **Password** associated with the username from step 3.

5. **Host:** Enter the IP address of the server you wish to connect to.
6. **Port**: Specify the port number used for the connection, **445** being the default port
7. Click **Next** in order to finish the process
8. From the left side menu, navigate to **Inventory**, click **Add New** > **Unit**, choose the previously created account and click **Next**



9. Give your unit an unique name so you can easily identify it later
10. Assign your unit to a Group
11. Indicate the specific directory path on the server from which you want to access files
12. Click **Next** in order to finish the process.

Info

To avoid authentication issues the Username should have one of the following formats:

- local users should be .\username [for example: .\Administrator]
- domain users should be username@domain or domain\username

Info

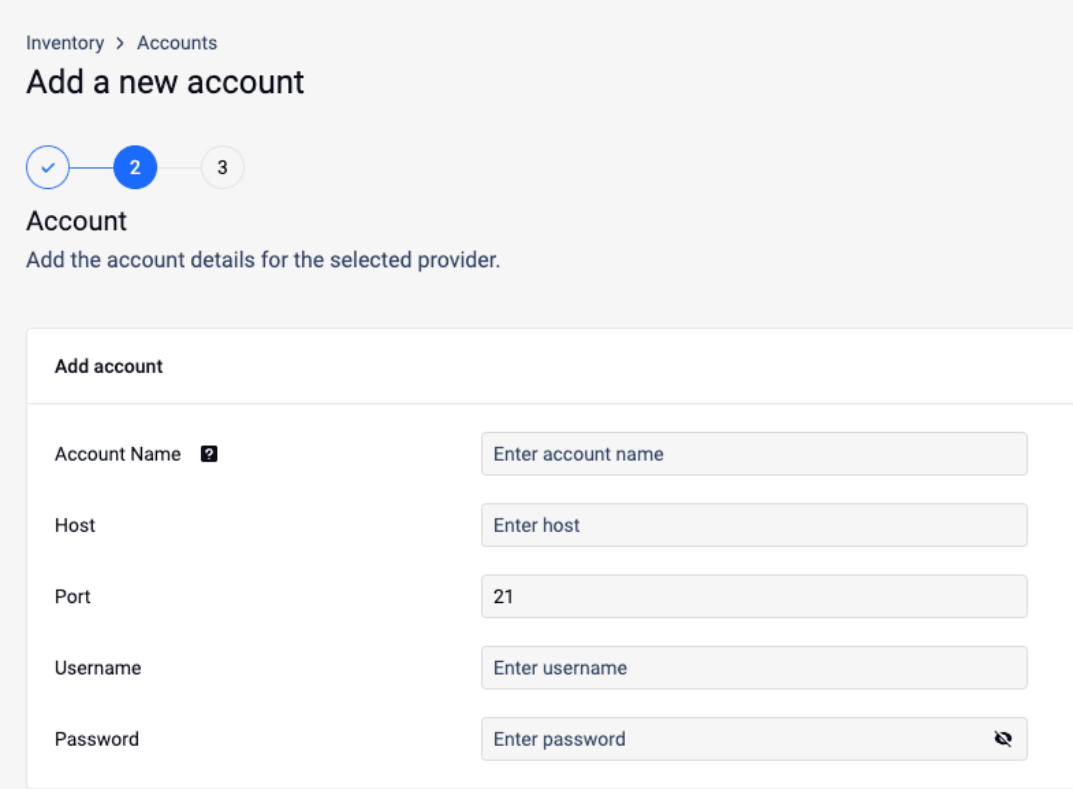
If you use a hostname in the host portion of IPv4/v6 socket address, the program may show a non-deterministic behavior. The socket address will be resolved differently into an actual IPv4/v6 address, depending on the results from DNS resolution and/or the host configuration. For deterministic behavior use a numeric address in host portion.

Add an FTP Storage

> **Info**
>
> We recommend increasing the maximum number of concurrent connections allowed by your FTP server from its default setting. This helps improve performance and prevent connection bottlenecks during high-load operations. After that, make sure to set the **FTP_MAX_CONNECTIONS** environment variable in MDSS

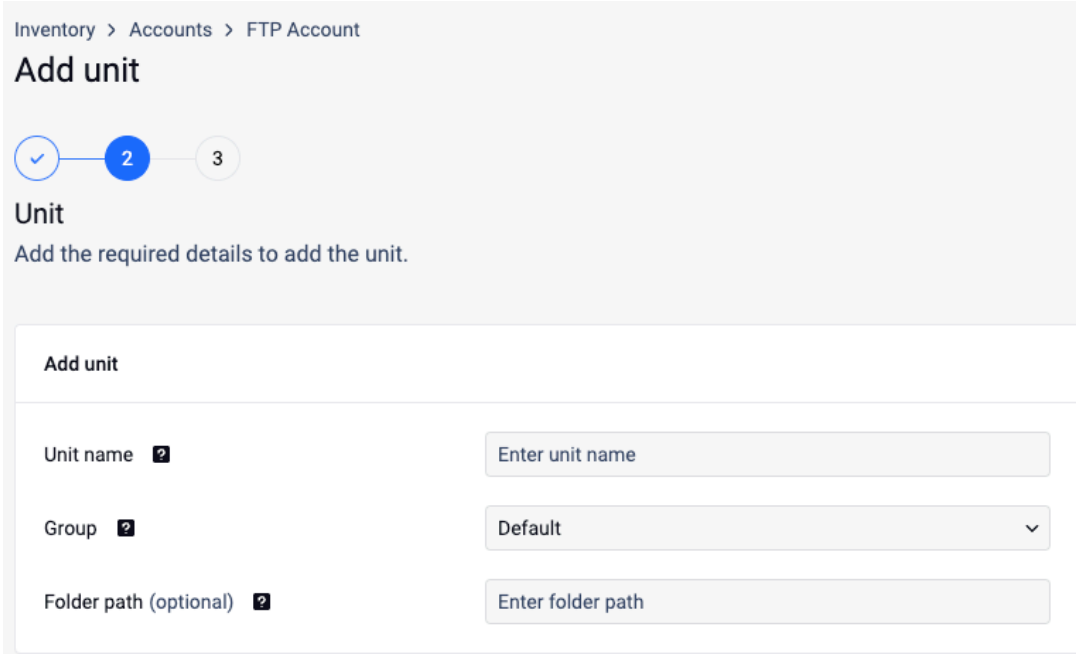
In order to integrate FTP with MetaDefender Storage Security please follow these steps:

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **FTP (File Transfer Protocol)** and click **Next**



2. Give your account a name so you can easily identify it later
3. **Host:** Enter the IP address of the FTP server you wish to connect to.
4. **Port**: Specify the port number used for the FTP connection, typically **21** for standard FTP
5. Input the **Username** required to authenticate your access to the FTP server. (The user should have at least read and write security permissions)
6. Provide the **Password** associated with the username provided at step 4.
7. Click **Next** to finish the process

8. From the left side menu, navigate to **Inventory**, click **Add New** > **Unit**, choose the previously created account and click **Next**.



8. Give your unit an unique name so you can identify it later
9. Assign your unit to a Group
10. Indicate the specific directory path on the FTP server from which you want to access files .
11. Click **Next** to finish the process

> **Info**
>
> The folder path should start from the root of the shared directory, not the absolute path of the server. For example, if the shared directory is `/shared_files` and you want to access a folder named `documents` within it, you would enter `documents` as the folder path, rather than the full path like `/home/user/shared_files/documents`

6. Click **Continue** in order to finish the process.

> **Info**
>
> If you use a hostname in the host portion of IPv4/v6 socket address, the program may show a non-deterministic behavior. The socket address will be resolved differently into an actual IPv4/v6 address, depending on the results from DNS resolution and/or the host configuration. For deterministic behavior use a numeric address in host portion.

Configuring a Windows IIS FTP Server for MDSS Scanning

Applies to: Microsoft IIS FTP on Windows Server · All MetaDefender Storage Security (MDSS) versions

MDSS scans FTP storage by opening a new passive data connection for every folder listing and every file it reads. One IIS setting governs how many of those connections the server can sustain — and its default is far too small for a scan. This page covers that setting, the connection limits that go with it, and how to verify the result.

---

1. Settings at a glance

Four server-side settings matter. Only the first one commonly breaks scans; the rest are usually already correct and just need confirming.

Setting	Recommended	IIS default	Priority
Data Channel Port Range	`10000-20000`	`0-0` (ephemeral)	**Must set**
Firewall rule for that range	Inbound TCP allow	none	**Must set**
Server Listen Backlog	`500`	`60`	Confirm
Max Connections	unlimited, or >= 200	`4294967295`	Confirm

> **Read this first**
>
> When an FTP scan fails, **Max Connections and Server Listen Backlog are the settings everyone checks, and they are almost never the cause** — IIS ships with Max Connections effectively unlimited. The setting that actually stops scans is the passive **Data Channel Port Range**, because it is consumed by connection turnover rather than by how many clients are connected. Check it first.

---

2. Set the data channel port range

Every listing and every file transfer takes one port from this range. When the transfer finishes, Windows holds that port in `TIME_WAIT` for 120 seconds before it can be reused. A range of a handful of ports therefore supports only a handful of operations every two minutes, which a scan exhausts in seconds.

This range is server-wide — one range covers every FTP site on the machine. Run all commands in an **elevated** PowerShell session on the FTP server.

```
& "$env:windir\system32\inetsrv\appcmd.exe" set config -
section:system.ftpServer/firewallSupport `
  /lowDataChannelPort:10000 /highDataChannelPort:20000
```

Choosing your own numbers is fine, as long as they satisfy all four constraints:

- **At least 1000 ports** — see How many ports you need.
- **Within 1025-65535** — IIS rejects anything lower.
- **Below 49152**, so the range never overlaps the Windows dynamic port range. Confirm your own with `netsh int ipv4 show dynamicport tcp`.
- **Clear of reserved and in-use ports.** Hyper-V, Docker and WinNAT silently reserve large blocks; a range overlapping one will fail to bind.

Check the range you picked is free:

```
netsh int ipv4 show excludedportrange protocol=tcp

Get-NetTCPConnection -State Listen |
  Where-Object { $_.LocalPort -ge 10000 -and $_.LocalPort -le 20000 } |
  Select-Object LocalPort, @{n='Process';e={(Get-Process -Id
$_.OwningProcess).ProcessName}}
```

No overlapping exclusions and no listeners returned means the range is yours to use.

---

3. Allow the range through the firewall

Widening the range achieves nothing if the firewall still blocks it. Scope the rule to the MDSS host so you are not opening 10,000 ports to the whole network.

```
New-NetFirewallRule -DisplayName "IIS FTP Passive Data Channel 10000-
20000 (MDSS)" `
  -Direction Inbound -Action Allow -Protocol TCP -LocalPort 10000-20000
`
  -RemoteAddress 10.0.0.50 -Profile Any
```

> **Use the MDSS host address, not the container address**
>
> On Docker deployments, the MDSS services that talk to the FTP server run in Docker, and their outbound traffic is source-NAT'd to the Docker host. The address to allow is the **MDSS host** IP — never the internal `172.x.x.x` container address.

To read the address the FTP server actually sees, run this while a scan or a storage **Test connection** is in progress:

```
Get-NetTCPConnection -LocalPort 21 -State Established |
  Select-Object -ExpandProperty RemoteAddress -Unique
```

If the MDSS host uses DHCP, give it a static address or a reservation first — otherwise the rule silently stops matching the next time the lease changes.

For multiple MDSS hosts, `-RemoteAddress` accepts a list `[@('10.0.0.50','10.0.0.51')]` or a subnet `['10.0.0.0/24']`. Scope the existing port 21 rule the same way — there is little point restricting the data channel while the control channel accepts connections from anywhere.

---

4. Confirm the connection limits

These two settings live on the FTP site rather than the server. Max Connections caps how many clients may be connected at once; Server Listen Backlog caps how many connection attempts can queue while the service is busy. MDSS connects in bursts, so a small backlog causes intermittent failures under load.

Substitute your own FTP site name:

```
$site = "Your FTP Site Name"

& "$env:windir\system32\inetsrv\appcmd.exe" set site /site.name:"$site"
`
  /ftpServer.connections.maxConnections:4294967295 `
  /ftpServer.connections.serverListenBacklog:500
```

`4294967295` is the IIS value for unlimited and is the default, so in most cases this command changes nothing but is harmless to run. Set it explicitly if a previous administrator lowered it. If your policy requires a real cap, size it to at least **200**: the FTP provider runs inside four MDSS services (Storages, Discovery, Remediations and Scanning) that each maintain their own connection pool, by default 8 sessions per credential set, so a single host can hold up to 32 sessions before you account for retries or a second MDSS instance.

If `appcmd` rejects the syntax on your IIS version, set the same two values through **IIS Manager → select the FTP site → Advanced Settings → ftpServer → connections**.

If the FTP server is behind NAT

The server tells the client which address to open the data connection on. Behind NAT it will advertise its private address unless you tell it otherwise, and every data connection will fail.

```
& "$env:windir\system32\inetsrv\appcmd.exe" set config -
section:system.applicationHost/sites/siteDefaults `
  /ftpServer.firewallSupport.externalIp4Address:"203.0.113.10"
```

Use the address MDSS connects to. Any intervening firewall must also forward the same data channel port range.

---

5. Restart and verify

IIS only picks up a changed data channel range when the FTP service restarts. This drops any FTP session in progress, so do it outside a scan window.

```
net stop ftpsvc
net start ftpsvc

Get-Service ftpsvc | Select-Object Name, Status

Select-String -Path
"$env:windir\system32\inetsrv\config\applicationHost.config" `
  -Pattern
'firewallSupport|maxConnections|serverListenBacklog|externalIp'
```

The service should report `Running`, and the configuration should show your new `lowDataChannelPort` and `highDataChannelPort`.

Now start a scan from MDSS and, while it runs, confirm data connections are landing in the new range:

```
Get-NetTCPConnection | Where-Object { $_.LocalPort -ge 10000 -and
$_.LocalPort -le 20000 } |
  Group-Object State | Format-Table Name, Count
```

A healthy result shows a handful of `Established` connections and a larger, **stable** count of `TimeWait` entries. Stable is the important word: a count that plateaus means ports are being recycled as fast as they are consumed. A count that climbs steadily toward the size of your range means the range is too small — widen it.

---

6. How many ports you need

Port demand is driven by connection turnover, not by how many files the storage holds:

```
ports needed  ~=  new data connections per second  x  120
```

The 120 is the Windows `TIME_WAIT` delay in seconds. A production MDSS scan measured roughly six new data connections per second, which held about 750 ports in use at steady state — against an IIS default that often leaves only a handful available.

Deployment	Range	Ports
Single MDSS host, light scanning	`10000-11000`	1001
Standard — recommended default	`10000-20000`	10001
Multiple MDSS hosts, or real-time plus on-demand	`10000-20000` with a shorter `TIME_WAIT`	10001

A bigger range costs nothing — IIS does not reserve the ports, it just draws from the range on demand.

For the heaviest workloads you can instead attack the other half of the formula and shorten `TIME_WAIT`, which multiplies the capacity of whatever range you chose:

```
New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters' `
  -Name 'TcpTimedWaitDelay' -PropertyType DWord -Value 30 -Force
```

Valid values are 30-300 seconds. Going from 120 to 30 quadruples effective capacity. Unlike the other changes on this page, this one needs a full reboot rather than an FTP service restart.

---

7. If scans still fail

The signature of a misconfigured data channel: logins succeed and the control channel works, but every data connection fails immediately. Work through these in order.

1. **Is the port range wide enough?** Run `appcmd list config -section:system.ftpServer/firewallSupport`. Anything under a few hundred ports is the cause.
2. **Did the FTP service restart** after the range changed? Without it, IIS is still using the old range.
3. **Does the firewall rule match the new range exactly,** and is it scoped to the right source address? A rule left pointing at an old narrow range fails identically.
4. **Watch the drops directly.** Enable logging, reproduce, then read it — dropped entries name the exact ports being refused.

```
Set-NetFirewallProfile -All -LogBlocked True

# reproduce the failure, then:
Get-Content "$env:systemroot\system32\LogFiles\Firewall\pfirewall.log"
-Tail 200 |
  Select-String 'DROP'

# turn logging back off when finished
Set-NetFirewallProfile -All -LogBlocked False
```

If the range and the firewall both check out, look at what the server advertises in its passive reply — a private or unexpected address there points back to the NAT setting in section 4. The IIS FTP logs under `%SystemDrive%\inetpub\logs\LogFiles\FTPSVC<n>` record every command, and the `sc-win32-status` column on failing `LIST` entries gives the underlying Windows error.

---

8. MDSS-side settings

This setting is optional and set in the MDSS `customer.env` file (see Custom configuration). It is not a substitute for the server configuration above. There is no dedicated FTP container: the FTP provider runs inside the Storages, Discovery, Remediations and Scanning services, so restart MDSS (`mdss -c restart`) for a change to take effect.

Variable	Default	Guidance
`FTP_MAX_CONNECTIONS`	unset	Unset, each service keeps up to 8 connections per FTP credential set. Setting it changes that limit (2-64). Set it only when the FTP server genuinely limits concurrent sessions; lowering it does not relieve data channel port exhaustion.

Every command on this page runs in an elevated PowerShell session on the FTP server

Add a Scality RING Storage

1. Prerequisites

In order to Add a **Scality RING** storage in MDSS, you will need the **Service URL**, **Access Key Id / Secret Access Key** and the **bucket name**.

2. Obtaining the Service URL

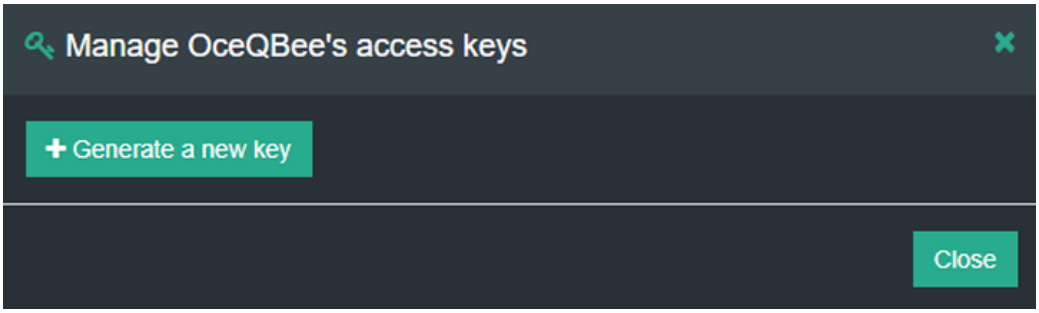
The Service URL is the S3 endpoint URL of your Scality RING, used to access the S3 services on the RING. It is typically available via the **Supervisor interface** under **S3 Services**.

3. Obtaining Access Keys

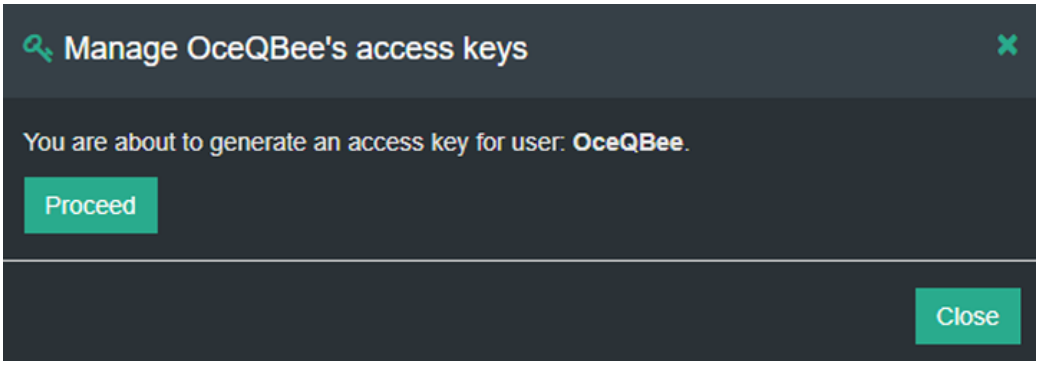
Access keys are managed through the Scality RING Supervisor's S3 Console.

1. Click the target account's Actions Key icon to call the Manage access keys dialog.



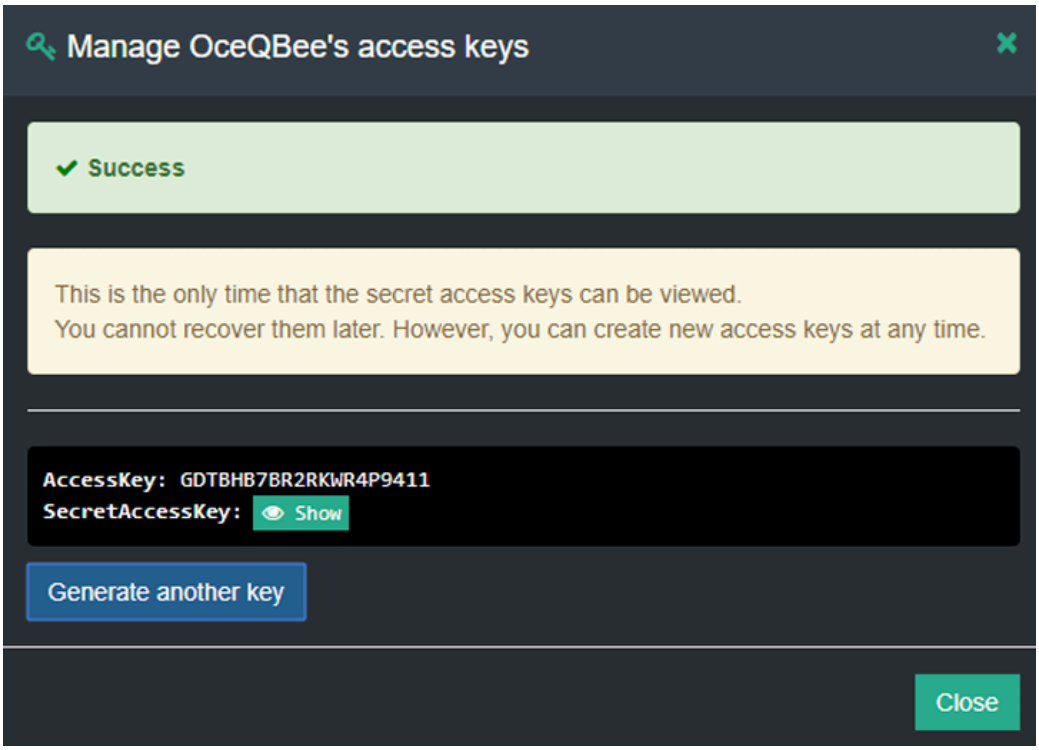


2. Click Generate a new key. A confirmation dialog will display.



3. Click Proceed to trigger account access key creation. The dialog will indicate the success of the operation, as well the AccessKey (click Show to view the

SecretAccessKey).



> **Warning**
>
> Copy and save SecretAccessKey to a secure location. It is not shown again and cannot be recovered later.

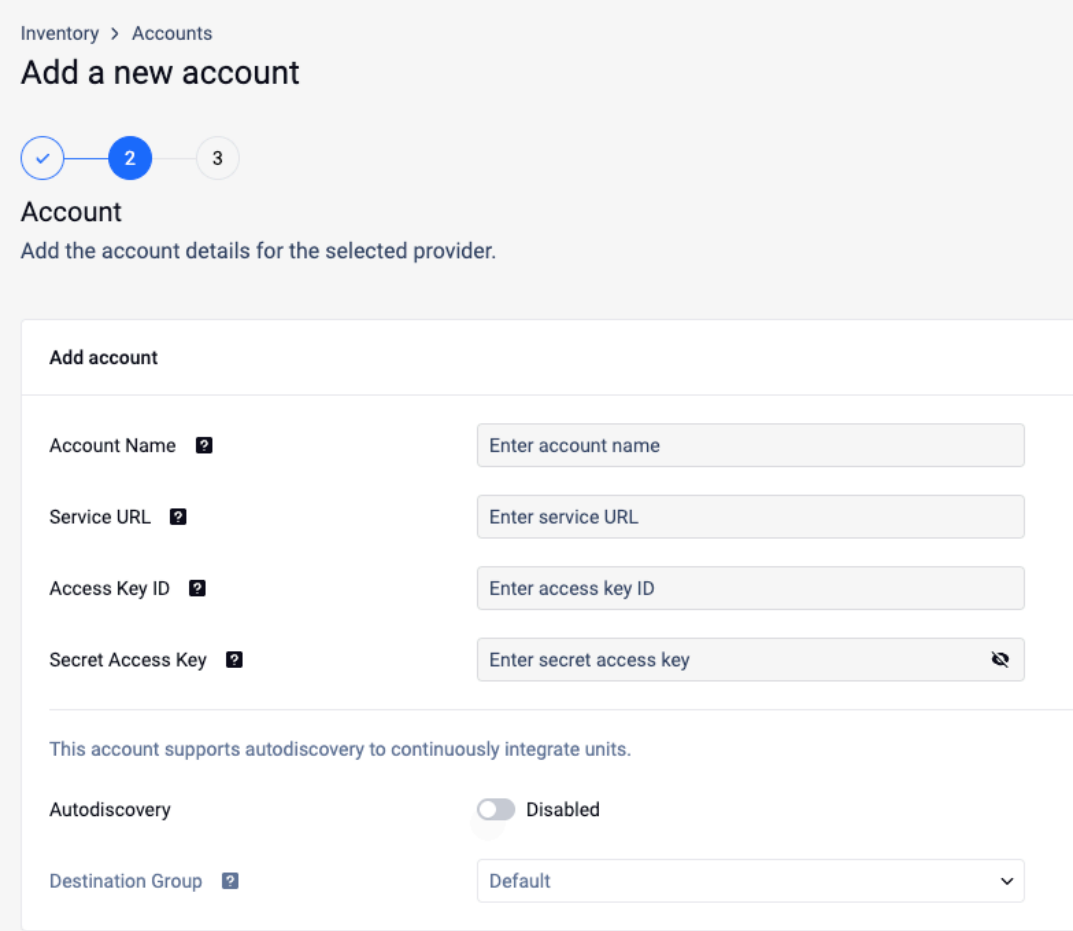
4. Obtaining the bucket name

1. **Go to Buckets:** Locate the “Buckets” section
2. **Identify Bucket:** Find your bucket in the list and note its name.

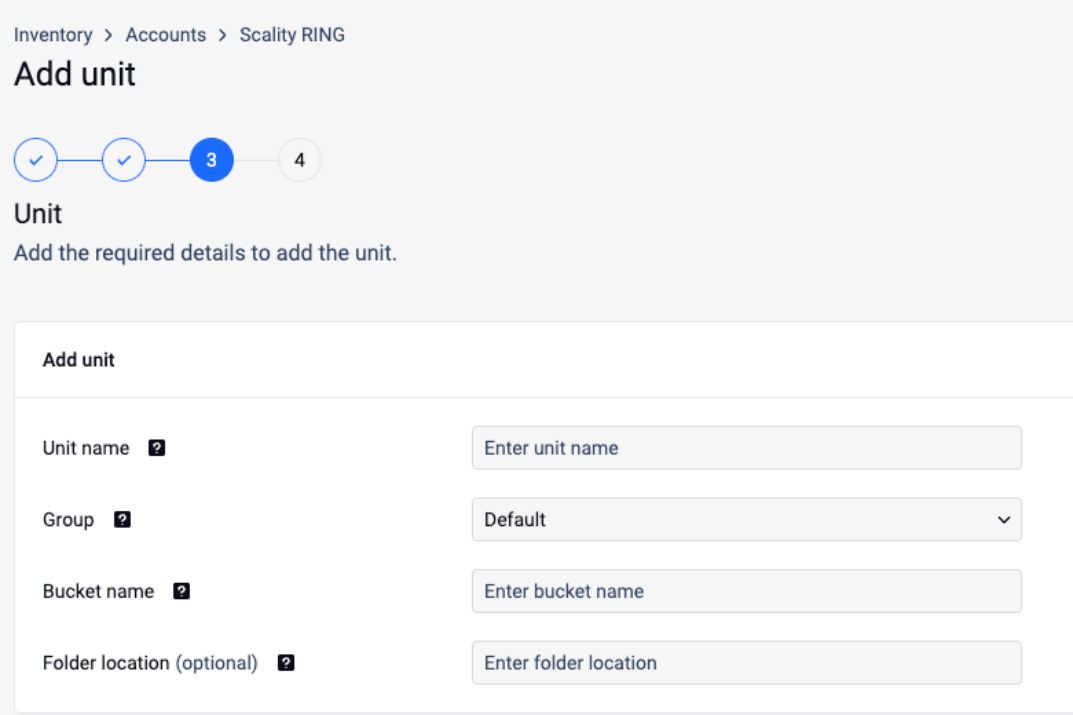
5. Adding Scality RING storage in MDSS

In order to integrate Scality RING with MetaDefender Storage Security please follow these steps:

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose **Scality RING** and click **Next**



2. Give your account a name so you can easily identify it later
3. Enter the Service URL for your Scality RING endpoint
4. Enter your **Access Key ID**. Scroll to the end of this page for more details.
5. Enter your **Secret Access Key.** Scroll to the end of this page for more details.
6. (Optional) Enable Autodiscovery
7. Once the account is successfully added, navigate to **Inventory**, click **Add New** > **Unit**, choose the previously created account and click **Next**



9. Give your unit an unique name
10. Choose Group
11. Enter a valid bucket name
12. Enter a Folder location (optional)
13.

Necessary policies

These are the necessary policies required in order to add a Scality RING Storage:

- `s3:GetObject`
- `s3:PutObject`
- `s3:GetObjectVersion`
- `s3:ListBucket`
- `s3:ListBucketVersions`
- `s3:GetObjectTagging`
- `s3:GetObjectVersionTagging`
- `s3:PutObjectTagging`
- `s3:PutObjectVersionTagging`
- `s3:DeleteObject`
- `s3:DeleteObjectVersion`

To be able to use client side encryption, additional AWS policies are required:

- `secretsmanager:GetSecretValue`

- `kms:DescribeKey`
- `kms:Decrypt`

To be able to use **server-side encryption with AWS Key Management Service (SSE-KMS)**, additional AWS policies are required:

- `kms:GenerateDataKey`
- `kms:DescribeKey`
- `kms:Decrypt`

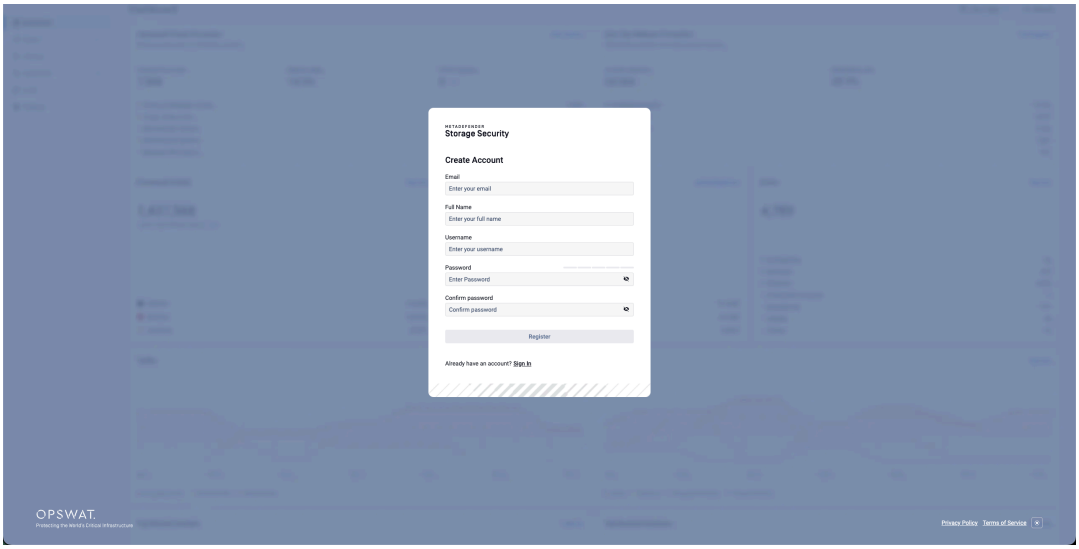
To be able to use **autodiscovery**, additional AWS policies are required:

- s3:ListAllMyBuckets

Operations

Onboarding

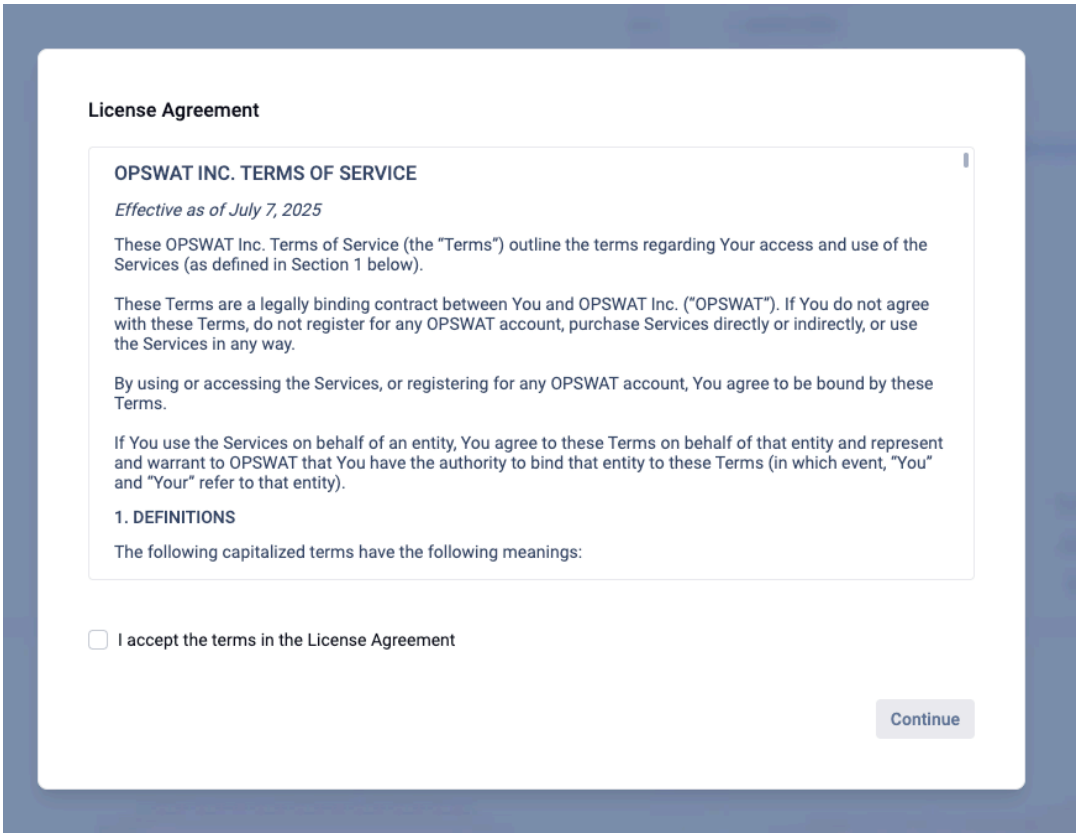
Once you've completed the installation you are ready to configure MetaDefender Storage Security. To begin, please navigate using your preffered browser to the MetaDefender Storage Security web interface which is located by default at http://:80. You will see the following screen:



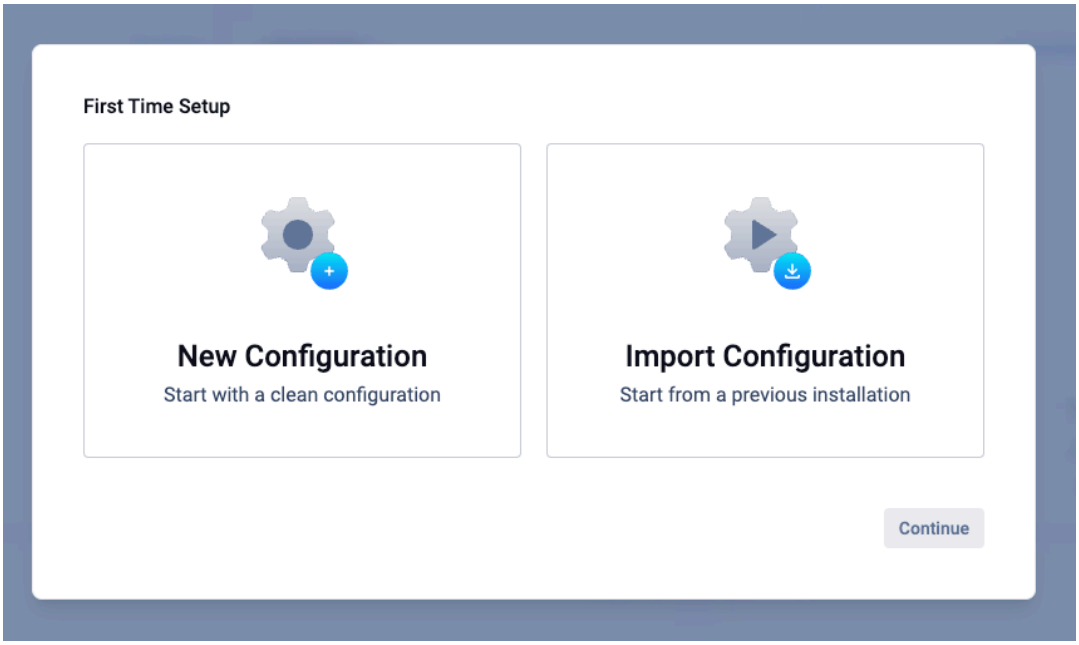
Create your account by filling the necessary information.

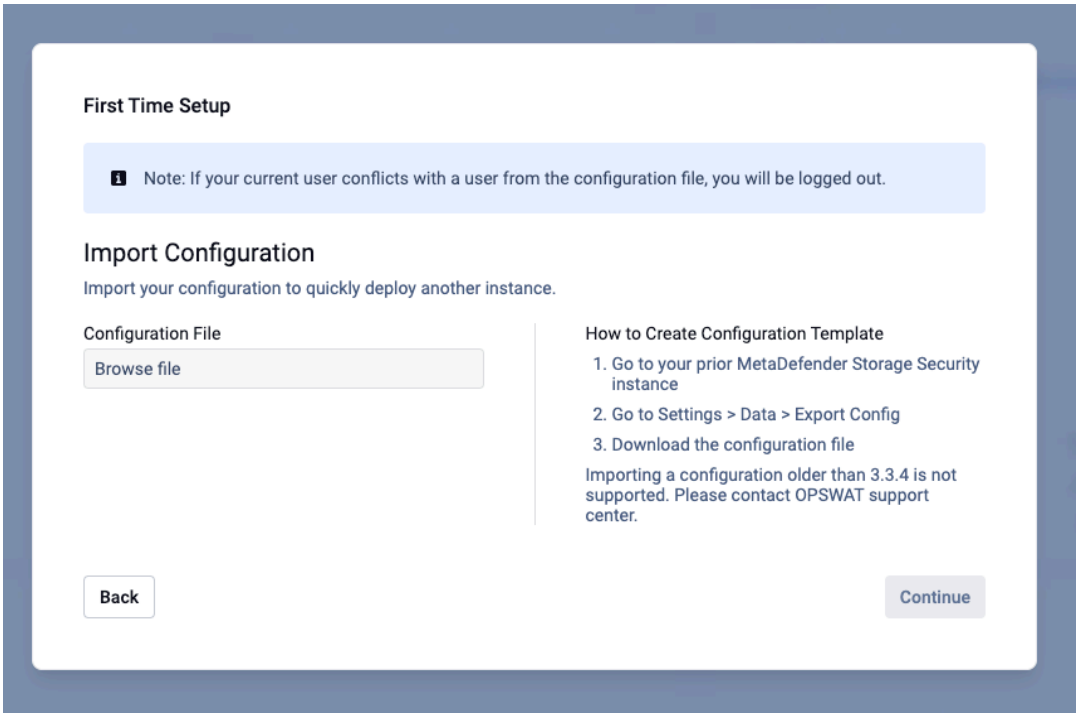
When you are ready, click **Register**. And then login using your newly created account. You are now ready to begin the initial configuration wizard.

Before starting the process, please accept our Licensing Terms of Service.

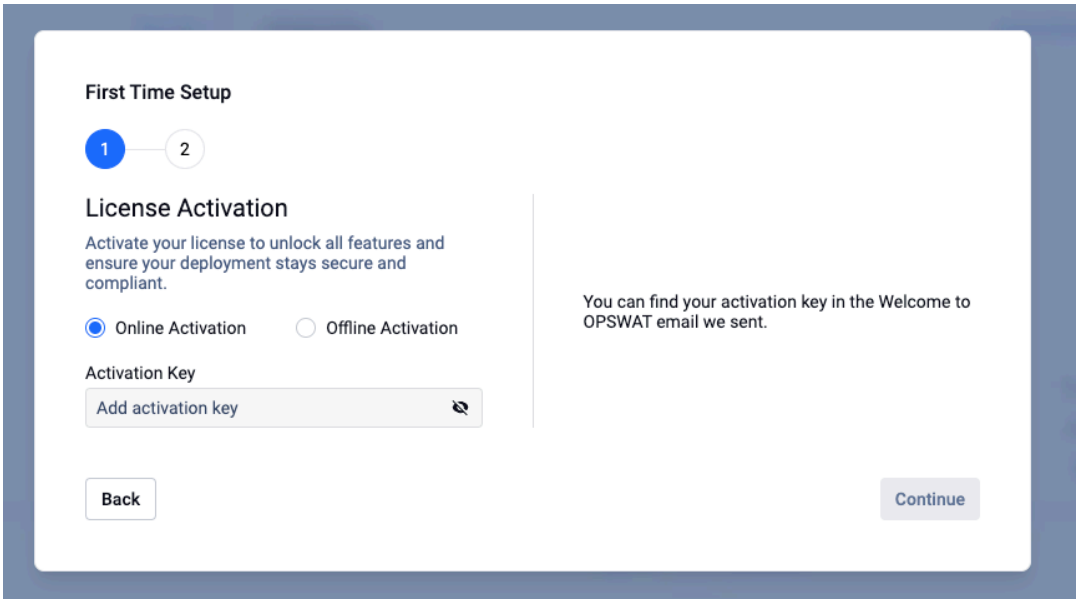


In this step, you can start with a new configuration, or import an existing configuration so that all your saved settings are restored.





The next step is to apply a license:



Choose between the following options:

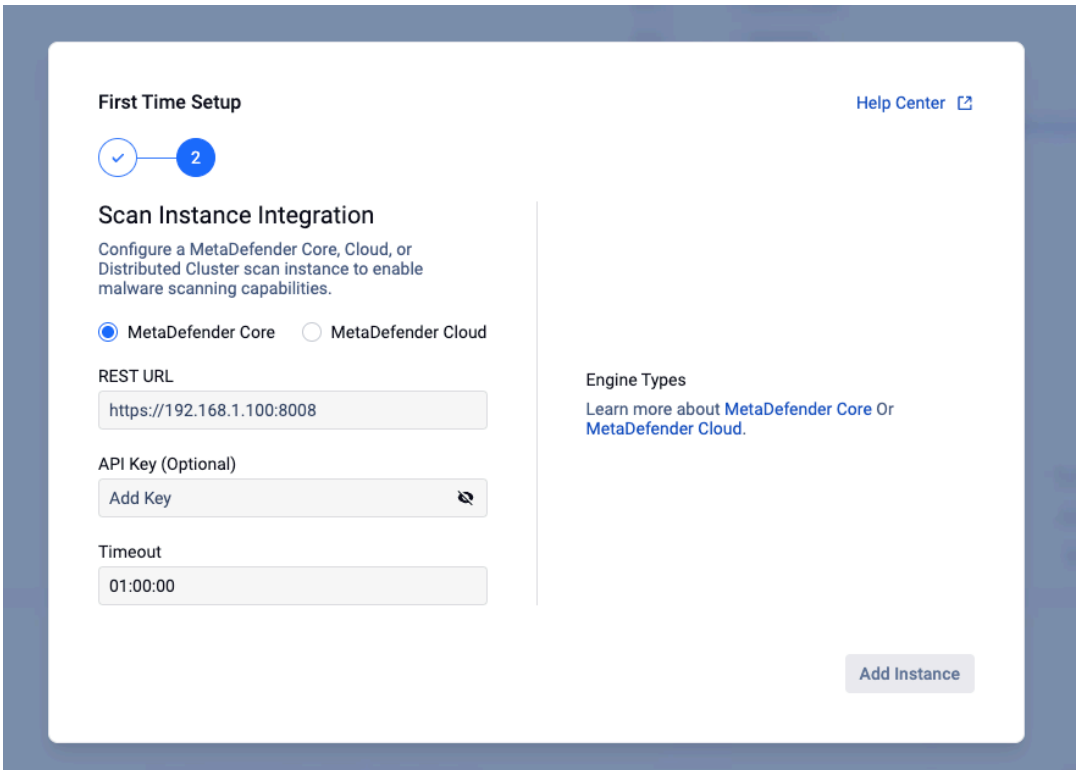
- Online activation: your license will be online validated against OPSWAT's activation server.
- Offline activation: you will upload a file that you previously generated from My OPSWAT
- Trial: if you don't have a license, please contact OPSWAT in order to receive one.

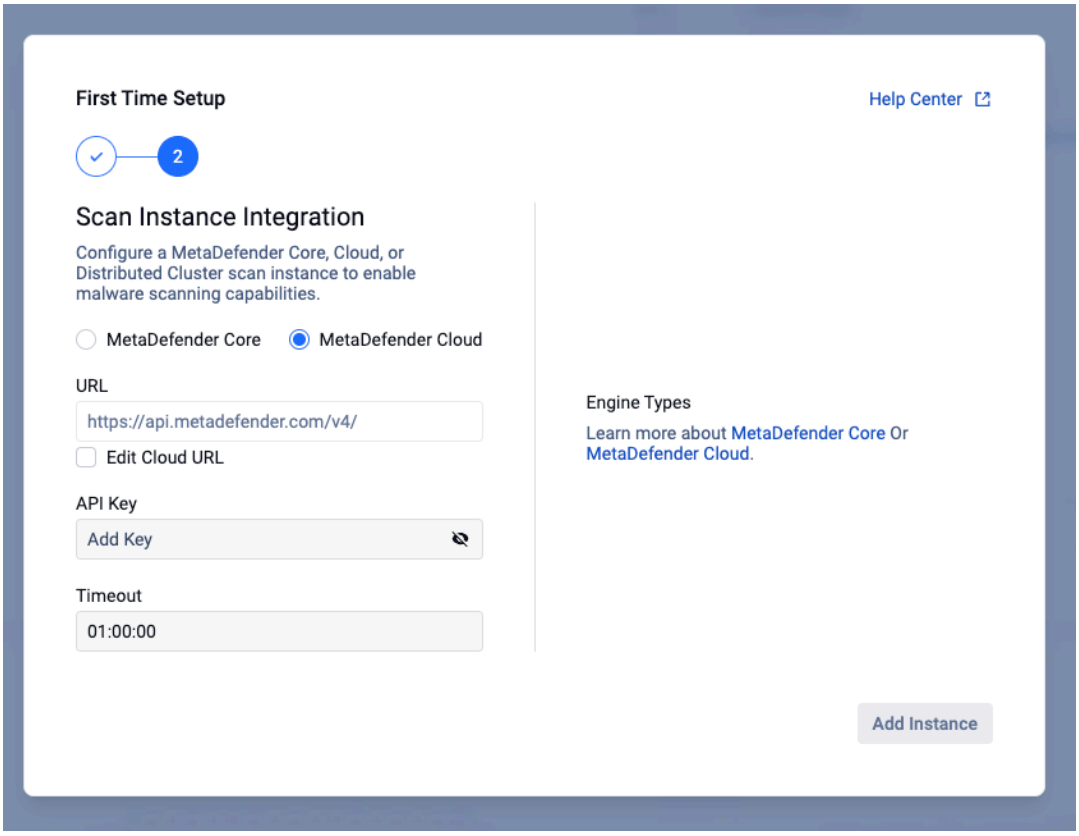
Once your instance of MetaDefender Storage Security is activated you can proceed to the next step.

In the next step, you'll be integrating MetaDefender Storage Security with MetaDefender Core or MetaDefender Cloud.

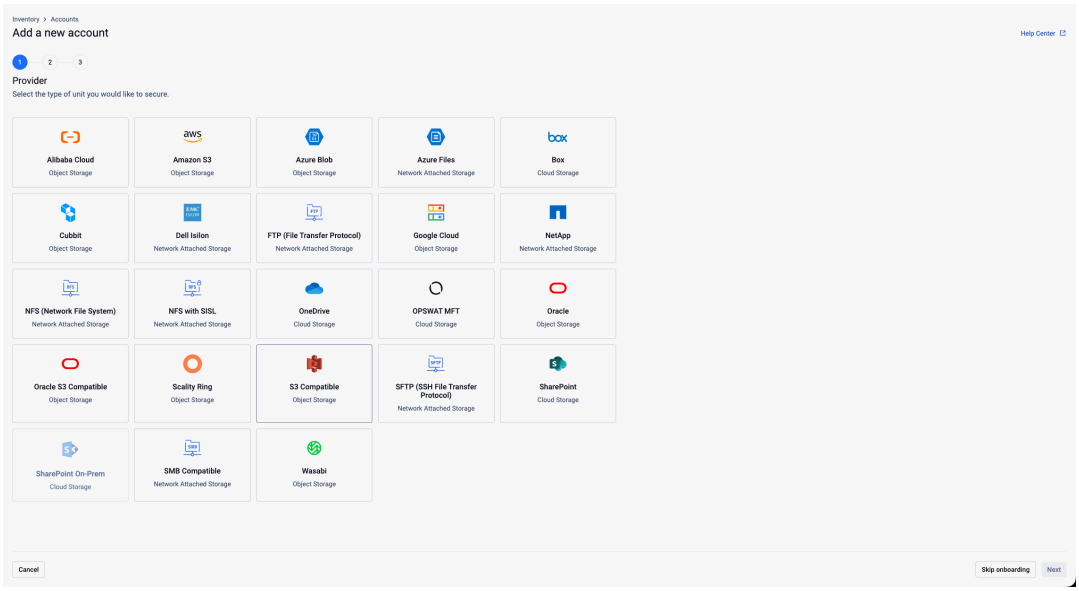
MetaDefender Core is an advanced malware protection and detection system that offers high-speed multi-scanning, deep content disarm and reconstruction (Deep CDR), and file-based vulnerability assessment technologies, preventing zero-day attacks and unknown threats as well as providing close to 100% known threat detection, without affecting performance. Click here to learn more about MetaDefender Core.

After choosing between MetaDefender Core or MetaDefender Cloud, you will be prompted with a form to complete the integration.





After MetaDefender Core/Cloud integration is done, you can start configuring an account or skipping the onboarding:



For a detailed tutorial on how to integrate a specific account and storage unit, please visit the Integrations section.

Browser Requirements for the Web Console

The following browsers are suggested to view the MetaDefender for Secure Storage Web Console:

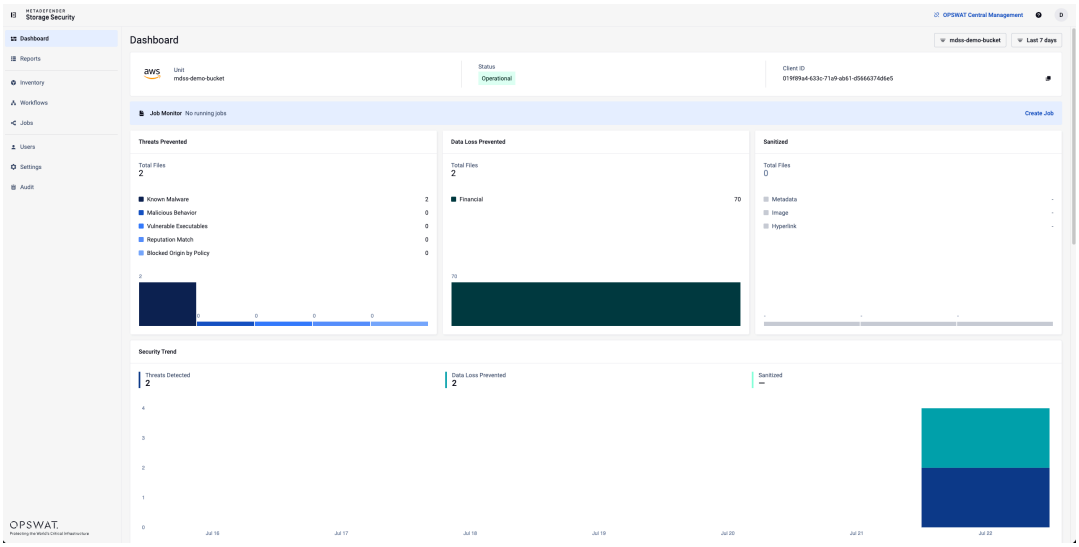
- Microsoft Edge
- Chrome
- Firefox
- Safari

Chrome, Firefox, Safari, and Edge browsers are tested with the latest available version at the time of release.

> **IMPORTANT**
>
> Note that Internet Explorer is no longer supported. Please upgrade to another supported browser (listed in the **Installation** section) to access the MetaDefender Storage Security web interface.

Dashboard

The MetaDefender Storage Security dashboard is the page where you can see an overview of your storage and find information about file processing and usage statistics as well as details about currently running processes or upcoming ones.



Key Features at a Glance

- The dashboard begins with a dynamic activity feed, displaying the most recent scan results, file processing statuses, and any detected threats. This immediate visibility allows you to quickly assess the current state of your storage security.
- Quick access to essential configuration settings, user management, and other key sections of MetaDefender Storage Security is available through the navigation menu on the right. Customize your setup, manage users, and fine-tune your security policies directly from this central location.
- Gain a quick understanding of your system's performance with key statistics displayed prominently in the center. See the total number of files processed, detected threats, and other critical metrics at a glance.
- Dive deeper into file processing details with a breakdown of processed, sanitized, and blocked files. Understand the efficiency of your security measures and identify any potential bottlenecks.
- Monitor threat detection effectiveness with clear visualizations of potentially malicious files, permanently unavailable files, and possible threats.
- Track system performance over time with an interactive graph. Analyze file processing trends and identify any performance anomalies.
- Access in-depth analysis of individual files, including threat assessment results and remediation actions. This granular view allows you to understand the specific threats detected and take appropriate action.

Why the Dashboard Matters

The MetaDefender Storage Security dashboard is designed to be your primary point of interaction with the system. It provides the essential information you need to:

- Stay informed about the real-time security status of your storage.
- Understand system efficiency and identify areas for optimization.
- Customize settings and policies to meet your specific security requirements.
- Quickly address detected threats and mitigate potential risks.

Dashboard Counters

Previously, the MetaDefender Storage Security dashboard provided a consolidated view of compliance and security metrics, offering insights into potential violations and threats. While these metrics remain critical for understanding your security posture, they are now presented in more detail within their respective technology sections: Multiscanning, Content Disarm and Reconstruction (CDR), and Data Loss Prevention (DLP).

Key Compliance and Security Metrics (Now Found in Dedicated Sections)

These categories, though no longer directly visible on the main dashboard, are still essential for comprehensive security analysis and are now located within the dedicated sections for each technology:

- Compliance Violations (DLP Section):
  - PCI-DSS (Payment Card Industry Data Security Standard): Tracks files containing Credit Card Numbers (CCN).
  - HIPAA (Health Insurance Portability and Accountability Act): Identifies files containing protected health information (e.g., full name, address, date of birth, phone number, email, SSN, bank account, passport).
  - GDPR (General Data Protection Regulation): Monitors files with personally identifiable information (PII) as defined by GDPR (e.g., full name, address, date of birth, phone number, email, bank account, passport).
- Security Alerts (Multiscanning and Vulnerability Assessment Sections):
  - Malware: Displays the number of potentially dangerous files detected by our multi-scanning technology.
  - Vulnerabilities: Shows the number of files with known vulnerabilities identified by our File-Based Vulnerability Assessment engine.
- PII Data (DLP Section) tracks files containing personally identifiable information (e.g., full name, bank account, passport, email, SSN).

Why These Metrics Still Matter

- Demonstrating adherence to industry regulations.
- Identifying potential malware and vulnerabilities.
- Protecting sensitive information from unauthorized access.

Where to Find These Metrics Now:

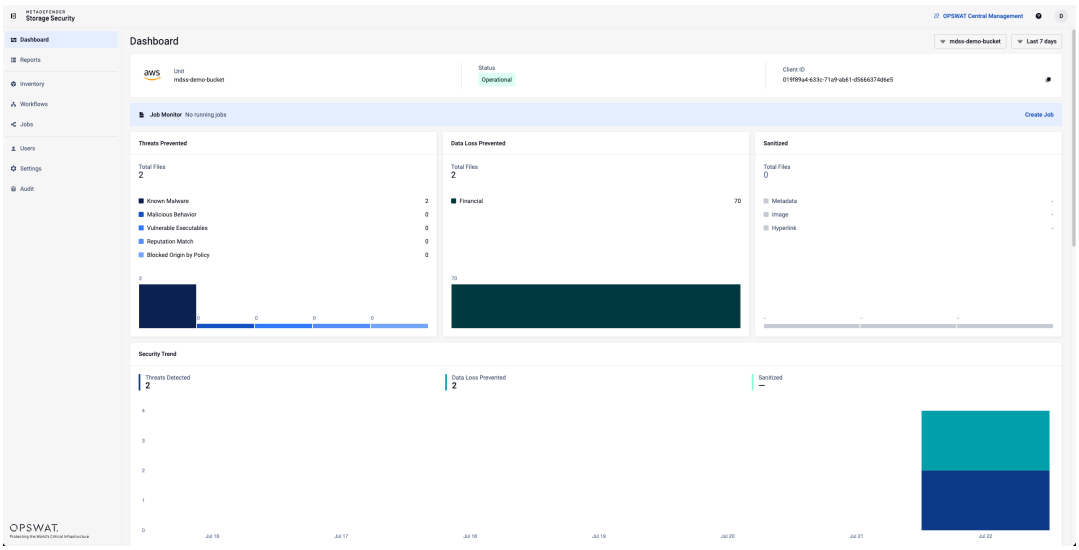
For detailed information on these metrics, please click on each dedicated sections for: Multiscanning, CDR and DLP.

> **Info**
>
> Please check the MetaDefender Core sample regular expression page for a list of rules that are being used to match full names, US addresses, birth dates, etc.

> **Info**
>
> The small arrows below each category shows you a percentage increase or decrease based on the previous scan.

Processing Overview

You can select to view data for the last week, month or even the last year. The height of the bar represents the volume of files processed by MetaDefender Storage Security. Hover over it to see more details:



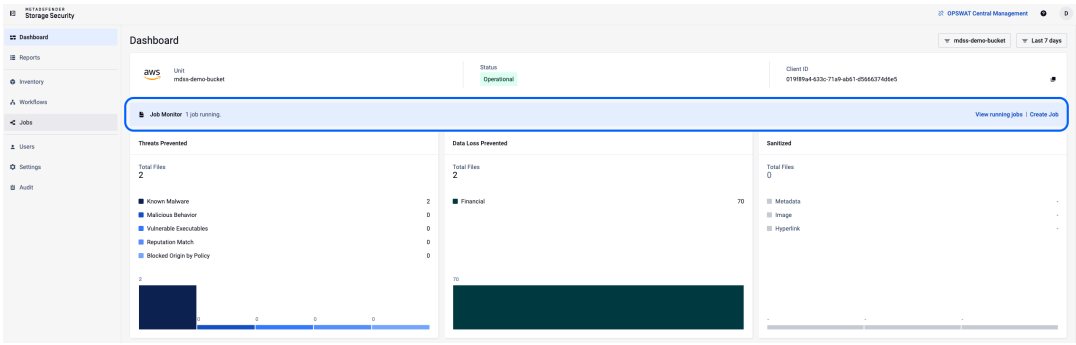
The processed quantity field represents the total amount of data processed with MetaDefender Storage Security. This quantity will exceed your actual storage data if you have triggered more than one process in the defined time unit (day or month). That is

because every time you process a file this quantity will be increased even if you've processed the same file multiple times.

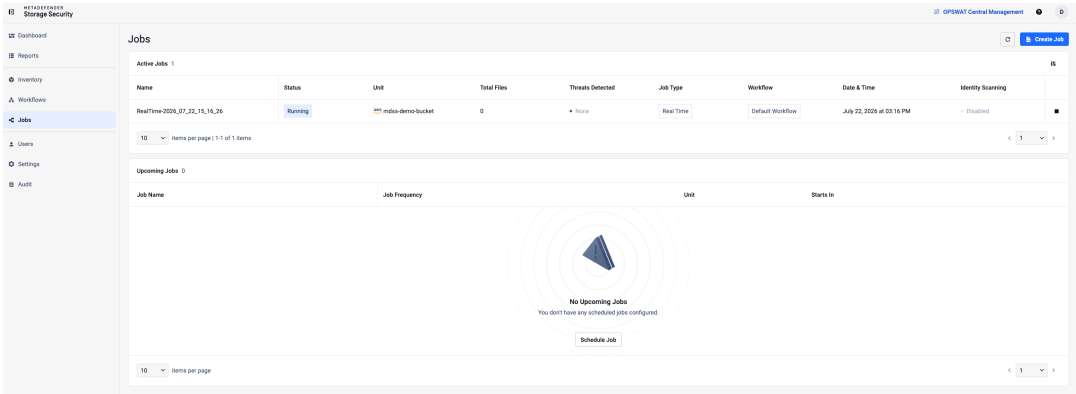
The number of suspicious files, as well as the counters on the top of the page, represent unique files. If the hash of the file has not changed, MetaDefender Storage Security will only count it once.

Active jobs

The dashboard displays an overview of all active jobs. To see them in detail, click **View Running Jobs**.

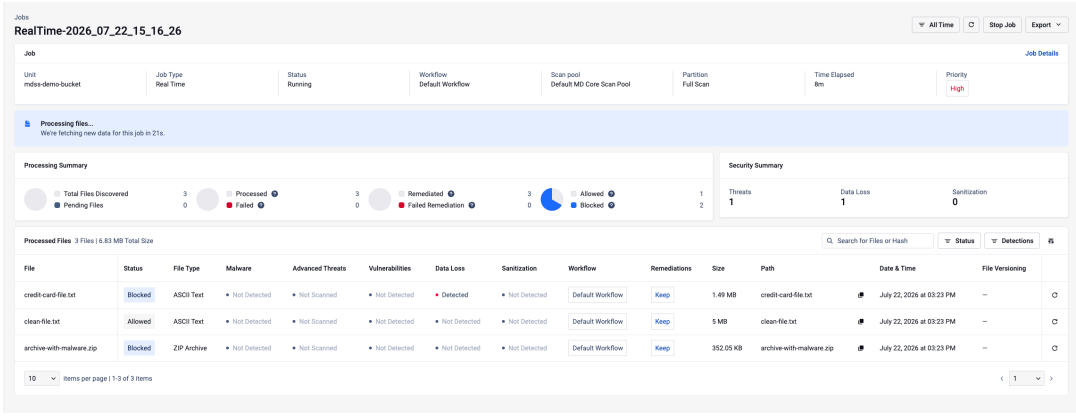


By clicking on **View Running Jobs**, you will be redirected to a page that contains the following information:



Ongoing Jobs details

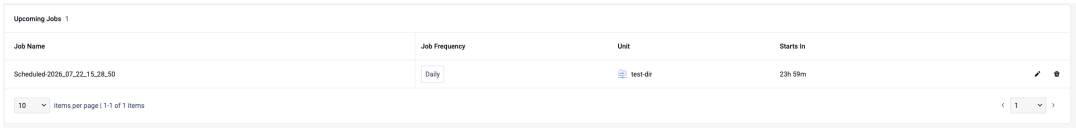
When clicking on the job you will be redirected to the ongoing job page:



Here you can find the information we previously mentioned as well as the time elapsed since this job was started. Click back if you want to go back to the dashboard.

Real time processing and upcoming processes

On the dashboard page, if you scroll down, you will find more information about real time processing and upcoming scheduled processes. This area, like the whole page, is updated in real time so you can always know what's happening.



> **Info**
>
> Real time processing requires a lot of internal communication. This could impact the performance of other scans

> **Warning**
>
> Starting real time processing and other type of scans in parallel with remediations that modify the files in storage could trigger a loop scanning.

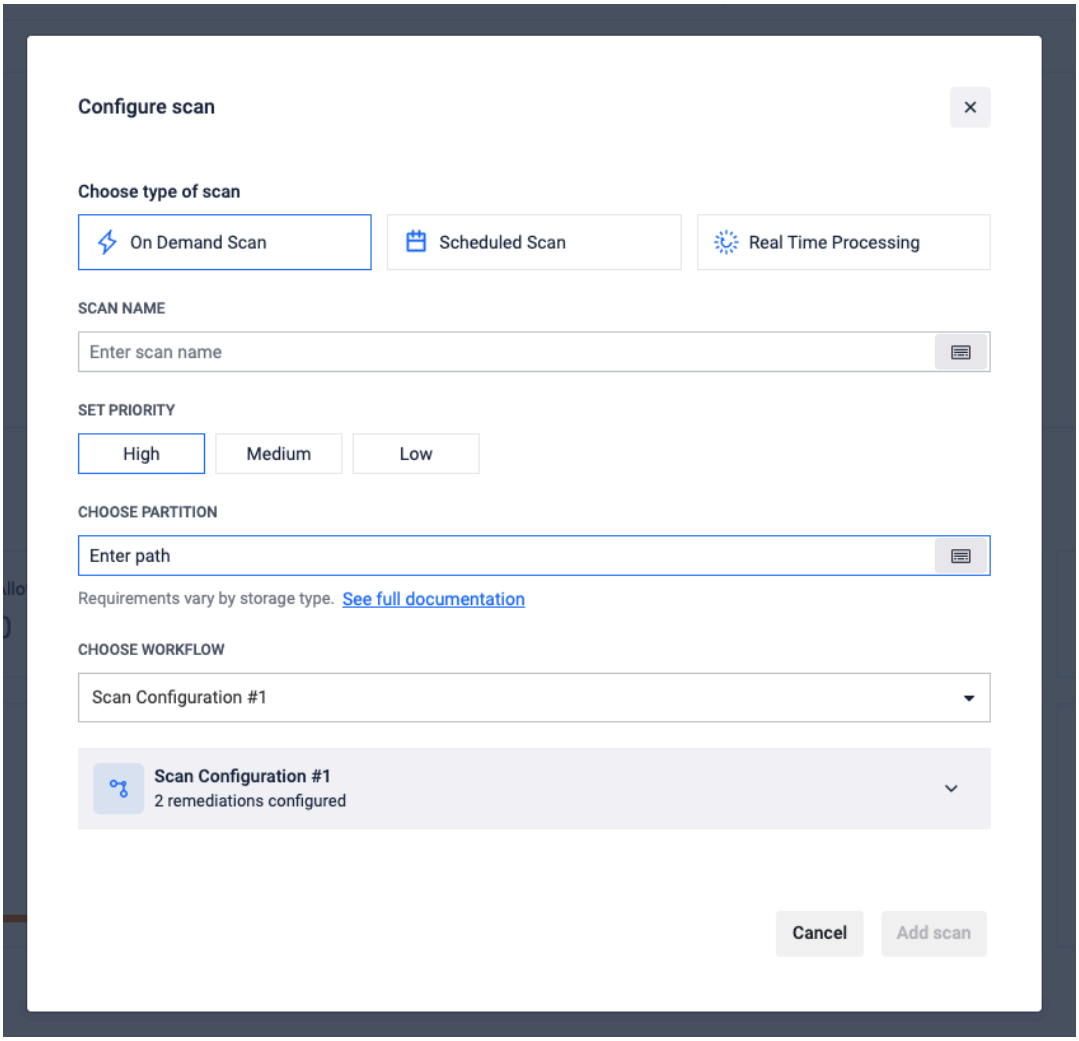
Scanning

This page explains how to configure scans in MetaDefender Storage Security (MDSS) and highlights its unique features.

Types of scanning

MetaDefender Storage Security offers three distinct types of scanning to meet your needs: **On-Demand Scan**, **Scheduled Scan**, and **Real-Time Processing**. Each type offers distinct configurations and capabilities to ensure effective file management.

1. On-Demand Scan

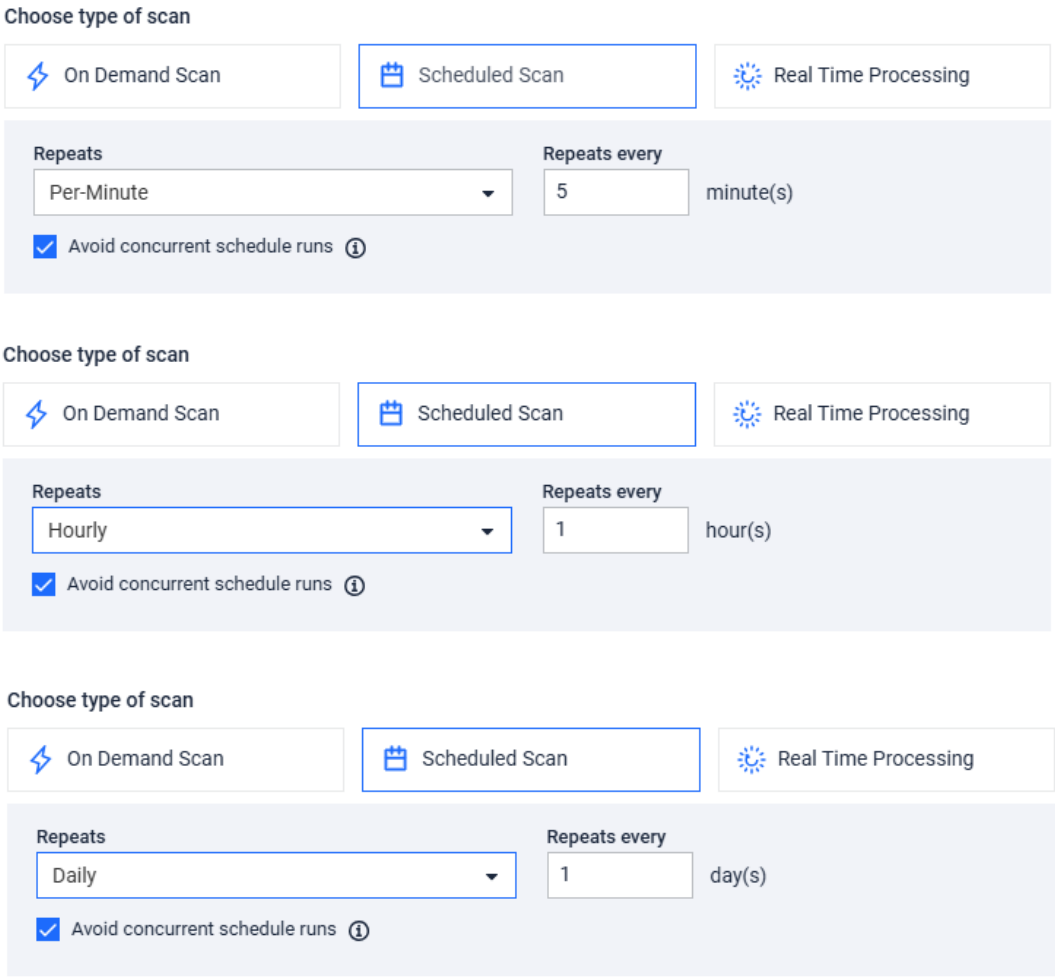


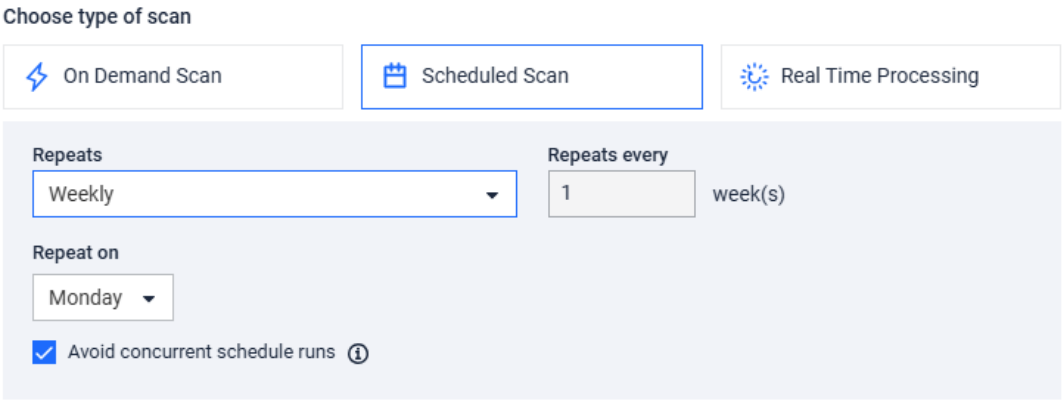
The On-Demand Scan allows you to initiate an instant scan of your selected storage. Here's how it works:

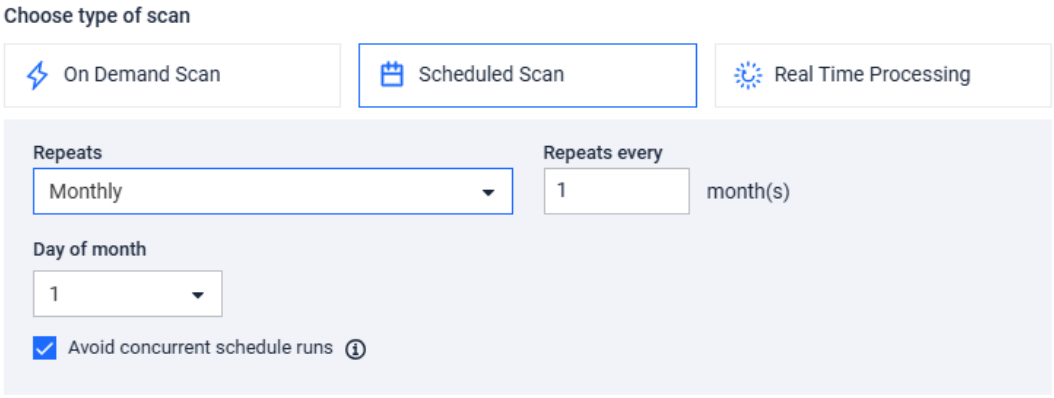
- **Input Scan Name**: Provide a name for your scan.

- **Set Priority**: Choose the priority level for the scan.
- **Choose partition** (optional): Specify a path from the storage root to scan instead of the integrated folder path.
- **Select Workflow**: Choose your preferred workflow (you can preview it's configuration).
- **Start the Scan**: Click "Add Scan" to begin processing all files in your storage integration immediately.

2. Scheduled Scan







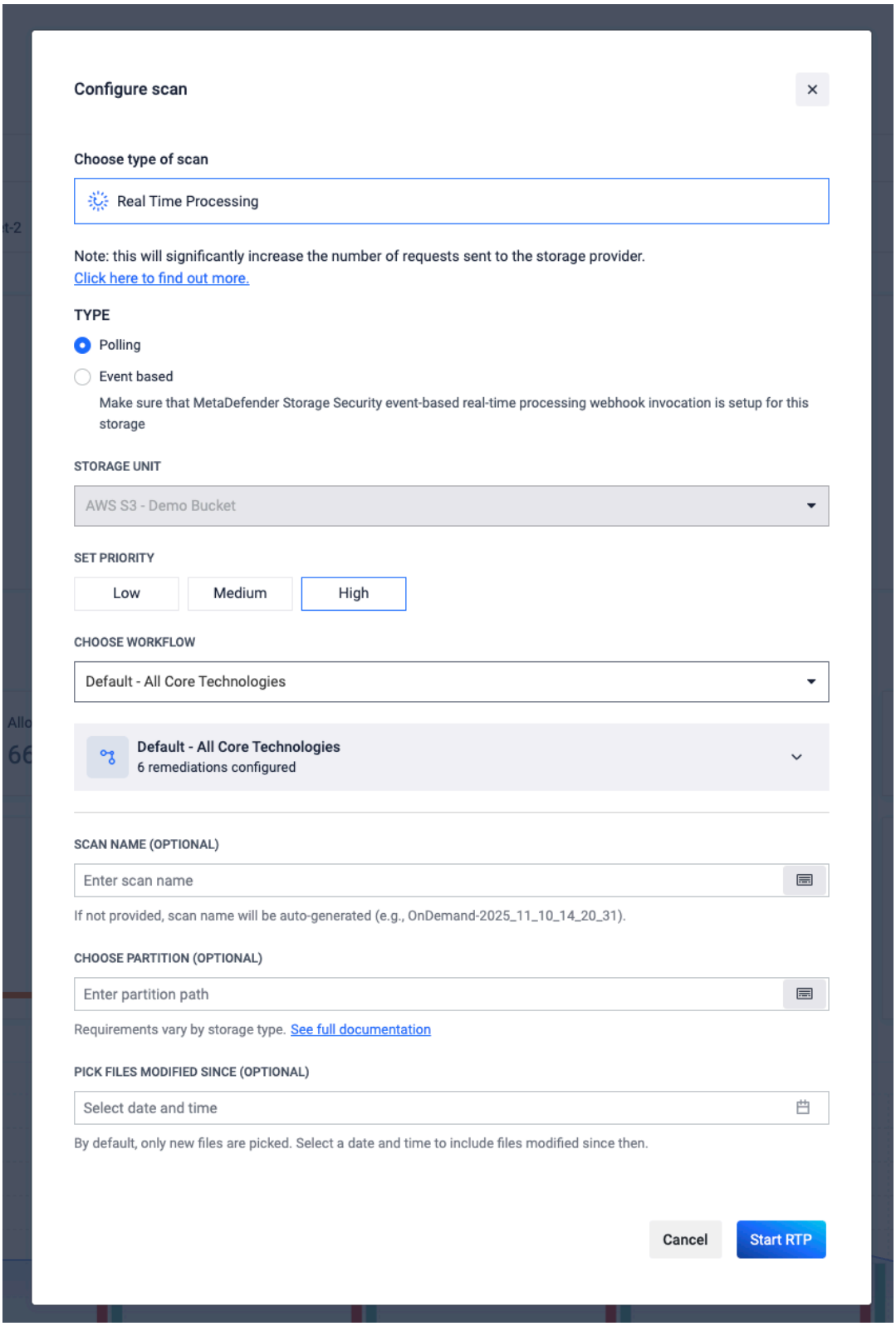
The Scheduled Scan enables you to set up recurring scans at specified intervals. You can configure the scan with the following options:

- **Input Scan Name**: Provide a name for your scan.
- **Set Priority**: Choose the priority level for the scan.
- **Choose partition** (optional): Specify a path from the storage root to scan instead of the integrated folder path.
- **Select Workflow**: Choose your preferred workflow.
- **Recurrence Options:** You can configure the scan to recur in one of four ways:
  - **Per-minute**: Set the scan to trigger every specified number of hours. The scheduling begins immediately upon clicking “Schedule Scan.”
  - **Daily**: Configure the scan to recur every specified number of days, starting from the moment you click “Schedule Scan.”
  - **Hourly**: Set the scan to trigger every specified number of hours. The scheduling begins immediately upon clicking “Schedule Scan.”
  - **Daily**: Configure the scan to recur every specified number of days, starting from the moment you click “Schedule Scan.”
  - **Weekly**: Set the scan to run every specified number of weeks, with the option to select the specific day of the week. The scan will commence at the time you click the button.

  - **Monthly**: Choose the frequency in months for the scan to repeat, along with the specific day of the month it will run.

- **Avoid concurrent schedule runs -** When enabled, the system will skip a scheduled scan run if the previous run is still in progress. This prevents multiple scans from running simultaneously.

3. Real-Time Processing



Real-Time Processing continuously monitors your storage integration for any new or modified files after the scan has started. This option includes:

- **Input Scan Name**: Provide a name for your scan.
- **Set Priority**: Choose the priority level for the scan.
- **Choose partition** (optional): Specify a path from the storage root to scan instead of the integrated folder path. Only one single Real-Time Processing scan is allowed per partition.
- **Choose monitoring date** (optional): By default only new files are picked starting from the moment real time processing is enabled. You can select a date from which the monitoring will start
- **Select Workflow**: Choose your preferred workflow.
- **Processing Methods**: You can select from two methods of Real-Time Processing:
  - **Polling Method**: This method checks for new files at specified intervals in the order of seconds.
  - **Event-Based Mode**: Available for certain storage integrations, this method allows you to configure a function that sends notifications to MetaDefender Storage Security via a generic webhook endpoint whenever a file is added to your storage For more information, check out this page Event-based handling.

Monitoring starting date

This feature will allow users to set a specific date in the past to start a real time processing.

> **Info**
>
> Retention configuration will alter this configuration on runtime to not process the same files after were removed from the report

Partition scanning

Partitions replace the integrated folder path for that scan only. Any folder path configured during storage integration will be ignored. The partition path takes full effect instead. Partitions are defined relative to the root of the storage source, and this behavior is consistent across all supported storage types.

For example, if your integrated storage is `my-bucket/` and you want to scan only `my-bucket/folder/subfolder/`, your partition should be set to `folder/subfolder/`.

> **Info**

> For **OPSWAT MFT, OneDrive, SharePoint, Box and SharePoint OnPrem**: If no specific users or document libraries are selected, the integration applies to **all users/doc libraries**, and **every one of them must contain the specified folder** or the partition scan will fail.

Priority scanning

MetaDefender Storage Security features a **Priority Scanning** system that allows you to assign priority levels to your scans, ensuring that critical tasks are processed more efficiently. This feature is available for all scan types, including On-Demand, Scheduled, and Real-Time Processing.

Priority Scanning Behavior:

While you can assign priority levels to any type of scan—On-Demand, Scheduled, or Real-Time Processing—it's important to understand that these priorities do not influence the execution order of scans. Here's how it works:

- **Same Priority, Same Type**: If you have two On-Demand scans with the same priority, they will run concurrently without any prioritization.
- **Different Priorities, Same Type**: When two scans of the same type are initiated with different priorities, the scan with the higher priority will be processed faster, but they will still run concurrently.
- **Different Types, Different Priorities**: For scans of different types, such as a Real-Time scan with Low priority and a Scheduled scan with High priority, the priority level will still be respected. The Scheduled scan will be processed faster than the Real-Time scan, regardless of their types.

Key Features of Priority Scanning:

- **Priority Levels**: You can choose from three priority options—**High**, **Medium**, and **Low**—when initiating any scan.
- **Efficient Processing**: Scans with higher priority levels are processed faster than those with lower priority levels, allowing you to manage your resources effectively and address urgent tasks promptly.
- **Concurrent Scanning**: Multiple scans can run simultaneously at the same priority level, ensuring flexibility in your scanning operations.

By leveraging the Priority Scanning feature, you can enhance your file management processes and ensure that your most critical scans are prioritized effectively. This not only improves overall performance but also enhances the efficiency of your scanning

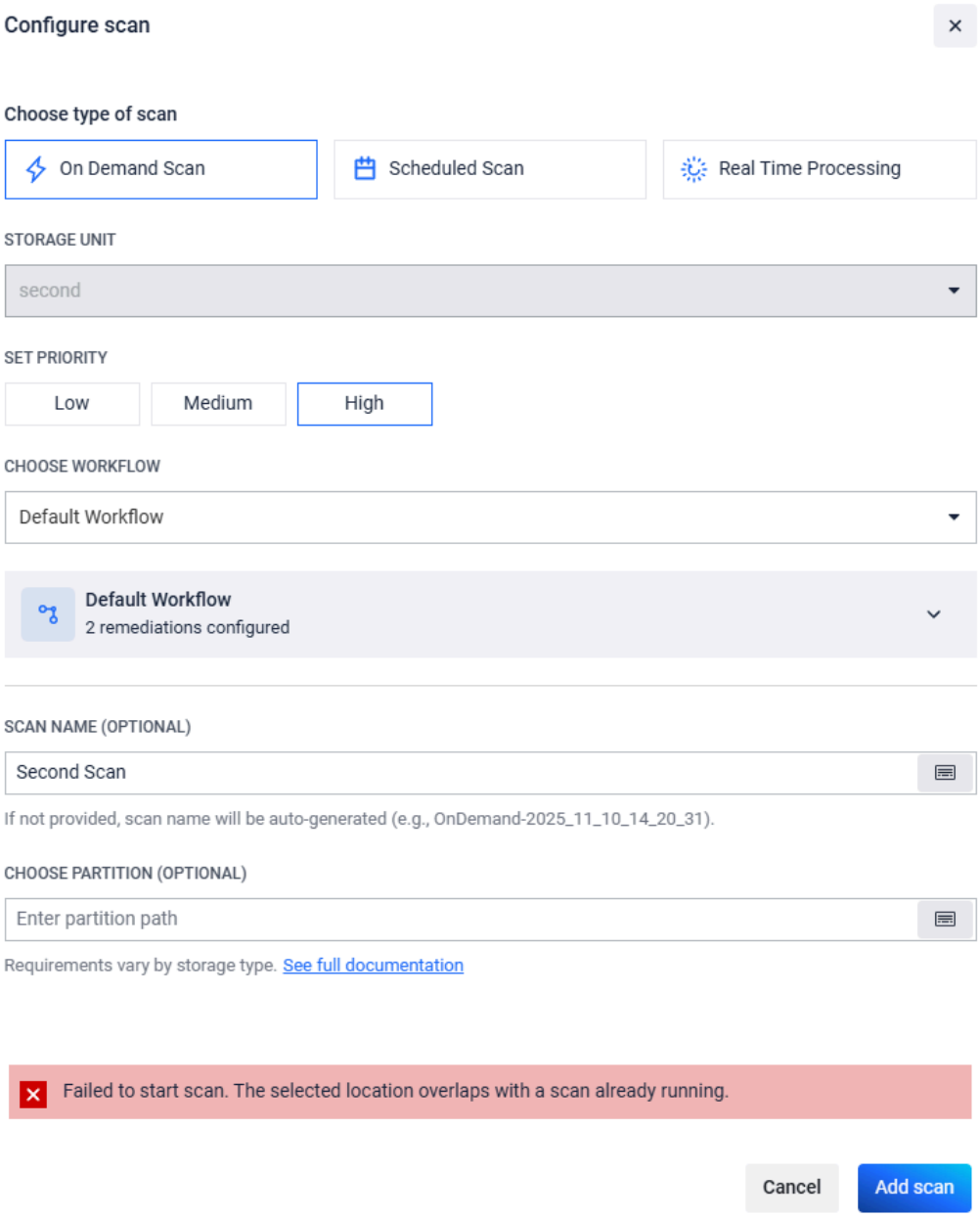
operations. For more ways to fine-tune performance and customize your deployment for speed, please check out this page: Performance Tuning Guide.

Scan Overlap Protection

MetaDefender Storage Security prevents two scans from running at the same time over the same physical storage location. The same location, regardless if it's a bucket, share, site, or folder, can be added to the product more than once (for example, a bucket root and a sub-folder within it, or the same location added twice with different credentials). Scanning both at once wastes resources by re-processing the same files and, when a tagging or remediation workflow is enabled, can create a repeating loop or unexpected behavior in general.

Behavior

This check takes place before starting any type of scan (**On-Demand**, **Real-Time Processing** or **Scheduled**) and will check if the storage location you are trying to start a scan for overlaps with a storage location from any other running scan. If it does overlap, you will receive this error:



Currently, this overlap detection compares locations within the same storage protocol type. It does not detect overlap between two different protocols pointing at the same physical storage (for example, the same NAS accessed over two different protocols). Path comparison is also case-sensitive, so locations that differ only in letter case will not be recognized as overlapping.

Skipping the scan overlap check

Some environments intentionally rely on running scans over overlapping locations and may need to preserve that behavior. For these cases, the overlap check can be disabled using the environment variable:

- **Variable**: SKIP_SCAN_OVERLAP_CHECK

- **Values**: *true* / *false*
- **Default**: *false*
- **Effect**: When *false*, overlap protection is active and overlapping scans are blocked. When *true*, the overlap check is skipped entirely and all scans are allowed to start regardless of overlap.

Password Protected Files and Encrypted Archives

Overview

Password-protected or **encrypted files**, such as Word documents, archives, and PDFs, cannot be scanned without providing the appropriate password to the scanner instance. To address this limitation, MetaDefender Storage Security offers several features, which we will detail below.

Password Storage

One option is to utilize the **Password Storage** feature from MetaDefender Core as explained here.

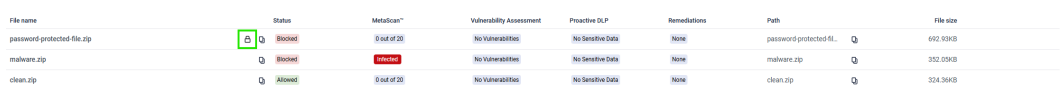
This feature enables you to define and manage a list of static passwords used for file decryption, making it particularly useful for automated workflow scenarios. When this feature is enabled, the scanner can decrypt and scan the file seamlessly, meaning it will not appear as password-protected in Storage Security.

Unlocking Encrypted Files within Storage Security using the GUI

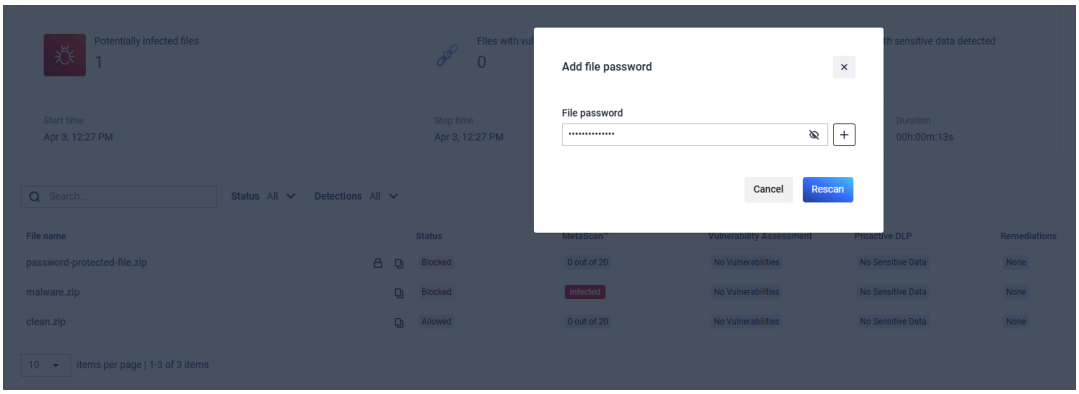
This feature empowers you to take control of sensitive data by enabling immediate scans of encrypted files. Simply provide the necessary passwords through the intuitive user interface and Storage Security will analyze the files, ensuring comprehensive protection against hidden threats.

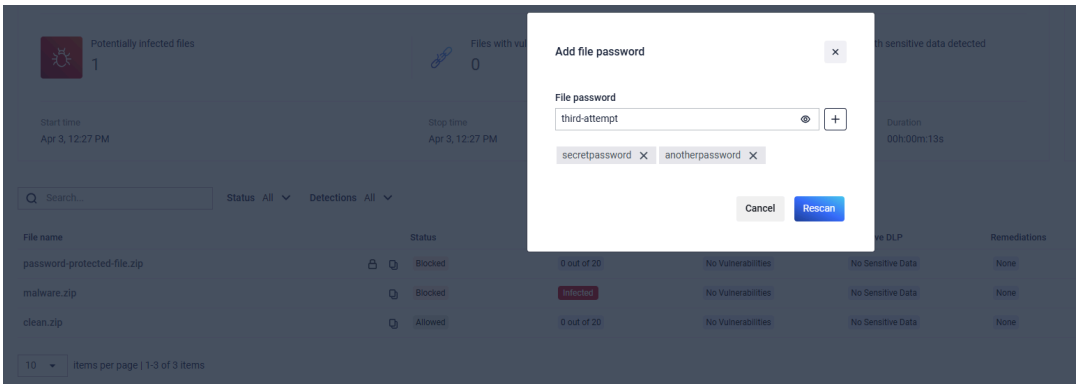
Unlocking Encrypted Files within MetaDefender Storage Security is simple **for users who prefer the GUI**:

1. Identify Encrypted Files: In the Reports page, look for files marked with a lock icon, indicating an “Encrypted” verdict. These files are automatically blocked to ensure security.



2. Provide Passwords: Click the lock icon to open a window where you can securely enter potential passwords. Need to try multiple passwords? Easily add more with the “plus” symbol. You can also visualize your entered passwords using the “eye” icon.
3. Initiate Rescan: Click the rescan icon to begin the decryption process. MDSS will attempt to decrypt the file using each password provided.





Currently, you can rescan one encrypted file at a time. We are actively working to enhance this feature, allowing you to provide passwords for multiple encrypted files simultaneously in future releases.

Unlocking Encrypted Files within Storage Security using the API

For **users who prefer the API integration**, MetaDefender Storage Security offers a flexible approach to On-Demand Scanning of encrypted archives:

The dedicated API endpoint allows you to initiate rescans of encrypted files programmatically. This means you can trigger a rescan at any time, regardless of whether you have passwords to provide.

If passwords are available, simply include them in your API request. MDSS will attempt decryption using the provided passwords and return the scan results. This enables

seamless integration of encrypted archive scanning into your automated workflows and security processes.

This API-driven approach provides advanced users with the control and flexibility they need to manage encrypted archives effectively within their existing systems. Full API documentation is available to guide your implementation.

Request Body:

```
{

        "fileId": "{string}",
        "scanId": "{string}",
        "storageProtocolType": "{int32}",
        "passwords": ["{array[string]...}"]
}
```

Reprocess files

MetaDefender Storage Security allows users to manually initiate a new scan for files. This feature provides control and flexibility, enabling rescans for the following scenarios:

- Investigating potential false positives
- Re-evaluating a file after anti-malware definition updates
- Re-processing a file after a policy configuration change
- Attempting to scan an encrypted file by providing the correct password(s)

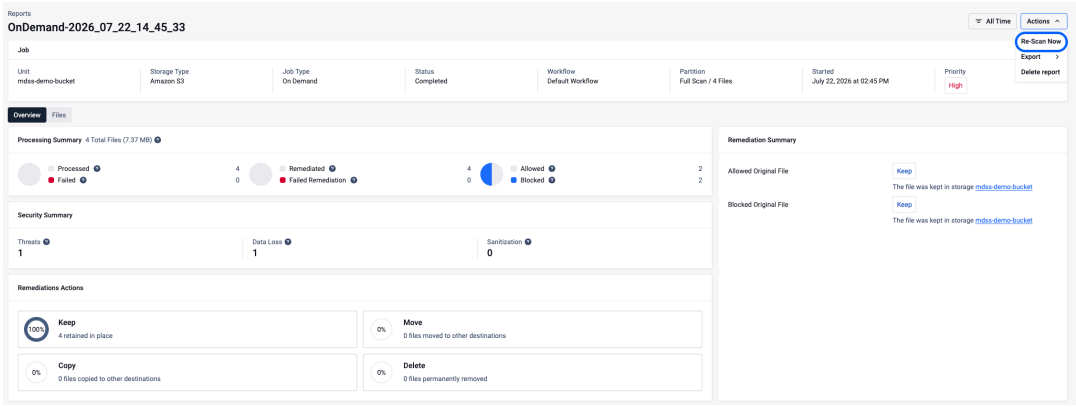
Files can be rescanned using either the **graphical user interface (GUI)** or the **Application Programming Interface (API)**.

> **Info**
>
> Reprocess actions have been designed for finished scans and are not applicable to running scans or active Real-Time Processing (RTP) scans.

Using the Report Reprocess Flow

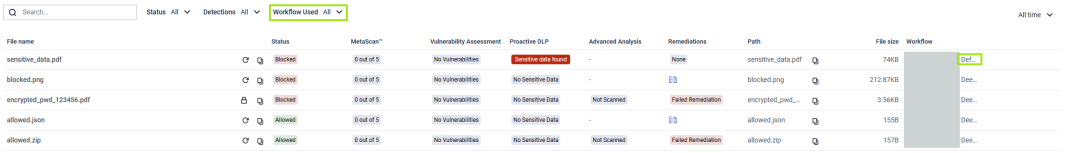
You can initiate a batch rescan directly from the MetaDefender Storage Security web interface:

1. From the scan report, go to **Actions -> Re-Scan Now**



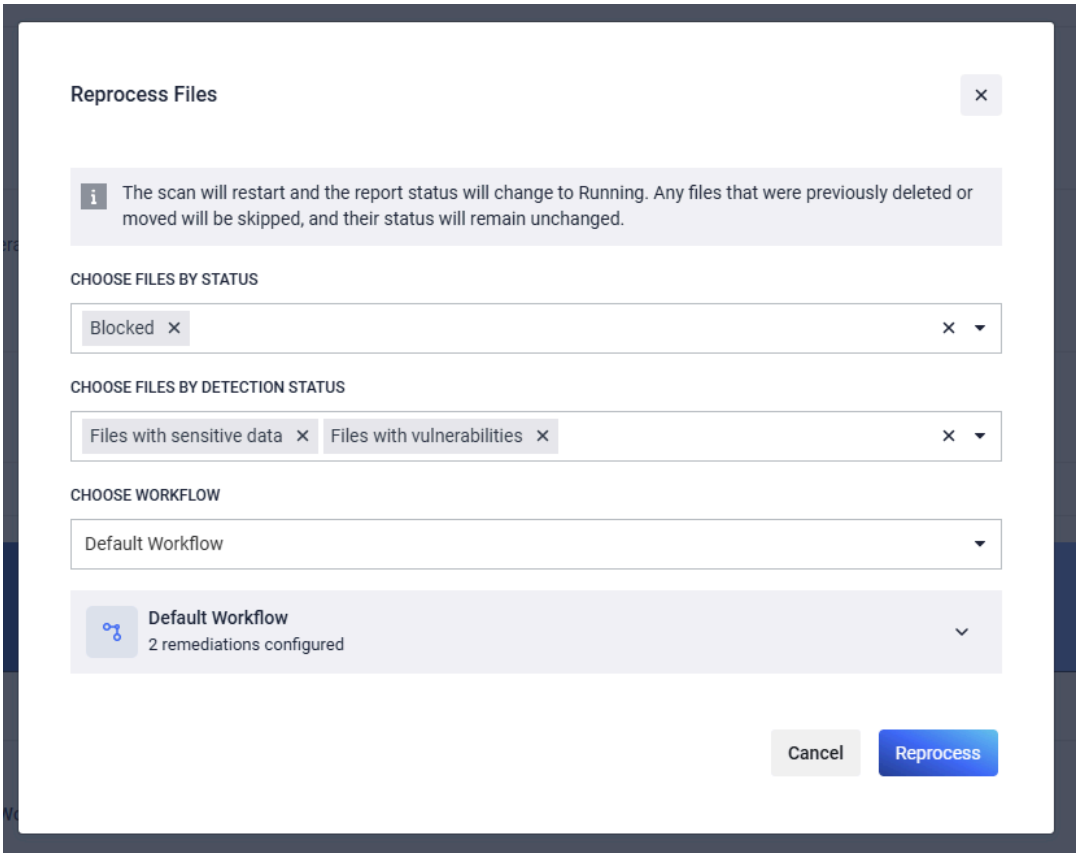
2. Configure the reprocess filters with the desired **Status**, **Detection**, and **Workflow** to be used by the rescan action.

3. After a batch rescan, the files of the report can be filtered by the Workflow and Date of the rescan action.



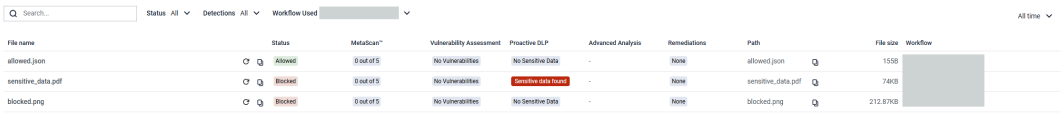
> **Info**
>
> The rescan process will filter for files that match both the STATUS and the DETECTION filters.
>
> Adding multiple DETECTION filters will scan will filter for files that match ALL DETECTIONS and not at least one of them.

Rescanning for **Blocked** files with **Sensitive Data - AND - Vulnerabilities** will not rescan any files because no files have triggered both **DLP** and **Vulnerability** detection flags in the scenario above.



Info

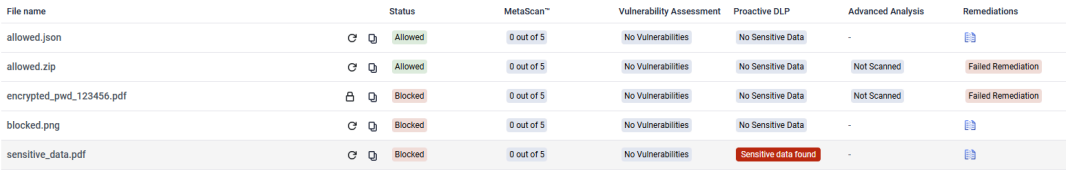
Reprocessing for Allowed and Blocked files will reprocess all files with Allowed and Blocked files but not any with Failed Remediations or other results.

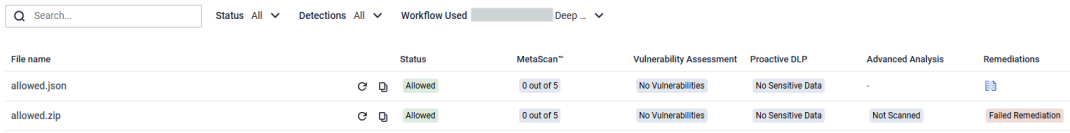


Warning

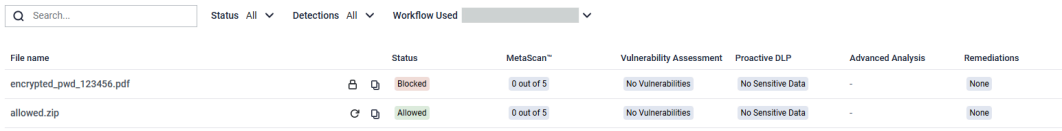
Allowed and Blocked file results are 1 tier above Failed, Failed Remediations, and Canceled file results.

Reprocessing for Allowed and Failed Remediations Files will NOT rescan Blocked files with Failed Remediation results because the rescan did not target Blocked files. Only Allowed files, with or without Failed Remediations, will be reprocessed.





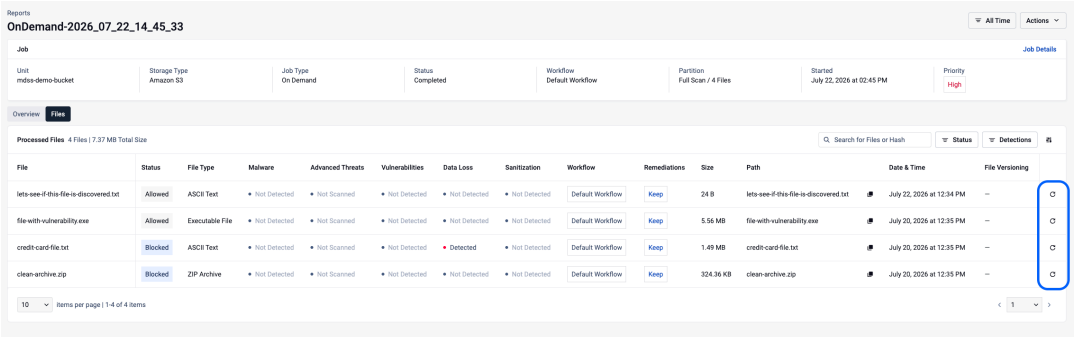
> **Info**
>
> Reprocessing for ONLY Failed Remediation files will reprocess both Allowed and Blocked files that have Failed Remediations



Using the Reports GUI

You can initiate a single file rescan directly from the MetaDefender Storage Security web interface:

1. Navigate to the **Active Scans** page or the **Reports** page.
2. Locate the specific file you wish to rescan within the list (status is indicated by icons or labels).
3. Click the **Reprocess File** button in the actions column for the blocked or failed file. For encrypted files, click the lock icon, enter the passwords, and click the rescan icon.
4. The file will be re-queued for processing using the currently applicable workflow.
5. Monitor the scan progress and final result on the Reports page.



Using the API

1. Files can be programmatically submitted for rescan using this API from the MetaDefender Storage Security API. This is useful for automation or integration with other systems.

2. You will need the unique identifier (`fileId`) of the blocked file you wish to rescan. This ID can typically be retrieved from scan reports or event logs associated with the initial block action.
3. **Missing Information :** the specific endpoint for triggering a rescan of any blocked file in MDSS based on `fileId` should be documented here / link to it
4. Send an API request (POST) to the determined endpoint. The request body should be in JSON format (below)
5. The API should return a response indicating whether the rescan request was successfully submitted (a success message, perhaps a new `data_id` or task ID for tracking or whatever it is). The actual scan results will be available asynchronously and can be retrieved via subsequent API calls or viewed in the GUI.

For standard rescan

```
{
        "fileId": "{string}",
        "scanId": "{string}",
        "storageProtocolType": "{int32}",
}
```

For encrypted blocked files

```
{
        "fileId": "{string}",
        "scanId": "{string}",
        "storageProtocolType": "{int32}",
        "passwords": ["password_1","password_2"]
}
```

Full API documentation is available to guide your implementation.

Identity Scanning

Identity Scanning speeds up file processing by identifying files that have already been scanned in previous runs. When MDSS (MetaDefender Storage Security) detects that a file hasn't changed since the last scan, it reuses the existing scan results instead of re-scanning the file.

This feature is useful for backup datasets where many files stay the same across multiple scan cycles, and for recurring scans of the same dataset.

Benefits

- Improves performance for backup datasets and recurring scans
- Reduces scanning time by skipping unchanged files
- Keeps complete reporting while avoiding redundant work

- Maintains security with customizable rescan intervals

How it works

Identity Scanning uses file metadata to check whether files have been scanned before. Here's what happens:

1. Mount backup repositories as NFS/SMB file shares to MDSS
2. Enable Identity Scanning in your workflow and set a scan interval (for example, 3 days)
3. First scan: All files are scanned (no stored results exist yet)
4. Later scans: Only new or changed files are scanned; unchanged files use cached results
5. You get better performance while keeping full security coverage

Rescan interval

To maintain security, Identity Scanning includes a rescan interval. Even files that haven't changed get rescanned after the time period you configure. This ensures nothing slips through the cracks.

> **IMPORTANT**
>
> Deep CDR and Real-Time Processing are currently not supported by Identity Scanning.

Things to know

- Identity Scanning works with On-Demand and Scheduled scans only
- You cannot start Real-Time Processing with Identity Scanning enabled
- The feature checks file metadata to identify files accurately

Configuration

Setting up identity scanning

Configure Identity Scanning from the Discovery tile in your workflow:

1. Create or edit a Standard Workflow with scanning enabled
2. Turn on "Enable Identity Scanning"
3. Set the "Scan Interval (Days)" value
4. Save your changes

Scan interval

Choose how often MDSS will rescan files to check for changes. For example, if you enter "3", MDSS will scan every 3 days to verify files that haven't changed.

Reading scan results

What happens during scans

First Scan

- All files in storage are scanned
- No identity matches exist in the database yet
- All files show as "Scanned"

Later Scans

- New or changed files are scanned
- Unchanged files use stored results and show as "Stored"
- Reports show all files, whether scanned or stored

What the labels mean

Identity: Scanned

- Files that were scanned in this run
- Includes new files and files changed since the last scan

Identity: Stored

- Files that were skipped during scanning
- These files haven't changed since the last scan
- Results come from stored data

Using Reports

Dashboard

When a scan with Identity Scanning runs, you'll see an "Identity Scan" indicator on the scan card.

Report list

Reports that used Identity Scanning show an icon next to the report name.

Report details

Inside a report, you'll find:

Summary Cards

- Identity Scanned - the number of files scanned in this run
- Identity Stored - the number of files that used stored results

Filters and Columns

- Filter by "Identity Status" to see only "Scanned" or "Stored" files
- Each file shows its status in the table

Active Scans

- See Identity Scanned and Identity Stored counts in real-time
- Filter the file table by Identity Status while scans run

Special cases

Files that are always scanned

Even with Identity Scanning turned on, some files always get scanned:

- Manual reprocessing - if you manually reprocess a file, it gets scanned regardless of identity status
- Different Scan Instance - if a file was scanned with a Scan Instance that's not in your current Workflow's Scan Pool, it gets scanned again

After the interval expires

When you run a scan after your configured interval (like 3 days), all files get scanned again even if they haven't changed. This keeps your security coverage complete.

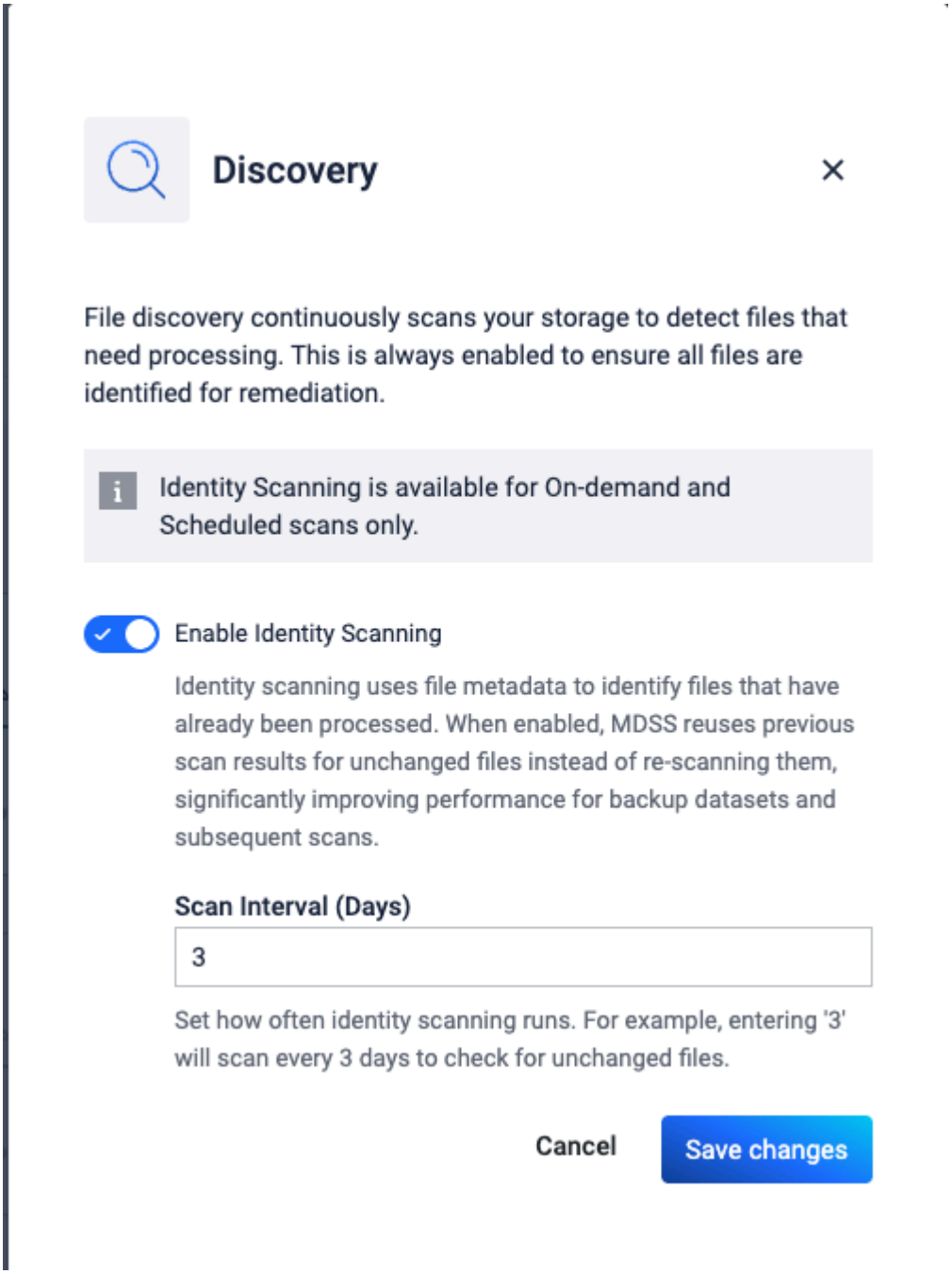
Example workflow

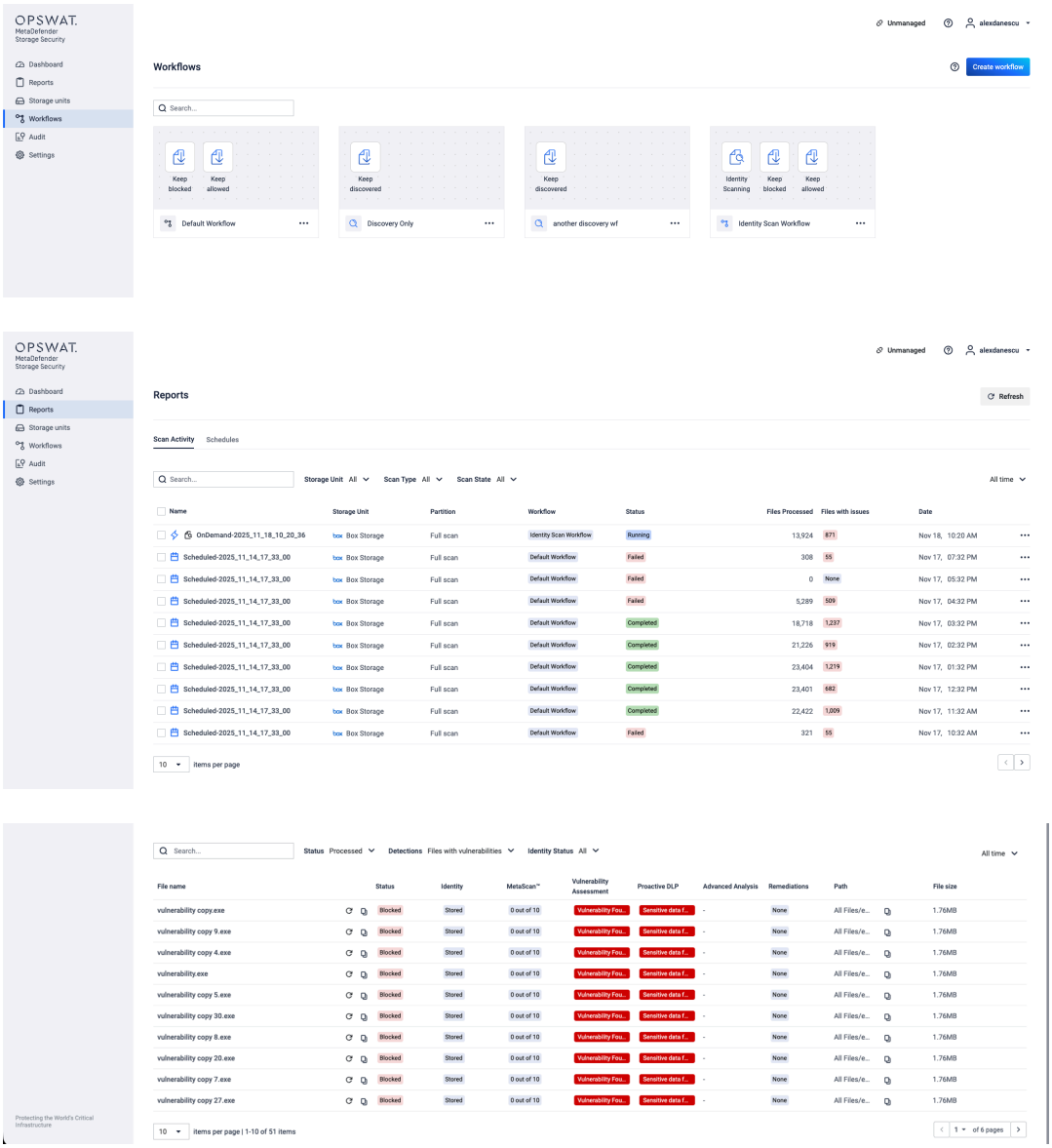
Here's how Identity Scanning works in practice:

1. Create a workflow with Identity Scanning turned on and a 3-day interval
2. Run the first scan
    - All files are scanned
    - Report shows all files as "Identity: Scanned"
3. Run a second scan (within 3 days)

- Unchanged files show as “Stored”
- Only new or changed files actually get scanned

4. Run a third scan (after 3 days)
   - All files get scanned again, whether they changed or not
   - This keeps your security coverage up to date





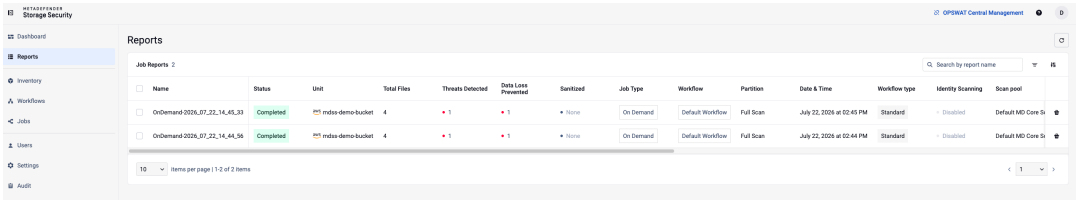
Tips for Using Identity Scanning

- Identity Scanning works well for **backup datasets** where most files stay the same
- Pick a scan interval that balances security needs with performance
- Look at Identity Scanned vs. Identity Stored numbers to see how your files change over time
- Use Identity Scanning with scheduled scans for automated security coverage

Reports

After a process is finished a report is produced for you by MetaDefender Storage Security. This report has information about the overall scan of your storage (i.e. the number of detected issues) as well as individual file analysis details.

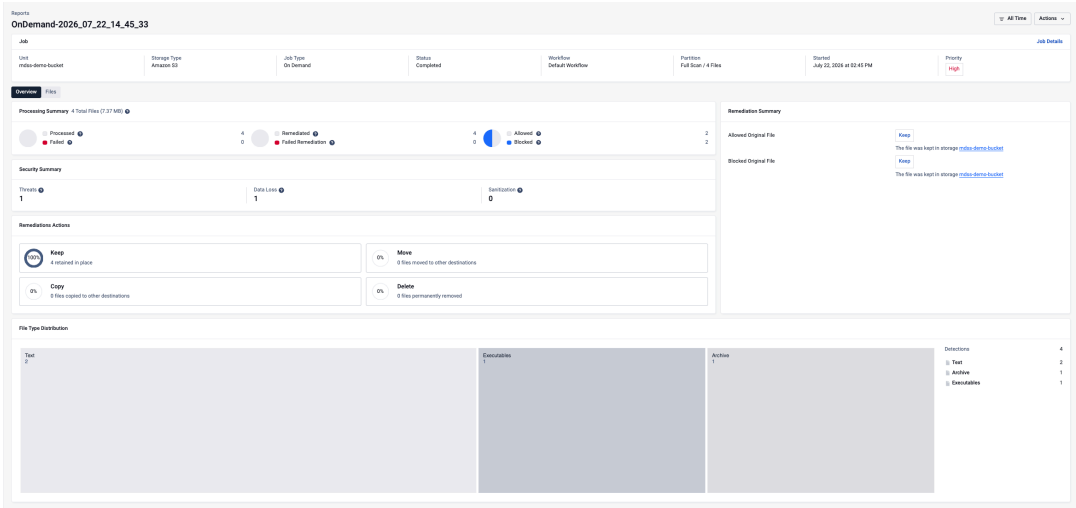
You can access all the reports generated for a specific storage by navigating from the left menu:



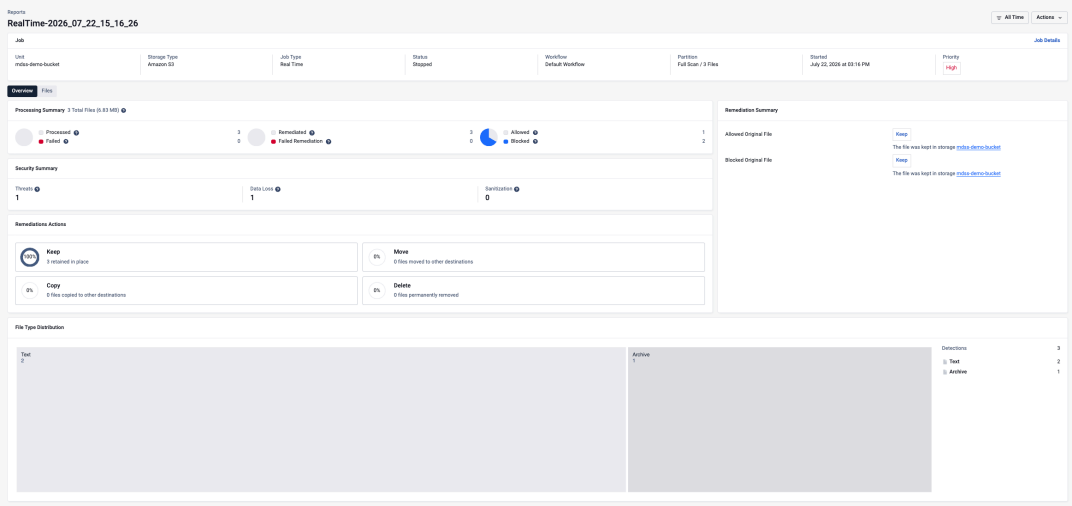
Here are some actions that are available for you:

- Search for a report or filters report based on type or a specific time range
- Click on a report in order to see more details and you will be redirected to the report page
- Delete a report using the three dots menu from the last column

Scan Report



Real Time Processing Report



Info

Cleaning a RTP report will not reprocess existing files.

In case one of the following, on cleaning the RTP report only the files that were not processed will be handled.

DISCOVERY_SERVICE _SMB_RTP_HANDLING=1

DISCOVERY_SERVICE_SFTP_RTP_HANDLING=1

DISCOVERY_SERVICE_NFS_RTP_HANDLING=1

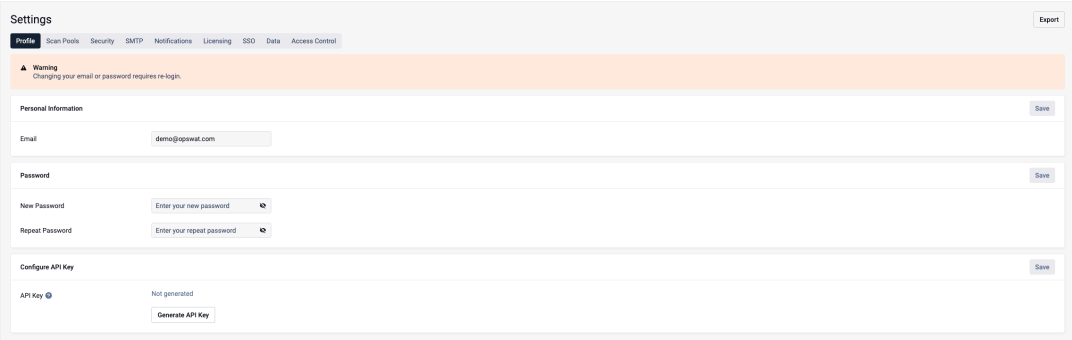
DISCOVERY_SERVICE_FTP_RTP_HANDLING=1

Settings

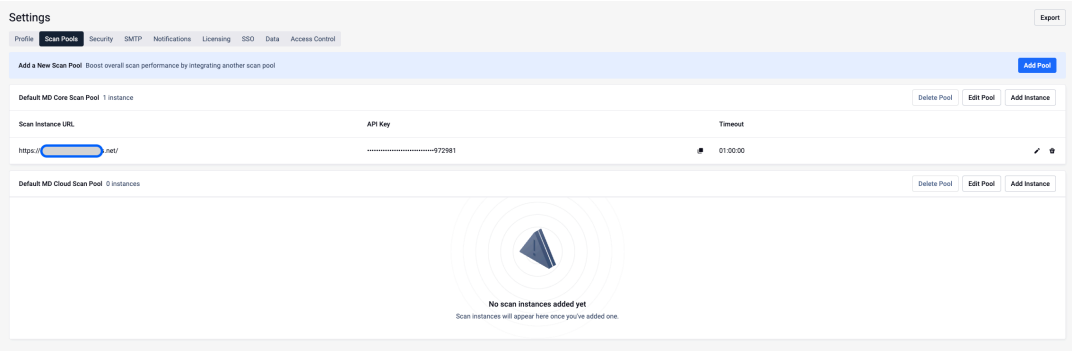
The “Settings” section of MetaDefender Storage Security provides a centralized location for configuring various aspects of the system to meet your specific security and operational needs.

Each tab provides a clear interface for configuring the corresponding settings. Make sure to review the specific documentation for each tab to fully understand the available options and their impact on your system.

The Settings page is organized into the following sections:



- Scan Pools
  - This tab allows you to manage and configure scan pools, which define the resources used for scanning files. You can add, edit, and delete scan pools, as well as specify the number of integrated scan instances for each pool. *[Referencing the image: this tab is currently selected in the screenshot.]*



- Export
  - This tab allows you to configure how scan results and other data are exported from MetaDefender Storage Security. You can specify export formats, destinations, and other export-related settings. You can find more information on this page.
- SMTP
  - This tab is used to configure SMTP settings for email notifications. You can specify the SMTP server, authentication details, and other email-related parameters to

ensure seamless delivery of notifications.

- Notifications
  - This tab allows you to configure various types of notifications, such as alerts for detected threats, scan completion notifications, and other system events. You can customize the content and delivery method for each notification type.
- License
  - This tab provides information about your MetaDefender Storage Security license, including expiration dates and feature availability. You can also use this tab to manage and update your license.
- Users
  - This tab allows you to manage user accounts and permissions within MetaDefender Storage Security. You can add, edit, and delete users, as well as assign roles and permissions to control access to different features.
- Data Retention
  - This tab allows you to configure data retention policies, specifying how long scan results and other data are stored in the system. You can define retention periods based on your organization's compliance requirements.
- External Loggers
  - This tab allows you to configure external loggers, such as syslog servers, to forward system logs for centralized logging and analysis.
- Single Sign On (SSO)
  - This tab allows you to configure Single Sign-On (SSO) integration with your organization's identity provider. You can enable SSO and specify the necessary settings to allow users to authenticate using their existing credentials.
- Security
  - This tab allows you to generate a new master encryption key that will re-encrypt all sensitive data. You can find more information on this page.
- Access Control
  - This tab allows you to view the roles defined in your system and the permissions associated with each. You can browse all roles, inspect their Read, Write, and Delete permissions per resource, and use the search bar to locate a specific role by name quickly

Importing/Exporting a configuration

> **Warning**

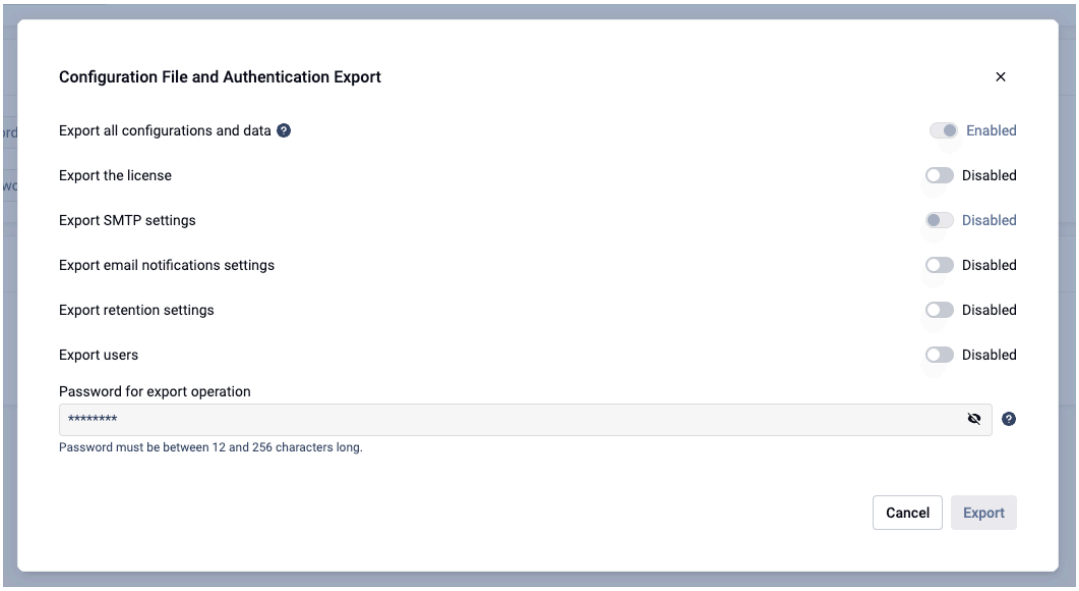
> Importing a configuration file from a version greater than 4.0.0 that includes SSO settings will require reconfiguring SSO, as it was not properly exported.

MetaDefender Storage Security allows administrators to export and import system configuration files. This feature is useful for backup, migration, or replicating settings across environments.

Exporting the Configuration File

Navigate to **Settings **page, click on **Export** button. The export includes all system settings such as accounts, groups, storage units, scan pools, workflows, and more.





Steps to Export

1. Choose which parts of the configuration you want to export (License, SMTP settings, email notifications, retention settings).
2. In the Export Authentication section, you must set a password for the export operation. This password will be required to import the configuration file later. Please store it

securely, as it cannot be recovered.

3. Click Export Configuration to download the configuration file. The file will be encrypted using the password you provided.

Export File Information

1. Configurations from version 4.0.0 and newer are exported as ZIP files. Older versions use JSON format.
2. Export files typically range from 10MB for basic setups to 100MB+ for complex configurations.
3. Store export files in secure locations with appropriate access controls.

Importing a Configuration File

On a fresh install of MetaDefender Storage Security, after creating the first user account and logging in, you can choose to import an existing configuration.

Version Compatibility

- **JSON files** for configurations older than version 4.0.0
- **ZIP files** for configurations from version 4.0.0 or newer
- **Cross-version imports** for importing older configurations into newer versions - has limited compatibility and may require manual review

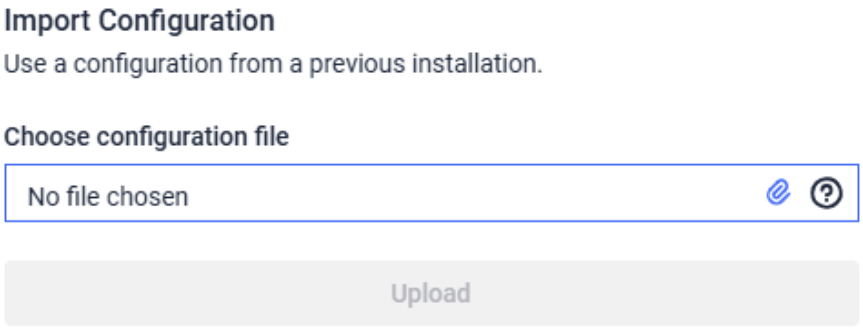
Steps to Import

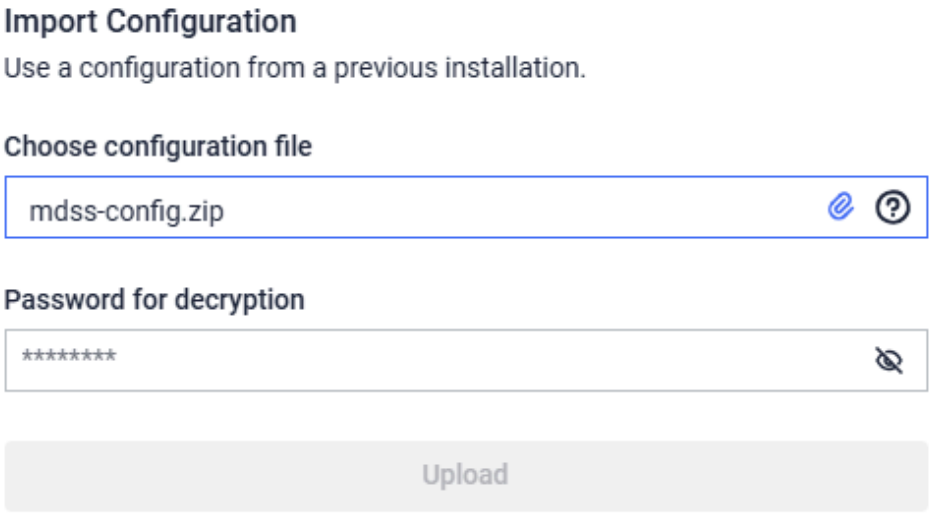
In the Import Configuration screen, you can click on the input field, browse and select a file to Import. The input will accept either a JSON file (if the configuration is older than 4.0.0) or a .ZIP file (if the configuration is 4.0.0 or newer).

If the file is password-protected, you will be prompted to enter the password that was set during export. Afterwards, click on the Upload button and the configuration import procedure will start.

Import timing

The import process typically takes 30 seconds to 2 minutes, depending on configuration size. If license transfer is enabled, allow additional time for online activation.





Post-Import Requirements

Important: After importing a configuration that contains advanced settings (such as HTTPS configurations, SSL certificates, or custom integrations), you must restart the following Windows services:

- MetaDefender Storage Security Service
- MetaDefender Storage Security Web Service

- MetaDefender Storage Security Scanning Service

Note: Service restart is only required if the imported configuration contains advanced settings. Basic configurations typically do not require a service restart.

Verification Steps

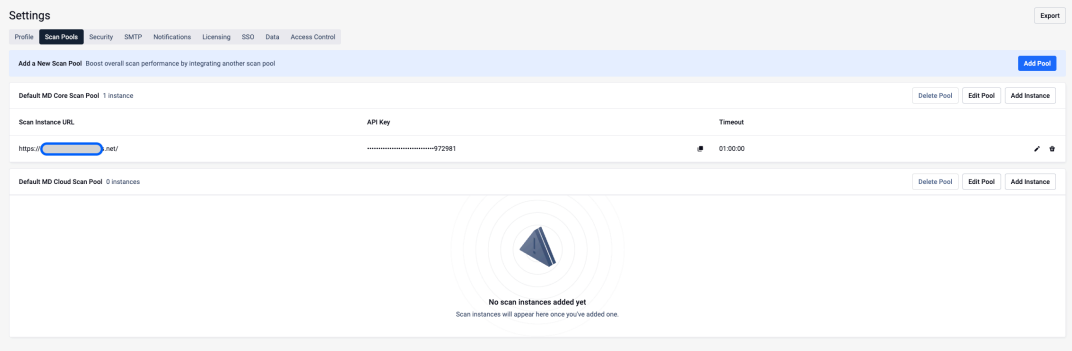
After import completion, verify the following:

- All services are running properly
- License is activated (if applicable)
- Storage units are accessible
- User accounts function correctly
- Email notifications work (if configured)

Scan Pools

MetaDefender Storage Security uses scan pools to group scan instances. A single pool can hold MetaDefender Core, MetaDefender Core Cluster, and MetaDefender Cloud instances together, in any combination.

Scan pools balance the load of scan requests across the instances they contain. A pool still records the type it was created with, but that value no longer limits which instances you can add to it.



The default scan pool

Every installation has one default scan pool, named **Default MetaDefender Scan Pool**. Earlier versions created two, one per instance type.

On upgrade both existing default pools are kept with all their instances, but only the MetaDefender Core one stays marked as default. No workflow changes the pool it points to. If you renamed a pool yourself, your name is kept.

Scan rules in a mixed pool

MetaDefender Core and MetaDefender Cloud use different rule names. A workflow now keeps its rules separately per engine type, and only the matching rules are sent with a scan request. A workflow saved before the upgrade keeps a single rule list until you edit and save it, and that list is sent to every instance in its pool.

If a workflow has no rule for the selected instance type, the scan is refused with an explicit message rather than submitted with rules the engine cannot interpret. Workflows created before the upgrade keep working unchanged as long as their pools hold one engine type. Before you add an instance of the other engine type to a pool, open each workflow that uses it, add rules for that engine, and save.

> **Supported technologies in a mixed pool**

> Technologies are reported from the instances in the pool. MetaDefender Cloud does not report which technologies a rule enables, so one MetaDefender Cloud instance **in the primary or failover pool** makes all technologies show as unknown for every workflow using that pool. This is expected: a file may be scanned there, so no guarantee is claimed.

Failover pools

A failover pool no longer has to be the same type as the primary pool.

> **MD Core API Endpoints and Permissions**
>
> To learn more about the required permissions and the endpoints used by MDSS for MD Core, please read the following KB article: Which API endpoints from MD Core does MDSS utilize?

> **MD Cloud Private Processing**
>
> MD Cloud instances use private processing by default to ensure maximum data privacy and security. Therefore, a non-Community API key must be used when integrating in MDSS. With private processing:
>
> - All submitted files are automatically deleted immediately after analysis completion
> - Scan results are only accessible to the original API key owner

Workflows

Each workflow consists of the following configurations:

- File discovery
- File filters
- Scan pool
- File tagging
- Deep CDR
- File remediations
- Other remediations

> **Configurations limitations**
>
> The available configurations are tied to the MetaDefender Cloud/Core available technologies. Providing an API Key when setting up the Scan Pools will list all the available technologies that are configured in MetaDefender Cloud/Core so that configurations can be successfully applied.

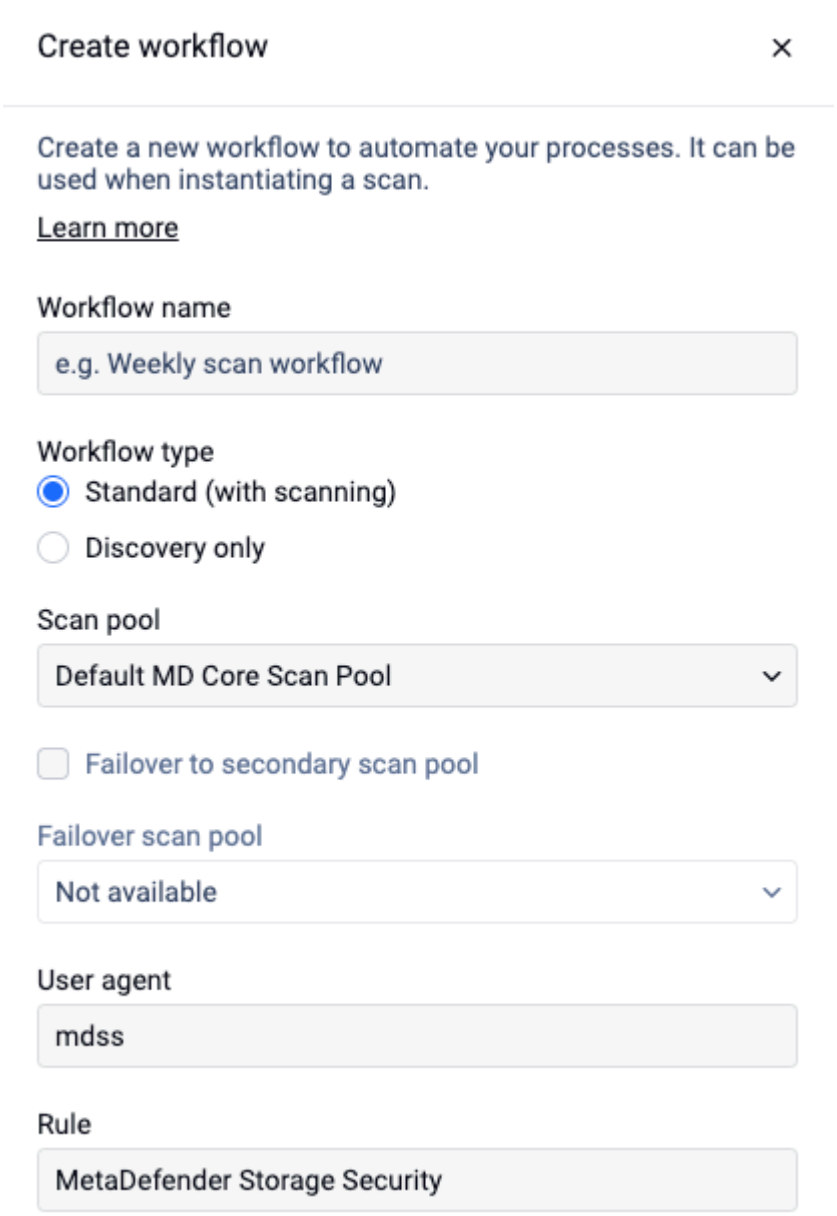
> **Prerequisites**
>
> In order to be able to create a workflow, a scan pool is required.

Creating a Workflow

As a first time user, there is a default workflow that is created during the onboarding process. There can be created as many workflows as required for different use-cases.

> **Default workflow**
>
> The default workflow can't be deleted. There must always be a default workflow configured in the system.

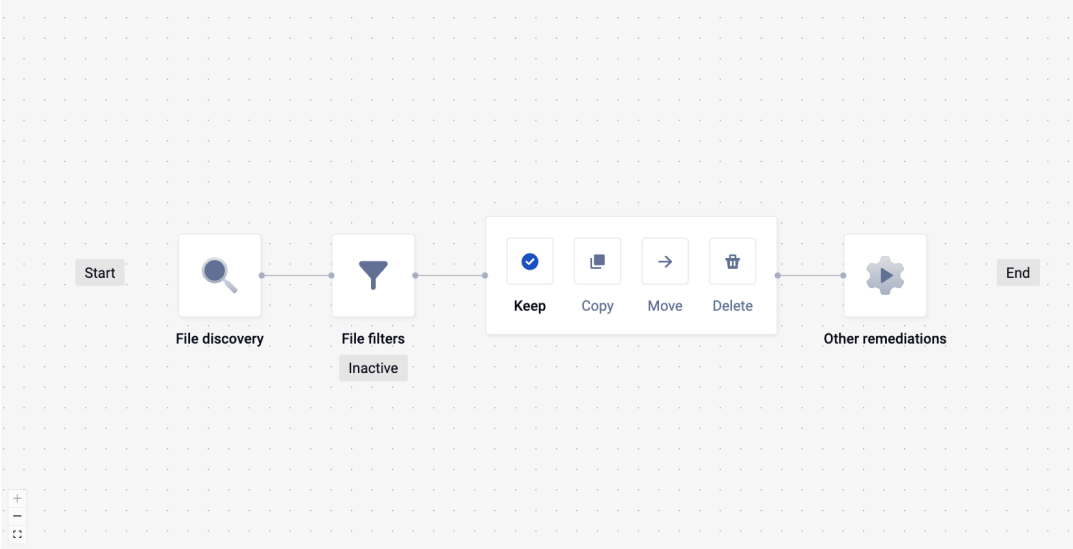
A workflow can be created by navigating to the Workflows page and clicking the "Add new workflow" button. This will reveal a dialog box where the new workflow name must be set and also the type of the workflow.



Once a name is given to the new workflow it will require to choose the type of workflow, afterwards the system will mark it as a draft.

Discovery only

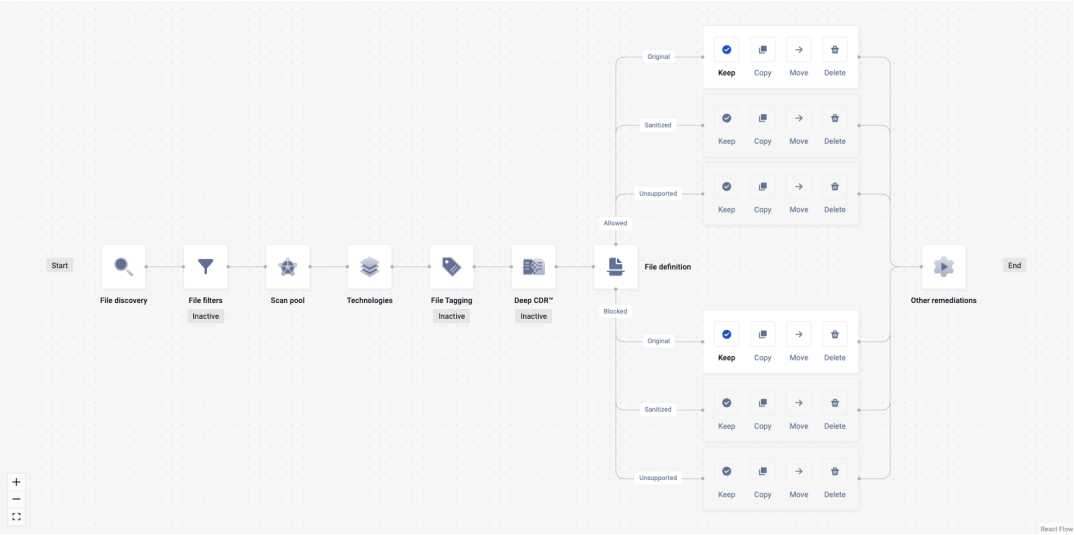
The Discovery only workflow will be created with the default settings after pressing Create button



This type of workflow it is used for storage interaction without scanning the file. This allows a more limited number of remediations and focus only on discovering folder structure and managing it

Standard

The standard workflow is saved in the system only after the “Scan pool” is properly configured. While in draft mode, all the other settings nodes are inaccessible until the Scan pool is successfully configured.

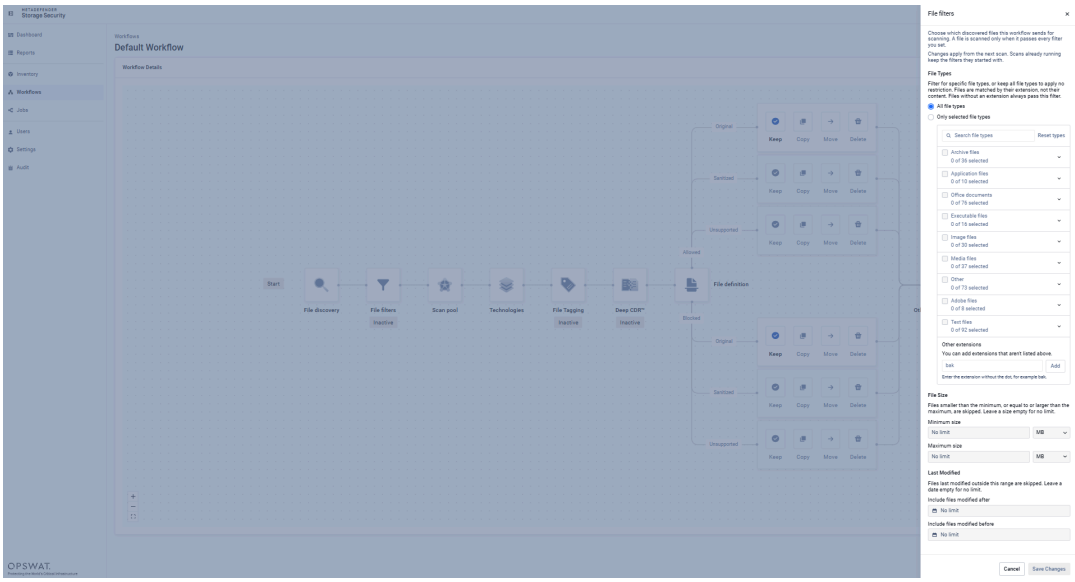


Once the Scan pool is successfully configured, the other settings can be updated by clicking on each one individually. The workflow is updated instantly once the “Save changes” button is clicked for each of the settings nodes.

File Filters

File filters decide which discovered files are sent for scanning. You can limit a workflow to selected file types, a size range and a last-modified date range. A file is scanned only when it passes every filter you set; files that don't are recorded in the scan report as **Skipped by filter**, with the reason, and are not scanned. The node shows **Active** when at least one filter is set.

File types are picked from categories (archives, Office documents, images, executables and others) or entered as custom extensions. The match is based on the file extension only, not on the file's content.



For detailed setup instructions and the full list of supported file types, refer to File Filters.

Scan pool

Each workflow must have a scan pool node that determines which scan pool is used for scanning. It connects to an existing scan pool - such as MetaDefender Core, MetaDefender Distributed Cluster, or MetaDefender Cloud - and allows you to specify a user agent and a workflow rule to control how scanning is performed. A failover scan pool can also be designated to provide an alternative scanning resource when the primary pool is unavailable or unresponsive.

MetaDefender Storage Security uses scan pools to provide a flexible way to initiate scanning processes, leveraging scanning services from MetaDefender Core, MetaDefender Distributed Cluster, or MetaDefender Cloud based on the configured workflow rule.

Each scan pool configuration is tied to a specific scan pool and workflow rule, ensuring consistent scanning behavior across MetaDefender Core, MetaDefender Distributed Cluster, and MetaDefender Cloud instances.

> **IMPORTANT**
>
> The default workflow rule for MetaDefender Core/Distributed Cluster is 'MetaDefender Storage Security'.
>
> For MetaDefender Cloud default rules are 'multiscan' and 'unarchive', please check the following documentation.

When configuring the scan pool step in a workflow, you can select a scan pool, optionally enable failover to a secondary scan pool for cases where all scan instances from the primary pool are unavailable or unresponsive, and specify the user agent and workflow rule to apply during scanning.



Technologies

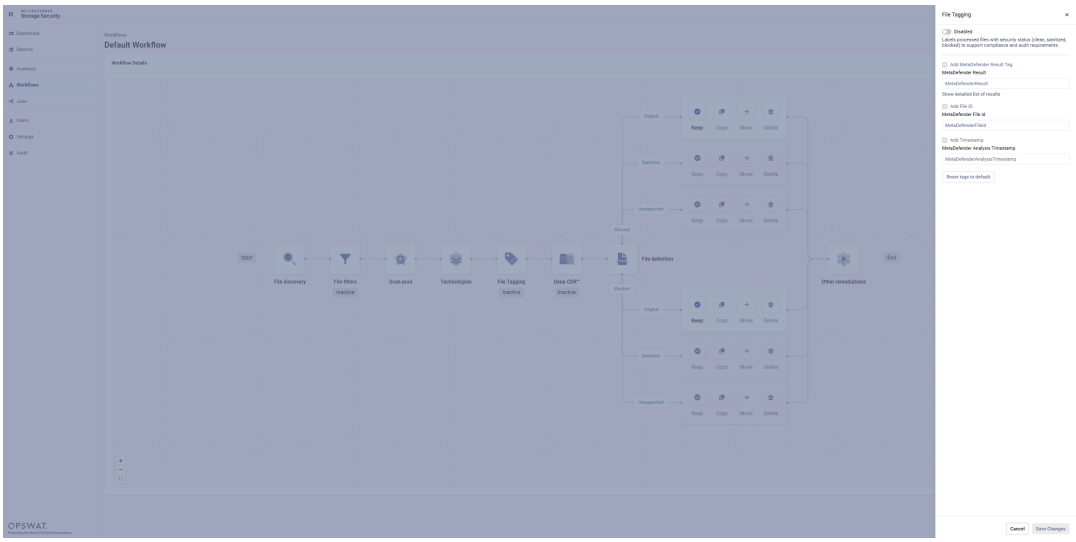
This lists all technologies available in the connected MetaDefender Core/Cloud/Distributed Cluster instance configured in the scan pool step. If settings have changed on the MetaDefender Core/Cloud/Distributed Cluster side, use the **Refresh technologies** button to load the latest available technologies



File Tagging

When enabled, File Tagging labels processed files with their security status (clean, sanitized, blocked) to support compliance and audit requirements. You can configure which tags are applied, including the MetaDefender result, file ID, and analysis timestamp and customize the tag name for each. Tags can be reset to their default values at any time using the **Reset tags to default** button.

For instructions on setting up the feature, refer to the Configure File Tagging.



Deep CDR

Applies OPSWAT's Deep CDR (file sanitization) technology to files.

Enabling the File Sanitization feature creates a secure version of a file by removing potentially harmful content. This process utilizes OPSWAT's Deep Content Disarm and

Reconstruction (Deep CDR) technology.

For detailed instructions on configuring this feature, refer to the Deep CDR user guide and Deep Content Disarm and Reconstruction (Deep CDR).



File Definition

What was previously known as Remediation Actions is now called File Definition, which dictates the outcome for each file based on its scan results. Files can be designated as Allowed (approved), Sanitized (cleaned), or Blocked (quarantined).

For detailed setup instructions, refer to the Configure File Definition.



File Remediations

File remediation actions are automated post-processing tasks performed on files after they have been scanned. The following remediation options are currently available: Keep,

Copy, Move, Delete.

For more information on setting up the feature, refer to the Keep, Copy, Move or Delete Files.

Important Considerations

- Make sure that the destination buckets for moved or copied files are properly configured and accessible within your environment (in this case Amazon S3)
- Carefully consider your organization's risk tolerance when configuring deletion actions. Implement safeguards like versioning or backups to prevent accidental data loss.
- Develop a clear tagging strategy to facilitate efficient file management and consistency.
- Regularly review and update your remediation actions and tagging configurations to align with your organizational evolving security needs and policies.



Other Remediations

Delete Empty Folders

Delete a directory if it is empty after remediation (does not delete if the folder was empty to begin with).

For detailed setup instructions, refer to the Delete Empty Folders.

> **Dependency**
>
> The delete empty folders remediation can be active only if there is at least one of another remediations configured with "Move the file" or "Delete the file".

Scan File Versions

> This remediation type is only supported by the SharePoint On-Prem storage unit type.

Retrieve & process all past versions of a file. This comes with a performance cost, as all versions will be scanned like normal files.

For more details about this feature, refer to Scan File Versions.



Error Handling

The remediation process includes automatic retry functionality to handle transient failures.

For detailed configuration steps, refer to the Remediation Process Retries.

Autodiscovery

Autodiscovery is a powerful feature in MetaDefender Storage Security designed to **simplify and accelerate the integration of your cloud storage**.

It automatically detects and adds storage units from supported cloud storage services, eliminating the need to manually configure each one.

This significantly streamlines the process of onboarding multiple storage units, saving valuable administrative time and reducing the potential for configuration errors.

Supported Storage Types

> **Info**
>
> Currently, Autodiscovery functionality is available for the following storage types:
>
> - **Amazon S3** — How to Enable Autodiscovery
> - **S3 Compatible**, **Wasabi**, **Oracle S3-Compatible**, **Cubbit** and **Scality RING** — these follow the same flow as Amazon S3; see the Amazon S3 guide above
> - **Azure Blob** — How to Enable Autodiscovery
> - **Google Cloud** — How to Enable Autodiscovery
> - **SharePoint Online** — How to Enable Autodiscovery

Note: MetaDefender Storage Security is continuously evolving. Support for additional cloud storage services may be introduced in future releases. Please refer to the latest release notes for the most up-to-date information.

How Autodiscovery Works

The Autodiscovery process involves several key stages:

1. **Prerequisites**

    - Before initiating autodiscovery, ensure the credentials you provide for your cloud storage service have the necessary permissions to list all available storage units. For example, an AWS IAM user or role needs the `s3:ListAllMyBuckets` permission for Amazon S3 and S3-compatible storages, and a Google Cloud service account needs the `storage.buckets.list` permission on the project (for example through a custom role or Storage Admin). The exact permissions for each storage type are listed on its “Add a ... Storage” page. Without these permissions, MetaDefender Storage Security cannot detect your storage units.

- Familiarize yourself with how accounts and groups are managed within MetaDefender Storage Security. This will help you understand how autodiscovered storage units are organized post-discovery. For comprehensive details, please review the Accounts & Groups documentation below.

2. **Initiating Storage Discovery**

- You provide valid credentials for a supported cloud storage service through the MDSS interface.

3. **Scanning and Detection**

- MDSS uses the provided credentials to securely connect to your cloud storage account
- It then scans the account to detect all available storage units (e.g. S3 buckets associated with the AWS account, Azure Blob containers in the storage account, or SharePoint Online sites in the tenant).

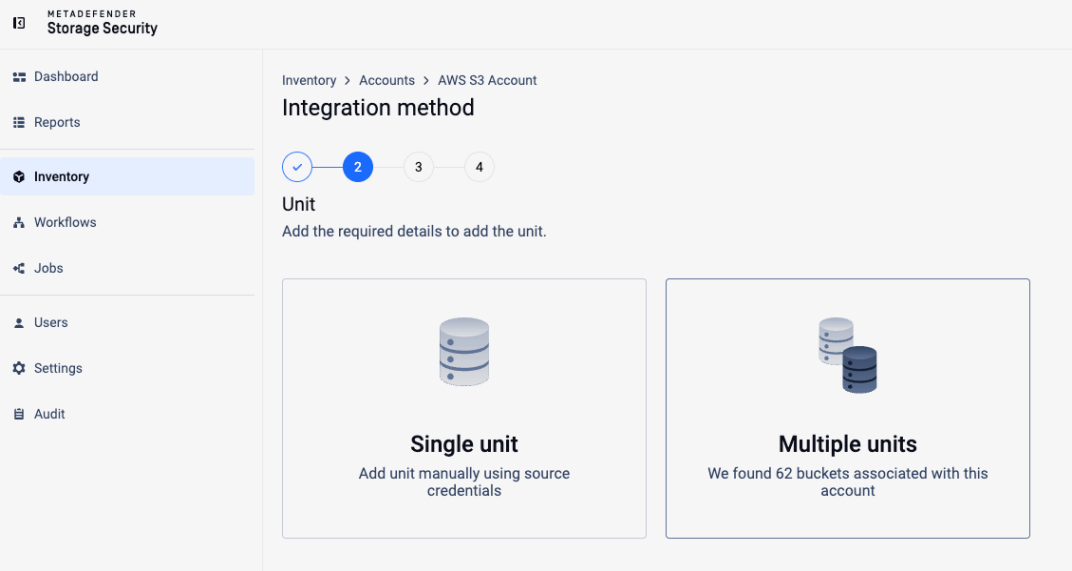
4. **Automated Integration & Configuration**

- Successfully detected storage units are automatically added to MetaDefender Storage Security.
- These autodiscovered storage units will typically be associated with a specific account and group within MetaDefender Storage Security, often based on the context of the autodiscovery setup.

> **LIMITATION**
>
> **All units discovered in a single operation will share the same initial account and group configuration.**

The Autodiscovery feature works by:

- **Storage Discovery**: When provided with valid credentials for a supported cloud storage service, the system scans the associated account to detect all available storage units (such as S3 buckets).



- **Automated Addition**: Depending on the chosen method, the system can either:
  - Automatically add all discovered storage units
  - Present a list of discovered storage units for manual selection
  - Allow individual adding of storage units
- **Periodic Synchronization**: For accounts with Autodiscovery enabled, the system periodically checks for new storage units and automatically adds them to MetaDefender Storage Security.

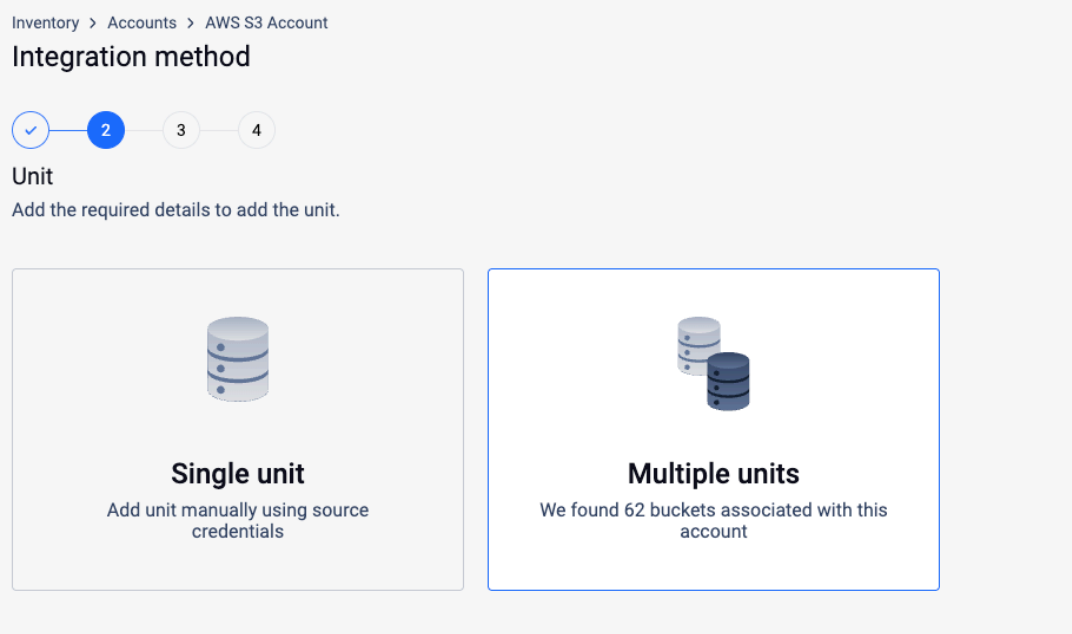
Using Autodiscovery

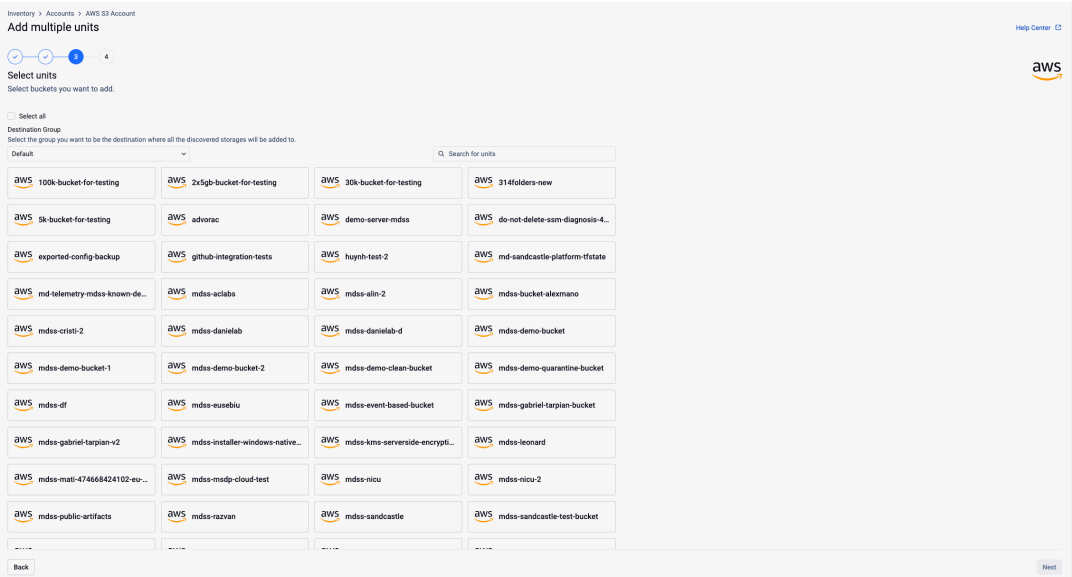
Method 1: When Adding an Account

1. From the left side menu, navigate to **Inventory**, click **Add New** > **Account**, choose one of the supported storage types listed above and click **Next**
2. Fill out the form with valid account credentials
3. Turn on the **Autodiscovery** switch and select the **Destination Group** where the discovered storage units will be added
4. Click **Next**. The **Summary** step shows Autodiscovery as **Enabled** together with the destination group, and the discovered storage units are added to that group. Click **View in inventory** or **Finish**

To add only specific storage units, leave **Autodiscovery** disabled, click **Add unit** on the **Summary** step, and choose:

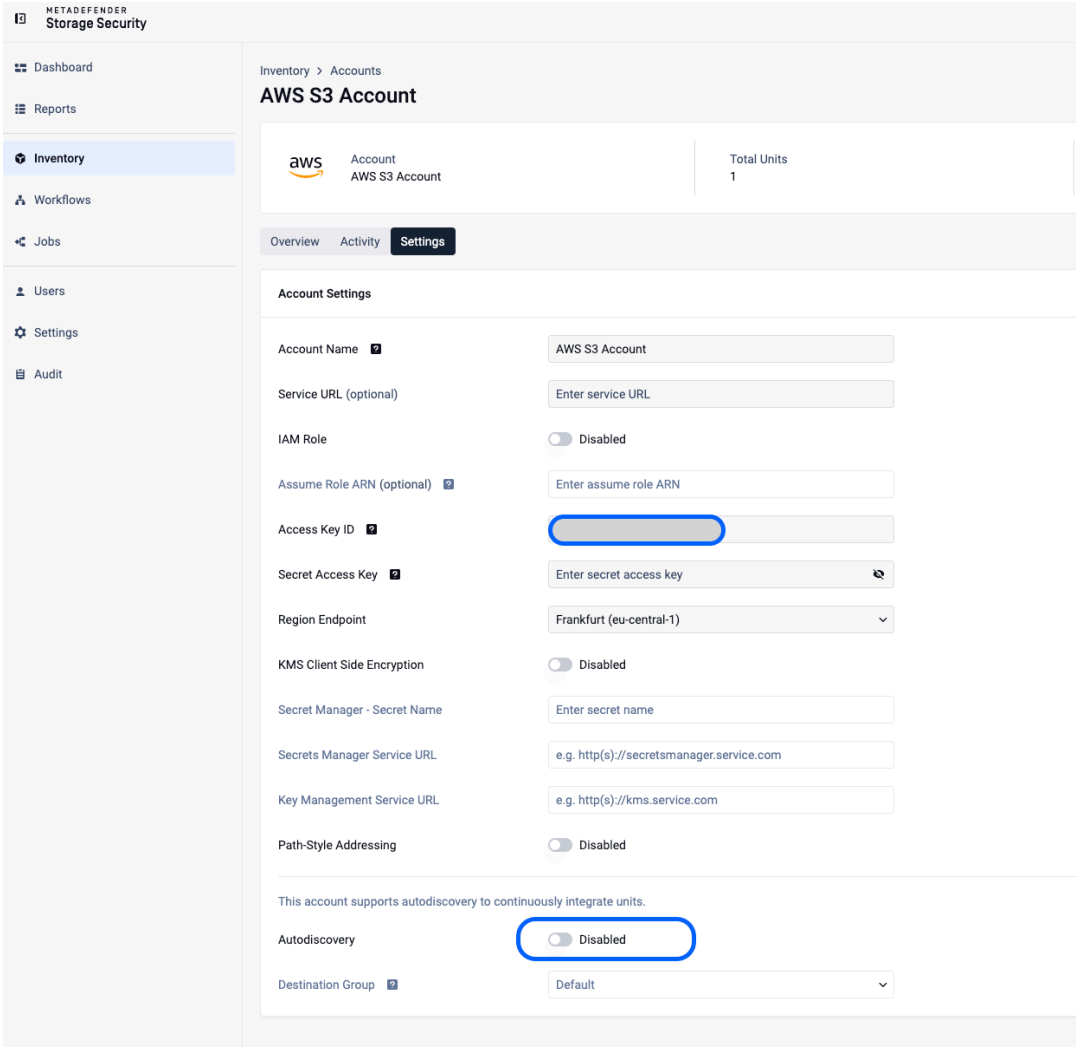
1. **Single unit** to add one storage unit with full configuration control
2. **Multiple units** to select which of the discovered storage units to add





Method 2: Enabling Autodiscovery for an Existing Account

1. Navigate to **Inventory** > **Accounts** and open the account (or select **Account Settings** from the account's three-dot menu)
2. Open the **Settings** tab
3. Turn the **Autodiscovery** switch on or off and select the **Destination Group**
4. Input the secret key or credentials file and complete the process by clicking **Update**



Common Issues

1. **No storage units discovered**
    1. Verify the credentials have sufficient permissions
    2. Check that the selected region is correct
    3. Confirm that the account contains storage units of the supported type
2. **Autodiscovery not working periodically**
    1. Verify the account's Autodiscovery setting is enabled
    2. Check system logs (e.g., Audit page) for any authentication or connection errors
3. **Storages not appearing in Inventory**
    1. If the storage units were added with **Add unit** > **Multiple units**, the Summary step lists the storage units that weren't added successfully, including the error why each wasn't added.

LIMITATIONS

> When storage units are autodiscovered, they will all be set up in the system sharing the same account and group details provided during the autodiscovery configuration.
>
> For autodiscovery to successfully identify your storage units, the credentials used must possess the necessary permissions to list all accessible storage units.

Inventory

The **Inventory** page is where you view and manage all the storage resources that MetaDefender Storage Security (MDSS) protects. From here you can see every storage unit registered for scanning, the groups they are organized into, and the accounts used to connect to your storage providers.

Access: Select **Inventory** in the left navigation menu.

Tabs

The page is organized into three tabs:

- **Units** — Individual storage locations registered for scanning (for example, an Amazon S3 bucket or an NFS share).
- **Groups** — Logical collections of units, used to apply shared settings and workflows.
- **Accounts** — The connections and credentials MDSS uses to reach each storage provider.

Units

Individual storage locations registered for scanning (for example, an Amazon S3 bucket or an NFS share).

To add a new storage unit, please follow the Integrations page.

Accounts

What are Accounts?

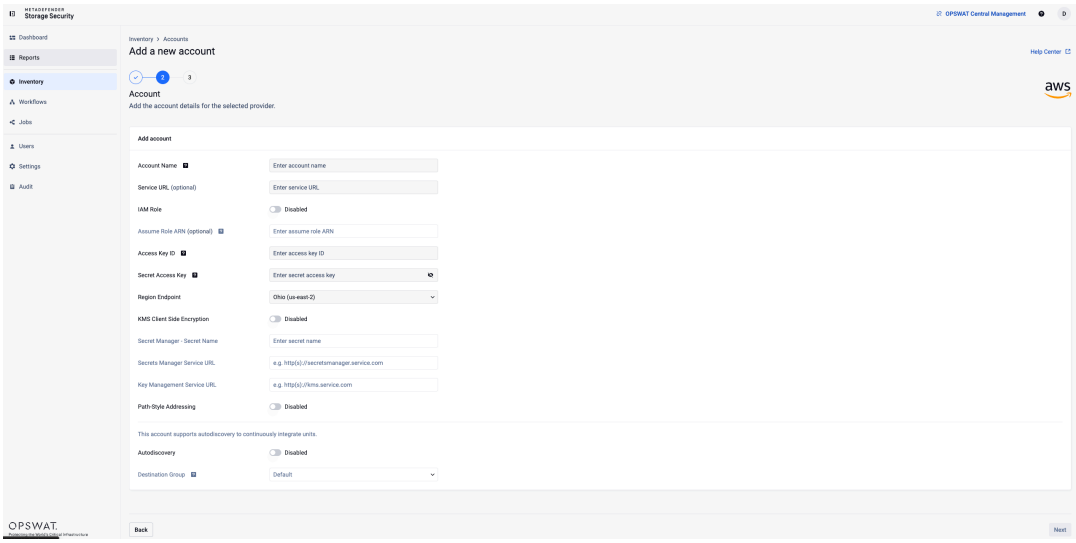
Accounts in MetaDefender Storage Security allow you to:

- Store and manage access credentials for storage services in one centralized location (credentials are encrypted and stored securely in the database)
- Associate multiple storage units with a single account for simplified management
- Enable automatic discovery of new storage units for supported services (see Autodiscovery)
- Organize storage resources within groups for better administration

> **Security Note**
>
> Role-based access control governs all account operations - Administrator, SSO Administrator, and Tenant Administrator roles have full permissions to create, modify,

> and delete accounts, while Read-Only Administrator and SSO Read-Only Administrator roles cannot perform these actions.

Creating a New Account



To create an account:

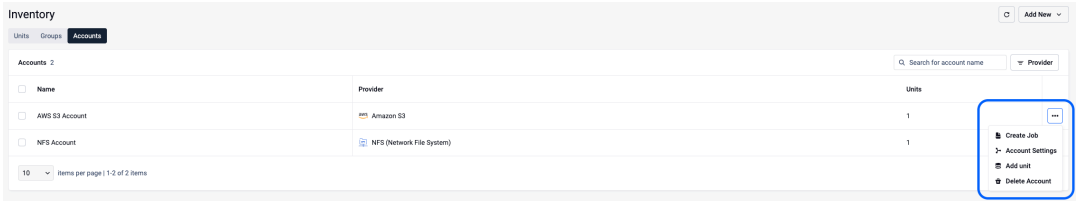
1. From the left side menu, navigate to **Inventory** and click **Add New** > **Account**
2. Select the service you want to integrate and click **Next**
3. Fill out the required account's credentials and click **Next**
4. On the **Summary** step, click **Add unit** to add a storage unit to the new account

> **Note**
>
> There are no limits on the number of accounts that can be created.

Managing Existing Accounts

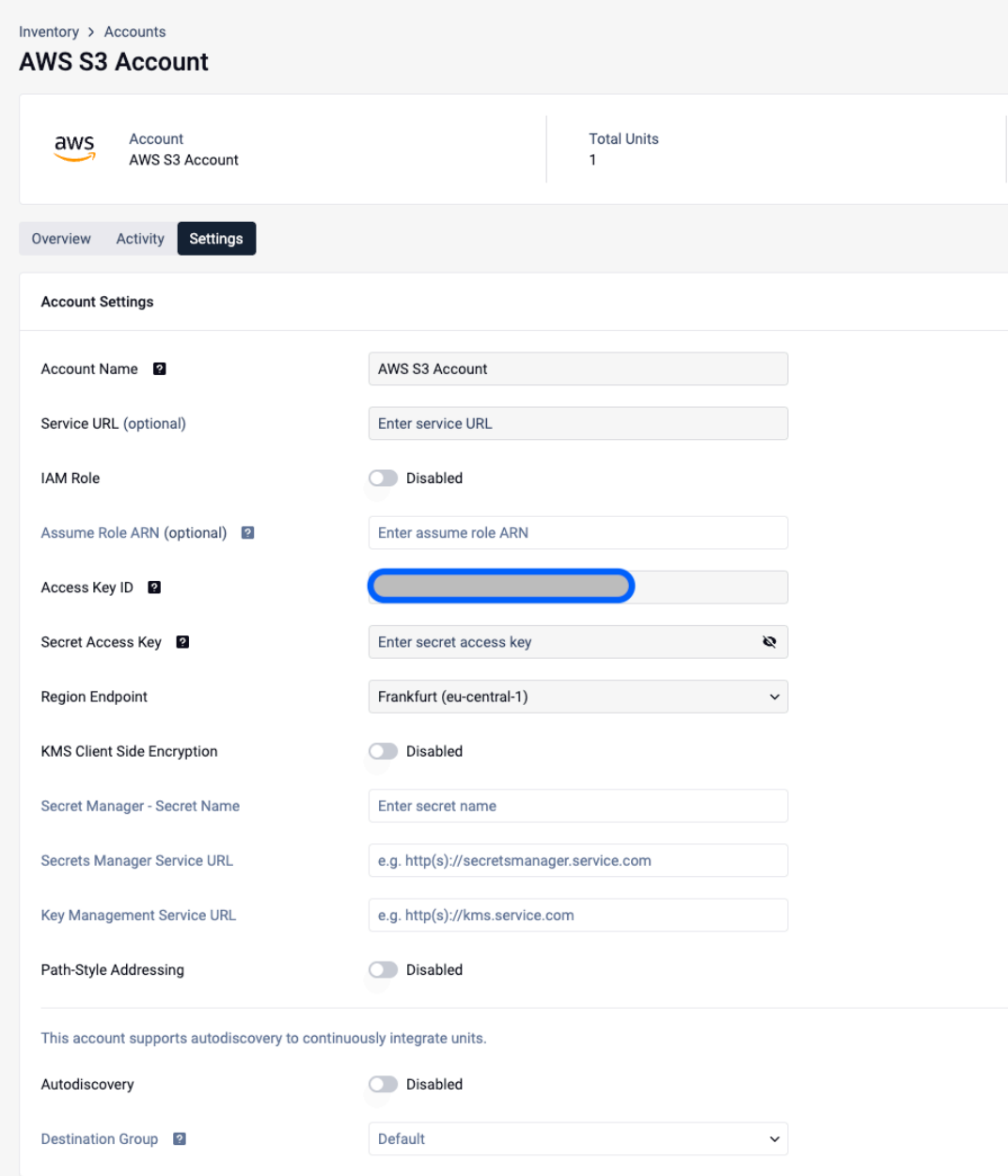
1. Navigate to **Inventory** and open the **Accounts** tab
2. Locate the account you want to manage
3. Click the **three-dot menu** next to the account name

Available Account Actions



- **Create Job -** Starts instant scans for all storage units under the account. These scans can be monitored through the ongoing scans interface. If individual scans fail, they are handled independently without affecting other scans in the batch.
- **Account Settings** - Opens the account's **Settings** tab to modify account credentials
- **Add unit** - Add a new storage unit in this account
- **Delete Account** - Removes the account

Account Settings



1. From the account's three-dot menu, select **"Account Settings"**
2. The **Settings** tab opens pre-populated with current account details
3. You'll be required to input credentials or credentials file to validate changes
4. Click **"Update"** to update the account

Delete Account

1. From the account's three-dot menu, select **"Delete Account"**

2. **Accounts with storage units:** The confirmation message shows the count of associated storage units, which are deleted together with the account
3. **Empty accounts:** Can be deleted after explicit confirmation
4. Confirm the deletion when prompted

> **IMPORTANT**
>
> When accounts are deleted, all associated reports and historical scan data are permanently removed and cannot be exported - there are currently no data export capabilities for this purpose.

Groups

What are Groups?

Groups in MetaDefender Storage Security provide an organizational structure for managing storage units and accounts. They enable logical grouping of related storage resources, simplifying administration and enhancing visibility across the system.

Key Features include ways to:

- Organize storage units and accounts in logical collections
- Perform actions on multiple storage units at once
- Clear visual separation of resources in the user interface

> **IMPORTANT**
>
> MDSS groups operate as a single-level organization system without nesting capabilities, and storage units cannot belong to multiple groups simultaneously. All storage units are mandatory to be assigned to a group.

Default Group

The system includes a default group. This ensures that all storage units and accounts always have a group assignment, even if a custom group is deleted.

- Storage units or accounts can be moved to the default group if their assigned group is deleted
- The default group has a predefined ID and name but can have its description modified

Creating and Managing Groups

Creating a New Group

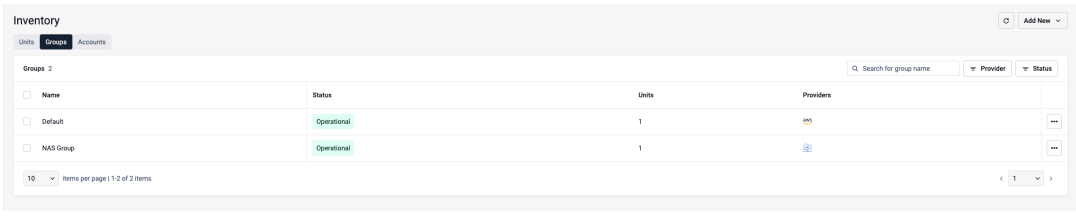


Groups can be created from the Inventory page

- Click **Add New** > **Group**
- Enter a **Group Name** (required field)
- Optionally **select existing storage units** in the **Units** field to include in the group
- Click **"Add group"** to finalize

> **Note**
>
> During storage unit creation, the first available group is pre-selected by default. There are no limits on the number of groups that can be created.

Managing Existing Groups



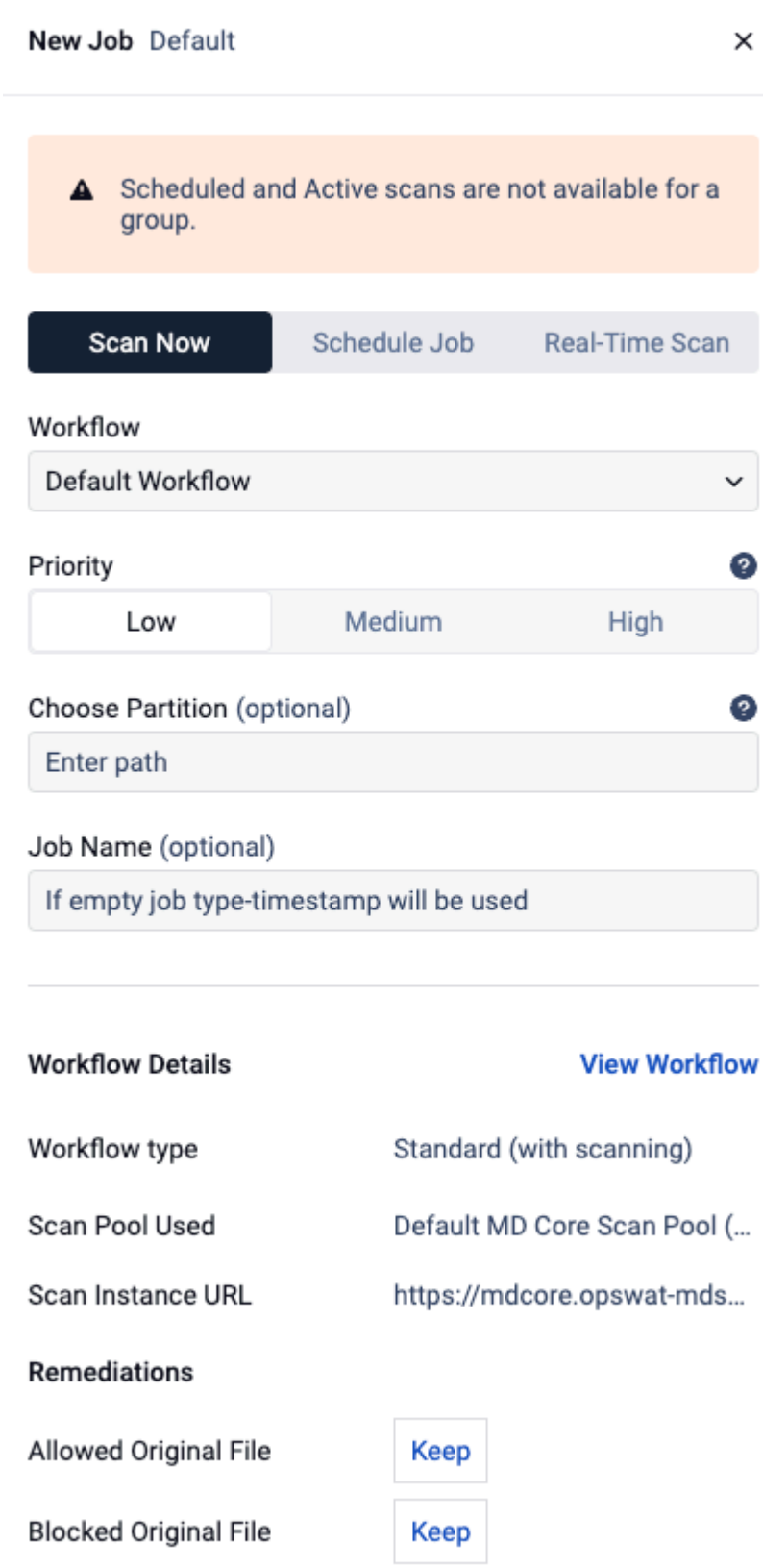
1. Navigate to **Inventory** and open the **Groups** tab
2. Locate the group you want to manage
3. Click the **three-dot menu** next to the group name

> **Security Note**

> Role-based access control governs all group operations - Administrator, SSO Administrator, and Tenant Administrator roles have full permissions to create, modify, and delete groups, while Read-Only Administrator and SSO Read-Only Administrator roles cannot perform these actions.

Available Group Actions

Scan All Group Items



- Starts instant scans for all storage units under the group. These scans can be monitored through the ongoing scans interface. If individual scans fail, they are

handled independently without affecting other scans in the batch.

Group Status Information

Storage Count Badge

- Shows the total number of storage units in each group with format: “X storage units”

Status Badges

Visual indicators showing:

- **Error Count** - Badge showing storage units with errors
- **Testing/Untested Count** - Badge showing storage units being tested or untested
- **Success State** - No badges when all storage units are operational

Moving Storage Units Between Groups

1. From any storage unit’s management options(three-dots), select **“Move to Group”**
2. Choose the destination group from the dropdown
3. Confirm the move to complete the transfer

Cross-Domain Transfer (Data Diode)

Move files across a one-way security boundary using MetaDefender Storage Security on both sides of an OPSWAT NetWall diode. The diode transfers files over SMB; MDSS accesses each SMB share by mounting it as an NFS storage unit.

Overview

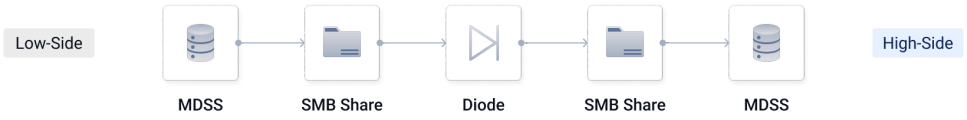
A data diode enforces one-way data flow: traffic goes from the low side (source) to the high side (destination) and never back. This physical isolation is why diodes protect critical environments — there is no return path, and no delivery acknowledgement.

How It Works (Concepts & Architecture)

The architecture is symmetric. The diode owns the cross-boundary mechanics over SMB; MDSS reads and writes its share through an NFS mount of that same SMB share. Complete flow:

1. Low-side MDSS pulls from a source storage (SharePoint, S3, or any integrated storage), scans/remediates, and writes approved files to the low-side share.
2. The NetWall diode polls that share over SMB, transmits each file across the optical link, and writes it to the high-side share.
3. High-side MDSS reads the high-side share, scans/remediates, and forwards files to the destination.

The diode is the SMB client on both sides and owns all cross-boundary mechanics: polling, transfer, and cleanup. MDSS only writes to the low-side share and reads from the high-side share, treating each as an ordinary NFS storage unit.



SMB vs FTP

SMB is recommended. It multiplexes transfers, sustains higher bandwidth, and has sub-second timestamps that MDSS uses to detect when a file is fully written. NetWall's native FTP transport is supported but under increased concurrent load the FTP channel resets without self-recovery.

Requirements

- An OPSWAT NetWall diode configured for native SMB file transfer. Set up the share on the diode per the NetWall Windows file share guide.
- Two SMB shares, one per side, reachable by the diode and the local MDSS host. Any standards-compliant SMB server works (e.g. Samba).
- Two MDSS instances, one per side, see Planning your deployment. Low-side MDSS reaches the source storage; high-side MDSS reaches the destination.

The diode speaks SMB, but MDSS connects to each share as an NFS storage unit: mount the SMB share on the MDSS host, then add it to MDSS as NFS. Do this on both sides:

- Add an SMB share as an NFS storage unit (Windows deployment)
- Add an SMB share as an NFS storage

Throughput is bounded by the diode license tier (e.g. 100 Mbps; gigabit raises file transfer to 500 Mbps) and by disk I/O on the shares.

Set Up the Low Side (Sender)

1. Mount the SMB share into low-side MDSS and add it as a NFS storage unit (see Requirements).
2. Create a workflow whose copy or move destination is that storage unit.
3. Start a scan on the desired storage unit with that workflow. Files are written to the share, and the diode polls them automatically.

Set Up the High Side (Receiver)

1. Mount the SMB share into high-side MDSS as an NFS storage unit (see Requirements).
2. Set the discovery debounce environment variable so MDSS never scans a file the diode is still writing; it skips any file modified within the window:`NFS_DISCOVERY_DEBOUNCE_TIME_SECONDS=30` 30s suits typical files; raise it for large files that take longer to cross the link.
3. Create a workflow whose copy or move destination is your target storage (e.g. SharePoint)
4. Start a real-time (RTP) scan on the storage unit with that workflow. As the diode writes files in, MDSS discovers, scans, and forwards them.

Monitor

- MDSS surfaces progress only for running scans, the low-side scans and the high-side RTP scan, from the UI.
- Use the diode's own monitoring for transfer status, queue depth, and channel health across the link.
- There is no delivery acknowledgement back to the low side. End-to-end reconciliation means comparing source vs. destination inventories.

Troubleshooting

Files get scanned mid-write, or scans come back partial. MDSS picked up a file before the diode finished writing it. Bump `NFS_DISCOVERY_DEBOUNCE_TIME_SECONDS` past how long your biggest file takes to cross.

Transfers die after the first 10–15 files. This shows up on old SMB servers (Samba 4.12.x). Upgrade to a modern version, if it is not an option, turning off oplocks and leases helps but the real fix is the newer server.

Low-side share keeps filling up. The diode is supposed to delete each file once it's across. If files pile up, check that its account can actually delete on the share and that it's still polling.

Files vanish under heavy load. When the destination can't keep up with the incoming rate, the high-side diode buffers to disk. Once that buffer is full, new files are dropped — silently, because a diode has no way to tell the sender to slow down. Give the destination enough headroom, cap per-file size, and split very large files on the source side.

Limitations

- **No delivery confirmation.** A diode has no return channel, so a full high-side buffer drops files silently. Size the destination and diode disk for peak load.
- **Retransmit on HA failover.** NetWall supports active/standby HA, but if the active high-side node fails with files still buffered, those files are lost and must be re-sent from the source today a manual reprocess. Plan source-side retention.
- **FTP is supported but not recommended.** channel resets under concurrent load and timestamps are minute-resolution. Use SMB.

Configuration and Management

Custom configuration

How to edit the configuration file on Linux

Custom application configuration can be achieved by editing the *customer.env* environment file.

The following steps are required to change different parameters in this file:

- Navigate to */etc/mdss/customer.env* and open the file
- If the parameter that needs to be changed is not present, add the new parameter with the desired value. If it already exists, just edit the value.

```
CONFIGURATION_PROPERTY_NAME=VALUE
```

- Save the file
- In order for the changes to apply, please restart the application by running the following command:

```
sudo mdss -c restart
```

How to edit the configuration file on Windows

Custom application configuration can be achieved by editing the *customer.env* environment file.

The following steps are required to change different parameters in this file:

- By default the file is located in C:\Program Files\OPSWAT\MetaDefender Storage Security\config\cutomer.env
- If the parameter that needs to be changed is not present, add the new parameter with the desired value. If it already exists, just edit the value

```
CONFIGURATION_PROPERTY_NAME=VALUE
```

- Save the file
- In order for the changes to apply, please restart the application by running the following command:

```
mdss.ps1 -c refresh
```

Increase the number of concurrent scans

To be able to scan multiple files at the same time, the event bus needs to be configured accordingly. Please add the following environment variable in your *customer.env* file (as described above):

Environment Variable	Default Value	Maximum Value	Description
RABBITMQ_SCANNING_PREFETCH_COUNT	20	200	Allow up to 50 files to be scanned at the same time.

Certificate validation for MetaDefender Core connection

This environment variable controls certificate validation for MetaDefender Core integration during HTTP(S) requests. It is useful when MetaDefender Storage Security needs to accept a partially valid certificate chain or a self-signed certificate.

Environment Variable	Default Value	Possible Values
MD_CORE_CERTIFICATE_VALIDATION	0	0 - Use the built-in validation (recommended) 1 - Ignore certificate chain errors 2 - Ignore all certificate errors

Description

- 0 → A MetaDefender Core server with invalid certificate can't be integrated
- 1 → A MetaDefender Core server with certificate chain errors can be integrated
- 2 → A MetaDefender Core server with invalid certificate can be integrated

MetaDefender Core unavailability period

Configure the period of time a MetaDefender Core server is considered unavailable when a scan failure occurs.

Environment Variable	Default Value	Description
LOAD_BALANCER_MD_CORE_UNAVAILABLE_TIME	5	When a scan fails, the respective MetaDefender Core server will be unavailable for this period of time (in minutes). The load balancer cannot choose it for another file during this time period.

Azure Blobs

Configure the number of blobs read in one request

Environment Variable	Default Value	Max Value	Description
AZURE_BLOB_PAGE_SIZE	100	5000	Number of blobs discovered in container in a single request.

Amazon S3

Configure the validity duration of presigned URL used for download files from storage

Environment Variable	Default Value	Min Value	Description
PRESIGNED_URL_EXPIRATION_SECONDS	259200	30	Value is defined in seconds

Web interface port

MetaDefender Storage Security's web interface normally uses port 80 (for regular web access) and port 443 (for secure web access). If you need to use different ports, follow these steps:

For Linux Users:

1. Open the Configuration File:
   - Having elevated privileges, use a text editor to open the file located at `/etc/mdss/customer.env`.
2. Change the Port Numbers:
   - Find or add these lines:
     - `WEB_PORT = 80`

        - `WEB_PORT_SSL = 443`
    - Replace `80` and `443` with the new port numbers you want to use.
3. Save the Changes:
    - Save the `customer.env` file.
4. Restart MetaDefender Storage Security:
    - Open a terminal and run this command: `sudo mdss -c restart`

For Windows Users:

Setting up a custom Web interface port can be done either during the installation process [Windows Installation] or after the installation by following the steps below:

1. Open the Configuration File:
2. - Having elevated privileges, open the file `C:\Program Files\OPSWAT\MetaDefender Storage Security\config\customer.env` using a text editor like Notepad.
3. Change the Port Numbers:
    - Find or add these lines:
        - `WEB_PORT = 80`
    - [Optional] If running on HTTPS
        - `WEB_PORT_SSL = 443`
    - Replace `80` and `443` with the new port numbers.
4. Save the Changes:
    - Save the `customer.env` file.
5. Apply the Changes and Restart:
    - Open powershell.
    - Run these commands:
        - `cd "C:\Program Files\OPSWAT\MetaDefender Storage Security\scripts"`
        - `.\mdss.ps1 -c refresh`
        - This command will refresh the environment variables and restart MetaDefender Storage Security.

Real Time Processing configuration for NAS

> **Info**

> This is applicable only to **NFS**, **SMB**, **SFTP, FTP**, **NetApp ONTAP** and **Dell Isilon** storage units.

> > **DISCOVERY_SERVICE_{PROTOCOL}_RTP_HANDLING** is no longer available starting with version 4.6.0. Please find below instructions on how to enable the same functionality from our Web UI.

When activating Real-Time Processing (RTP) for your storage resources, please note the following operational characteristics:

Scope of Detection

Real-Time Processing monitors and detects only files that are newly created or modified subsequent to scan initiation. The system does not retroactively process pre-existing files that were present in your storage prior to enabling RTP.

Comprehensive File Assessment

To ensure complete analysis of your storage repository, including all pre-existing files, we recommend implementing either:

- Scheduled scans - Configured at predetermined intervals
- On-demand scans - Initiated manually as needed

These comprehensive scan types will process the entirety of your storage contents, including files that predate your RTP implementation. This differentiated approach allows for both continuous monitoring of new content and periodic comprehensive assessment of your complete data repository.

Timestamp Behavior and Detection Implications

When using certain file transfer tools such as WinSCP and similar GUI-based applications, the original file's **LastModifiedDate** is typically preserved during the copy operation. This behavior means:

- Files copied via WinSCP to SMB/NFS/SFTP/FTP shares will retain their original modification dates
- A file last modified in January 2023 and copied in June 2024 will still display January 2023 as its modification date

As a result, Real-Time Processing (RTP) may not detect these files if their modification timestamps predate the RTP scan initialization. For example, if a file with a modification

date of “January 2023” is copied to your storage in “June 2024” after RTP has been activated, standard detection mechanisms will not identify this file as newly added content.

Advanced Configuration for Comprehensive Detection

To ensure all newly copied files are properly detected regardless of their modification timestamps, we have implemented enhanced detection capabilities:

1. Create a Job -> RealTime Scan
2. Select an older date for **Pick files modifies since**



   This means that we will process all the files that have the LastModified newer than the selected.

For example, date is selected to January 24, 2022. When we discover a file that has LastModified on February 1, 2022, we will process it. If we discover a file that has LastModified on January 20,2022, that file will be skipped.

Temporary local files created during remediation

Overview

MDSS creates temporary files during specific remediation operations to ensure reliable data processing. These temporary files are generated during:

- File transfers between storage units
- File sanitization operations using our DeepCDR technology

To assist with troubleshooting and compliance auditing, MDSS provides configurable logging of these temporary file operations.

Temporary File Locations

Linux deployments

Each storage integration runs as a single Docker container named mdintegrations_ [for example mdintegrations_nfs], which hosts the Discovery, Storage, and Remediation services for that storage type. Temporary files are created inside the container of the destination storage type, in the standard Linux temporary directory** /tmp/**.

Operation	Container location
Moving from any storage to NFS	**mdintegrations_nfs**
Moving from any storage to SMB	**mdintegrations_smb**
Sanitization with DeepCDR	Container of the source storage(**mdintegrations_nfs **for NFS files)

Windows deployments

All temporary files will be created in the standard Windows temporary directory **C:\Windows\Temp**

Configuration

The temporary file logging feature can be enabled or disabled by setting the **LOG_CREATION_OF_TEMP_FILES** variable in your **customer.env** file:

- 0: logging disabled (default value)
- 1: logging enabled

To update your **customer.env** file, please follow this guide.

Example:

```
LOG_CREATION_OF_TEMP_FILES=1
```

```
[2025-03-13 14:44:40Z] [Information] remediations.nfs.service: Local
temporary file created. Path: C:\Windows\Temp\4mn110iw.gzn. Size:
1048624. Reason: Upload non-seekable stream. AutoDelete: True.
Destination: credit-card-file-
v2_sanitized_by_OPSWAT_MetaDefender_9619874ae98542129d386871a8cb0268.tx
t
```

Log file locations

Linux deployments: **/etc/mdss/services_logs/**

Windows deployments**:** **{InstallDir}\services**

Frequently Asked Questions

Are my files safe during this process?

Yes. Temporary files exist only within isolated Docker containers (Linux deployment) or in locations accessible only by administrators (Windows). Access to these files is restricted to MDSS processes.

Does this affect performance?

The impact is minimal. Our system is optimized to handle temporary files efficiently and clean them up promptly.

Can I disable temporary file creation?

No. Temporary files are essential for reliable data transfer between microservices in the MDSS architecture.

What happens if a container crashes?

Temporary files exist only within the container's filesystem. If a container crashes or restarts, the temporary files are automatically deleted as part of Docker's container lifecycle.

Are the temporary files using additional storage beyond what is already allocated?

No. The temporary files are created within the allocated storage for your Docker containers. They do not consume additional storage beyond what is already allocated to your deployment.

Best Practices

For optimal performance and security:

1. Ensure your Docker containers have adequate storage allocation. Please follow our guide to properly allocate disk space
2. Monitor container resource usage during large data transfers

Performance Tuning Guide

About This Guide

This guide is designed to assist you in fine-tuning various configuration settings to achieve desired throughput and performance levels. By default, MetaDefender Storage Security operates in a balanced mode, optimizing both speed and functionality for maximum value. However, in scenarios where certain features may be unnecessary and system throughput is paramount, this guide will provide you with the necessary steps to adjust your settings for optimal performance outcomes.

Scanning Performance

The throughput of the system largely depends on the speed of the scan instance. If you are using MetaDefender Cloud, our SaaS offering, you are automatically benefiting from optimal performance. For users of MetaDefender Core, we recommend visiting the MetaDefender Core Performance Tuning Guide to explore ways to reduce the response time of your scan instance.

Increasing the number of concurrent file scans

A critical configuration parameter in MetaDefender Storage Security that significantly affects throughput is the number of concurrent scan requests sent to the scan instance. By default, the system processes 20 files simultaneously. As each file completes scanning, another is promptly queued. If your scan instance has additional capacity, we recommend gradually increasing this count to optimize throughput until you reach the desired performance level.

> **IMPORTANT**
>
> When adjusting the number of parallel scan requests, it's important to monitor the scan instance queue closely. An excessive number of messages in the queue indicates that the setting is too high, which can lead to performance degradation due to overloading the scan instance. Please adjust accordingly to maintain optimal performance.

To adjust this setting please modify the **RABBITMQ_SCANNING_PREFETCH_COUNT**environment variable as described in: Custom Configuration | Increase the number of concurrent scans.

Discovery Performance

Discovery Throttling

In scenarios where system load is exceptionally high, you may encounter a notification that discovery has been temporarily throttled. This measure halts the discovery of new files when the queue already contains a large volume of pending files. It is designed to optimize system resources and prevent excessive RAM usage. By setting this limit, we ensure efficient processing and reduce the risk of an endlessly growing queue, which can occur if the rate of file discovery exceeds the scanning capacity.

The threshold for maximum pending files is controlled by two environment variables:

- `DISCOVERY_THROTTLER_MAX_PENDING_FILES` , which defaults to 5000 for workflows with scanning
- `DISCOVERY_THROTTLER_NOT_SCANNED_MAX_PENDING_FILES` , which default to 50000 for discovery-only workflows

These values can be adjusted up or down based on your system's specifications to optimize memory usage.

When the system is throttled, it does not stop indefinitely - it waits a moment, then checks whether it is ready to continue. Two settings let you control how long it keeps trying before moving on:

- `DISCOVERY_THROTTLER_MAX_RETRY_ATTEMPTS` - the number of times the system will check before resuming discovery. Defaults to `10`.
- `DISCOVERY_THROTTLER_MAX_DELAY_MINUTES` - the maximum wait time between each check, in minutes. Defaults to `10`.

You can raise these values if you want the system to wait longer during periods of high load, or lower them if you prefer it to resume sooner.

Performance Mode

Several configuration options are available that can accelerate system performance while slightly reducing functionality. When certain features are not required, enabling **Performance Mode** can be an effective way to enhance speed.

To activate Performance Mode, modify the environment variable `PERFORMANCE_MODE`as outlined in Custom configuration and set the value to `PERFORMANCE_MODE=true` .

Functionality Impacted with Performance Mode Activated

Before enabling Performance Mode, please review the table below, which details all areas that will be impacted. This information will help you make an informed decision about whether the trade-offs in functionality are acceptable for your needs.

Functionality	Details
Amazon S3	Tags are not being retrieved automatically for all files which means they are not available on the response of `/api/file/{fileId}` and `/api/file/{scanId}/{startIndex}/{count}` APIs.
Core max file size validation	The WorkflowRules Job which updates cache with the max file size configured on each Core Instance will not run, and files which are bigger than this limit will be sent for scanning and fail

Rabbitmq configuration

One of the key ways to adjust MetaDefender Storage Security performance is by changing default configuration of Rabbitmq. The following are three environment variables that can help increase performance

`RABBITMQ_CONSUMER_MULTIPLIER`

Controls how many RabbitMQ consumers are created per standard (non-RPC) subscribe queue. The value is applied per priority tier with built-in multipliers:

- Low = N
- Medium = N × 3
- High = N × 6

When to increase:

When a service's standard queues show growing message backlogs and the service has spare CPU headroom. Each additional consumer adds a concurrent message-processing goroutine per priority channel.

Recommended values (16 CPU / 32 GB RAM VM): 1–3.

Valid range: 1–10. Default: 1.

`RABBITMQ_RPC_CONSUMER_MULTIPLIER`

Controls how many RabbitMQ consumers are created per RPC (request/reply) subscribe queue. RPC consumers handle synchronous-style calls from the API Gateway and other services, so they typically need higher concurrency than fire-and-forget consumers.

Valid range: 1–50. Default: 10.

Recommended values (16 CPU / 32 GB RAM VM): 10–20.

When to increase:

When RPC calls time out under load due to all consumers being busy, not because of slow downstream processing. Check RabbitMQ management for consumer utilization on RPC queues before raising this

`RABBITMQ_CONNECTION_CONSUMER_CONCURRENCY`

Sets **ConsumerDispatchConcurrency** on the RabbitMQ connection, which controls the maximum number of messages the client dispatches concurrently to consume callbacks on a single connection. This is a connection-level cap — it limits the total in-flight message processing across ALL channels and consumers on the connection.

Default: 50.

When to increase:

When the service has many queues/consumers (especially after raising ConsumerMultiplier or RpcConsumerMultiplier) and you observe that throughput plateaus while CPU is not used at full capacity. A low concurrency cap starves consumers even when there are messages ready.

Recommended values (16 CPU / 32 GB RAM VM): 50–100.

Configuring HTTPS

MetaDefender Storage Security can be configured to support HTTPS communication when accessing the Web UI and REST API.

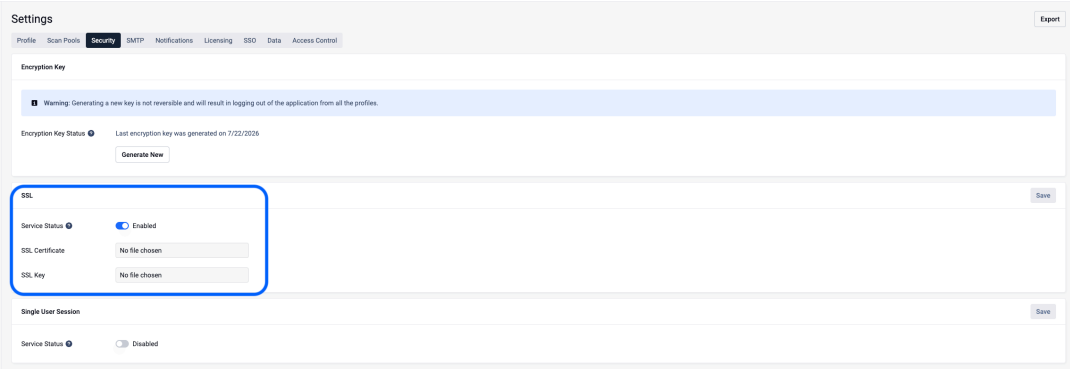
Which method applies to your deployment?

Deployment type	Method
Linux (Docker-based)	CLI — `mdss -u enable_https`
Windows (native services)	CLI — `mdss.ps1 -u enable_https`
Any (non-Kubernetes)	UI — Settings → Security tab
Kubernetes	See the Kubernetes deployment guide

Enabling HTTPS from the UI

Enabling HTTPS from the UI is now available for non Kubernetes deployments. On the Settings page, under the Security tab, you can enable HTTPS directly from the UI by entering the certificate file and the key file. The certificate must be a `.crt` file while the key must be a `.key` file.

After configuring HTTPS from UI, for the configuration to apply, restarting MDSS manually is required using the `mdss` command on Linux or the `mdss.ps1` script on Windows.




```
sudo mdss -c restart
```

```
.\mdss.ps1 -c restart
```

Enabling HTTPS from the Command Line

Prerequisites (both platforms)

- Certificate and private key must be in **PEM format** (`.crt` or `.pem` for the cert, `.key` for the key).
- If your certificate chain includes intermediate certificates, concatenate them into the `.crt` file (server cert first, then intermediates).
- Only one `.crt`/`.pem` and one `.key` file may be present in the certificate directory at a time.
- If no certificate is available, skip the certificate placement steps — the script will auto-generate a self-signed certificate (Linux only; Windows requires a certificate to be provided).

> **Info**
>
> **Self-signed certificates**: Browsers will show a trust warning for self-signed certificates. This is expected in lab or internal environments. To suppress the warning, add the certificate to your browser's or OS's trust store.

Enable HTTPS on Windows-based deployments

In order to setup HTTPS communication please follow these steps:

1. Place your certificate (`.crt` or `.pem`) and key (`.key`) files in: `C:\Program Files\OPSWAT\MetaDefender Storage Security\config\nginx\certificates` The files can have any name — the script renames them to `ssl.crt` and `ssl.key` automatically.
2. Navigate to the scripts folder: `C:\Program Files\OPSWAT\MetaDefender Storage Security\scripts`
3. Open a PowerShell prompt and run:

```
PS> .\mdss.ps1 -u enable_https
```

4. The script restarts the services automatically. Verify HTTPS is working by navigating to `https://<your-host>:443` (or the port configured in `WEB_PORT_SSL` in your .env file).
5. Verify that HTTPS is configured correctly by navigating with your browser.

Disable HTTPS on Windows-based deployments

For disabling HTTPS communication please follow these steps:

1. Navigate to the *scripts* folder (by default, in *C:\Program Files\OPSWAT\MetaDefender Storage Security\scripts*)
2. Run the following command in PowerShell:

```
PS> .\mdss.ps1 -u disable_https
```

3. The script patches the nginx configuration and restarts services automatically. The certificate files remain in `config\nginx\certificates` — they are not deleted.

Enable HTTPS on Unix-based deployments

In order to setup HTTPS communication please follow these steps:

1. Place your certificate (`.crt` or `.pem`) and key (`.key`) files in `/etc/mdss/webclient/` The files can have any name — the script renames them to `ssl.crt` and `ssl.key` automatically.
2. Run:

```
sudo mdss -u enable_https
```

3. The script reloads nginx automatically — **no MDSS restart is required**. Verify HTTPS is working by navigating to `https://<your-host>:443` (or the port configured in `WEB_PORT_SSL` in your .env file)
4. Verify that HTTPS is configured correctly by navigating with your browser.

Disable HTTPS on Unix-based deployments

> **Info**
>
> **Note:** Disabling HTTPS on Linux removes the certificate files from `/etc/mdss/webclient/`. Back up your certificates beforehand if needed.

For disabling HTTPS communication, run the following command:

```
sudo mdss -u disable_https
```

The script reloads nginx automatically — no restart required.

Port configuration

By default, MDSS listens on:

- **HTTP**: port `80`

- **HTTPS**: port 443

If ports 80 or 443 are unavailable, you can change them in your .env file:

```
WEB_PORT=8080
WEB_PORT_SSL=8443
```

Custom CA certificates

Custom certificates can be loaded in each MDSS container by placing them in the appropriate configuration directory depending on the host OS. If the directory does not exist, an empty one is created at startup. Any number of certificates can be placed in the directory and they will be loaded into MDSS.

Configuring custom certificates on Linux

Certificates are loaded from the following path:

```
/etc/mdss/ca_certificates/
```

Notes:

- All certificates should be in .crt format
- The full SSL certificate chain should be loaded in MDSS, especially if the certificate wasn't issued by a trusted CA
- Root CAs and intermediate CAs can be loaded using the same procedure

Configuring custom certificates on Windows

Certificates are loaded from the host certificate store. In order to add new CA certificates you'll need to do the following:

- Obtain the CA Certificate: Get the root CA certificate file (usually a .cer or .crt file) from your CA administrator.
- Open Certificate Manager:
- Press the Windows Key + R, type certlm.msc, and press Enter. (This opens the local machine certificate store)
- Import the Certificate:
- Expand the Trusted Root Certification Authorities folder.
- Right-click on Certificates.
- Select All Tasks -> Import.

- Follow the Certificate Import Wizard, browsing to the CA certificate file. Make sure to place the certificate in the Trusted Root Certification Authorities store

Enable Real-Time Processing (RTP)

Real-Time Processing is a scan type that processes files immediately upon upload.

This scan mode offers two file discovery methods:

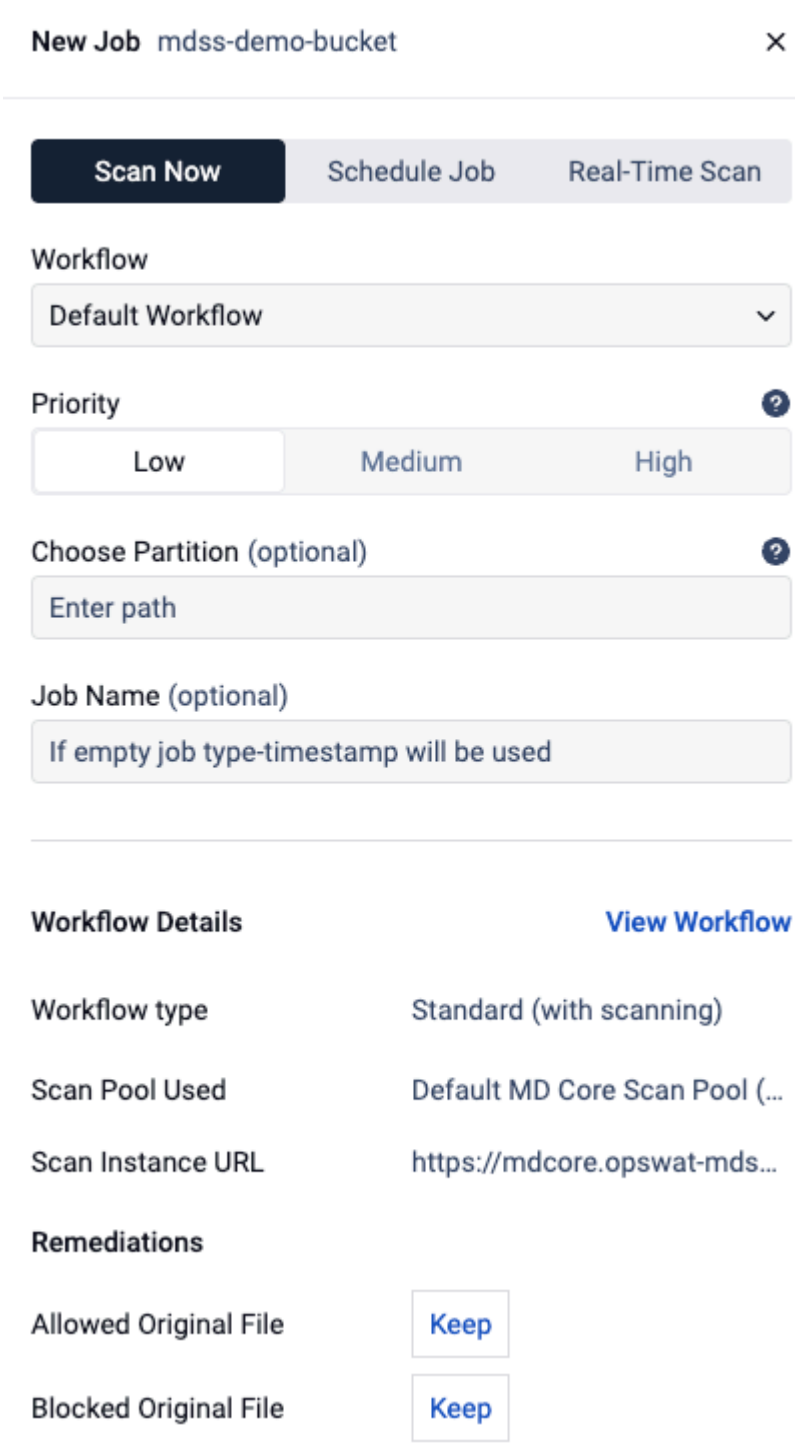
- Polling
- Event-Based

This feature is disabled by default. To enable it:

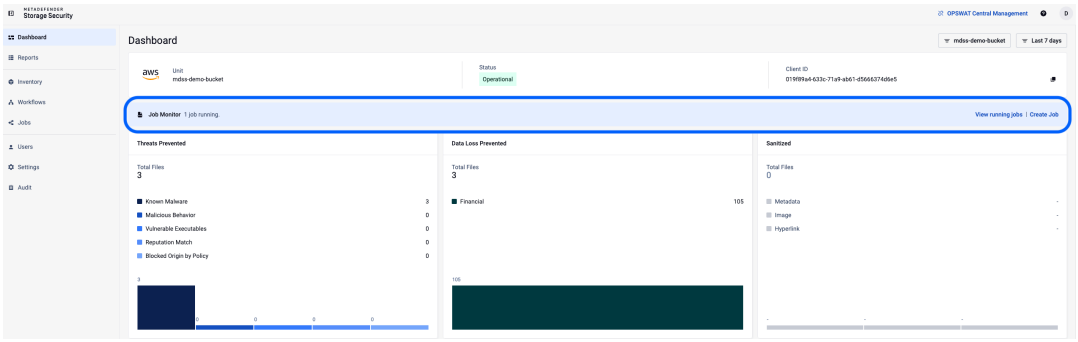
1. In the MetaDefender Storage Security web interface, navigate to "**Storage Units**."
2. Select the desired storage unit and click the "More Options" button (represented by three dots) to choose "**Add New Scan**."

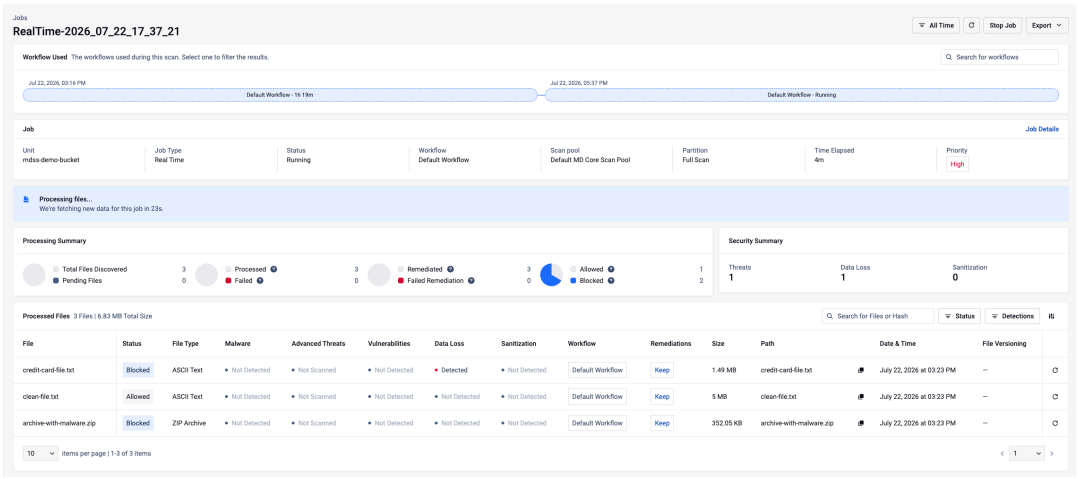
3. Select "Real-Time Processing" and choose the file discovery method (Polling or Event-Based, if the storage supports event-based real-time processing webhook invocation).

> **Info**
>
> Pick files modified since (Optional) field is a date/time picker that defines a cutoff point for file discovery; by default, only new files are picked up, but selecting a date causes the scan to also include files that were modified on or after that date and time.
>
> The field is **disabled** in the following cases:
>
> - `DISCOVERY_SERVICE_<PROTOCOL>_RTP_HANDLING = 1` (for SMB, SFTP, NFS or FTP)
> - **Event-based RTP**: event-based processing handles all incoming events regardless of file modification date, so this filter is not applicable.



To monitor active processes, go to the “Dashboard”, select your storage unit and click the “Active Process” card. This will open the scan details page, providing an overview of scan settings, discovery, and options to stop the scan, view, export, or clear the scan report from the audit log.





> **Info**
>
> Starting a second RTP scan on the same storage partition will update the existing scan and not start one more. Each storage supports multiple RTP scans, but only one RTP scan for each defined partition.

Polling handling

The **Polling file discovery method** retrieves and processes new or modified files directly from the storage.

Key characteristics include:

- Each identified file is processed independently.
- Files are processed only once, preventing redundant scans.
- The storage is checked for updates **every 10 seconds**, or immediately after a complete storage discovery cycle.

> **Note:**
>
> Using the **Polling** method significantly increases the number of requests to the storage system, ensuring timely detection of file changes.

Special considerations for Dell Isilon / SMB Compatible Storage

Dell Isillon / Smb Compatible

Real-time processing for Dell Isilon and SMB-compatible storage units employs a distinct approach compared to other storage types.

The behavior of this processing mode is governed by environment variables.

Refer to the the custom configuration page for detailed information on these settings.

Event-based handling

> **Event-Based File Discovery**
>
> Event-based real-time processing triggers scans immediately upon file creation or modification notifications.
>
> This method is supported for the following storage units: **Amazon S3**, **S3 Compatible**, **Azure Blob**, **Alibaba Cloud**, **Google Cloud**, **Box**, **SMB Compatible**, **NFS**, **SFTP, FTP**, **SharePoint Online** and **NetApp ONTAP**.

Generic Webhook Endpoint

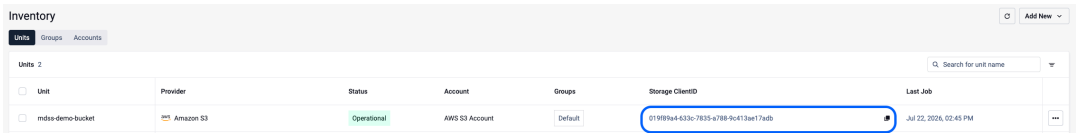
MetaDefender Storage Security provides a generic webhook endpoint **http[s]://{baseurl}/api/webhook/realtime** for integration with serverless functions from AWS, Azure, Alibaba Cloud, and Google Cloud Platform.

> **Important Note for BOX Integration**
>
> For Box, the API endpoint is `https://{baseurl}/api/webhook/realtime/public/{storageClientId}`. The generic webhook endpoint is **NOT** compatible with BOX. Refer to the Event-based handling for Box documentation for specific instructions.

Integrating Cloud Provider Event Notifications

To integrate event notifications from AWS, Azure, Alibaba Cloud, and Google Cloud Platform with MetaDefender Storage Security's real-time processing, the following parameters must be included as request headers in the event payload:

- `storageclientid` - Obtain this value from the Dashboard, once you select the Storage Unit desired



- `apikey`- Generate an API key for authentication in MetaDefender Storage Security under “Settings” -> “Users” -> “User (More Options).” Refer to the “Generate an API Key” documentation for detailed instructions.
- `apiendpoint`- Use the generic real-time webhook endpoint: `http(s)://{baseurl}/api/webhook/realtime`

For examples of the request format, please visit this page.

For detailed configuration instructions, please consult the Event-Based Real-Time Cloud Functions Examples documentation.

Event-Based request format

Overview

This document describes the real-time webhook API endpoint that external integrators can use to notify the system about file events. The endpoint accepts different payload formats depending on the storage type.

Endpoint Information

URL: `https://{baseurl}/api/webhook/realtime`

Method: POST

Content-Type: application/json

> **Important note for Box integration**
>
> For Box, the API endpoint is **https://{baseurl}/api/webhook/realtime/public/{storageClientId}.** Please refer to this article for specific instructions.

Request Format (with examples)

The request body differs based on storage type. Below are the specifications for each supported storage:

Amazon S3 and S3 Compatible

```
{
    "metadata": "{\"s3\": { \"object\": {\"key\": \"{Object Path}\"
}}}",
    "storageClientId": "{Storage Client Id}"
}
```

```
{
    "storageClientId": "0c5454e3-d576-44c5-8ce4-cdc078011d0b",
    "metadata": "{\"s3\": { \"object\": {\"key\": \"test-file.txt\"
}}}"
}
```

```
{
    "storageClientId": "0c5454e3-d576-44c5-8ce4-cdc078011d0b",
    "metadata": "{\"s3\": { \"object\": {\"key\":
\"folder1/folder2/test-file.txt\" }}}"
}
```

Alibaba Cloud

```
{
    "metadata": "{\"oss\": {\"object\": {\"key\": \"{Object
Path}\"}}}",
    "storageClientId": "{Storage Client ID}"
}
```

```
{
    "metadata": "{\"oss\": {\"object\": {\"key\": \"test-
file.pdf\"}}}",
    "storageClientId": "0c5454e3-d576-44c5-8ce4-cdc078011d0b"
}
```

```
{
    "metadata": "{\"oss\": {\"object\": {\"key\":
\"folder1/folder2/test-file.pdf\"}}}",
    "storageClientId": "0c5454e3-d576-44c5-8ce4-cdc078011d0b"
}
```

Google Cloud, OracleSdk, Azure Blob, SMB Compatible, NFS, SFTP, FTP and NetApp ONTAP

```
{
    "metadata": "{\"Name\": \"{Object Path}\" }",
    "storageClientId": \"{Storage Client ID}\"
}
```

```
{
    "metadata": "{\"Name\": \"test-file.exe\" }",
    "storageClientId": "0c5454e3-d576-44c5-8ce4-cdc078011d0b"
}
```

```
{
    "metadata": "{\"Name\": \"folder1/folder2/test-file.exe\" }",
    "storageClientId": \"0c5454e3-d576-44c5-8ce4-cdc078011d0b\"
}
```

Scanning with a partition

If you have one or more partition Event-Based scans active, a new parameter called `scanId` is required with every request and the format will become:

```
{
    "metadata": "{Metadata format specific for every storage}",
    "storageClientId": \"{Storage Client ID}\",
    "scanId": "Scan Id for the scan with partition"
}
```

Response Formats

Success

When a webhook is successfully processed, the system returns:

```
{
    "correlationId": "67ea85a36f2f5732a82834d0"
}
```

Field	Description
correlationId	A unique identifier that references the submitted file in our database. You can use this ID for tracking the file's processing status or for future API calls related to this file.

Error

When an error occurs during processing, the system returns:

```
{
    "responseKey": "REST_API_MSG_FAILED_STORAGE_RTP_NOT_ENABLED",
    "responseMessage": "Real-Time Process is not enabled."
}
```

Field	Description
responseKey	Machine-readable error code that identifies the specific error condition.
responseMessage	Human-readable explanation of the error.

Common error responses

Response Message	Description	Recommended Action
The storage could not be found	The specified storage client ID does not exist in the system.	Verify the storageClientId is correct and the storage has been properly configured in the system.
Real-Time Process is not enabled.	Real-time processing has not been activated for this storage client.	Enable real-time processing for the storage client in your account settings.
Real-Time Process is enabled, but it is not set to Event Based.	The storage is configured for real-time processing, but is using polling rather than event-based processing.	Stop the current RTP scan and start another one using Event-Based

Event-Based Real-Time Cloud Functions Examples

Google Cloud Function Setup

1. Configure the Cloud Run with the `google.cloud.storage.object.v1.finalizedtrigger` to process newly added objects.
2. Python Function Example:

```
import functions_framework
import json
import requests
import os

@functions_framework.cloud_event
def hello_gcs(cloud_event):
    try:
        name = cloud_event.data['protoPayload']['request']
['insertObjectSpec']['resource']['name']
    except KeyError as e:
        print(f"Missing key in event data: {e}")
        return

    request_body = {
        'storageClientId': os.getenv('STORAGECLIENTID'),
        'metadata': json.dumps({'name': name})
    }

    requests.post(
        os.getenv('APIENDPOINT'),
        headers={'ApiKey': os.getenv('APIKEY')},
        json=request_body
    )
```

3. Requirements.txt example:

```
functions-framework==3.*
requests
```

Alibaba Cloud Function Setup

1. Follow the official Alibaba Cloud documentation to create a compute function with an OSS trigger: https://www.alibabacloud.com/help/en/function-compute/latest/configure-an-oss-trigger
2. Specify the bucket to monitor and subscribe to the `oss:ObjectCreated:*` event.
3. Python Function Example:

```
import oss2, json, os
import requests

def handler(event, context):
    for eventRecord in json.loads(event)['events']:
        requests.post(os.getenv('APIENDPOINT', ""), headers=
{'ApiKey':os.getenv('APIKEY', "")}, json = {'metadata':
json.dumps(eventRecord), 'storageClientId':
os.getenv('STORAGECLIENTID', "") })
```

Wasabi Function Setup

1. Follow the official Wasabi documentation to create an event notification: https://docs.wasabi.com/v1/docs/event-notifications-bucket
2. Establish a connection with a service capable of sending requests to MetaDefender Storage Security (MDSS).
3. The Wasabi documentation example uses AWS SNS, which can be integrated with AWS Lambda (see Amazon S3 Lambda Function Setup: How do I configure Event Notifications on my Wasabi bucket using AWS SNS?

S3 Compatible function setup

- Event-based real-time processing configuration varies for different S3-compatible services.
- Most S3-compatible services offer event notification similar to Wasabi.
- The service must send a request to the MDSS endpoint: `http(s)://{baseurl}/api/webhook/realtime` with the appropriate request body:

```
{
    "storageClientId": [Storage Client Id],
    "metadata": "{'s3': { 'object': {'key': [Object Path] }}}"
}
```

Event-based handling for Box

MetaDefender Storage Security has a generic webhook endpoint that can be invoked through the Box Webhooks. These monitor Box content for events and notify you whenever an event occurs. There are two types of Box Webhooks: V1 and V2, but only V2 is supported by MetaDefender Storage Security

> **Webhook limitation**
>
> Box V2 webhooks only support webhooks set on specific files/folders and not ones set at the root. Because of that limitation from Box itself, documented on this page, event-based scans work only on Box storage units set to a specific user or custom app (optionally narrowed with a folder path). Storage units set to all users or to a specific group do not receive events.

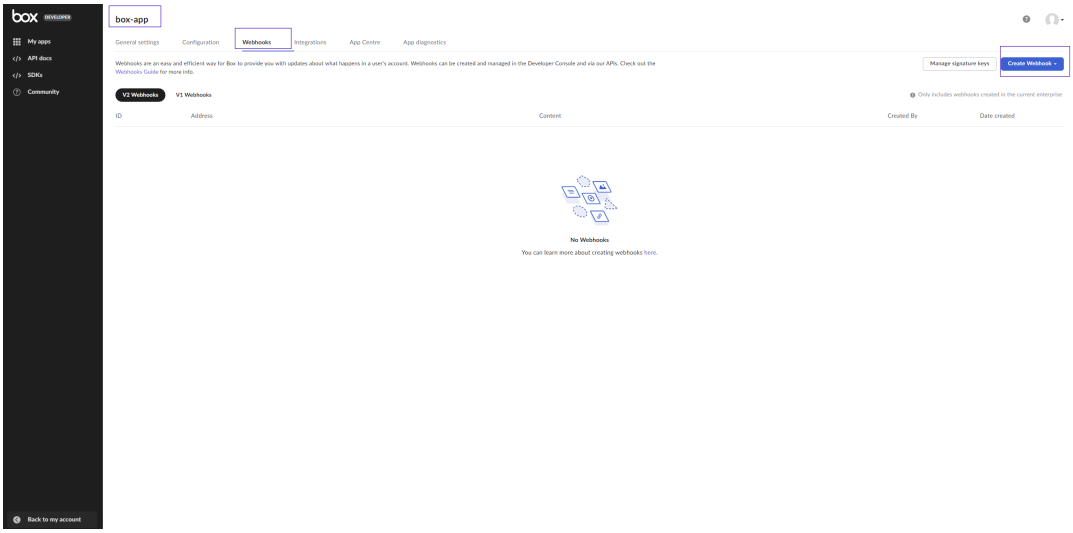
How to create Box Webhooks

1. Log in to Box and then go to Developer Console

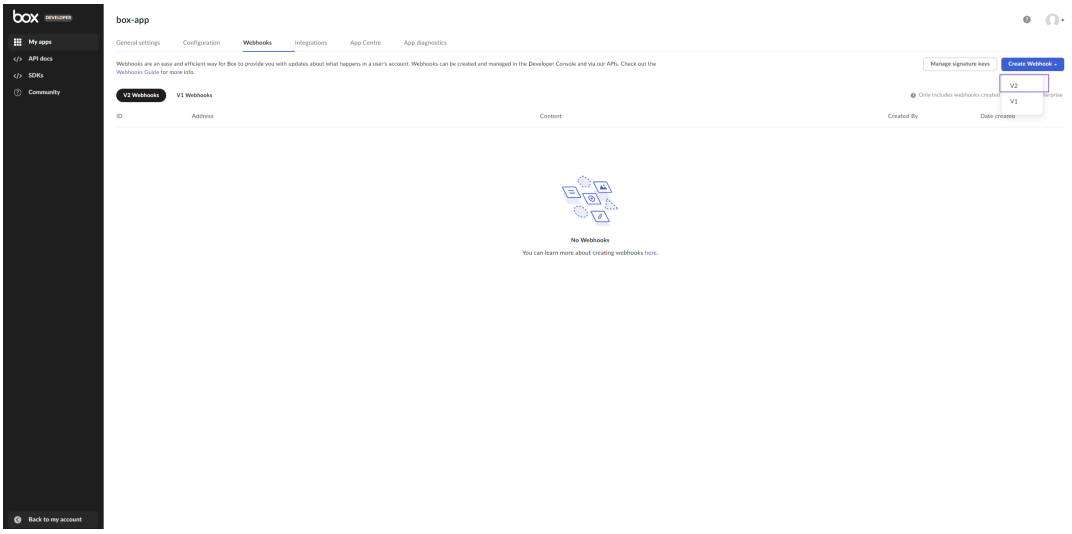
2. If you do not have an existing app, create a new one (Please check How can I generate a public/private key pair)

> **WARNING**
>
> Box V2 Webhooks can only be created when the scope Manage Webhooks is selected and the application has been authorized

3. Navigate to your application
4. Select the **Webhooks** tab
5. Click the **Create webhook** button

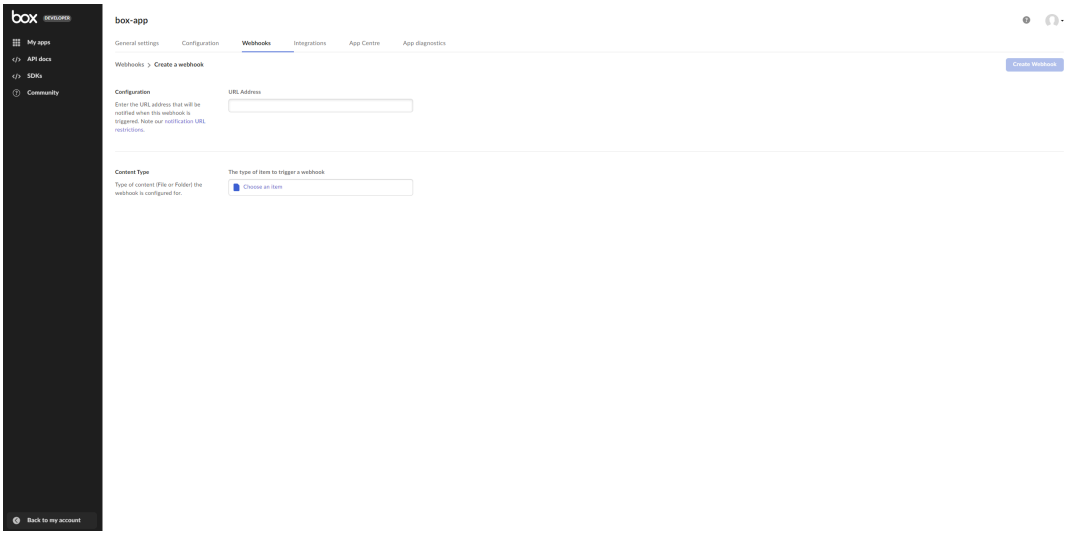


6. Select **V2** from the drop-down list



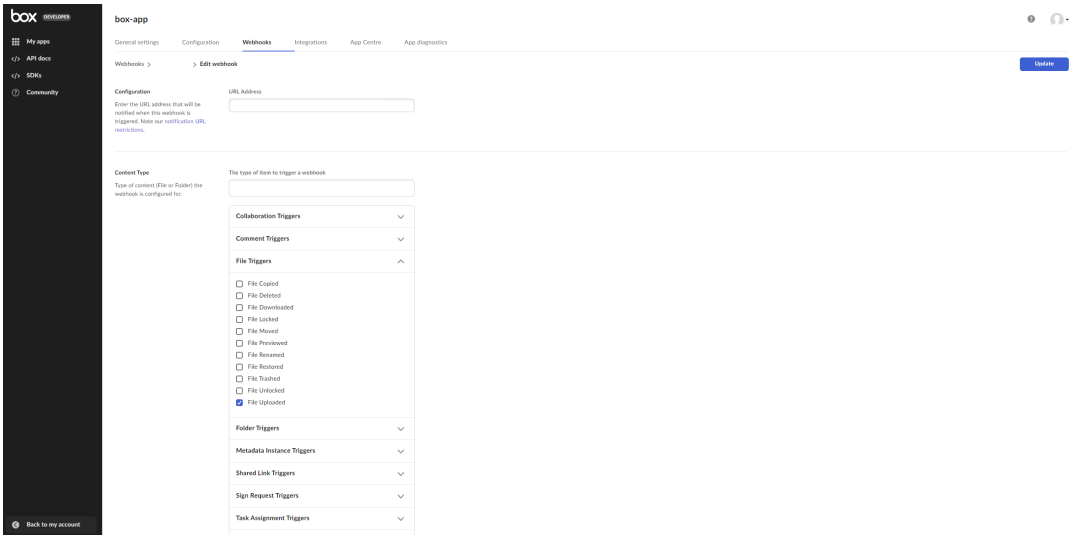
7. Fill in the form:
    1. Enter the **URL address** that will be notified when the webhook is triggered (the address where MetaDefender Storage Security is hosted)

> **WARNING**
>
> URL Address must have a secure URL (HTTPS)

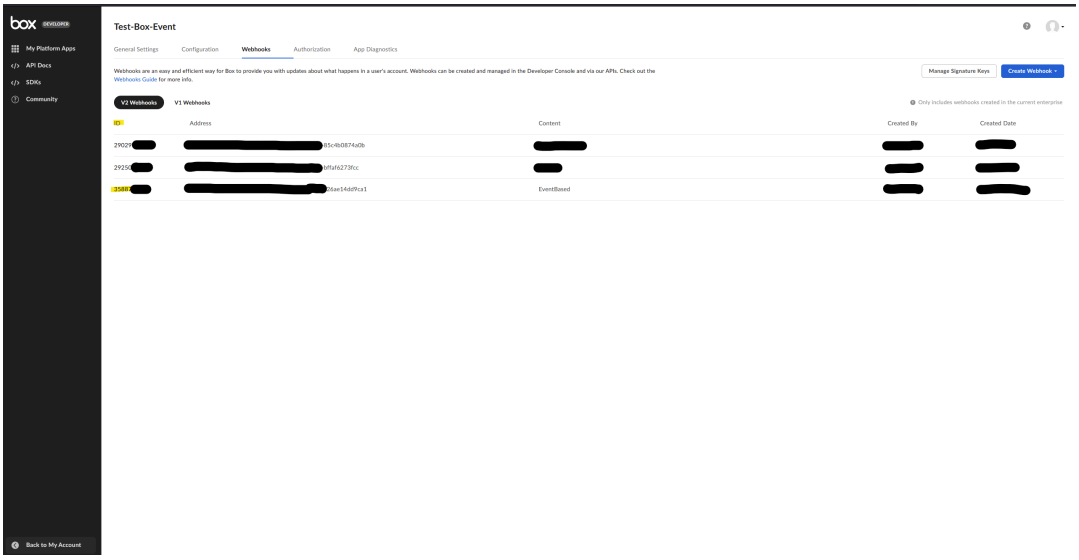


b. Select the folder that you have set in MetaDefender Storage Security as the element that triggers the webhook

c. Choose **File Uploaded** as the trigger for the selected folder



8. Click the Create webhook button to save your changes
9. **Copy the ID** of the generated webhook to enter into MDSS



10. Enter the copied webhook ID into the edit storage form for your Box storage

> **WARNING**
>
> If an event-based scan is started on a Box storage unit but the webhook ID on the requests does not match the webhook ID configured on the Box account, a warning will be shown on the scan.

Event-based handling for SharePoint Online

MetaDefender Storage Security supports event-based handling for SharePoint Online, enabling real-time scanning of content as it is added.

Unlike other storage providers, **MDSS automates the setup of the Microsoft Graph subscriptions** needed for monitoring SharePoint events. This makes the configuration process simpler and faster for SharePoint Online users.

Advantages

- MDSS automatically creates, renews, and manages all necessary Microsoft Graph API subscriptions. **No manual intervention is required for this critical infrastructure**
- Get real-time scanning up and running faster, without the complexities of manual Graph API configuration
- Files are scanned immediately upon upload or modification in SharePoint Online, providing continuous security

How it works

MDSS leverages Microsoft Graph API webhooks to receive notifications from SharePoint Online whenever content is added or modified. MDSS handles the entire lifecycle of these webhook subscriptions on your behalf. For this to work correctly, your MDSS instance must be reachable from the internet to receive these notifications from Microsoft.

Prerequisites

Before MDSS can automatically configure event-based handling, your environment must meet the following crucial prerequisites:

1. MDSS Hosted and Accessible via HTTPS
    - SharePoint Online sends event notifications (webhooks) to your MDSS instance. These notifications require a secure HTTPS connection to a trusted endpoint
    - Your MDSS instance must be served over HTTPS
    - The machine running MDSS needs a valid SSL certificate from a trusted Certificate Authority. While self-signed certificates can technically work, they are **strongly discouraged** as they require complex trust configurations within your SharePoint environment and can lead to notification failures if not managed perfectly
2. Publicly Accessible MDSS Port
    - Microsoft Graph API needs to send event notifications directly from Microsoft's cloud services to your MDSS instance. If MDSS is not publicly accessible, these critical notifications cannot be delivered
    - The HTTPS port used by MDSS (default is 443) must be open and reachable from the public internet
    - Configure your firewall, router, load balancers, and any other network security infrastructure to allow incoming HTTPS traffic on this port, directing it to the MDSS host machine
3. Configured SharePoint Online Connection in MDSS
    - MDSS needs to authenticate with your SharePoint Online environment to register the webhooks for event notifications and access file content for scanning
    - Your SharePoint Online storage must be successfully added and properly configured within MDSS.
    - Make sure that MDSS has the necessary permissions to access your SharePoint site(s). These permissions are typically granted via OAuth during the initial storage addition process and must include the ability to manage Graph API subscriptions for the targeted resources

> **Warning**
>
> The document library's display name must match its name in the web address (URL). For example, if the URL is `.../sites/[SiteName]/[LibraryName]/`, the name should also be **[LibraryName]**. If they differ, SharePoint event-based handling will not work, and MDSS won't process file events for that library.

Once these prerequisites are met, MDSS will automatically handle the setup and ongoing management of event-based scanning for your configured SharePoint Online storage

Unsupported SharePoint Content Types

Certain SharePoint content types fall outside the scope of items protected by our solution. Administrators should be aware of the following exclusions when reviewing coverage of their SharePoint environments.

Form Templates Library

The **Form Templates** library that appears under a SharePoint site's contents is a hidden system library automatically provisioned by SharePoint on every site. Despite its visibility in the site contents view, it is not a standard document library and is not backed by a Drive resource in Microsoft Graph. As a result, this library is **not protected** by our solution.

Lists

SharePoint items of type **List** (for example, custom lists, task lists, and contact lists) are not backed by a Drive resource in Microsoft Graph and are therefore **not protected** by our solution.

Site Pages and Other Page Libraries

Content stored in libraries of type **Page Library** (such as the default *Site Pages* library) is **not protected** by our solution. These libraries host SharePoint pages rather than user documents and are managed through SharePoint's page infrastructure rather than as standard document storage.

Content Type	Microsoft Graph Drive	Protected
Document Library	Yes	Yes
Form Templates (system library)	No	No
List	No	No
Page Library (e.g., Site Pages)	No	No

Event-Based Real-Time handling for AWS

AWS Lambda Function with S3 Trigger Setup Guide

Before proceeding with the setup, ensure the following requirements are met:

1. IAM User Permissions

- Your IAM user must have the necessary permissions to:
  - Create Lambda functions
  - Assign IAM roles
  - Configure S3 triggers
  - Access the target S3 buckets

2. Account-Level Verification

- Confirm there are no account-level naming restrictions that would prevent your chosen Lambda function name
- Verify your AWS account has sufficient permissions and service quotas to create new Lambda functions in the desired region
- Ensure you have access to the target AWS region where the Lambda function will be deployed

3. S3 Bucket Configuration

- Verify that S3 bucket policies are configured to allow access from the Lambda function's execution role
- Required S3 permissions may include:
  - `s3:GetObject` - to read objects from the bucket
  - `s3:PutObject` - to write objects to the bucket (if applicable)
  - `s3:DeleteObject` - to delete objects from the bucket (if applicable)

4. IAM Execution Role

- Ensure you have a pre-configured IAM role with:
  - Lambda execution permissions
  - S3 access permissions for your target bucket(s)
  - CloudWatch Logs permissions for function monitoring

Step 1: Creating the Lambda Function

1.1 Navigate to Lambda Service

1. Open the AWS Management Console
2. Navigate to the **Lambda** service

1.2 Initialize Function Creation

1. Click **"Create function"**
2. Select **"Author from scratch"**

1.3 Configure Basic Settings

1. **Function name**: Enter a descriptive name for your Lambda function
   - Example: `MetaDefenderStorageSecurityProcessor`
   - Ensure the name complies with AWS naming conventions and any account-specific policies
2. **Runtime**: Select **Python** (choose the latest compatible version)

1.4 Configure Permissions

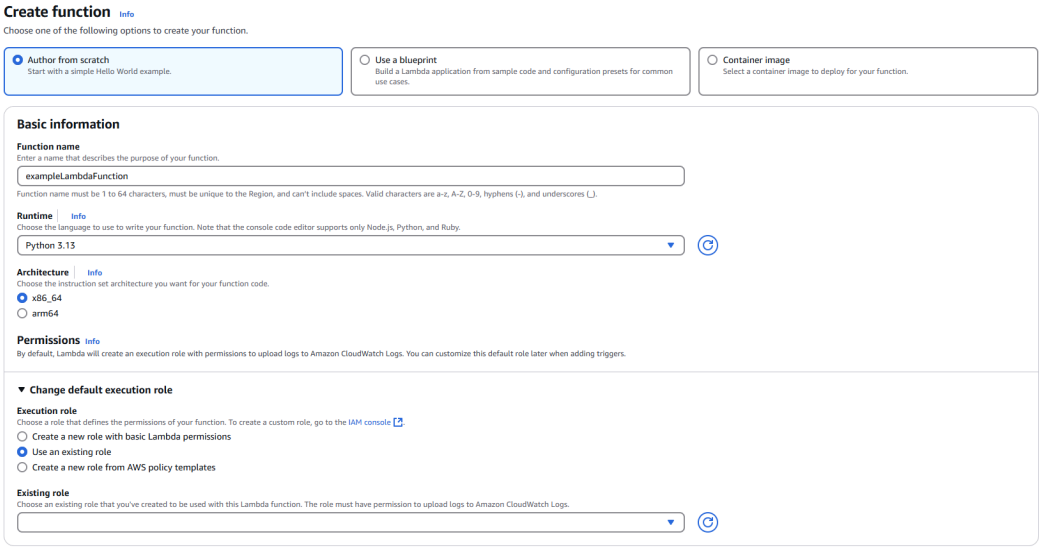
1. Under **"Change default execution role"**, select **"Use an existing role"**
2. From the **"Existing role"** dropdown, choose your pre-configured IAM role
3. **Verify role permissions**:
   - Navigate to the IAM console and review the selected role
   - Confirm **Permission policies** include necessary access to S3, CloudWatch Logs, and other required services
   - Verify **Trust relationships** allow the Lambda service (`lambda.amazonaws.com`) to assume the role

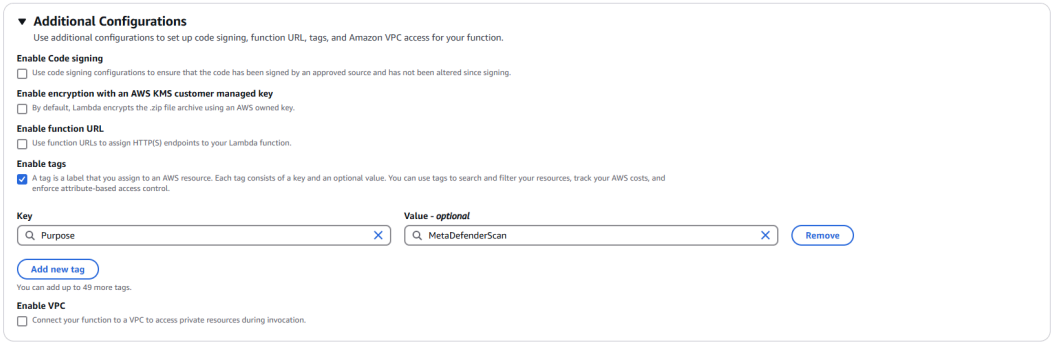
1.5 Configure Advanced Settings (Optional but Recommended)

1. Expand **“Advanced configuration”**
2. Enable **“Tags”**
3. Add relevant tags for resource organization:
   - **Key**: `Purpose` | **Value**: `MetaDefenderScan`
   - **Key**: `Environment` | **Value**: `Production/Development`
   - **Key**: `Owner` | **Value**: `[Your Team/Department]`

1.6 Create the Function

1. Click **“Create function”**
2. Wait for the function to be successfully created





Step 2: Adding S3 Trigger

2.1 Add Trigger Configuration

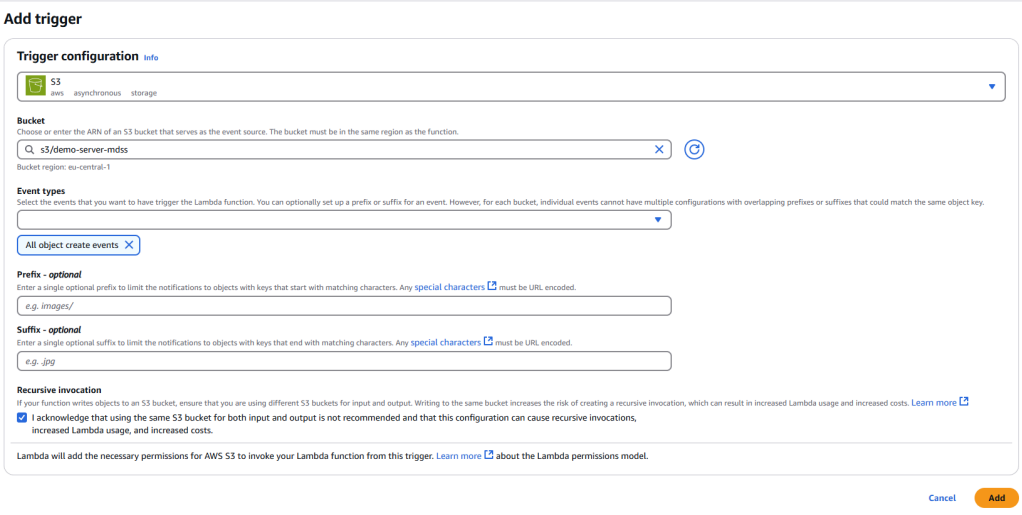
1. In your newly created Lambda function, click **“Add trigger”**
2. Select **“S3”** as the trigger source

2.2 Configure S3 Trigger Settings

1. **Bucket**: Select your target S3 bucket from the dropdown menu
2. **Event types**: Choose the appropriate event type
   - Default: **“All object create events”**
   - Alternative options: Object create, delete, or restore events based on your requirements
3. **Prefix** (Optional): Specify a prefix to filter objects by path
   - Example: `uploads/` to only trigger on objects in the uploads folder
4. **Suffix** (Optional): Specify a suffix to filter objects by file extension
   - Example: `.pdf` to only trigger on PDF files
5. **Recursive invocation**: Check this option to acknowledge potential recursive invocations

2.3 Finalize Trigger Setup

1. Review your trigger configuration
2. Click **“Add”** to create the S3 trigger



3.1 Navigate to Code Section

1. In your Lambda function console, navigate to the **“Code”** tab

2. Replace the default code with the following implementation:

```
import os
import json
import urllib.request
from urllib.parse import unquote

def lambda_handler(event, context):
    for eventRecord in event['Records']:
        eventRecord['s3']['object']['key'] = unquote(eventRecord['s3']
['object']['key'].replace("+", " "))

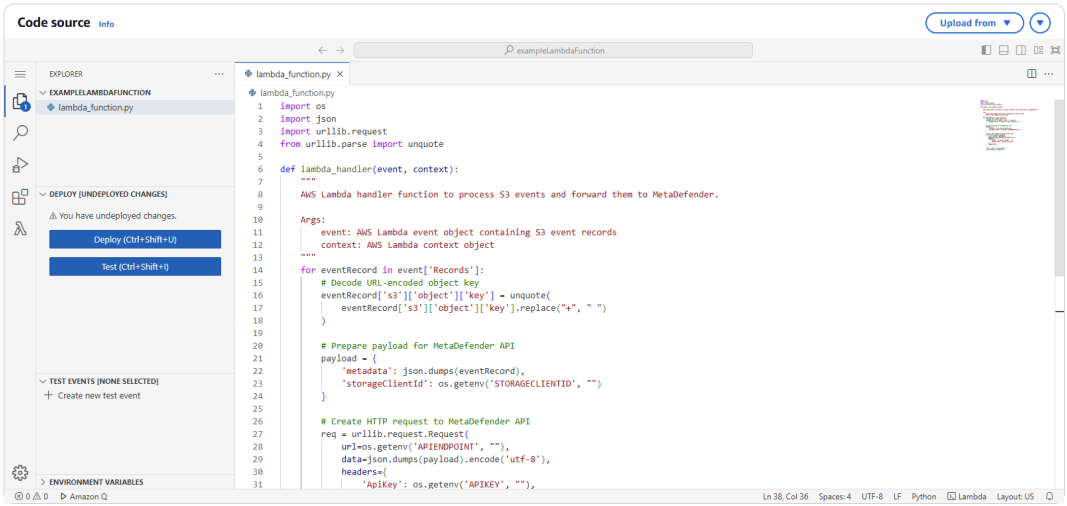
        payload = {
            'metadata': json.dumps(eventRecord),
            'storageClientId': os.getenv('STORAGECLIENTID', "")
        }

        req = urllib.request.Request(
            url=os.getenv('APIENDPOINT', ""),
            data=json.dumps(payload).encode('utf-8'),
            headers={
                'ApiKey': os.getenv('APIKEY', ""),
                'Content-Type': 'application/json'
            },
            method='POST'
        )

        urllib.request.urlopen(req)
```

3.2 Deploy the Code

1. Click **“Deploy”** to save and deploy your function code



Request Format (with examples)

The request body differs based on storage type. Below are the specifications for each supported storage:

Amazon S3 and S3 Compatible

```
{
    "storageClientId": '{Storage Client Id}',
    "metadata": "{'s3': { 'object': {'key': '{Object Path}' }}}"
}
```

```
{
    "storageClientId": "0c5454e3-d576-44c5-8ce4-cdc078011d0b",
    "metadata": "{'s3': { 'object': {'key': 'test-file.txt' }}}"
}
```

```
{
    "storageClientId": "0c5454e3-d576-44c5-8ce4-cdc078011d0b",
    "metadata": "{'s3': { 'object': {'key': 'folder1/folder2/test-
file.txt' }}}"
}
```

Response Formats

Success

When a webhook is successfully processed, the system returns:

```
{
    "correlationId": "67ea85a36f2f5732a82834d0"
}
```

Field	Description
correlationId	A unique identifier that references the submitted file in our database. You can use this ID for tracking the file's processing status or for future API calls related to this file.

Error

When an error occurs during processing, the system returns:

```
{
    "responseKey": "REST_API_MSG_FAILED_STORAGE_RTP_NOT_ENABLED",
    "responseMessage": "Real-Time Process is not enabled."
}
```

Field	Description
responseKey	Machine-readable error code that identifies the specific error condition.
responseMessage	Human-readable explanation of the error.

Common error responses

Response Message	Description	Recommended Action
The storage could not be found	The specified storage client ID does not exist in the system.	Verify the storageClientId is correct and the storage has been properly configured in the system.
Real-Time Process is not enabled.	Real-time processing has not been activated for this storage client.	Enable real-time processing for the storage client in your account settings.
Real-Time Process is enabled, but it is not set to Event Based.	The storage is configured for real-time processing, but is using polling rather than event-based processing.	Stop the current RTP scan and start another one using Event-Based

Step 4: Configuring Environment Variables

4.1 Navigate to Configuration

1. In your Lambda function console, click on the **“Configuration”** tab
2. Select **“Environment variables”** from the left sidebar

4.2 Add Required Environment Variables

Click **“Edit”** and add the following environment variables:

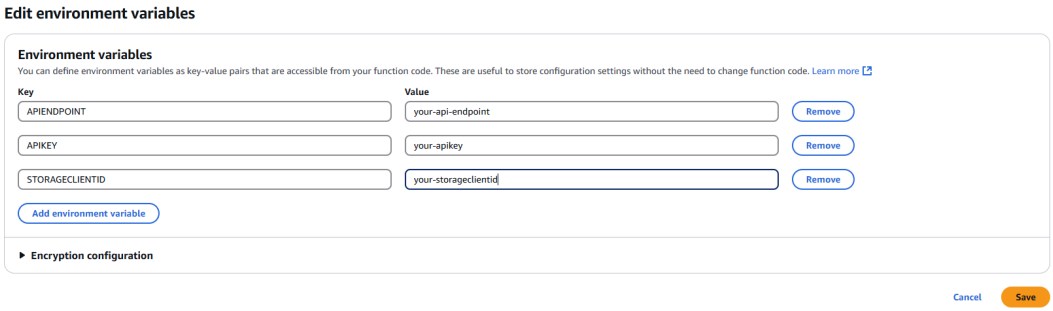
1. **APIENDPOINT**
   - **Key**: `APIENDPOINT`
   - **Value**: Your MDSS URL + `/api/webhook/realtime`
   - **Example**: `https://mdss-example.com/api/webhook/realtime`
2. **APIKEY**

- **Key**: `APIKEY`
- **Value**: Your MDSS user API key
- **Note**: Ensure this key has appropriate permissions for webhook operations

3. **STORAGECLIENTID**
   - **Key**: `STORAGECLIENTID`
   - **Value**: Your storage client ID from MDSS
   - **To obtain**: Navigate to your desired storage configuration and copy the `storageClientId`

4.3 Save Configuration

1. Click **"Save"** to apply the environment variable changes



Step 5: Testing and Validation

5.1 Test the Function

1. Upload a test file to your configured S3 bucket
2. Monitor the Lambda function's execution in the **"Monitor"** tab
3. Check CloudWatch Logs for any execution errors or successful processing

5.2 Verify Integration

1. Confirm that file events are being sent to your MDSS instance
2. Verify that scans are initiated as expected
3. Review MDSS logs for successful webhook reception

Event-Based Real-Time handling for Azure

Azure Blob function app setup

1. Deploy the Azure function app using the Terraform script: https://github.com/OPSWAT/metadefender-k8s/tree/main/terraform/azure-function-docker
2. Configure `STORAGECLIENTID`, `APIKEY`, and `APIENDPOINT` variables in the `.tvars` file:

```
resource_group_name = "" #The name of the resource group in which the
function app will be created."
service_plan_name = "" #The name of the app service plan
storage_account_name = "" #The name of the storage account to be
created
docker_registry_server_url = ""
docker_registry_server_username = "" #optional
docker_registry_server_password = "" #optional
docker_image_name = ""
docker_image_tag = ""
AzureWebJobsBlobTrigger = "" #The storage account connection string
that triggers the function
CONTAINERNAME = "" #The blob container that needs to be scanned
fn_name_prefix = "" #function name
location = "" #azure region
STORAGECLIENTID  = ""
APIKEY = ""
APIENDPOINT = ""
```

Request Format (with examples)

The request body differs based on storage type. Below are the specifications for each supported storage:

Amazon S3 and S3 Compatible

```
{
    "storageClientId": '{Storage Client Id}',
    "metadata": "{'s3': { 'object': {'key': '{Object Path}' }}}"
}
```

```
{
    "storageClientId": "0c5454e3-d576-44c5-8ce4-cdc078011d0b",
    "metadata": "{'s3': { 'object': {'key': 'test-file.txt' }}}"
}
```

```
{
    "storageClientId": "0c5454e3-d576-44c5-8ce4-cdc078011d0b",
    "metadata": "{'s3': { 'object': {'key': 'folder1/folder2/test-
file.txt' }}}"
}
```

Response Formats

Success

When a webhook is successfully processed, the system returns:

```
{
    "correlationId": "67ea85a36f2f5732a82834d0"
}
```

Field	Description
correlationId	A unique identifier that references the submitted file in our database. You can use this ID for tracking the file's processing status or for future API calls related to this file.

Error

When an error occurs during processing, the system returns:

```
{
    "responseKey": "REST_API_MSG_FAILED_STORAGE_RTP_NOT_ENABLED",
    "responseMessage": "Real-Time Process is not enabled."
}
```

Field	Description
responseKey	Machine-readable error code that identifies the specific error condition.
responseMessage	Human-readable explanation of the error.

Common error responses

Response Message	Description	Recommended Action
The storage could not be found	The specified storage client ID does not exist in the system.	Verify the storageClientId is correct and the storage has been properly configured in the system.
Real-Time Process is not enabled.	Real-time processing has not been activated for this storage client.	Enable real-time processing for the storage client in your account settings.
Real-Time Process is enabled, but it is not set to Event Based.	The storage is configured for real-time processing, but is using polling rather than event-based processing.	Stop the current RTP scan and start another one using Event-Based

Azure Blob Event Grid RTP configuration

Refer to the example for detailed configuration: https://github.com/OPSWAT/metadefender-k8s/tree/main/terraform/CloudFunctions/Azure/webhook-notification

> **WARNING**
>
> Event Notifications for Page and Append blob are **NOT** supported.
>
> Events for these blob types are triggered upon the first block commit, potentially before the upload is complete.

Event-Based Real-Time handling for OracleSDK

Oracle Rule with custom Function trigger Setup Guide

This guide explains how to configure **Oracle Cloud Functions** and **Object Storage Events** to enable **real-time event-based processing** in MetaDefender Storage Security (MDSS) when using the **Oracle Native SDK** integration.

Prerequisites

Before proceeding with the setup, ensure the following requirements are met.

1. Required Permissions

Your Oracle Cloud Infrastructure (OCI) environment only needs the following **IAM permissions if you plan to deploy the function from an OCI repository**. Deploying from the repository is optional; these permissions are **not required** if you use other deployment methods.

Dynamic Group Permissions
Allow dynamic-group <your_dynamic_ group> to to read repos in tenancy
Allow dynamic-group <your_dynamic_ group> to use repos in tenancy

These permissions enable your function to **read and execute code from the OCI repository**.

2. Network and Access Requirements

- The function's subnet must allow outbound access to the public MDSS domain. No specific protocol (HTTP/HTTPS) is enforced by MDSS.
- Ensure your **Object Storage bucket** is in the same **tenancy and region** as your Function Application (recommended for lower latency).
- Verify that your **Object Storage bucket emits object events** (see Step 5).

Step 1: Create an Application and Function

1. In the Oracle Cloud Console, navigate to **Developer Services → Functions → Applications**.
2. Click **Create Application** and provide:
   - **Name**: e.g. `MetaDefenderStorageProcessorApp`
   - **VCN/Subnet**: Select a subnet with outbound internet access.
3. Once created, open your application and click **Create Function**.
4. Set:
   - **Function Name**: `MetaDefenderStorageSecurityProcessor`
   - **Runtime**: **Python**
   - **Entry Point**: `handler`

You can create and deploy your function either from the **OCI Console Code Editor** or using the **OCI CLI / Fn Project CLI**.

Step 2: Implement Function Code

Use the following Python function implementation. This code listens to Object Storage create events and sends real-time metadata to your MDSS instance for processing.

```
import io
import os
import json
import urllib.request
from urllib.parse import unquote
from fdk import response

def handler(ctx, data: io.BytesIO = None):
    event_body = json.loads(data.getvalue())
    object_key = event_body.get('data', {}).get('resourceName', '')
    object_key = unquote(object_key.replace("+", " "))

    payload = {
        "metadata": json.dumps({"Name": object_key}),
        "storageClientId": os.getenv("STORAGECLIENTID", "")
    }

    req = urllib.request.Request(
        url=os.getenv("APIENDPOINT", ""),
        data=json.dumps(payload).encode("utf-8"),
        headers={
            "ApiKey": os.getenv("APIKEY", ""),
            "Content-Type": "application/json"
        },
        method="POST"
    )

    urllib.request.urlopen(req, timeout=5)

    return response.Response(
        ctx,
        response_data=json.dumps({"status": "ok"}),
        headers={"Content-Type": "application/json"}
    )
```

Step 3: Configure Function Environment Variables

1. Open your **Function details** page in the Oracle Cloud Console.
2. Click **Configuration → Environment Variables → Edit**.
3. Add the following key-value pairs:

Key	Value	Description
APIENDPOINT	Your MDSS URL + `/api/webhook/realtime`	The MDSS real-time processing API endpoint
APIKEY	Your MDSS user API key	Your MDSS API key with webhook permissions
STORAGECLIENTID	Your storage client ID from MDSS	Navigate to your desired storage configuration and copy the `storageClientId` in order to obtain

4. Click **Save Changes**.

Step 4: Deploy the Function

If you are using the Code Editor in the OCI Console, click **Deploy**.

If you are using the CLI:

```
fn deploy --app MetaDefenderStorageProcessorApp
```

Once deployment completes, confirm your function appears in the **Functions** list and has an active **endpoint**.

Step 5: Create Object Storage Rule and Enable Event Emission

You will now link your Object Storage bucket to your Oracle Function so that MDSS receives file events in real time.

5.1 Create a Rule

1. In the OCI Console, navigate to **Observability & Management → Events Service → Rules**.
2. Click **Create Rule** and configure:
   - **Name**: `ObjectCreateTriggerRule`
   - **Condition**:
     - **Service**: *Object Storage*
     - **Event Type**: *Object - Create*
   - **Actions**:

- Select **Functions** and choose your **MetaDefenderStorageSecurityProcessor** function.

3. Click **Create Rule**.

This ensures that every time a new object is created in your bucket, an event triggers the Function.

5.2 Enable Object Event Emission on the Bucket

1. Navigate to **Object Storage → Buckets → [Your Bucket Name]**.
2. Under **Events**, click **Emit Object Events** → **Enable**.
3. Save your configuration.

Step 6: Testing and Validation

1. Upload a new file to your configured **Object Storage bucket**.
2. Open the **Functions → Metrics & Logs** tab to confirm the function execution was triggered.
3. Check the **MDSS instance** to verify the event was received and the file processing started.
4. If successful, the MDSS dashboard should show the new file scan in progress.

Request and Response Format

Request Body (Example)

```
{
    "storageClientId": "{Storage Client Id}",
    "metadata": "{'Name': 'uploads/sample.pdf'}"
}
```

Successful Response Example

```
{
    "correlationId": "67ea85a36f2f5732a82834d0"
}
```

Field	Description
correlationId	A unique identifier that references the submitted file in our database. You can use this ID for tracking the file's processing status or for future API calls related to this file.

Error

When an error occurs during processing, the system returns:

```
{
    "responseKey": "REST_API_MSG_FAILED_STORAGE_RTP_NOT_ENABLED",
    "responseMessage": "Real-Time Process is not enabled."
}
```

Field	Description
responseKey	Machine-readable error code that identifies the specific error condition.
responseMessage	Human-readable explanation of the error.

Common error responses

Response Message	Description	Recommended Action
The storage could not be found	The specified storage client ID does not exist in the system.	Verify the storageClientId is correct and the storage has been properly configured in the system.
Real-Time Process is not enabled.	Real-time processing has not been activated for this storage client.	Enable real-time processing for the storage client in your account settings.
Real-Time Process is enabled, but it is not set to Event Based.	The storage is configured for real-time processing, but is using polling rather than event-based processing.	Stop the current RTP scan and start another one using Event-Based

Summary

After completing this setup:

- Oracle **Object Storage** emits events on object creation.
- **Oracle Cloud Function** receives the event and forwards it to **MDSS** via webhook.
- **MDSS** immediately begins scanning the uploaded object in real time.

This configuration enables secure, low-latency, event-driven protection for files stored in Oracle Object Storage.

Workflows

Workflows are configurable processing pipelines that define how files should be scanned, analyzed, and remediated. A workflow is required to be able to start a new scan with a specified set of rules. Each workflow consists of a set of remediation actions.

Remediation actions are automated post-processing tasks performed on files after they have been scanned. The following remediation options are currently available:

Name	Description	Documentation
File Tagging	Adds metadata tags to files in S3-like storage units, providing processing information.	File Tagging
Deep CDR	Applies OPSWAT's Deep CDR (file sanitization) technology to files.	Deep Content Disarm and Reconstruction (Deep CDR)
Keep, copy, move or delete	Keeps, copies, moves or deletes files based on configured rules, such as blocked or sanitized status.	Keep, Copy, Move or Delete Files
Other remediations	Delete a directory if it is empty after remediation (does not delete if the folder was empty to begin with).	Delete Empty Folders

REMEDIATION ACTION PROCESSING ORDER

The following order is used when applying remediation actions:

1. File Tagging
2. Sanitized Files (Deep CDR, Move or Delete)
3. Blocked Files (Keep, Copy, Move or Delete)
4. Allowed Files (Keep, Copy, Move or Delete)

NOTE

If you require a post-processing action not listed above, please let us know.

We may be able to add it in a future update.

File Tagging

IMPORTANT

File Tagging is only available for Amazon S3, S3-compatible storage units (where supported) and Google Cloud Platform.

Info

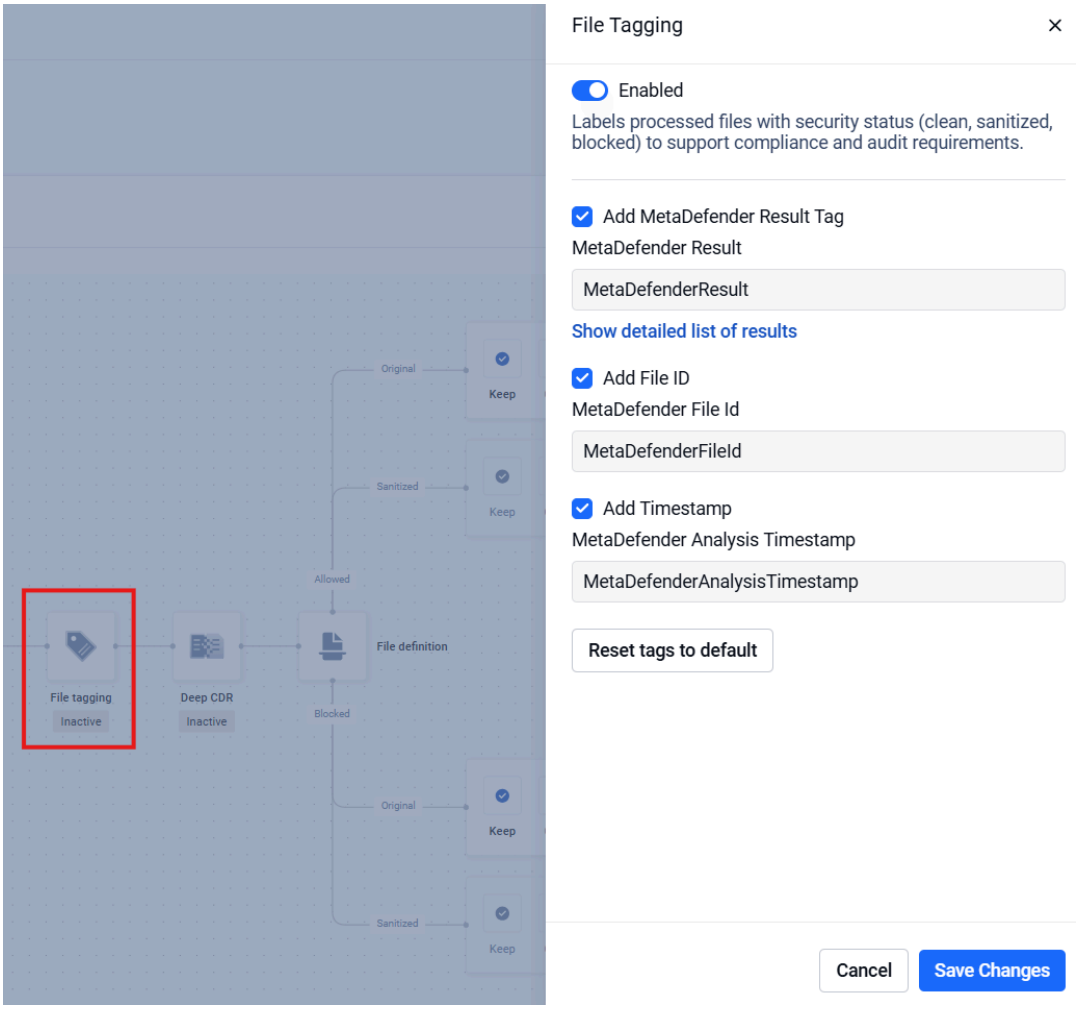
Google Cloud does not support applying tags directly to files; however, it does support metadata. Therefore, we treat metadata as tags and apply them in the same way as we apply tags in Amazon S3 and S3-compatible.

Warning

Metadata is preserved only when moving files between Google Cloud storage units.

Configuring File Tagging

- From the left-side navigation menu, go to **Workflows**
- Select the workflow for which you wish to configure **File Tagging** post action
- Select which tags to be added by checking/unchecking the boxes. (**Note:** at least one tag should be enabled if File Tagging is on)

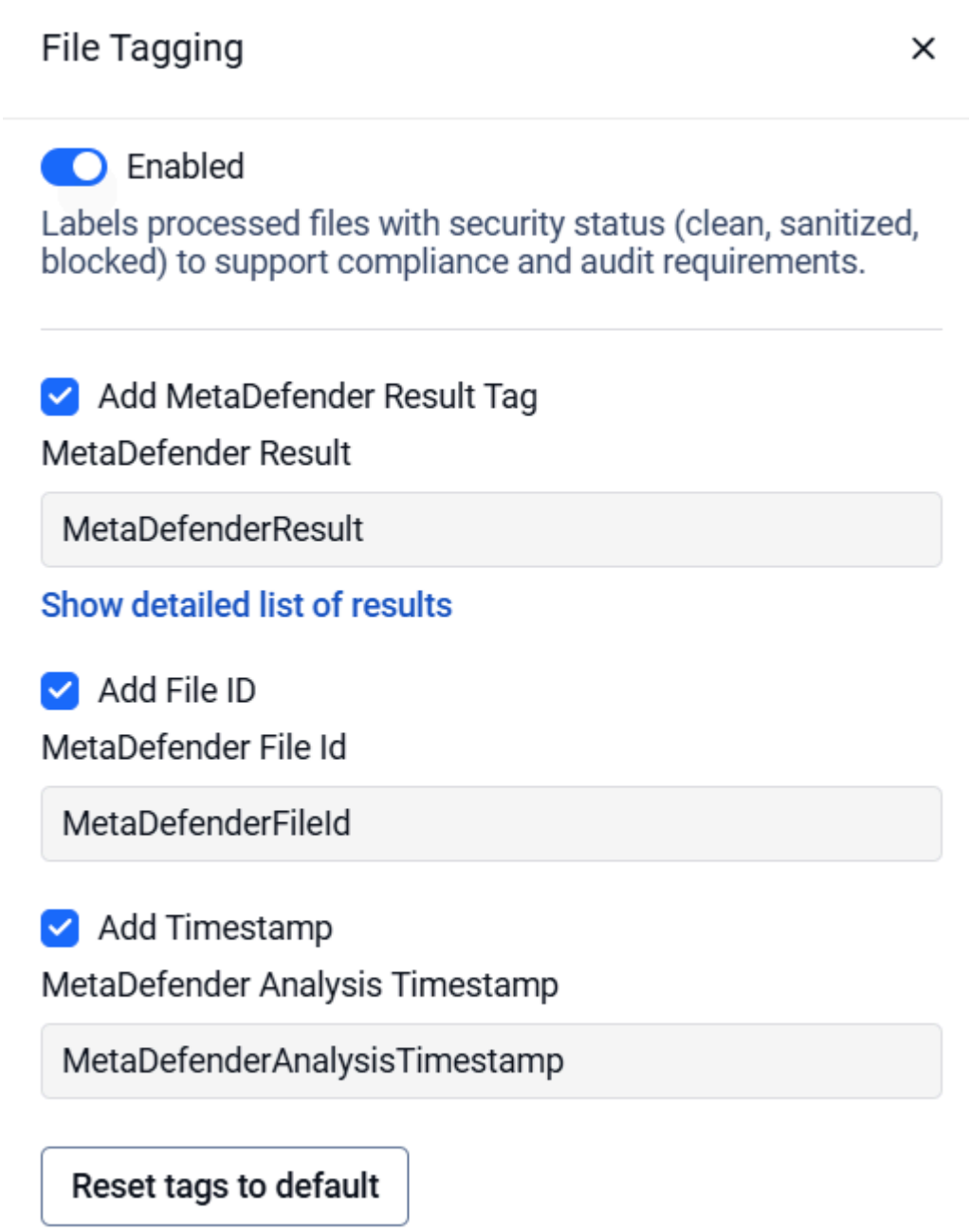


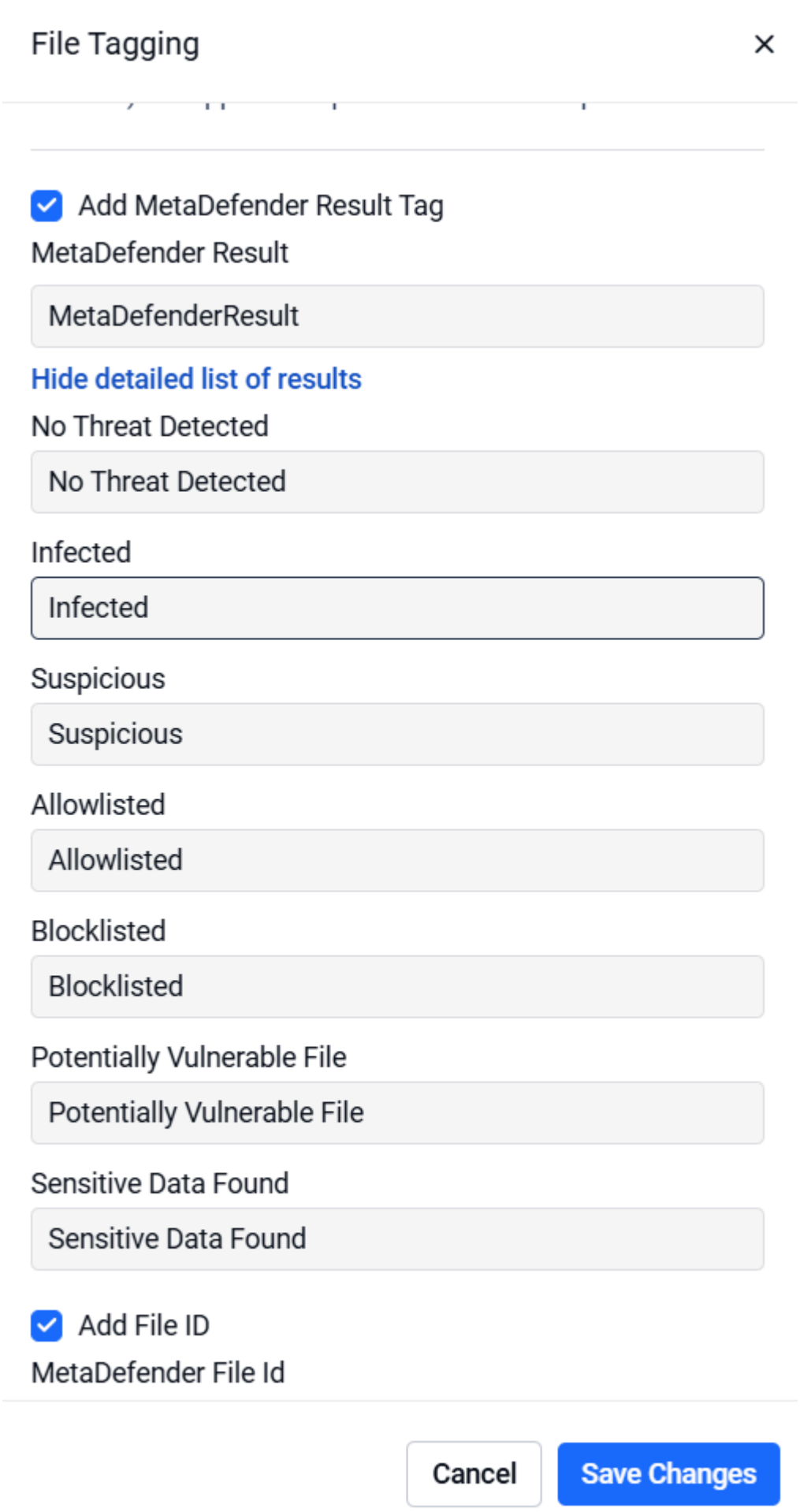
Customize file tags

There are three tags which can be customized and added to the file:

1. **MetaDefenderResult**
2. **MetaDefenderFileId**
3. **MetaDefenderAnalysisTimestamp**

To customize the **key** of a tag, please modify the values and if you want to keep it as a default value, click on the reset tags to default button.


File Tagging
Enabled
Labels processed files with security status (clean, sanitized, blocked) to support compliance and audit requirements.
Add MetaDefender Result Tag
MetaDefender Result
MetaDefenderResult
Show detailed list of results
Add File ID
MetaDefender File Id
MetaDefenderFileId
Add Timestamp
MetaDefender Analysis Timestamp
MetaDefenderAnalysisTimestamp
Reset tags to default


File Tagging
Add MetaDefender Result Tag
MetaDefender Result
MetaDefenderResult
Hide detailed list of results
No Threat Detected
No Threat Detected
Infected
Infected
Suspicious
Suspicious
Allowlisted
Allowlisted
Blocklisted
Blocklisted
Potentially Vulnerable File
Potentially Vulnerable File
Sensitive Data Found
Sensitive Data Found
Add File ID
MetaDefender File Id
Cancel
Save Changes

For the MetaDefender Result you can view the detailed list of customizable values. Below are some examples for the keys and the values of the tags.

Name	Value	Description
MetaDefenderResult (or the customized key associated with this name)	No Threat Detected / Infected etc (or the customized value associated with each result)	The final result of the processing
MetaDefenderFileId (or the customized key associated with this name)	5e7e15f132efdd0006b0e132	A unique file identifier that can be used for further REST API queries
MetaDefenderAnalysisTimestamp (or the customized key associated with this name)	2020-03-28T21:43:40.1500000Z	Date and time in UTC format when file processing has started

Deep Content Disarm and Reconstruction (Deep CDR)

Enabling the File Sanitization feature creates a secure version of a file by removing potentially harmful content. This process utilizes OPSWAT's Deep Content Disarm and Reconstruction (Deep CDR) technology.

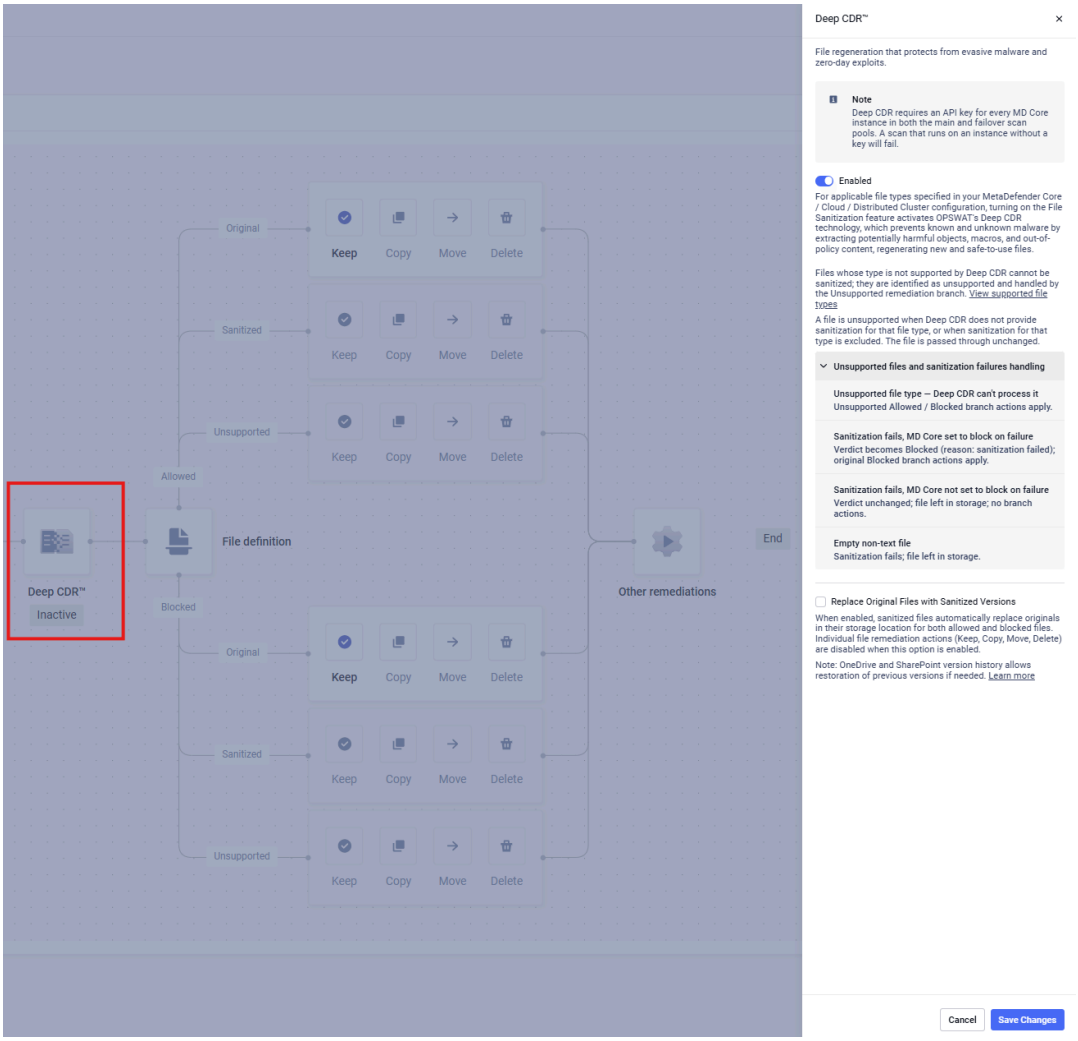
For detailed configuration instructions, refer to the Deep CDR user guide.

Learn more about OPSWAT's Deep CDR technology:
https://www.opswat.com/technologies/data-sanitization

Enabling Deep CDR

> **Dependency**
>
> In order to be able to use the Deep CDR functionality, Deep CDR must be active in MetaDefender Core instance or Metadefender Cloud and each scan instance both in the main and failover scan pools must have an API key.

1. Navigate to **Workflows** from the left-side navigation menu.
2. Select the workflow for which you want to enable Deep CDR.
3. In the **Workflow Diagram**, click on the "**Deep CDR**" node
4. In the modal window that's opened, select the **Deep CDR** tab and toggle **Active** to enable File Sanitization



Info

Enabling Deep CDR, also enables the configuration nodes for sanitized files that were allowed or blocked, which define what will happen with the sanitized file. It also enables the configuration nodes for files that are not supported by Deep CDR technology. Disabling Deep CDR disables the four nodes.

Unsupported branches

Files whose type is not supported by Deep CDR cannot be sanitized; they are identified as **Unsupported **and handled by the **Unsupported **remediation branch. For the list of supported file types, refer to the Deep CDR supported file types.

A file is considered unsupported when Deep CDR has no sanitization support for that file type, or when sanitization for that type has been disabled in the MetaDefender configuration. Unsupported files are not processed by Deep CDR and are left in their original state.

Unsupported files and sanitization failures handling

1. Unsupported file type - Deep CDR does not support this file type and cannot process it. The file is routed through the Unsupported branch, where the configured Allowed or Blocked actions apply.
2. Sanitization fails, and scan instance is configured to block the file on failure - The file verdict is set to Blocked with “Sanitization failed” as the reason. The original file is handled by the Blocked branch actions (move, copy, delete, or keep).
3. Sanitization fails, and scan instance is not configured to block the file on failure - The original file verdict is preserved. The file stays in its original location. No branch actions are applied.
4. For empty non-text files, sanitization fails - Sanitization cannot be applied to empty non-text files. The original file verdict is preserved and it is marked with a “Sanitization failed” remediations status. The file is left in its original location.

Replace Original Files with Sanitized Versions

Info

This is currently only supported for **OneDrive** and **SharePoint** storage types.

Overview

This feature enables **true in-place replacement** of original files with their sanitized counterparts produced by Deep CDR.

Previously, when Deep CDR was configured to *keep the sanitized version* and *delete the original*, the system performed these actions as two separate steps:

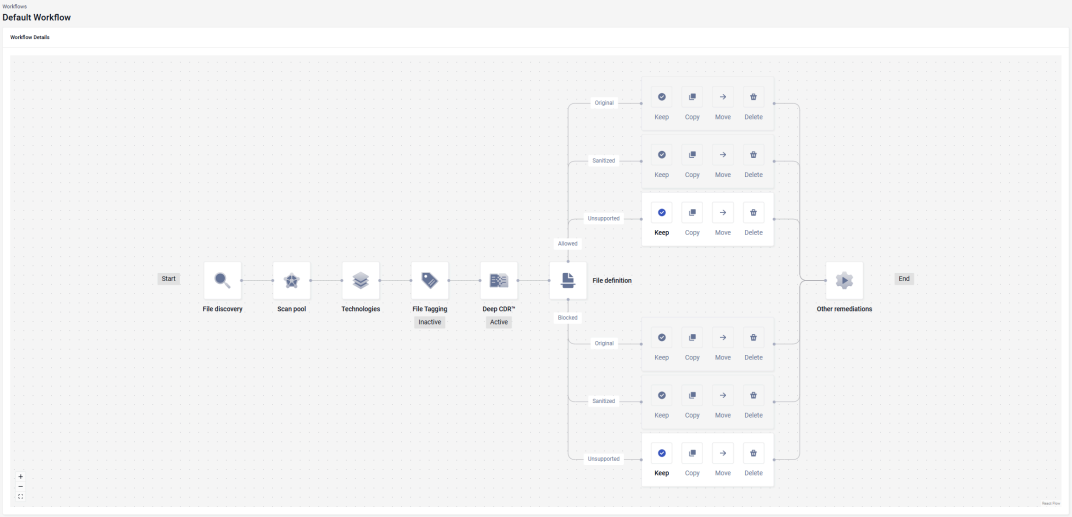
- A **new sanitized file** was uploaded
- The **original file** was deleted

With Replace Original Files enabled, the original file is **replaced directly** with the sanitized content, preserving file continuity by creating a new version of the same file rather than a new file.

How to Enable

1. Enable Deep CDR in the MetaDefender Storage Security Workflow
2. Check **“Replace Original Files with Sanitized Versions”**
3. Click Save Changes

When Replace Original Files with Sanitized Versions is enabled, all other **file remediation options are disabled and locked**. The only enabled options are those for files that are unsupported by Deep CDR.



File Version Cleanup (Optional)

By default, when Deep CDR replaces an original file with its sanitized version, the previous file version is retained in the storage unit's version history. This applies to **OneDrive** and **SharePoint Online**, which maintain automatic version histories for all files. To automatically delete the older version after replacement, set the following environment variable in the customer.env file:

```
DELETE_FILE_VERSIONS_AFTER_REPLACE=1
```

This setting is disabled by default (`0`). When enabled, the pre-sanitization version is removed from the version history after a successful replacement, keeping storage clean. If sanitization fails, the original file is left unchanged and no version deletion occurs.

Keep, Copy, Move or Delete Files

Configuring Keep/Copy/Move/Delete Remediation Actions

- Navigate to **Workflows** from the left-side navigation menu.
- Select the workflow for which you want to configure these remediation actions.
- Select the node for which you want to configure these remediation actions (Allowed Original, Allowed Sanitized, Blocked Original, Blocked Sanitized)
- Click on the box of the specific node.

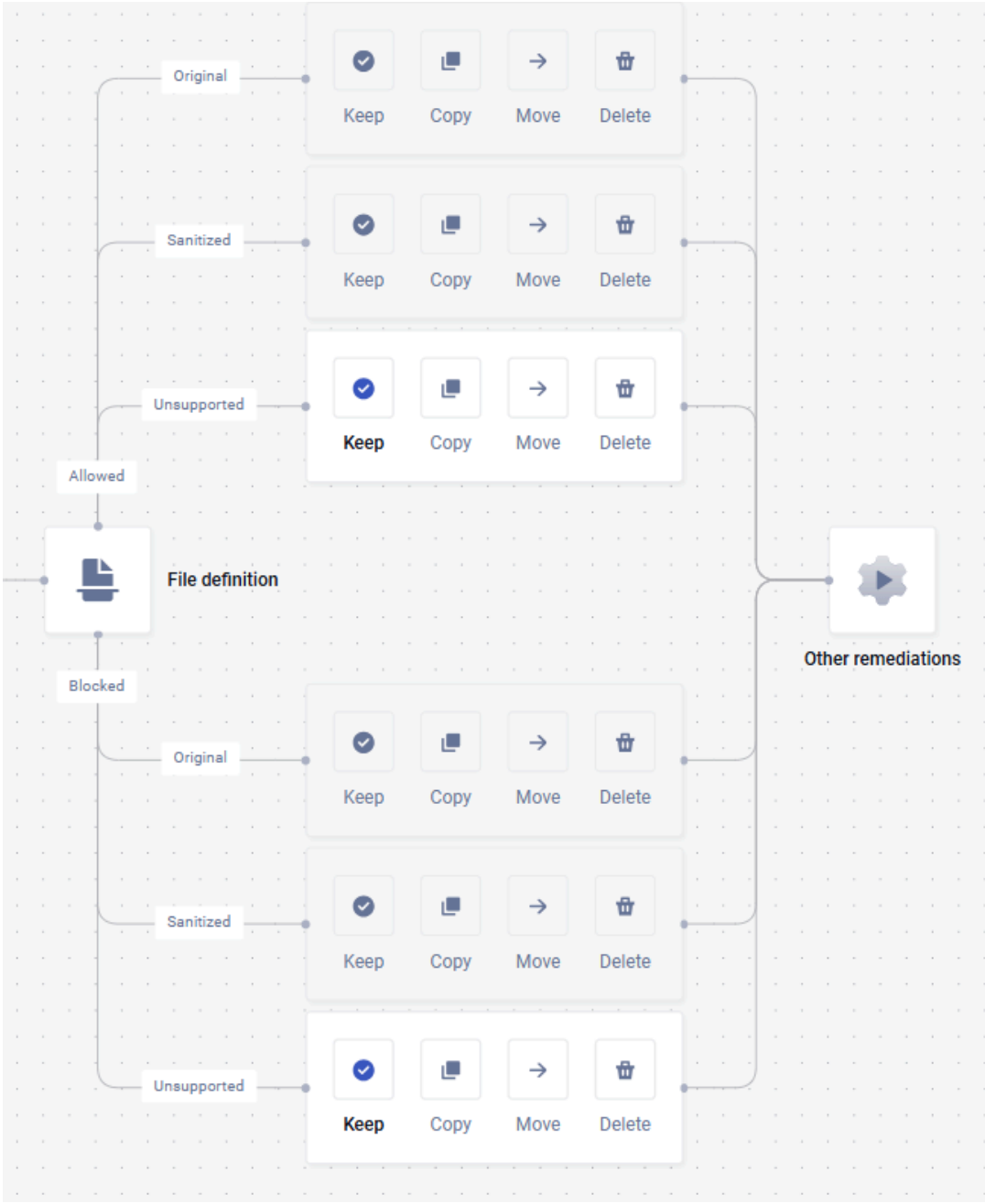
> **Note**

1. Sanitization fails, and MD Core is configured to block the file on failure - The file verdict becomes Blocked with Sanitization failed as the reason; in this case, the original Blocked branch actions apply
2. Sanitization fails, and MD Core is not configured to block the file on failure - The file verdict remains unchanged; the file is left in the storage unit, no branch actions are applied

Move operations are performed as a copy-then-delete sequence rather than a native move; if the upload step fails, the source file is left unchanged and the deletion is not attempted.

Technical Considerations

- Move and Re-processing **->** when a sanitized file is moved from Storage A to Storage B, it will not be re-sanitized on Storage B, even if Deep CDR is enabled there.
  - Similarly, if Real-Time Processing is enabled on Storage B, the moved file will not be processed again.
- File Overwriting **->** moving a file with the same name and path as an existing file on the destination storage will overwrite the existing file.
- By default, “Keep” is the selected option -> which will keep the file as is.
- For the sanitized versions (Allowed Sanitized, Blocked Sanitized), Copy and Delete options are unavailable.



Keeping/Moving Sanitized Files

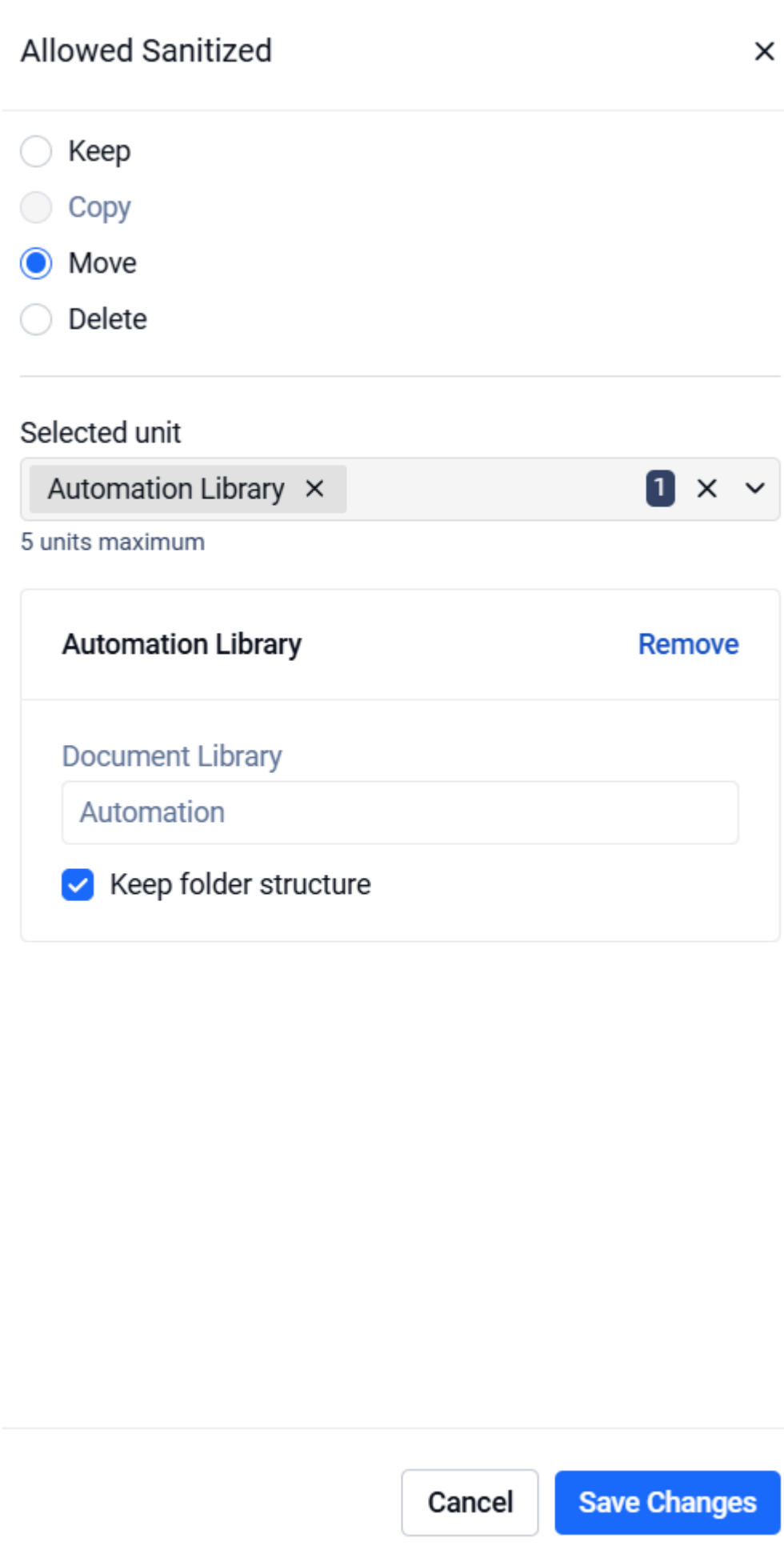
Sanitized files, created by OPSWAT's Deep CDR, can be moved to one or more storage units (up to a maximum of 5 destinations per remediation type).

Note: The File Sanitization feature must be enabled to configure this action - see Enabling Deep CDR: Deep Content Disarm and Reconstruction (Deep CDR)

1. Click on the **Sanitized Remediation (Allowed Sanitized or Blocked Sanitized)** node to open the modal window
2. Choose the action:

1. **“Keep the file”** -> the sanitized file is kept in the source storage location
2. “**Move the file**” -> the sanitized file is moved to another storage unit (or units). Configure the destination storage(s) as described below.

3. Configure the **destination storage** (only for option b):
   - Select one or multiple storage unit destinations out of the dropdown list
     - Additionally, narrow down the destination for certain types of storages (SharePoint Cloud, OneDrive, Box, SharePoint On-Premises), depending on how they are integrated
   - Note: A valid connection with the destination storage(s) is required.
   - Select **Keep folder structure** to maintain the source’s structure.

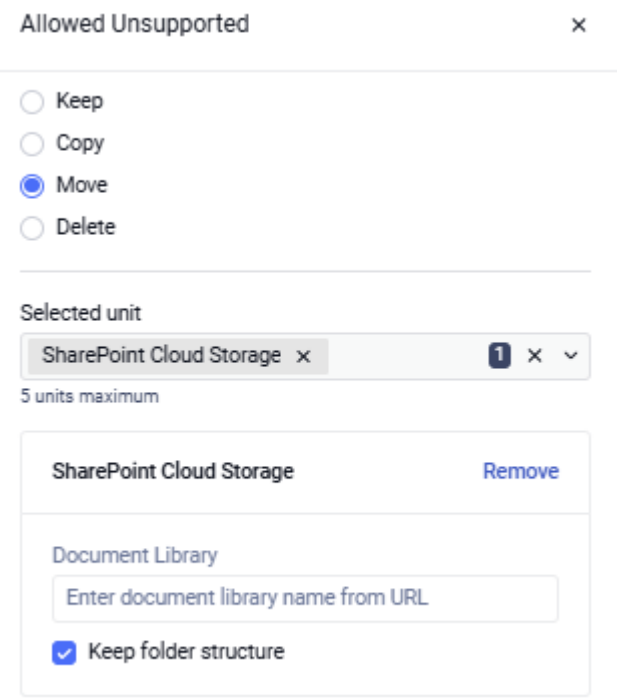

Allowed Sanitized
Keep
Copy
Move
Delete
Selected unit
Automation Library
1
5 units maximum
Automation Library
Remove
Document Library
Automation
Keep folder structure
Cancel
Save Changes

Keeping/Copying/Moving/Deleting Files that are Unsupported by Deep CDR

Deep CDR does not support all file types. Unsupported files can be kept, deleted, moved or copied to one or more storage units (up to a maximum of 5 destinations per remediation type).

Note: The File Sanitization feature must be enabled to configure this action - see Enabling Deep CDR: Deep Content Disarm and Reconstruction (Deep CDR)

1. Click on the **Unsupported Remediation (Allowed Unsupported or Blocked Unsupported)** node to open the modal window
2. Choose the action:
    1. **"Keep the file"** -> the file is kept in the source storage location
    2. **"Copy the file"** -> the file is copied to another storage unit (or units). Configure the destination storage unit(s) as described below
    3. "**Move the file**" -> the file is moved to another storage unit (or units). Configure the destination storage unit(s) as described below
    4. **"Delete the file"** -> the file is deleted
3. Configure the **destination storage** (only for options b and c):
    - Select one or multiple storage unit destinations out of the dropdown list
        - Additionally, narrow down the destination for certain types of storages (SharePoint Cloud, OneDrive, Box, SharePoint On-Premises), depending on how they are integrated
    - Note: A valid connection with the destination storage(s) is required.
    - Select **Keep folder structure** to maintain the source's structure.



Keeping/Copying/Moving/Deleting Allowed Original Files

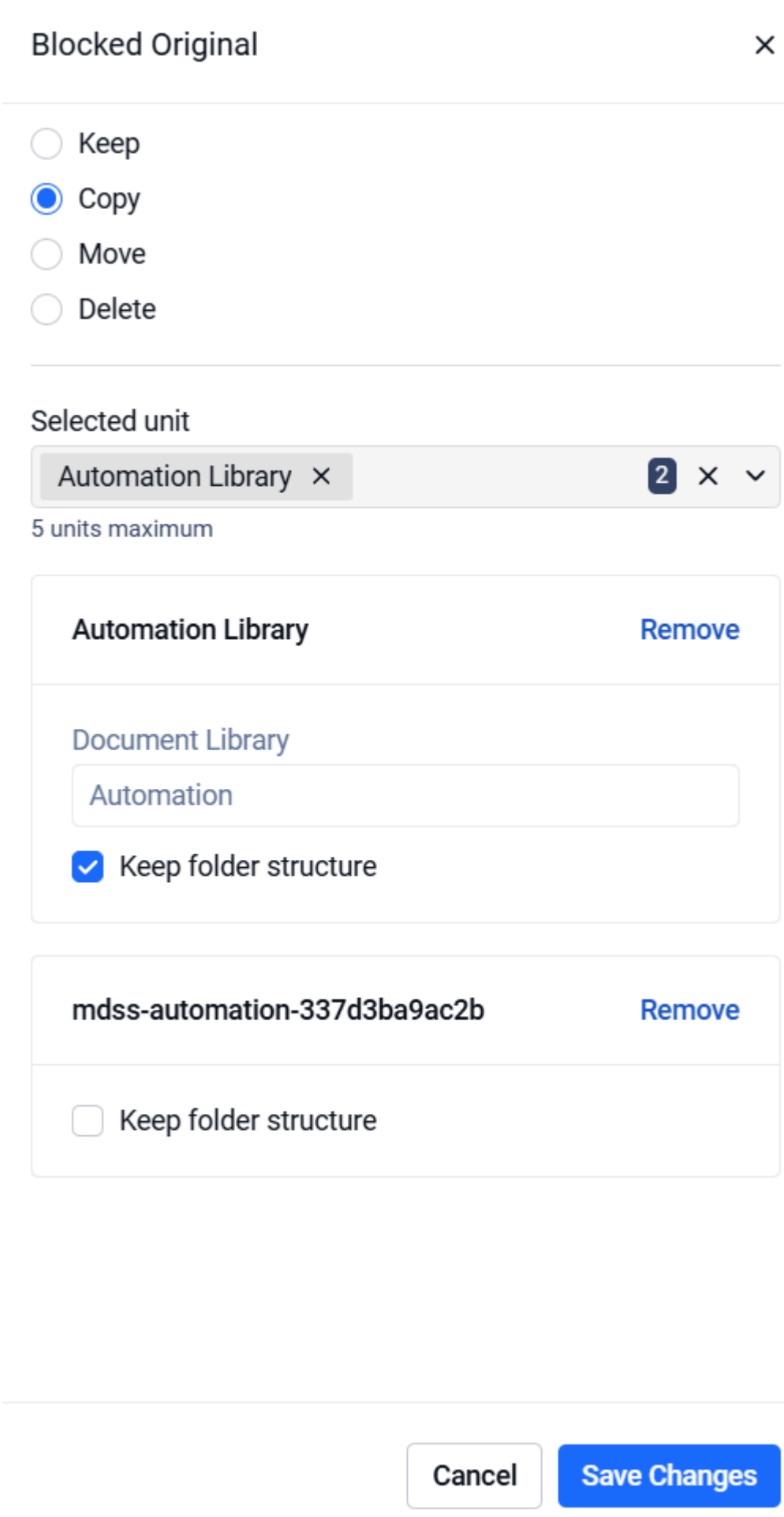
Files marked as **allowed** during processing can be deleted, copied or moved to one or more storage units (up to a maximum of 5 destinations per remediation type).

1. Click on the **Allowed Original Remediation** node to open the modal window
2. Choose the action:
    1. **Keep the file** -> the allowed file is kept in the same location
    2. **Copy the file** -> the allowed file is copied to another storage unit (or units). Configure the destination storage(s) as described below.
    3. **Move the file** -> the allowed file is moved to another storage unit (or units). Configure the destination storage(s) as described below.
    4. **Delete the file** -> the allowed file is deleted.
3. Configure the destination storage (only for options b and c):
    - Select one or multiple storage unit destinations out of the dropdown list
        - Additionally, narrow down the destination for certain types of storages (SharePoint Cloud, OneDrive, Box, SharePoint On-Premises), depending on how they are integrated
    - Note: A valid connection with the destination storage(s) is required
    - Select **Keep folder structure** to maintain the source's structure.

Keeping/Copying/Moving/Deleting Blocked Original Files

Files marked as **blocked** during processing can be deleted, copied or moved to one or more storage units (up to a maximum of 5 destinations per remediation type).

1. Click on the **Blocked Original Remediation** node to open the modal window
2. Choose the action:
    1. **Keep the file** -> the blocked file is kept in the same location
    2. **Copy the file** -> the blocked file is copied to another storage unit (or units). Configure the destination storage(s) as described below.
    3. **Move the file** -> the blocked file is moved to another storage unit (or units). Configure the destination storage(s) as described below.
    4. **Delete the file** -> the blocked file is deleted.
3. Configure the destination storage:
    - Select one or multiple storage unit destinations out of the dropdown list
        - Additionally, narrow down the destination for certain types of storages (SharePoint Cloud, OneDrive, Box, SharePoint On-Premises), depending on how they are integrated
    - Note: A valid connection with the destination storage(s) is required
    - Select **Keep folder structure** to maintain the source's structure.


Blocked Original
Keep
Copy
Move
Delete
Selected unit
Automation Library
2
5 units maximum
Automation Library
Remove
Document Library
Automation
Keep folder structure
mdss-automation-337d3ba9ac2b
Remove
Keep folder structure
Cancel
Save Changes

Storage types with additional configuration needed

INFO

For a number of the possible storage integrations - SharePoint Cloud, OneDrive, Box, SharePoint On-Premises - extra specifications may be necessary when selecting them as destinations as follows:

- If the **SharePoint Cloud** integration only has the **Site** specified, you will be asked to enter the name of the **Document Library** to which the files should be moved/copied
- If the **OneDrive** integration only has the **Group** specified, you will be asked to enter the name or email of the **User** to which the files should be moved/copied
- If the **SharePoint On-Premises integration** only has the **Site URL** specified, you will be asked to enter the name of the **Document Library** to which the files should be moved/copied
- If the **Box** integration has the **All Users** or **A specific group** selected, you will be asked to enter the name or email of the **User** to which the files should be moved/copied

Restrictions

Destination storage restrictions

Marking a storage as a destination for a remediation restricts certain actions on that storage such as deleting it, or altering its configuration in a way that makes the remediation invalid (in the case of destination storages that support users or libraries as a remediation configuration).

Scan restrictions

Starting a scan requires a valid workflow configuration.

The configuration can become invalid if one of the technologies that was enabled and used on the workflow is no longer available on the configured Scan Pool.

Handling Remediation Issues

When remediation actions are configured in a workflow and a user initiates a scan using that workflow, there is a possibility that some files may be discovered and scanned by the

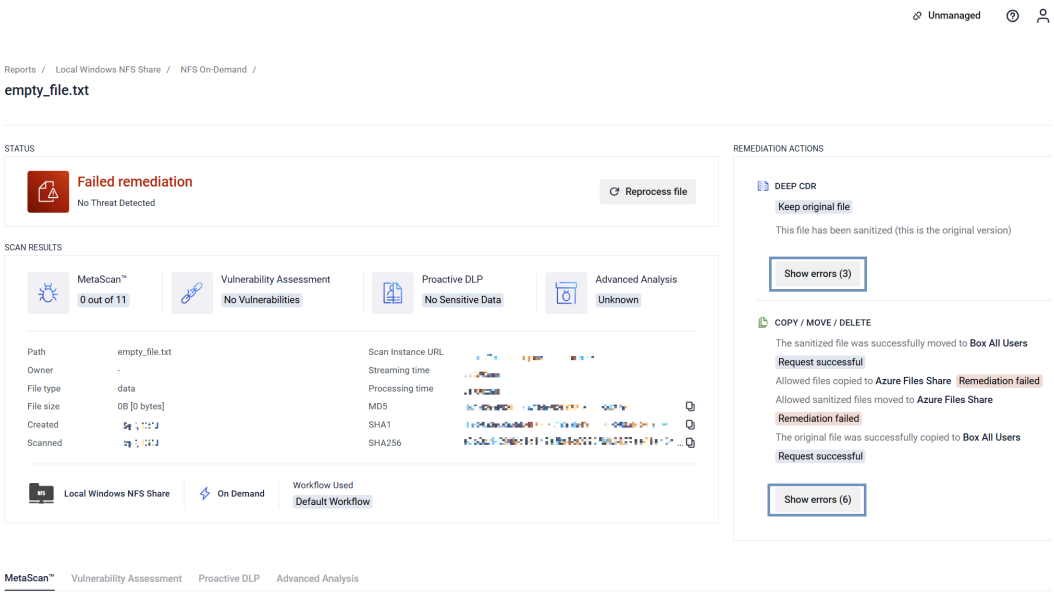
Scan Instance but fail during the application of remediation actions.

In such cases, the scan report will indicate a **Remediation Failed** status.

File name	Status	MetaScan™	Vulnerability Assessment	Proactive DLP	Advanced Analysis	Remediations	Path
presentation.pptx	Allowed	0 out of 11	No Vulnerabilities	No Sensitive Data	Unknown		presentation...
alireza.txt	Allowed	0 out of 11	No Vulnerabilities	No Sensitive Data	Unknown		alireza.txt
empty_file.txt	Failed	0 out of 11	No Vulnerabilities	No Sensitive Data	Unknown	Failed remediation	empty_file.txt
receipts.csv	Allowed	0 out of 11	No Vulnerabilities	No Sensitive Data	Unknown		receipts.csv

To obtain further details about this issue, users can filter the files marked as **Remediation Failed** and click on individual files to access their specific reports.

On the right side of the report, the **Remediation Actions** tab will display all remediation actions taken on the file, including their success or failure.



To diagnose the cause of the failure, users can click on the failed remediation to view the specific exception that led to the failure.

Typically, for Move or Copy remediation actions, the issue is related to the destination storage. If sanitization has failed, refer to the Unsupported files and sanitization failures handling section for details on which branch actions apply.

Troubleshooting Reference

Issue	Common causes	First steps
Copy/Move failures	Storage permissions, network connectivity, disk space	Check storage health, verify credentials
Sanitization failures	File type not enabled, sanitized version upload issue	Check Scan Instance configuration
Performance issues	Multiple destinations, large files, network latency	Review remediation complexity, optimize storage placement
Configuration Errors	Invalid storage selections, missing permissions	Validate workflow configuration, test connections

Integration considerations

Storage requirements

- Make sure the destination storages have sufficient capacity (recommend 2x source storage size)
- Verify network bandwidth between source and destination storages

Security Best Practices

- Use service accounts with minimum required permissions
- Enable encryption for sensitive file transfers
- Regular security audits of remediation workflows

Next steps after configuring remediation actions:

Test Workflows -> run test scans with sample files to verify remediation behavior

Plan scaling -> consider future storage and performance requirements

Delete Empty Folders

Automatic Empty Folder Deletion After Remediation

The **Delete Empty Folders** feature automatically removes directories that become empty following remediation actions. This feature helps maintain a clean and organized file system by eliminating unnecessary empty folders.

> **SUPPORTED STORAGE TYPES**
>
> NFS, SMB and SFTP

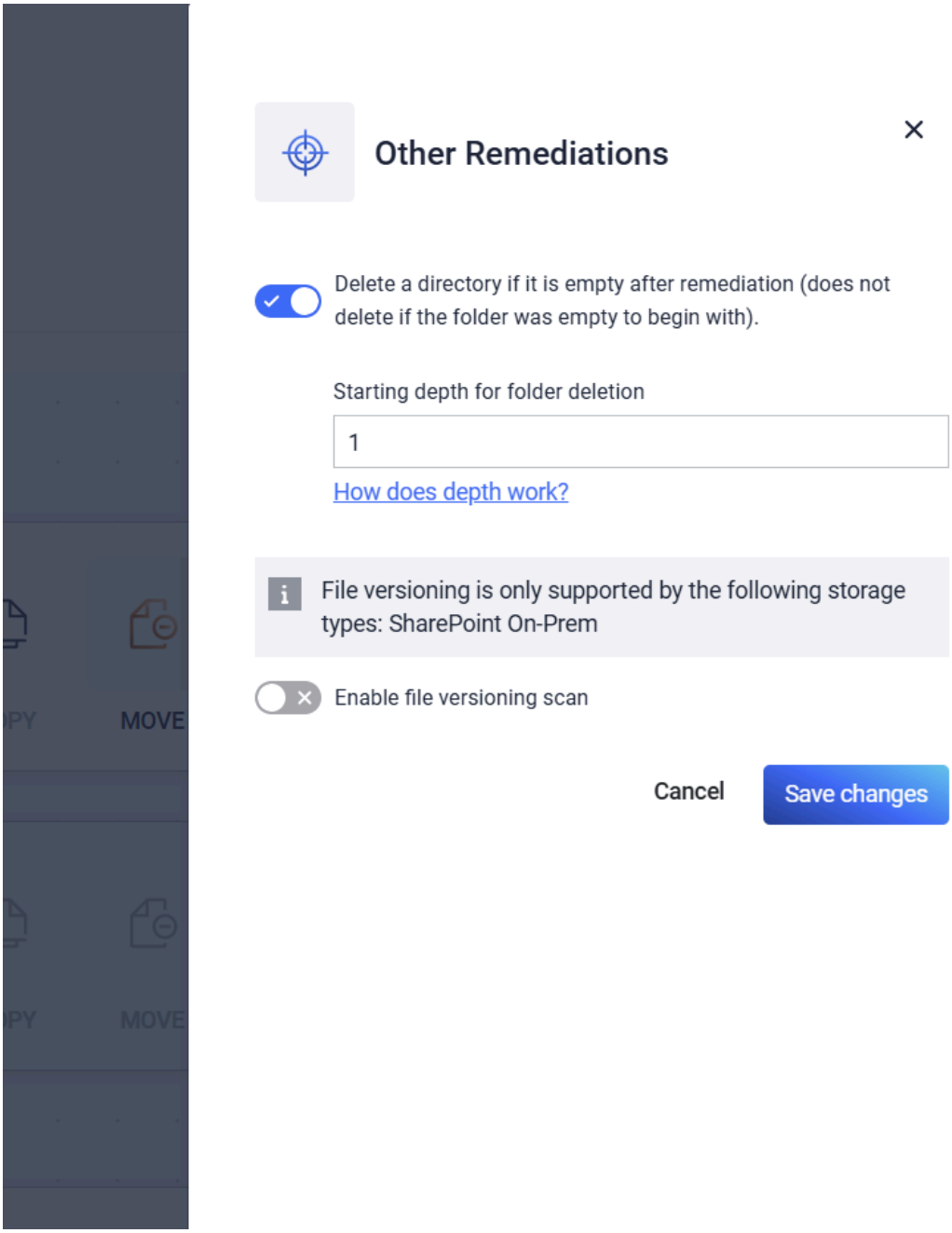
Deletion Behavior

1. Empty folder deletion occurs after a scan completes and then every 10 minutes.
2. A folder will **NOT** be deleted if it contains any files or folders after the remediation.
3. A folder will **NOT** be deleted if it was empty before the remediation.

Configuring Empty Folder Deletion

> **Limitations**
>
> The node is disabled if there is no prior remediation configured with Move/Delete. If Delete Empty Folders is enabled, and there is only one remediation configured with Move/Delete, that remediation cannot be changed to Keep/Copy.

- Navigate to **Workflows** from the left-side navigation menu
- Select the workflow for which you want to configure empty folder deletion
- Click the **Other remediations** node
- In the modal window opened, toggle the “Delete a directory if it is empty after remediation” option
- Enable the option: **Delete a directory if it is empty after remediation** - this does not delete if the folder was empty to begin with.



Depth Parameter

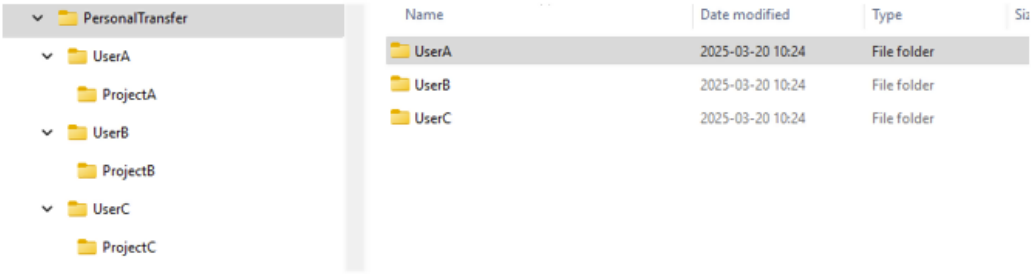
The depth parameter is used to determine where deletion of empty folders starts. It is a value in the range 1-10000

For example in the picture below we have a folder structure. Let's say that there is an nfs storage that has the sharePath PersonalTransfer.

Default - Depth = 1 -> PersonalTransfer Folder (which is also root) is kept, the rest of the empty folders are deleted

Depth = 2 -> UserA, UserB and UserC are kept, the rest of the empty folders are deleted

Depth = 3 -> UserA, UserB, UserC, ProjectA, ProjectB and ProjectC are kept, the rest of the empty folders are deleted



Remediation Process Retries

The remediation process includes automatic retry functionality to handle transient failures. When processing each file, the system will attempt up to **3 retries** if the initial remediation attempt fails. There is a **30-second interval** between each retry attempt to allow for temporary issues to resolve.

The retry interval is configurable through the `POLLY_POST_ACTION_RETRY_TIME` environment variable, allowing you to adjust the timing based on your specific requirements and infrastructure constraints.

Configuration Example

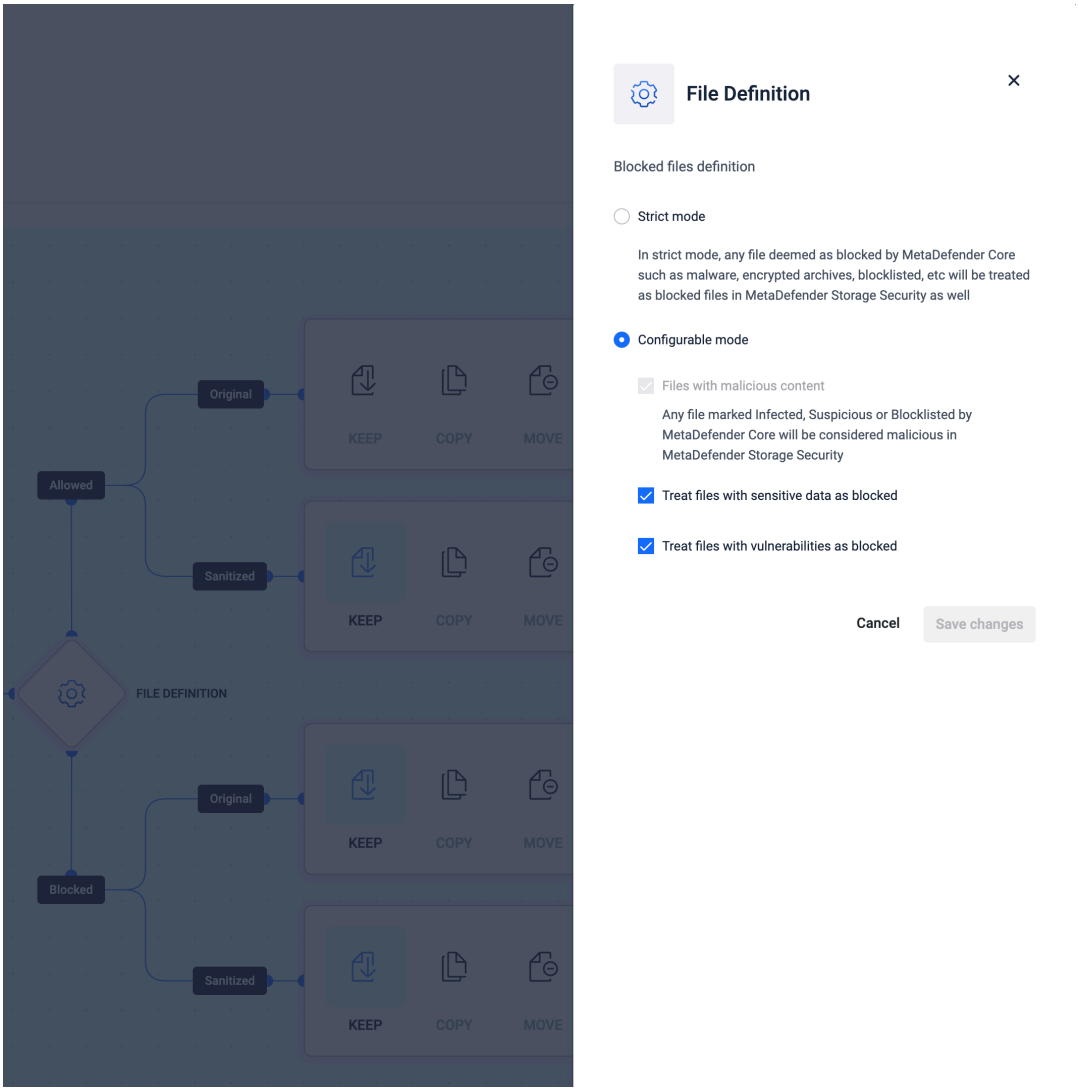
To extend the retry interval to 60 seconds, add the following to your `customer.env` file (instructions here):

```
POLLY_POST_ACTION_RETRY_TIME=60
```

File Definition

Blocked File Treatment Options

1. **Strict** - all files blocked by MetaDefender Core (e.g., malware, encrypted archives, blocklisted files) are treated as blocked
2. **Configurable -** allows selecting additional file types to treat as blocked.
   - **Files with malicious content** - always enabled (Infected, Suspicious, Blocklisted).
   - **Files with sensitive data -** enabled if checked.
   - **Files with vulnerabilities** - enabled if checked.



Technologies

MDSS monitors the status of the following technologies from MetaDefender Core:

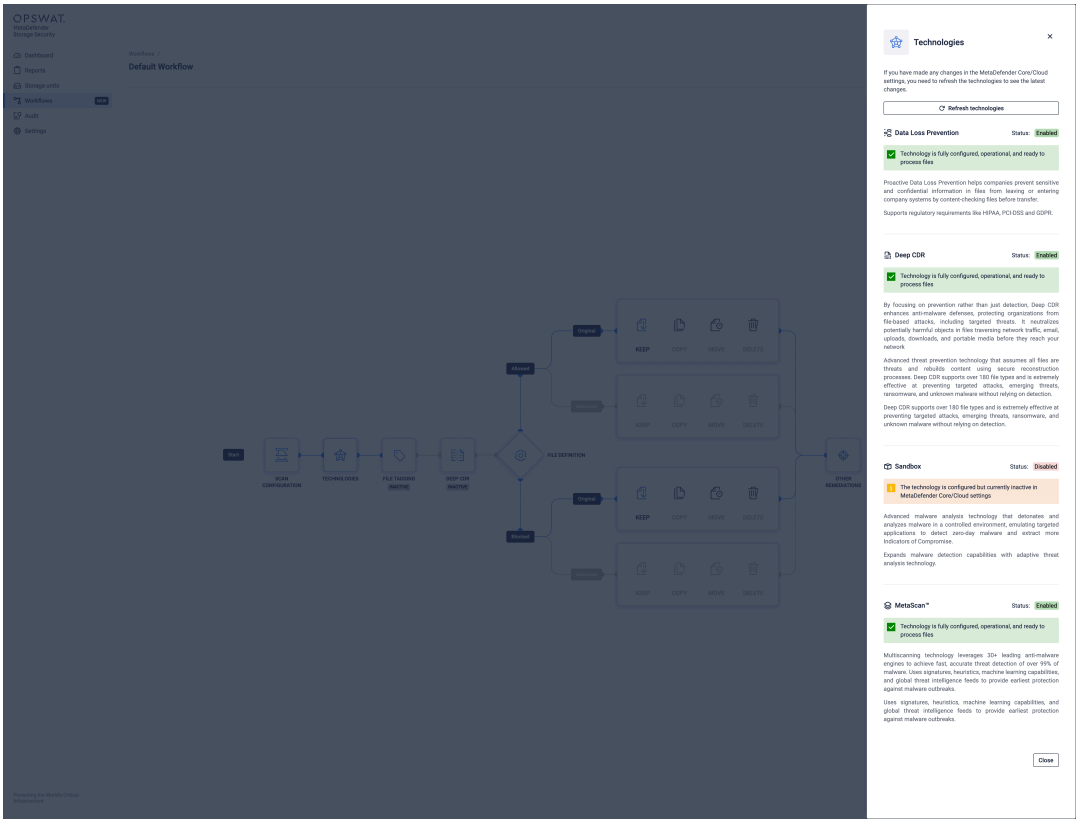
- Data Loss Prevention
- Deep CDR
- Sandbox
- Metascan

These monitoring is done every 5 minutes to see what changes are done on MetaDefender Core for these technologies. A manual refresh is available as well to have this check done on demand

This is not available on MetaDefender Cloud

> **Technology Status**
>
> There are three statuses for the technologies:

- **Enabled**
  - A valid ApiKey was provided for the scan pool and the technology is **available** in the MetaDefender Core instance.
- **Disabled**
  - A valid ApiKey was provided for the scan pool and the technology is **unavailable** in the MetaDefender Core instance.
  - This will prevent allowing to set up actions that are not available in the MetaDefender Core instance.
- **Unknown**
  - No ApiKey was provided for the scan pool. The status of the technology from MetaDefender Core instance is unknown.



Scan File Versions

Overview

The File Version Scanning remediation capability enables MDSS to locate and examine **every version of a file** that exists during scanning operations. This functionality helps users more effectively pinpoint problematic or restricted file versions.

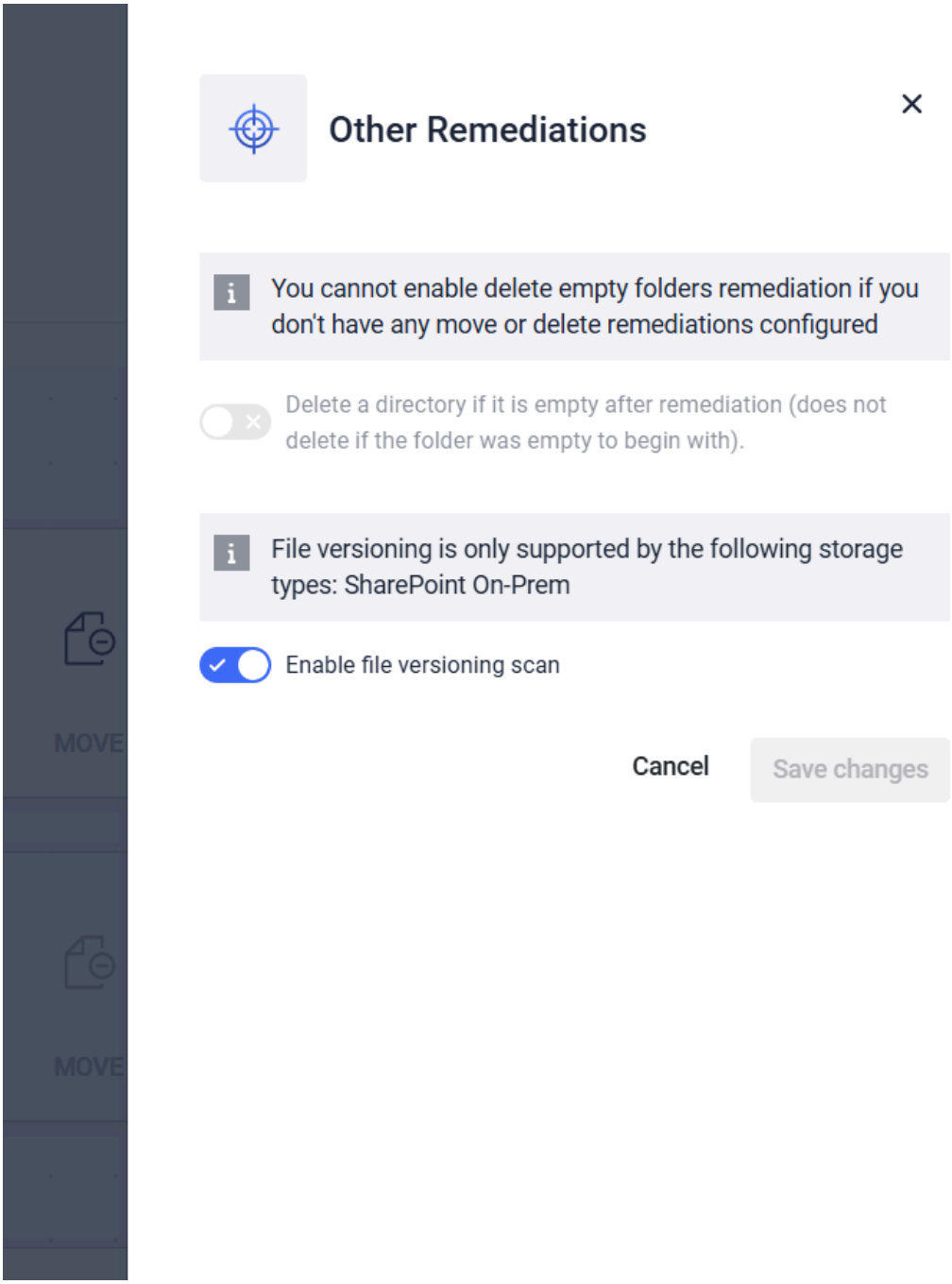
Requirements

File versioning must be activated on your SharePoint On-Premises environment before utilizing this feature. The scanning process cannot identify and report on multiple file versions without versioning being enabled.

> **Compatible Storage Systems/Storage Unit**
>
> SharePoint On-Premises

Configuration

1. Navigate to **Workflows** from the left-side navigation menu.
2. Select the workflow you want to configure scan file versions.
3. Click the **Other Remediations** node.
4. In the modal window that opens, toggle the **Scan File Versions** option.
5. Click **Save Changes** to apply the changes.



Use Case

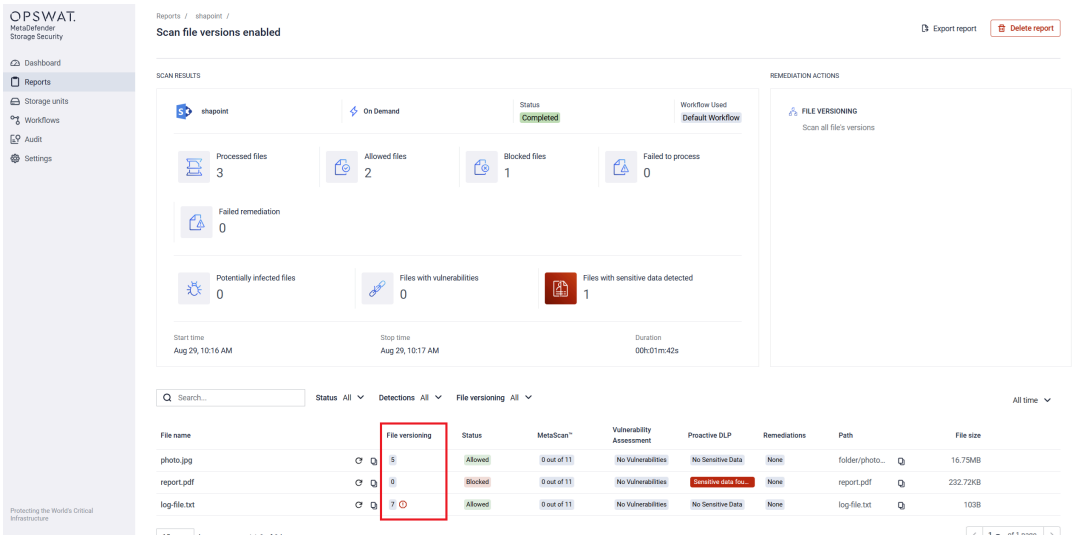
This remediation type is particularly useful when:

- Multiple versions of the same file may exist across different environments.
- You need to ensure that all versions comply with security and compliance requirements.
- You want quick visibility into which versions are safe and which are blocked.

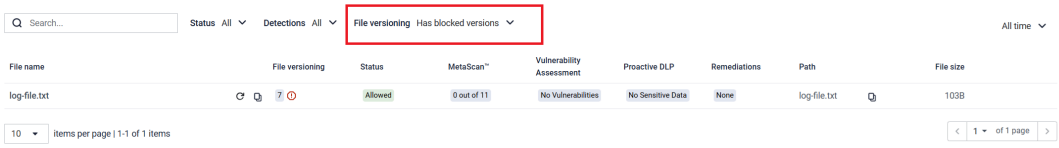
Reporting

During or after a scan with **Scan File Versions** enabled, the report provides additional insights:

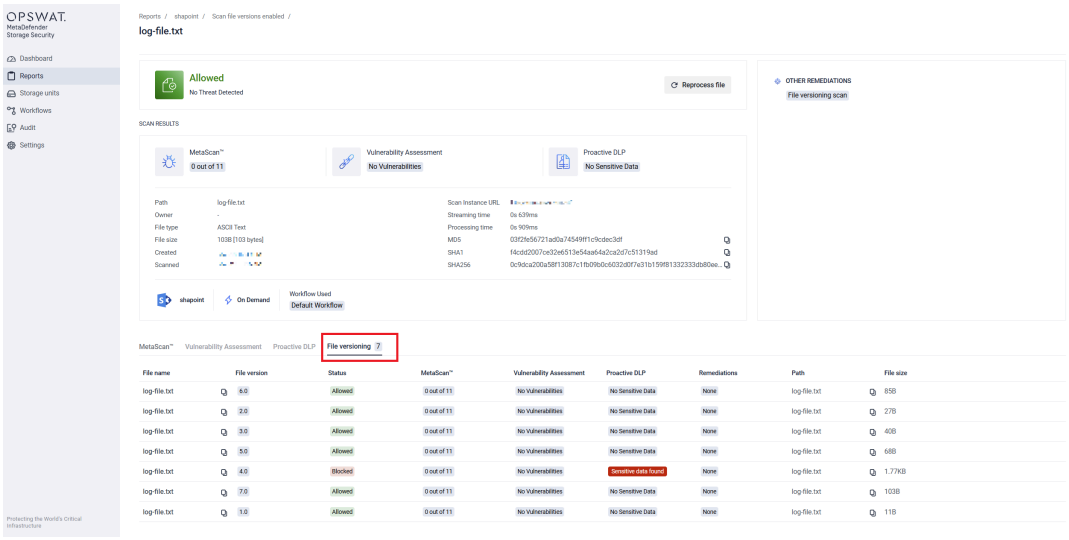
- **New Column – File Versioning**: A dedicated column displays the number of detected versions for each file and indicates if any of those versions are blocked.

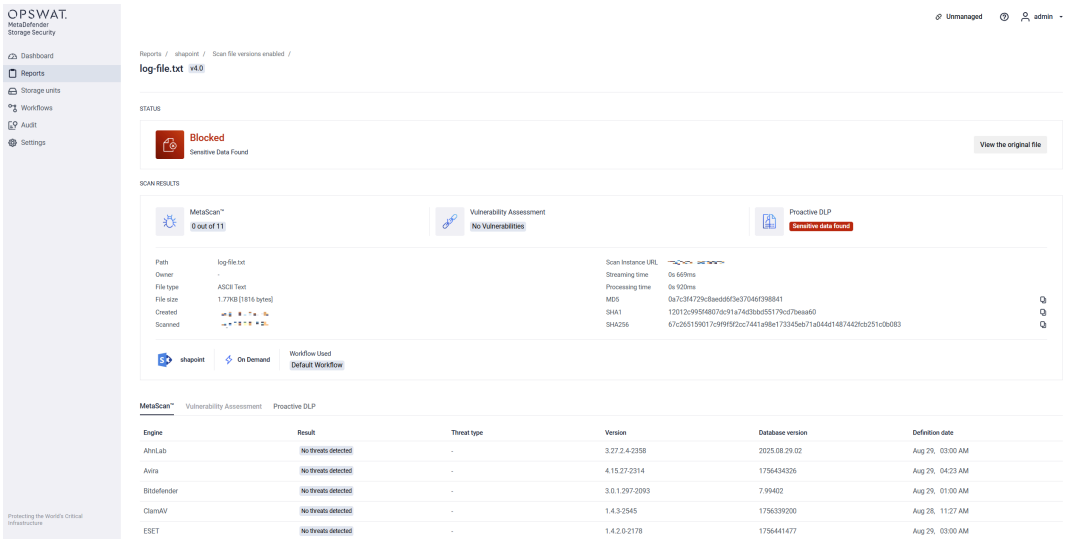


- **Filtering**: The report includes a filter to show only files that contain blocked versions.



- **Individual File Report**: Within a file's detailed report, a new column lists all discovered versions. Clicking on a version displays its scan result.





File Filters

File filters decide which discovered files a workflow sends for scanning. Use them to keep MetaDefender Core/MetaDefender Cluster/MetaDefender Cloud capacity for the files you care about: skip logs, backups or VM images, scan only recent files, or ignore files above a certain size. A file is scanned only when it passes every filter you set. Files a filter rejects are recorded as **Skipped by filter**, with the reason, and are not scanned.

There are three filters you can set on each workflow:

Filter	Skips a file when
File types	its extension is not in the list you selected
Size	it is smaller than the minimum, or equal to or larger than the maximum
Last modified	it was last modified outside the date range you set

A fourth check, for empty files, depends on how your deployment is configured. See Empty files.

> **Extension-based, not true file type detection**
>
> The file type filter looks only at the file name. It compares the text after the last dot with your list and never inspects the file's content. A file renamed from `report.exe` to `report.pdf` is treated as a PDF, and a file without an extension always passes the file type filter.
>
> Use the file type filter to reduce scanning load, not as a security control. True file type detection happens later, in MetaDefender Core/MetaDefender Cluster/MetaDefender Cloud, for the files that are scanned.

Before you start

- Filters are part of the workflow, so they apply to every storage and scan that uses that workflow.
- Changes apply from the next scan. A scan that is already running keeps the filters it started with.
- You need permission to edit the scan configuration. Without it, the File filters panel opens read-only.

Set up filters

1. Go to **Workflows** and open the workflow.
2. In the workflow diagram, select the **File filters** node. On standard workflows it sits between File discovery and Scan pool; on Discovery only workflows it comes before the discovered files step.
3. Set any combination of file types, size and last modified, as described below. Leave a filter empty to apply no restriction for it.
4. Select **Save**.

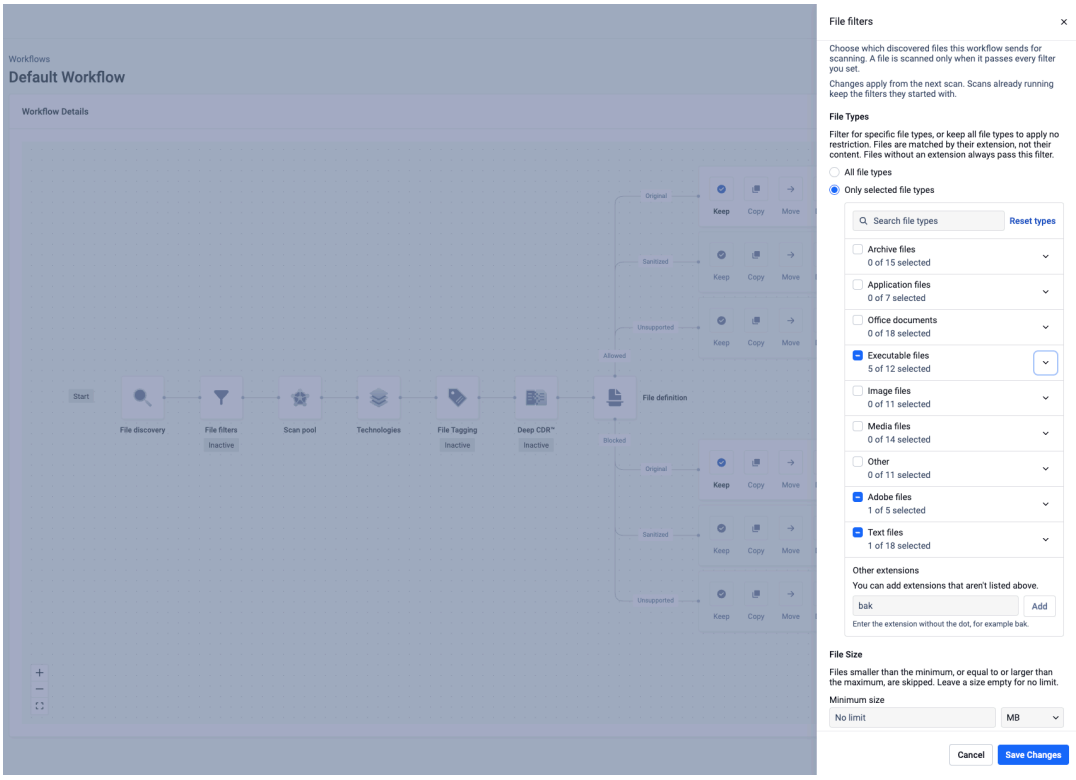
The node shows **Active** when at least one filter is set, and **Inactive** when none is.

File types

Choose **All file types** to apply no file type restriction, or **Only selected file types** to scan only the types you pick:

- Tick a category to select all of its extensions, or expand it to pick single extensions.
- Use **Search file types** to find an extension or a category by name.
- If an extension is not listed, add it under **Other extensions**. Enter it without the dot, for example `bak`. Extensions can use lowercase letters, numbers and the characters `_ - + ~ $`, up to 32 characters. Uppercase input is converted to lowercase.

When you choose **Only selected file types**, select at least one type.



See How file names are matched for how a file's extension is read, and Supported file types for the extensions in each category.

Size

Set a **Minimum size**, a **Maximum size**, or both. Sizes use decimal units (1 KB = 1,000 bytes, 1 MB = 1,000,000 bytes) up to 100 TB.

- A file exactly at the minimum size is scanned.
- A file exactly at the maximum size is skipped.
- The maximum must be larger than the minimum.

For example, a minimum of 1 KB and a maximum of 2 KB scans files from 1,000 bytes up to 1,999 bytes.

Last modified

Set **Include files modified after**, **Include files modified before**, or both.

- Dates use your browser's time zone. A date means midnight at the start of that day: "after 1 January 2025" includes files modified from 00:00 on 1 January 2025, and "before 1 January 2026" stops at 00:00 on 1 January 2026.
- A file modified exactly at either boundary is skipped.
- Dates in the future can't be selected, and the "before" date must be later than the "after" date.

Last-modified times come from the storage. File-system storages such as SMB and NFS keep the original modification time. Most object storages (Amazon S3, Azure Blob and similar) set it to the time the object was uploaded, so a date filter on those storages works on upload time.

Empty files

Your deployment can be configured to skip zero-byte files (setting `DISCOVERY_SERVICE_IGNORE_EMPTY_OBJECTS=1`). It is off by default, in which case empty files are treated like any other file. This setting applies to all workflows and is not part of the File filters panel.

When a file fails more than one filter

The filters are checked in this order, and the first one that fails is the reason recorded for the file:

1. Empty file (only when your deployment skips empty files)
2. Size
3. Last modified
4. File type

For example, a 500-byte `.xyz` file from 2020, checked against a 1 KB minimum size, a date range in 2025 and a PDF-only file type list, is reported as skipped for its size.

How file names are matched

The file type filter takes the text after the last dot in the file name and compares it with your list, ignoring case.

File name	Matched as	Notes
`report.pdf`, `Report.PDF`	`pdf`	Case doesn't matter
`invoice.pdf.exe`	`exe`	Only the last part counts
`archive.tar.gz`	`gz`	Select `gz`, not `tar.gz`
`file.with.many.dots.docx`	`docx`	
`noextension`	no extension	Always passes the file type filter
`.gitignore`	no extension	A name that only starts with a dot has no extension

Files without an extension always pass the file type filter. They are still subject to the size and last-modified filters.

Because only the last part of the name counts, an entry can't contain a dot: `tar.gz` would never match a file.

Supported file types

The categories offer the extensions below. Any other extension can be added under **Other extensions**.

Category	Extensions
Archive files (36)	7z, ace, alz, arj, asar, asice, asics, blz, bz2, cab, egg, gz, gzip, iso, jar, lz, lz4, lzh, lzma, mar, pck, rar, tar, taz, tgz, tse, tsec, tsez, vsix, war, wsp, xz, yz1, z, zip, zstd
Application files (10)	air, cpl, dll, drv, jnlp, lnk, ocx, sys, url, vsto
Office documents (76)	abw, cell, doc, docm, docx, dot, dotm, dotx, fodt, hwp, hwpx, hwt, jhd, jtd, jtdc, mpp, odf, odg, odp, ods, odt, one, otg, otp, ots, ott, oxps, pbix, pot, potm, potx, ppam, pps, ppsm, ppsx, ppt, pptm, pptx, pub, rtf, scrivx, show, sldm, sldx, stc, sti, story, stw, sxc, sxd, sxg, sxi, sxw, twbx, vdx, vsd, vsdm, vsdx, vssm, vssx, vstm, vstx, vsx, vtx, xbd, xct, xdw, xlam, xls, xlsb, xlsm, xlsx, xlt, xltm, xltx, xps
Executable files (16)	apk, appx, bat, cmd, com, exe, msi, msix, php, pl, ps1, py, scr, sh, vbs, wsf
Image files (30)	apng, avif, bmp, cur, dcm, emf, emz, gif, heic, heif, ico, jp2, jpeg, jpf, jpg, jpg2, jpm, jpx, mng, nef, png, svg, tga, tif, tiff, tiff64, wdp, webp, wmf, wsq
Media files (37)	3gp, aac, acm, aiff, asf, avc, avi, bw64, bwf, caf, fla, flv, hevc, jpgm, m4a, m4v, mj2, mjp2, mkv, mov, mp3, mp4, mpc, mpd, mpeg, mts, mxf, ogg, opus, paf, rf64, swf, w64, wav, webm, wma, wmv
Other (73)	3dml, 3ds, 3mf, aem, aff4, b3dm, base64, bfo, cap, cer, crl, crt, dae, dbf, der, dmp, drc, dwf, dwfx, dwg, dws, dwt, dxf, eak, easm, edi, eml, es3, esz, et3, etv, gpkg, ifc, jt, jwc, jww, key, lyrx, mbx, mdzip, msg, oft, ova, ovf, p12, p21, pcap, pcapng, pem, pfx, ppk, prn, pst, qcow2, ribc, ribcd, rvm, scdoc, sfc, shp, shx, spf, stp, tnef, trc, u3d, utib, vhd, vhdx, vmdk, x3p, yenc, zir
Adobe files (8)	ai, ait, eps, fdf, idml, indd, pdf, psd
Text files (92)	1st, aiml, apml, apt, asc, aspx, atom, bml, cfg, classpath, cnf, conf, config, css, csv, dat, dtd, err, faq, ficml, gml, gpx, har, hta, htm, html, ics, inf, ini, jrxml, json, jsp, jspf, jspx, kml, loc, log, lxfml, mathml, me, mht, mml, musicxml, mxl, note, now, plain, plist, pom, prm, project, psmdcp, rdf, rdl, rdlc, rdlx, readme, rels, resx, rng, rss, sdl, sgml, sln, smil, tab, tbm, tds, text, tmp, tsv, ttml, twb, txt, unx, vcf, vcs, vml, vtxt, x3dv, xaml, xht, xhtml, xml, xsd, xsl, xslt, xspf, xul, yaml, yml, zei

See what was skipped

Skipped files stay in the scan report:

1. Open the scan's report and go to the file list.
2. Filter the status by **Skipped by filter**.
3. Open a file to see why it was skipped, for example "This file was skipped because its size is outside the range allowed by a scan filter."

Skipped by filter is different from **Skipped (cached scan)**, which lists files that were not scanned again because an earlier result for the same file was reused.

Configure filters through the API

Filters are the `filters` object of a workflow. Every field is optional; a field you leave out means no restriction.

Field	Type	Meaning
`fileTypeExtensions`	array of strings	Only scan files with these extensions. Leave out, or send `null`, to scan all file types.
`minimumSize`	integer (bytes)	Skip files smaller than this. Default `0`.
`maximumSize`	integer (bytes)	Skip files of this size or larger. Default `109951162777600` (100 TiB), which means no limit.
`minimumDateTime`	ISO 8601 date-time, UTC	Skip files modified at or before this time.
`maximumDateTime`	ISO 8601 date-time, UTC	Skip files modified at or after this time.

Get the catalog the UI offers:

```
GET /api/Workflow/fileTypeCategories
```

Set the filters on an existing workflow. The workflow ID goes in the body, and the request replaces the whole `filters` object, so include every filter you want to keep:

```
PATCH /api/Workflow
Content-Type: application/json

{
  "id": "<workflow-id>",
  "filters": {
    "fileTypeExtensions": ["pdf", "docx", "bak"],
    "minimumSize": 1000,
    "maximumSize": 2000000000,
    "minimumDateTime": "2025-01-01T00:00:00Z",
    "maximumDateTime": "2026-01-01T00:00:00Z"
  }
}
```

The same `filters` object can be sent when creating a workflow with `POST /api/Workflow`, and it is included in settings export and import.

Dates sent through the API are used exactly as given, in UTC. The UI instead sends midnight in the browser's time zone, so the same calendar date can mean a different moment for users in different time zones.

Rules and errors

Extensions are cleaned up before they are checked: spaces around them and a leading dot are removed, they are lowercased, and duplicates are dropped. `" .PDF "` is stored as `pdf`.

Request	Result
An extension with a dot, a space or another character outside `a-z 0-9 _ - + ~ $`, or longer than 32 characters	`REST_API_MSG_FILE_TYPE_FILTER_UNSUPPORTED` with the rejected value in the `extension` parameter
More extensions than the configured maximum (500 by default, set by `WORKFLOWMANAGER_MAX_FILE_TYPE_FILTER_EXTENSIONS`)	`REST_API_MSG_FILE_TYPE_FILTER_TOO_MANY_EX`
`maximumSize` not larger than `minimumSize`, `maximumDateTime` not later than `minimumDateTime`, or a date in the future	`REST_API_MSG_FILTERS_NOT_CORRECTLY_CONFIG`

Nothing is saved when a request is rejected.

Limits

Limit	Value
Extension length	1 to 32 characters
Characters in an extension	`a-z, 0-9, _ - + ~ $`
Extensions per workflow	500 by default (configurable)
Maximum size	100 TB
Dates	Not in the future

User Management

> **IMPORTANT**
>
> Sharing administrative accounts is strongly discouraged.
>
> For users who only need to review reports, assign read-only permissions.

For organizations requiring multiple individuals to access MetaDefender Storage Security or MetaDefender Storage Security Cloud for report review or storage integration, you can create individual user accounts.

MetaDefender Storage Security supports various user roles with different permission levels.

Local Administrator

- Accounts with administrative privileges have full control over each available configuration and all features
- Exclusive to one individual per deployment
- Hidden from other users in the system
- Configure and set up the MDSS deployment with all necessary parameters
- Modify any settings related to the deployment as required
- Facilitate the creation and management of additional user roles for customer use

Permissions

- Add, delete, or modify scan instances, scan pools, and workflows
- Manage the product license settings
- Configure data retention settings
- Configure the possibility of external logging (to Kafka)
- Modify the SSO configuration if any alterations are necessary
- Access user management-related settings
- View the settings related to Workflows
- View the status of scan pools and individual scan instances
- View the configuration of data retention policies
- Set up and manage webhook notification settings to facilitate external integrations
- Add, delete, or modify storage units and remediation actions
- Initiate instant, scheduled, or real-time scans on specific storage units
- Access and review generated reports for insights into scan results
- View the audit page to track changes and activities within the system

- Monitor the dashboard for an overview of storage activity and key metrics

Tenant Administrator

- Available only on MDSS Cloud
- Highest privilege role available to standard users of MDSS Cloud
- Multiple users can hold this role simultaneously
- Utilize the MDSS features and functionalities without altering the overall configuration
- Manage common operations and user interactions within the application

Permissions

- Access user management-related settings
- View the settings related to Workflows
- Set up and manage webhook notification settings to facilitate external integrations
- Add, delete, or modify storage units and remediation actions
- Initiate instant, scheduled, or real-time scans on specific storage units
- Access and review generated reports for insights into scan results
- View the audit page to track changes and activities within the system
- Monitor the dashboard for an overview of storage activity and key metrics

Read-Only

- designed for users with minimal access privileges
- Multiple users can hold this role simultaneously
- Use the MDSS functionality without the ability to make any modifications or changes to MDSS
- Provides basic monitoring and visibility into system operations

SSO Read-Only

- Designed for users with minimal access privileges
- Multiple users can hold this role simultaneously
- This role can be used when logging in via SSO

Permissions

- View the audit page to track changes and activities within the system
- Monitor the dashboard for an overview of storage activity and key metrics

SSO Administrator

- Designed for users with minimal access privileges who need scan capabilities
- Multiple users can hold this role simultaneously
- This role can be used when logging in via SSO

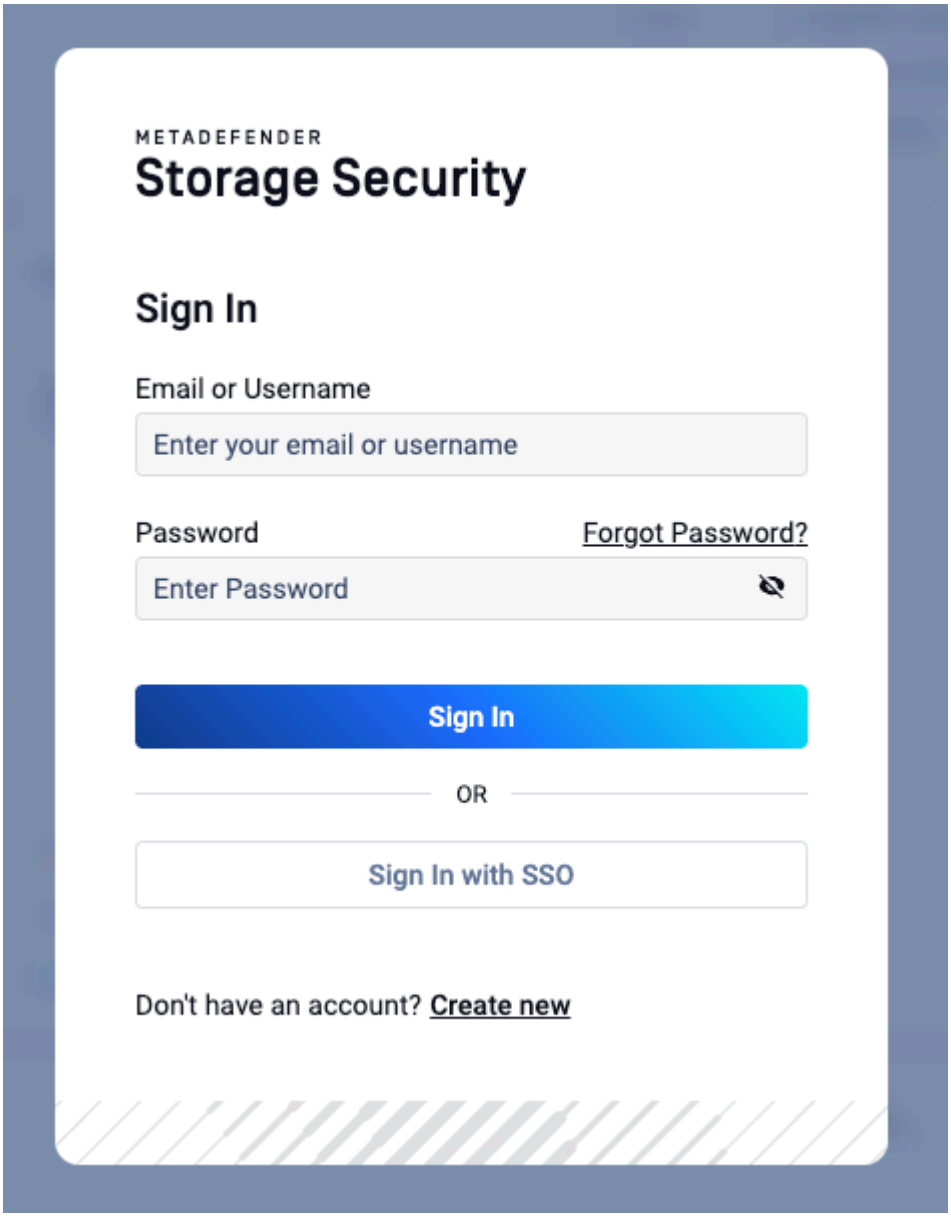
Permissions

- Add, delete, or modify Workflows
- Manage the product license settings
- Configure data retention settings
- Configure the possibility of external logging (to Kafka)
- Initiate instant, scheduled, or real-time scans on specific storage units
- Access and review generated reports for insights into scan results

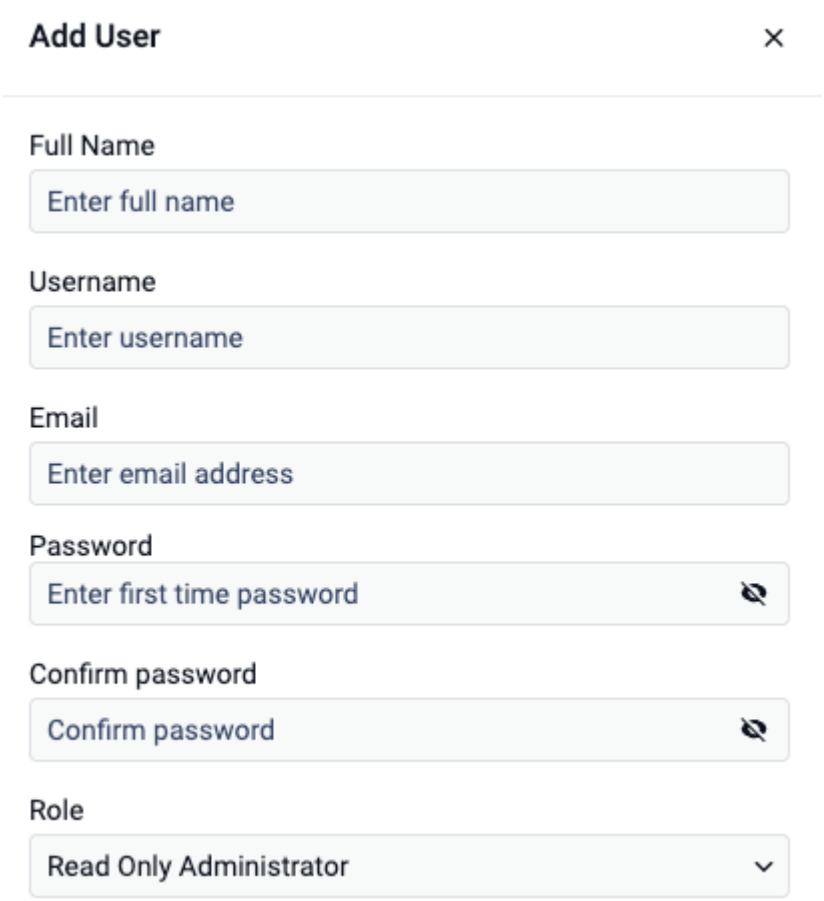
Creating New User Accounts

There are two methods to create user accounts:

- **User Self-Registration** - users can create their own accounts by clicking **Sign Up** on the web interface login page.



- **Administrator Creation** - administrators can create accounts through **Settings/Users** by clicking **Add new user** by clicking **Add new user**

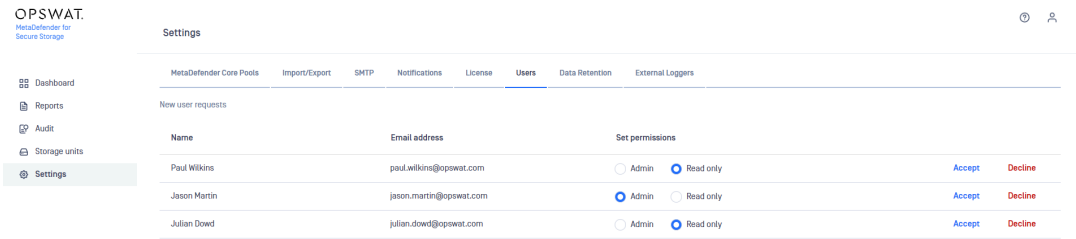


Approving or Denying User Registration Requests

When a user registers via the login page, their account is created but initially inactive. Administrators must approve these requests to grant access.

To manage registration requests:

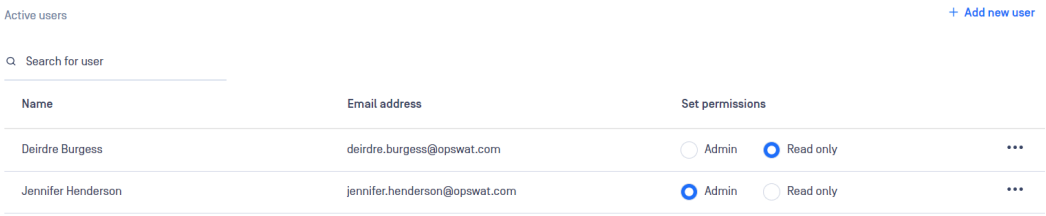
1. Navigate to Settings / Users.
2. Select the user request.
3. Choose the desired permissions for the account.
4. Click **Accept** to activate the account or **Decline** to deny access.



Modifying Permissions for Existing Users

To change the permissions of an existing user:

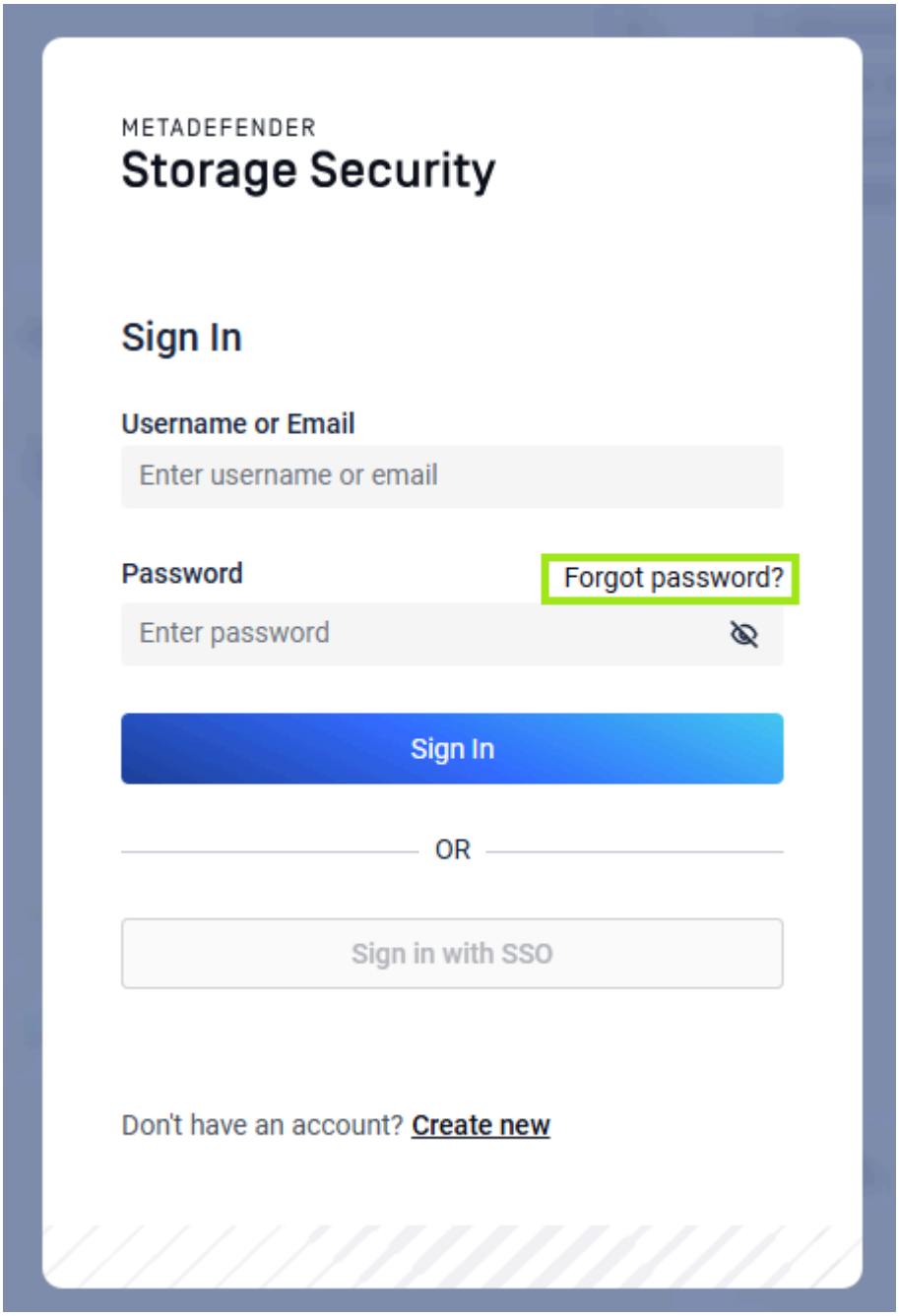
1. Navigate to Settings/Users.
2. Select the user account.
3. Choose the new permission level.



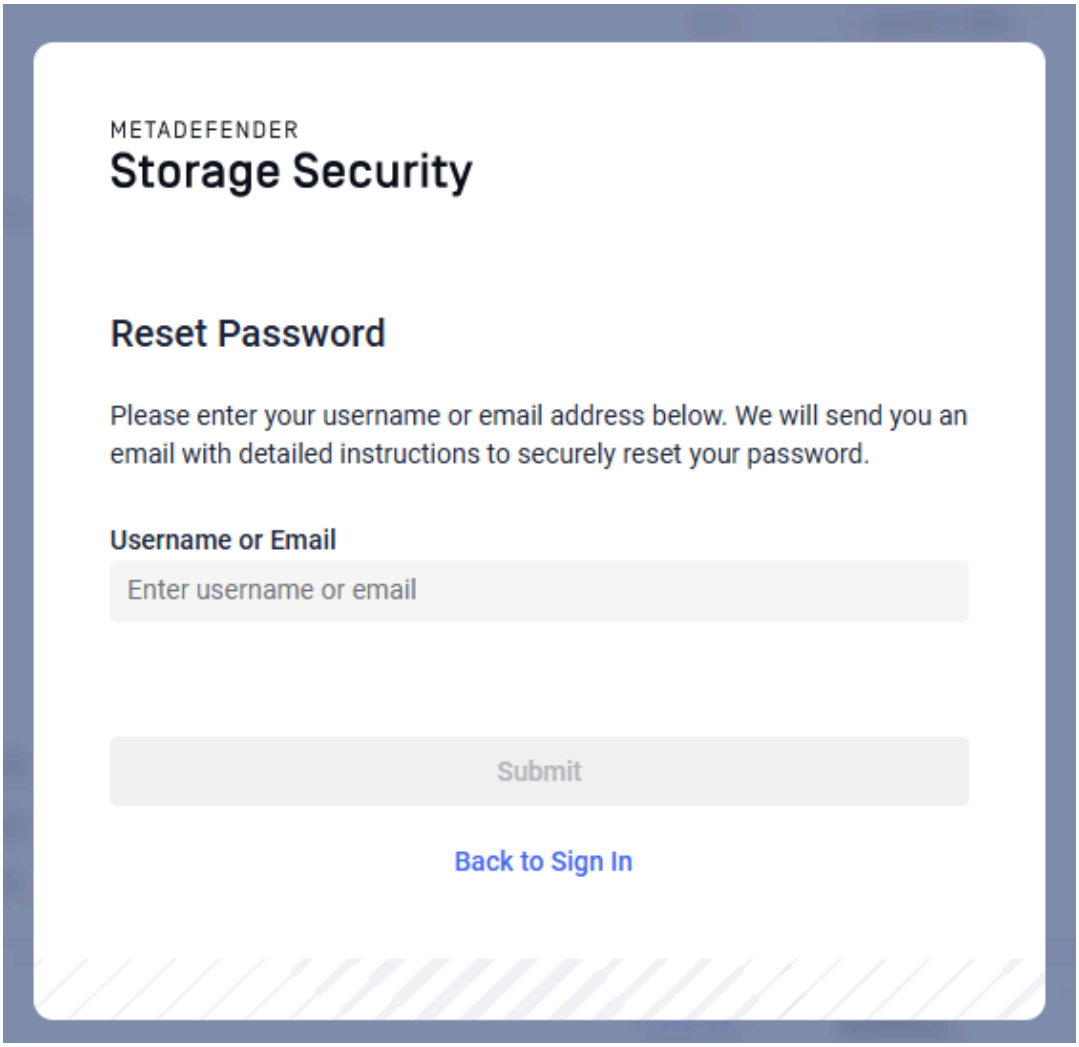
Reset/forgot password

> **INFORMATION**
>
> This functionality is available only when SMTP is configured and enabled. How to configure SMTP here

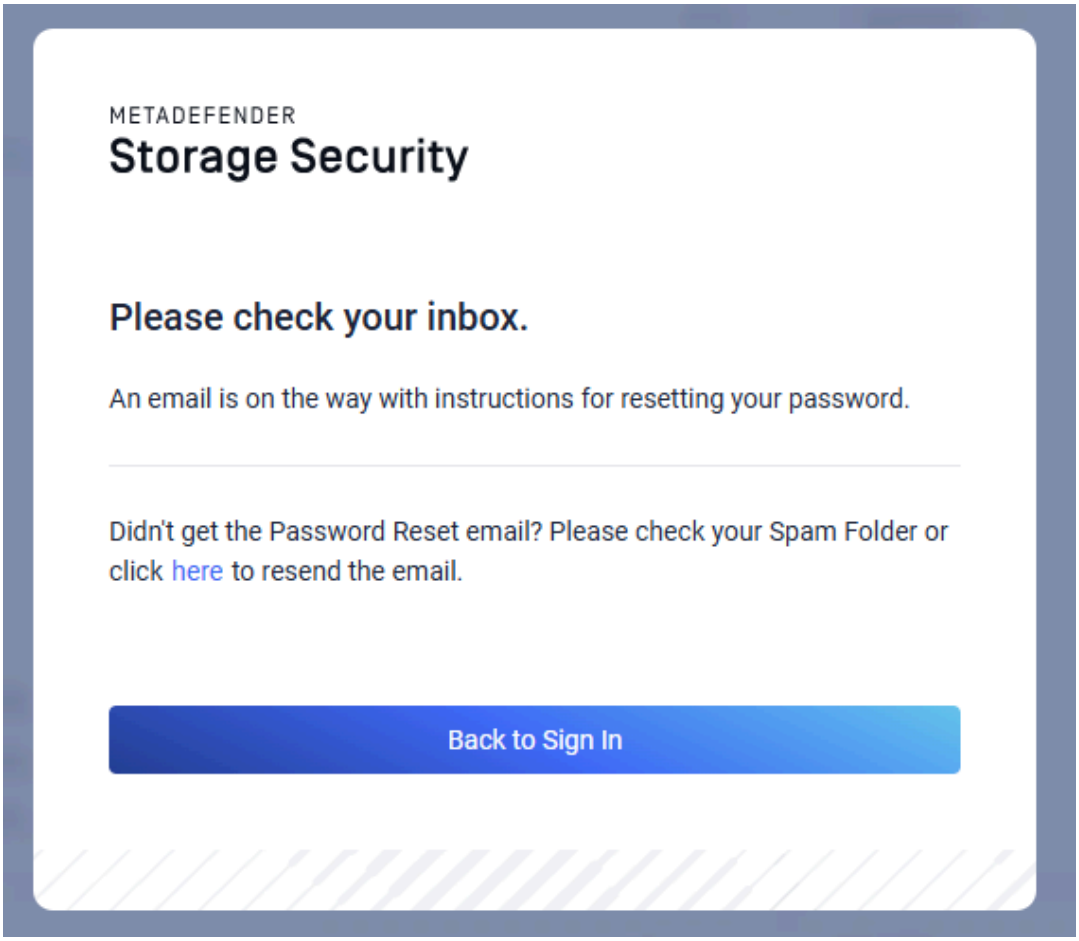
1. From the Sign in page, click on **Forgot password?**



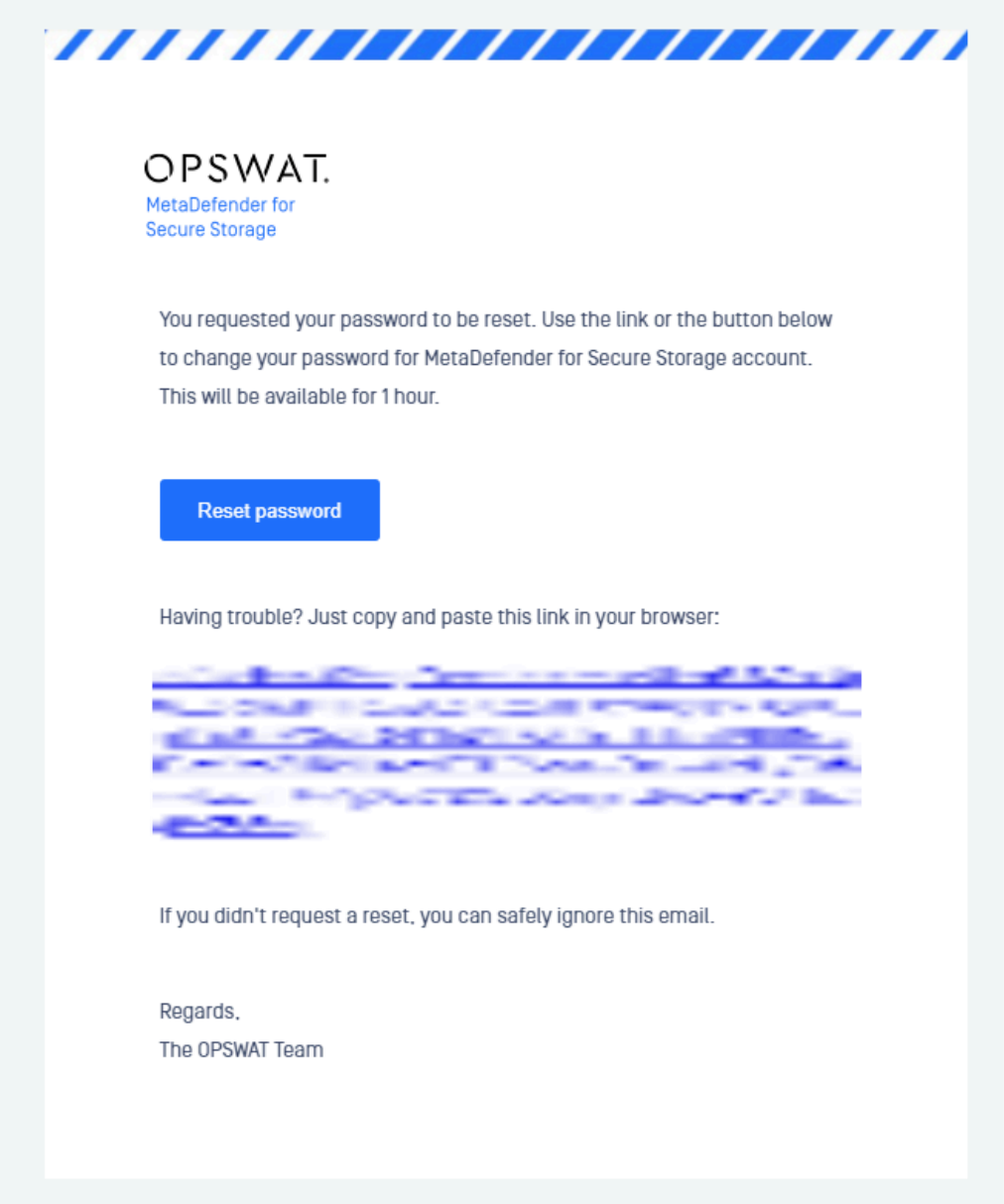
2. This will redirect you to a page where you can provide either the **username** or the **email address** associated with the account you want to change the password for. To continue, please click **Submit**.



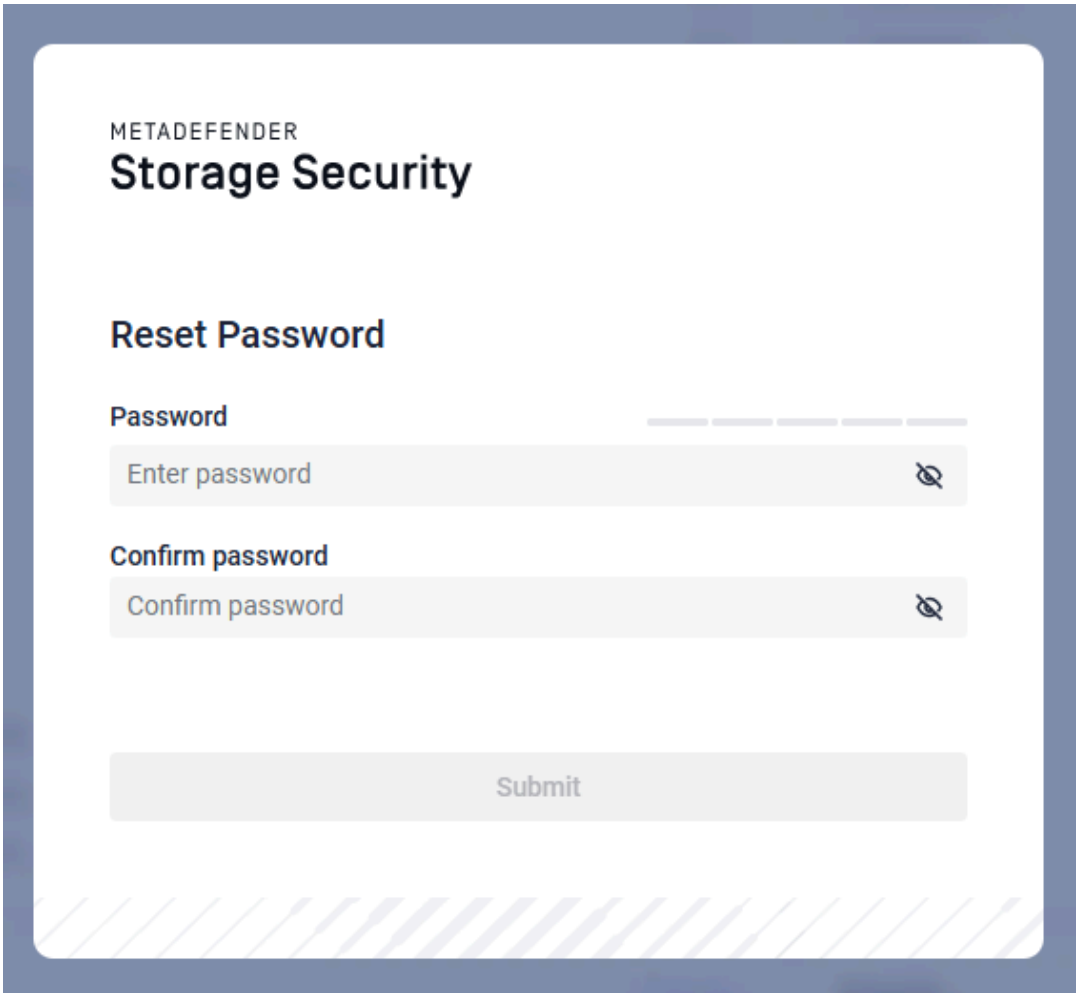
3. If the request is successful, an informative message will appear on the screen.



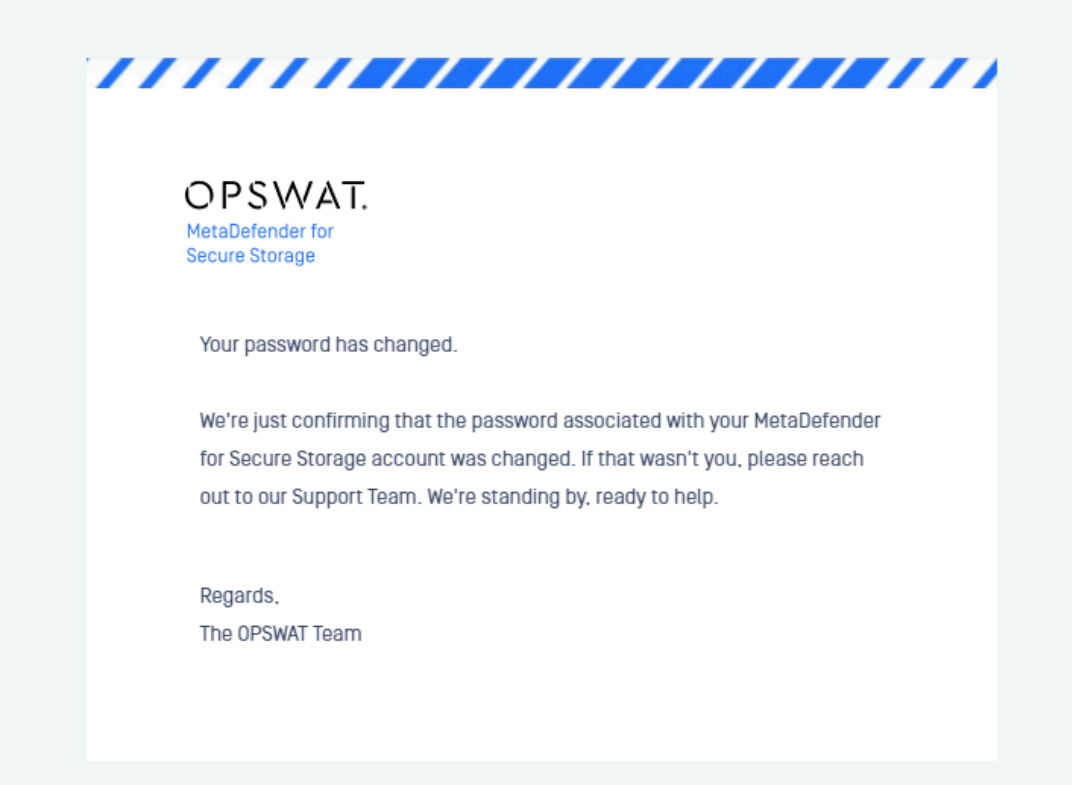
4. An email will be sent to the email address associated with the account. By clicking on the **Reset password** button or directly on the link, you will be redirected to a new tab in browser to continue with the password reset.



5. In the new tab, you are asked to provide the new password and to confirm it. Please click on **Submit** to finish the password reset.

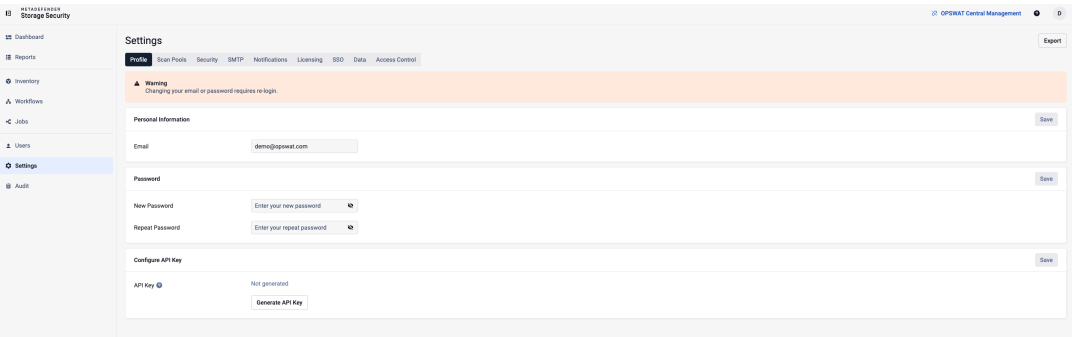


6. After resetting the password, you will be redirected to the login page and a confirmation email will be sent to your email address.



Change your account information

Navigate to **Settings** tab -> **Profile**.



Change password

1. Please enter the current password associated with the account.
2. After validating it, you will be allowed to provide the new password and to confirm it.
3. To finish changing the password, please click on **Update password**.

Change email

1. Please enter the current password associated with the account.
2. After validating it, you will be allowed to provide a new email address.

3. To finish changing the email, please click on **Update email**.

Change full name

1. Provide the new full name you wish to use for the account.
2. To finish changing the full name, please click on **Update full name**.

Generate API Key

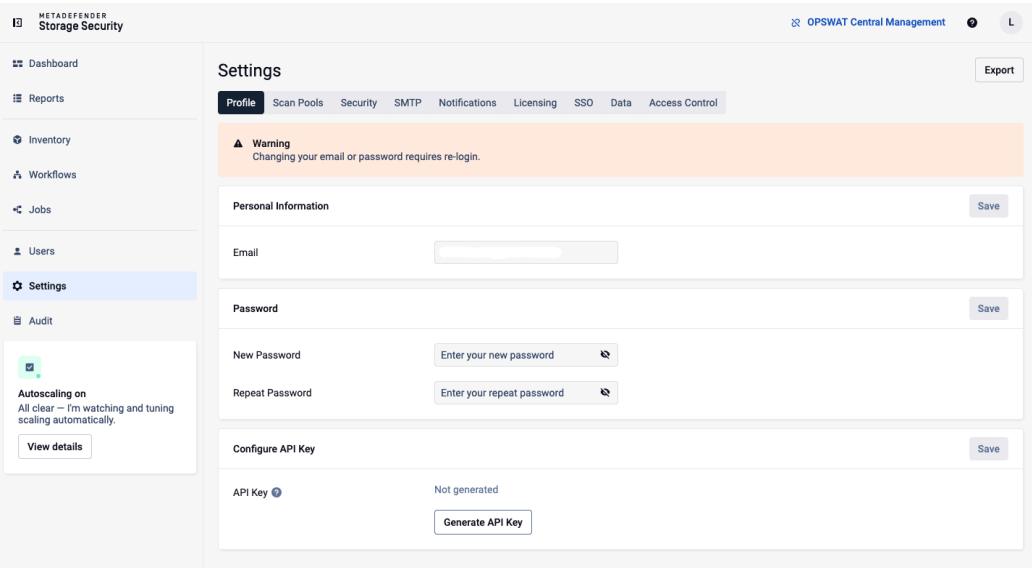
1. Click on **Generate API Key** button
2. Copy the generated API Key
3. Click on **Save** button

Generate an API Key

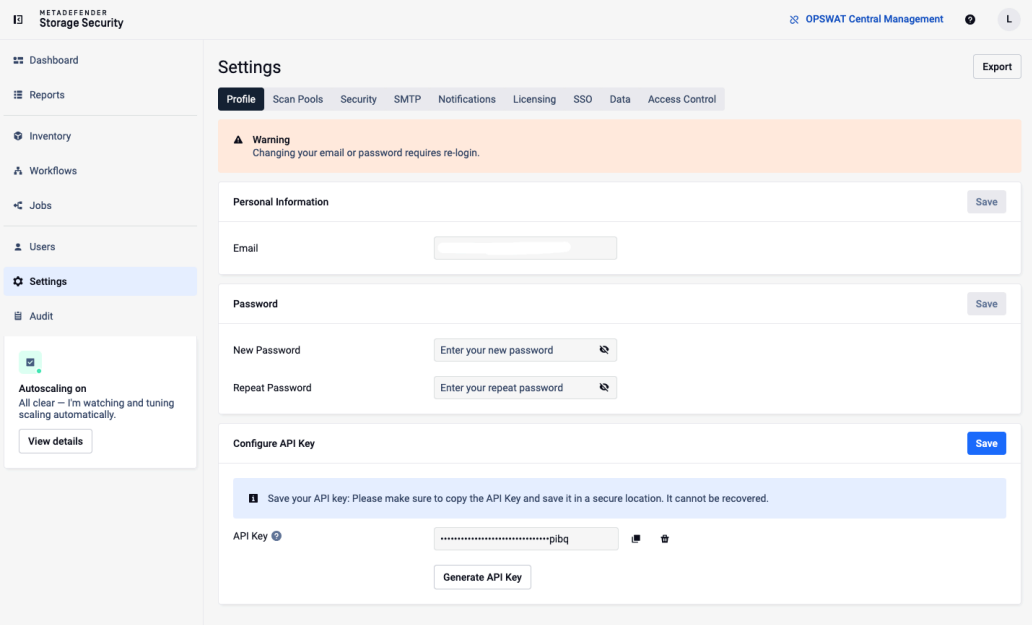
MetaDefender Storage Security users can generate **one API Key** for their user account. This key can be used as an alternative method to authenticate REST API requests to MetaDefender Storage Security — with the exception of user and API key management operations.

To generate an API key, follow these steps:

1. Navigate to **Settings** tab -> **Profile**



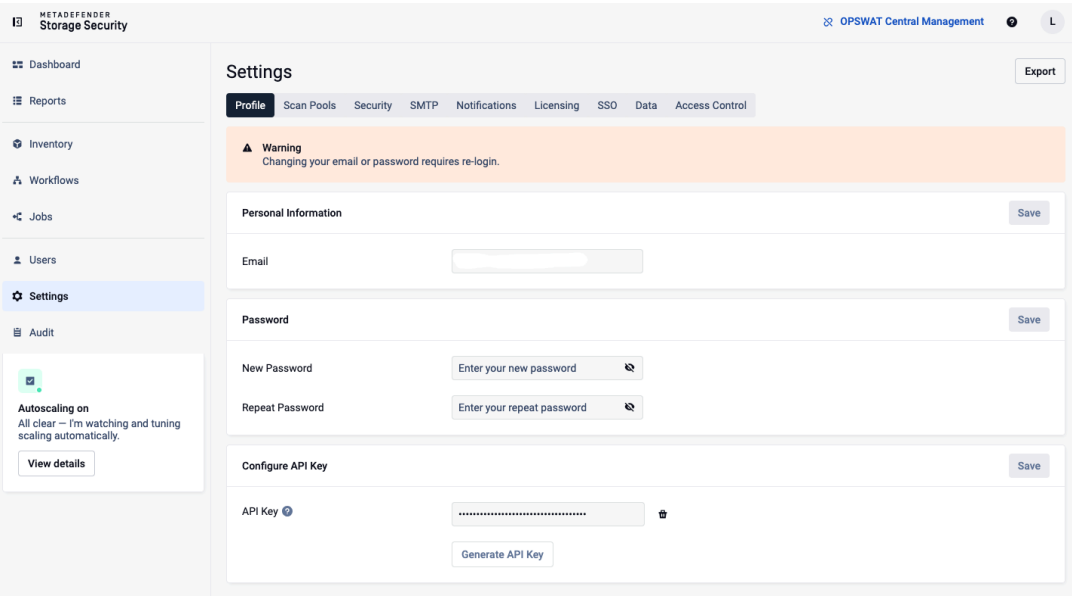
2. Select **Generate API Key**



3. Click the copy button and store your API Key somewhere secure. Afterwards, click on **Save**.

Information

The API key can be copied only at this step and is saved only after clicking the save changes button. If an API Key already exists, it can be deleted or updated to a freshly generated one.



Warning

API keys of a user are invalidated on a password change or reset operation.

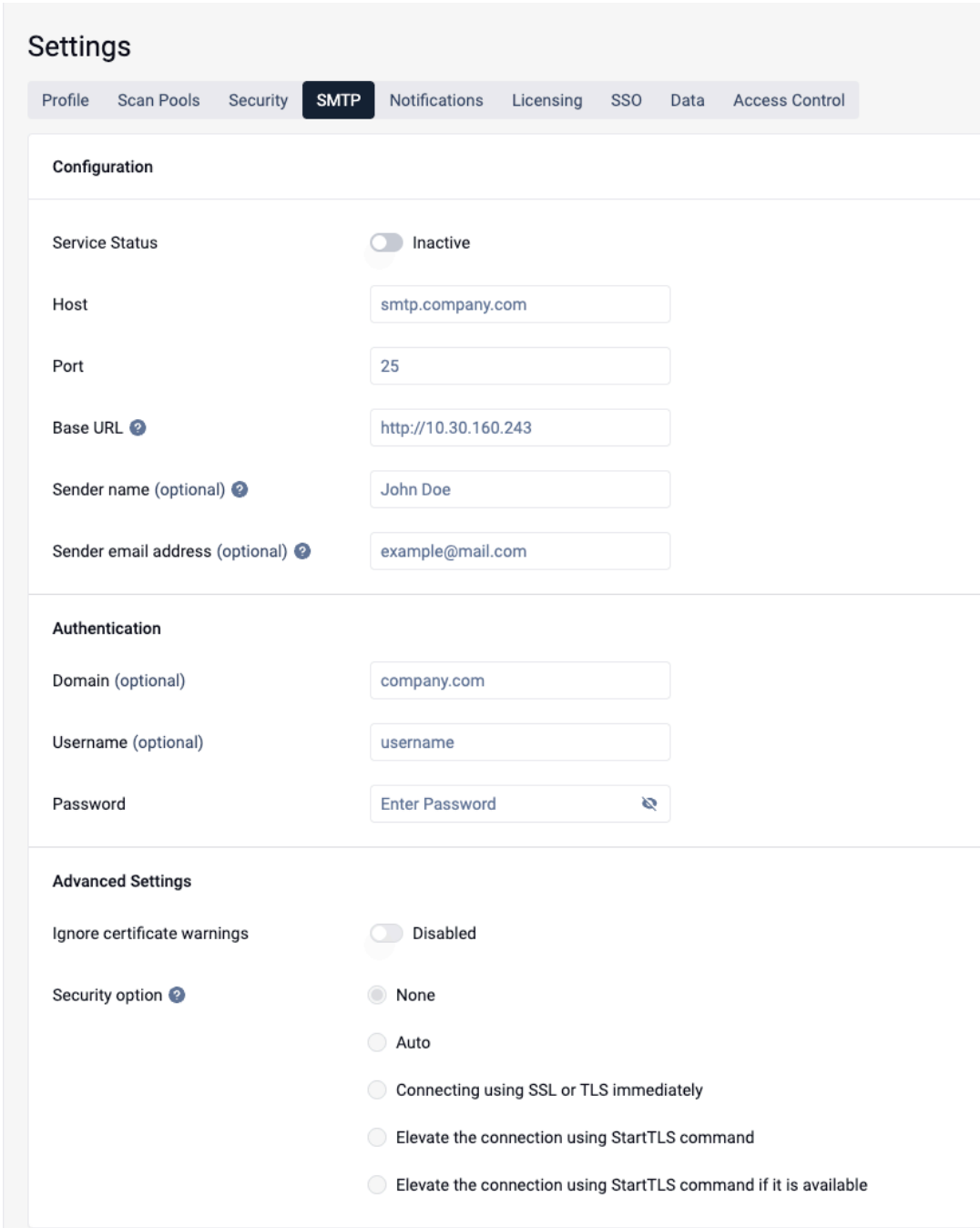
Warning

The following operations are not allowed using an API key:

- Generate an API key: **GET /api/apikey/generate**
- Retrieve an API key: **GET /api/apikey**
- Add an API key: **POST /api/apikey**
- Update an API key: **PUT /api/apikey**
- Delete an API key: **DELETE /api/apikey/{id}**
- Enumerate active users: **GET /api/user/active/{startIndex}/{count}**
- Create an user: **POST /api/user/create**
- Update user role: **PUT /api/user/role**
- Enumerate user requests: **GET /api/user/requests**
- Remove a user: **DELETE /api/user/{userId}**
- Update user: **PUT /api/user**
- Validate password: **POST /api/user/validation**

SMTP

In order to generate email reports, MetaDefender Storage Security needs to connect to an SMTP server. To configure this integration, please go to **Settings → SMTP**.



Setup the integration

1. Enable SMTP by clicking the toggle
2. Input the **Host** address of your SMTP server (i.e smtp.company.com)

3. Input the **Port** of your SMTP server
4. **Base URL**: the address to use in the email for links; this address should point to the MetaDefender Storage Security web interface
5. **Sender Email Address:** email address to be used as *From:* field when sending notifications [**Note:** if left empty, we will use the default email address: *mdss@opswat.com*]
6. **Sender Name**: display name to be used as *From Name:* field when sending notifications [**Note:** if left empty, we will use the default sender name: *MetaDefender Storage Security*]
7. [Optionally] Input a **domain, username,** and **password** if your SMTP requires authentication
    1. You can leave these fields empty if your SMTP server does not require authentication
    2. You can leave the domain empty if your SMTP provider requires only a username and a password
8. Advanced Settings
    1. **Security option:** provides a way of specifying the SSL and/or TLS encryption that should be used for a connection. Please select one of the following options [**Note:** default option is None]
        1. **None:** No SSL or TLS encryption should be used
        2. **Auto:** Allow the system to decide which SSL or TLS options to use [default]. If the server does not support SSL or TLS, then the connection will continue without any encryption
        3. **Connect using SSL or TLS immediately:** The connection should use SSL or TLS encryption immediately
        4. **Elevate the connection using StartTLS command:** Elevates the connection to use TLS encryption immediately after reading the greeting and capabilities of the server. If the server does not support the STARTTLS extension, then the connection will fail
        5. **Elevate the connection using StartTLS command if it is available:** Elevates the connection to use TLS encryption immediately after reading the greeting and capabilities of the server, but only if the server supports the STARTTLS extension
    2. **Ignore Certificate Warnings:** select this option if your SMTP does not have a valid certificate but wish to use SSL
9. **Save** your settings when you are done.

> **Info**

> In order to test the configuration, MetaDefender Storage Security will attempt to send a test email using the email address of the currently logged-in user. Please make sure that your account has a valid email configured.

License

A MetaDefender Storage Security license contains the following fields

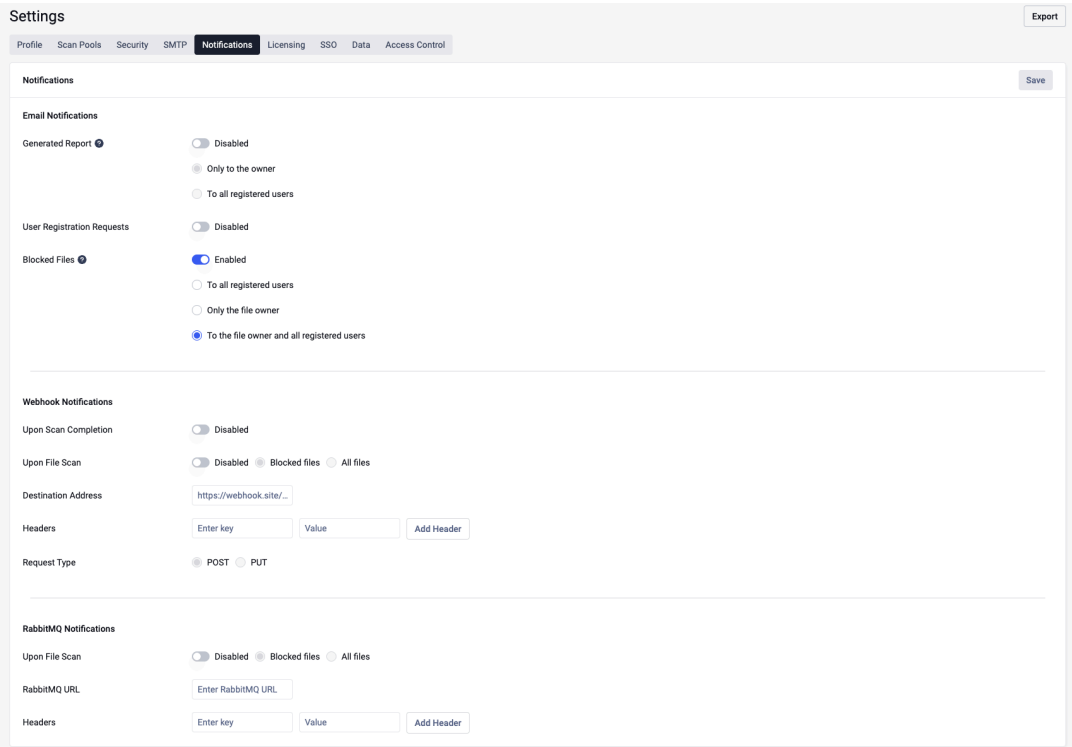
Field	Description
Deployment ID	A uniquely generated value that identifies your deployment.
Product ID	A unique product identifier
Expiration Date	The date until your license is valid
Name	The name of the license
Activation Key	The key with which the license is activated

Notifications

> **Info**
>
> In order to send email notifications, make sure that an SMTP server is available. Please see SMTP for more details.

To enable notifications follow these steps:

1. Navigate to **Settings -> Notifications**
2. Choose the event for which you wish to receive notifications. There are two types of notifications, each having different possible configurations:
    1. Email notifications, which can be sent:
        1. When a report is generated. Please see Email reports.
        2. When a user registration request is pending an administrator approval
        3. When a file is blocked while processing a storage. Please see Blocked file notifications.
    2. Webhook notifications, which can be sent:
        1. When a scan is completed. Please see Scan completed Notifications
        2. When a file is completed. Please see File scanned Notifications
    3. RabbitMQ notification which can be sent when a file is completed. Please see File scanned RabbitMQ Notifications

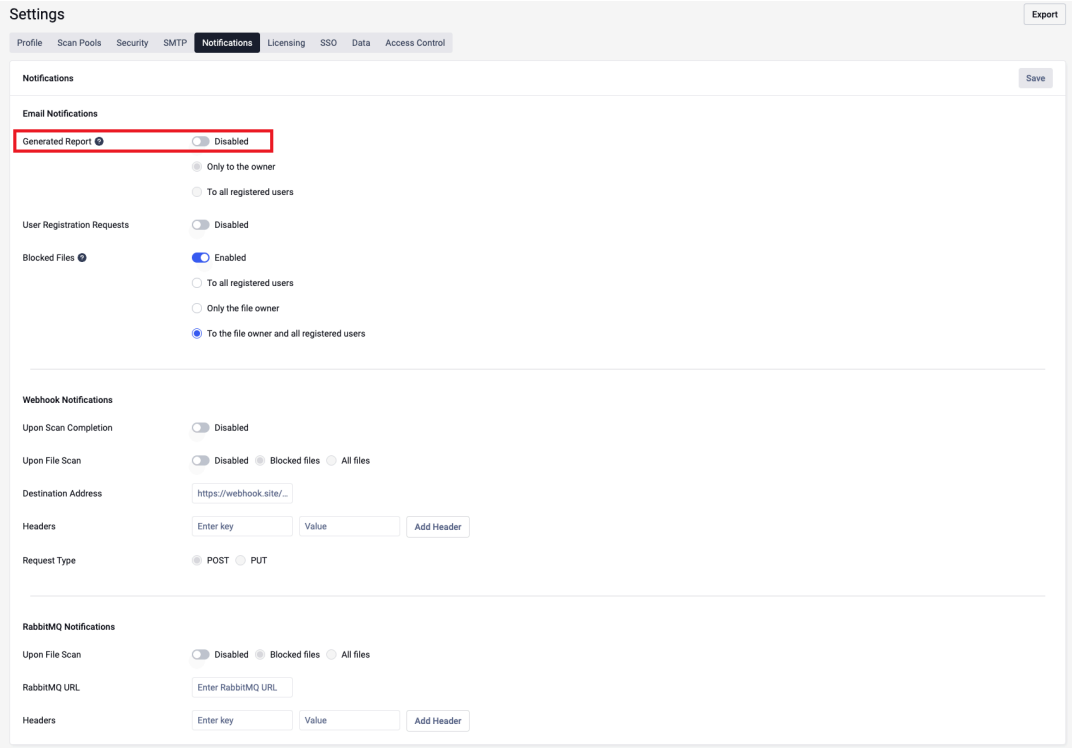


Email reports

Email reports can be generated after a storage process completes. The report contains vital information about multi-scanning results, potential compliance violations as well as detected vulnerabilities or other issues.

You can choose to send the email report:

- Only to the owner - the user that started the process
- All registered users - all MetaDefender Storage Security users



An email report contains the following information:

- A list of Top Risk Files
- An overview of the report
- Engine versions for Deep CDR and Vulnerability engine
- Details engine information for multi-scanning engines

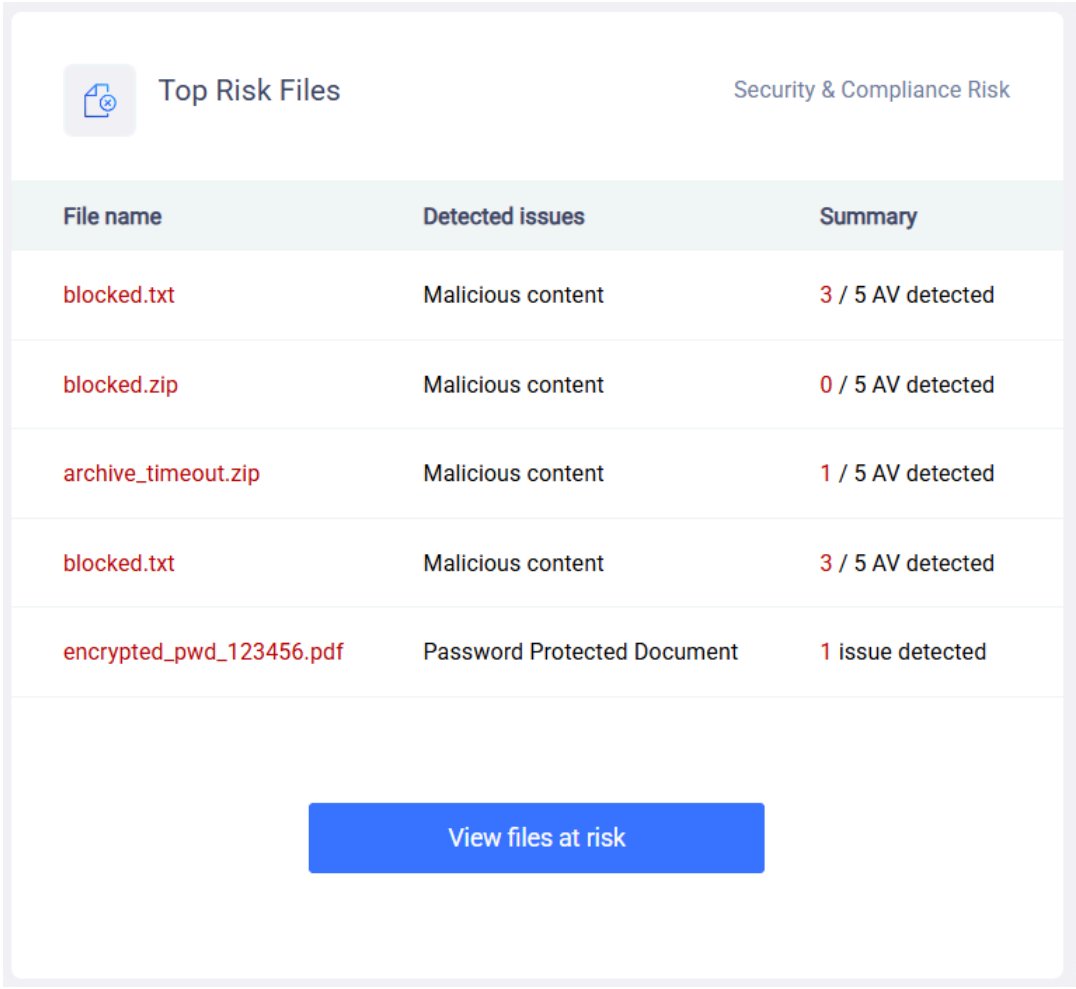
OPSWAT.
MetaDefender
Storage Security

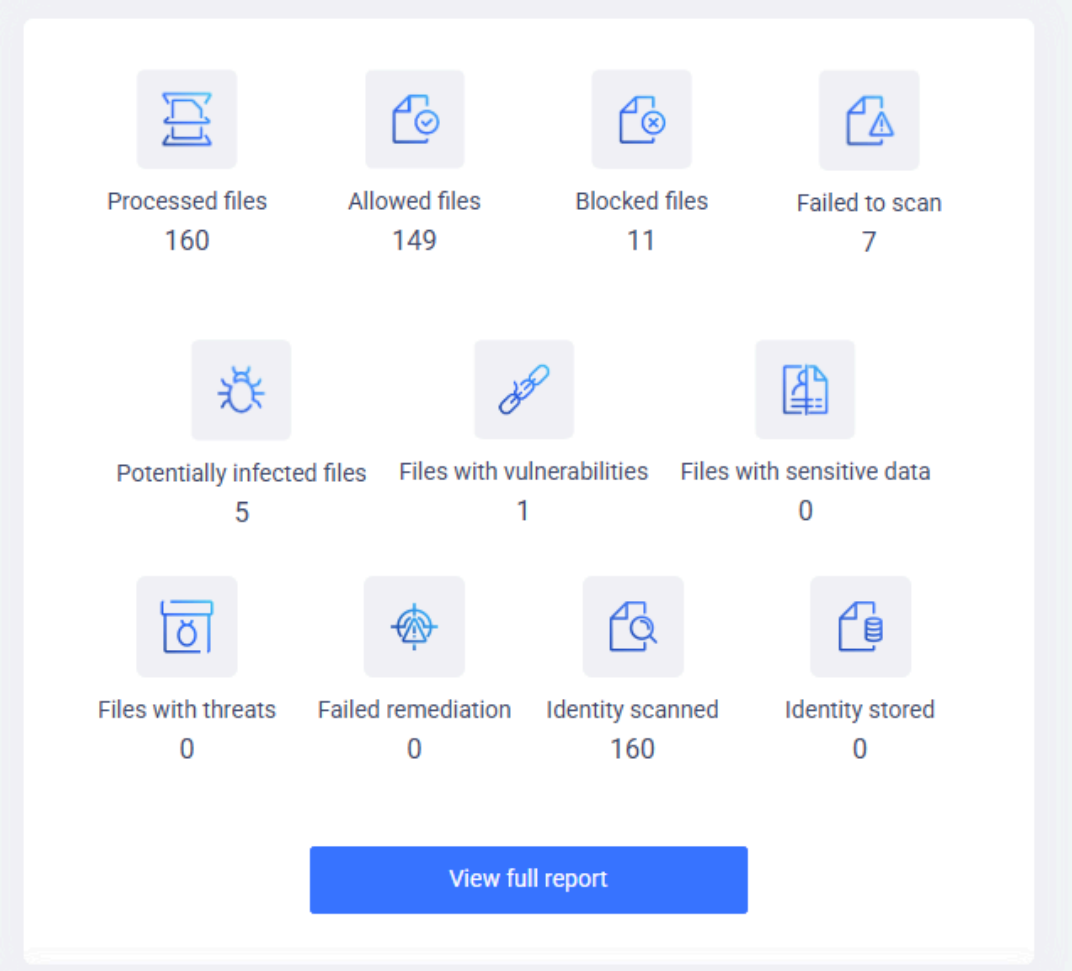
Report generated
Aug 5, 2026 10:48 UTC

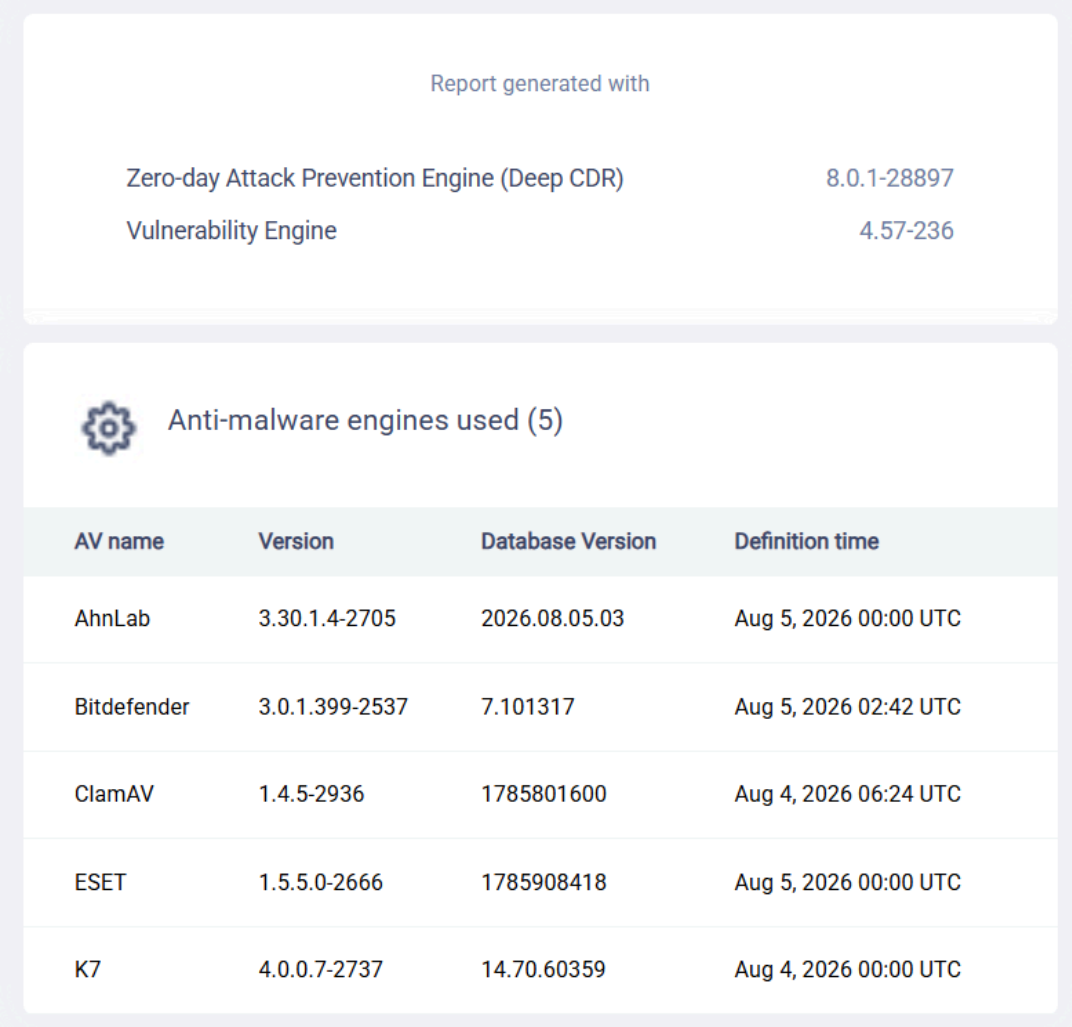
Protecting storage **Automation Library**
Scan name **OnDemand-2026_08_05_13_47_30**
Workflow Used **Default Workflow**

Type	Status
Instant	Completed

Start Time	Stop Time	Duration
Aug 5, 2026 10:47 UTC	Aug 5, 2026 10:48 UTC	0d 00h 00m 44s


Top Risk Files
Security & Compliance Risk
File name
Detected issues
Summary
blocked.txt
Malicious content
3 / 5 AV detected
blocked.zip
Malicious content
0 / 5 AV detected
archive_timeout.zip
Malicious content
1 / 5 AV detected
blocked.txt
Malicious content
3 / 5 AV detected
encrypted_pwd_123456.pdf
Password Protected Document
1 issue detected
View files at risk


Processed files
160
Allowed files
149
Blocked files
11
Failed to scan
7
Potentially infected files
5
Files with vulnerabilities
1
Files with sensitive data
0
Files with threats
0
Failed remediation
0
Identity scanned
160
Identity stored
0
View full report



Report generated with

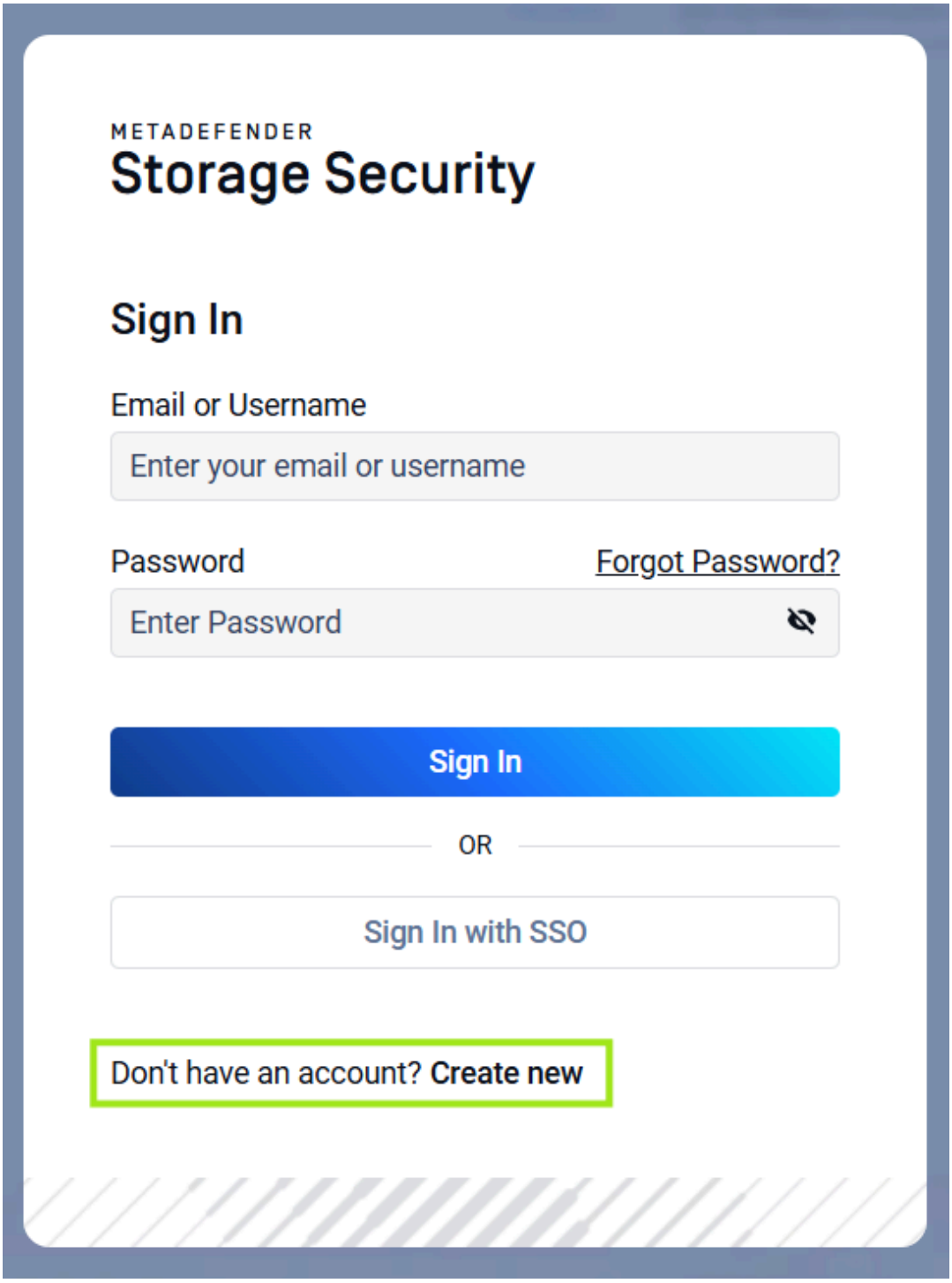
Zero-day Attack Prevention Engine (Deep CDR)	8.0.1-28897
Vulnerability Engine	4.57-236

Anti-malware engines used (5)

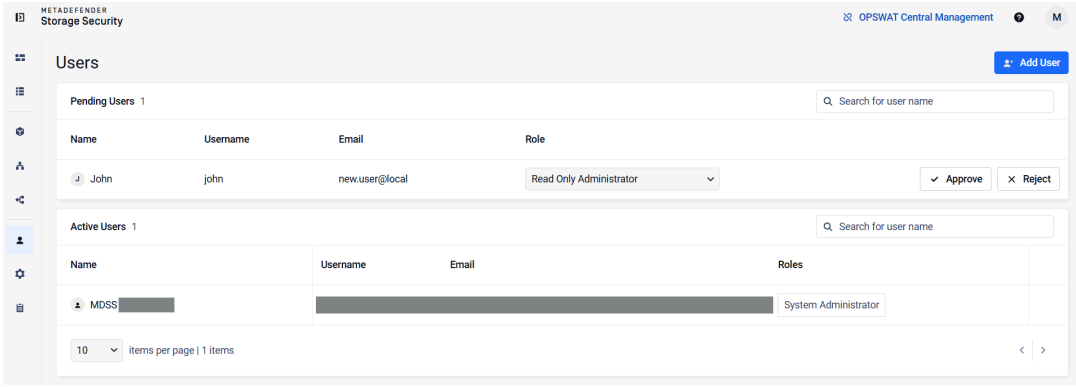
AV name	Version	Database Version	Definition time
AhnLab	3.30.1.4-2705	2026.08.05.03	Aug 5, 2026 00:00 UTC
Bitdefender	3.0.1.399-2537	7.101317	Aug 5, 2026 02:42 UTC
ClamAV	1.4.5-2936	1785801600	Aug 4, 2026 06:24 UTC
ESET	1.5.5.0-2666	1785908418	Aug 5, 2026 00:00 UTC
K7	4.0.0.7-2737	14.70.60359	Aug 4, 2026 00:00 UTC

User registration request notifications

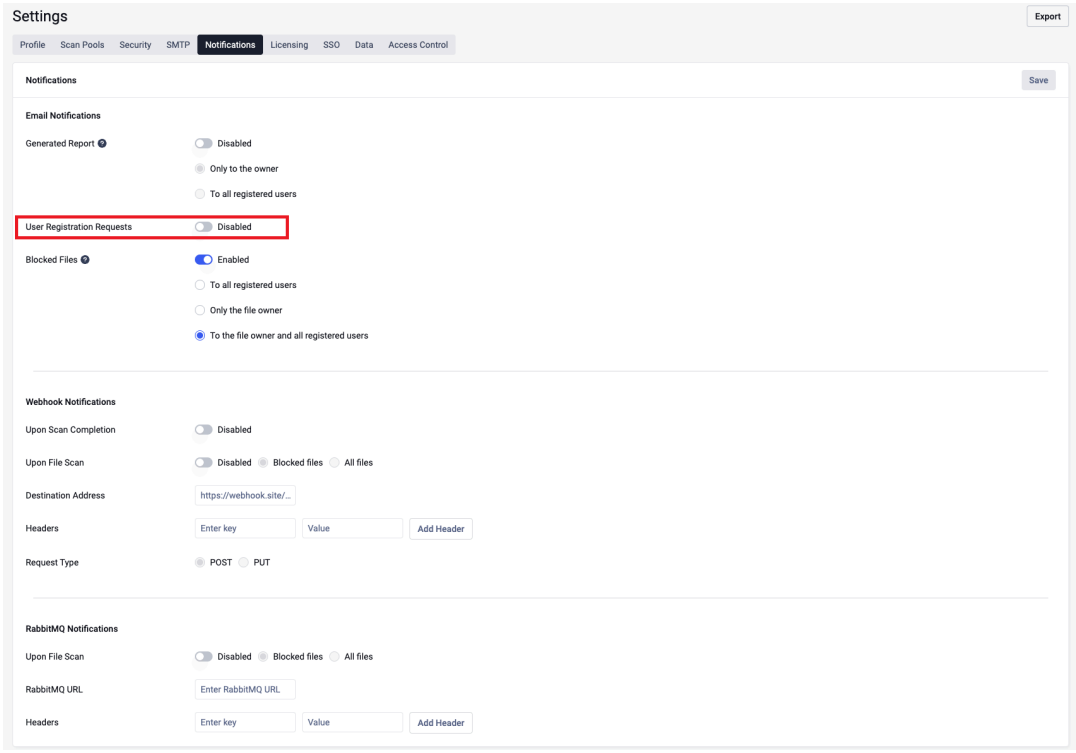
New users can attempt to register accounts from the login page:



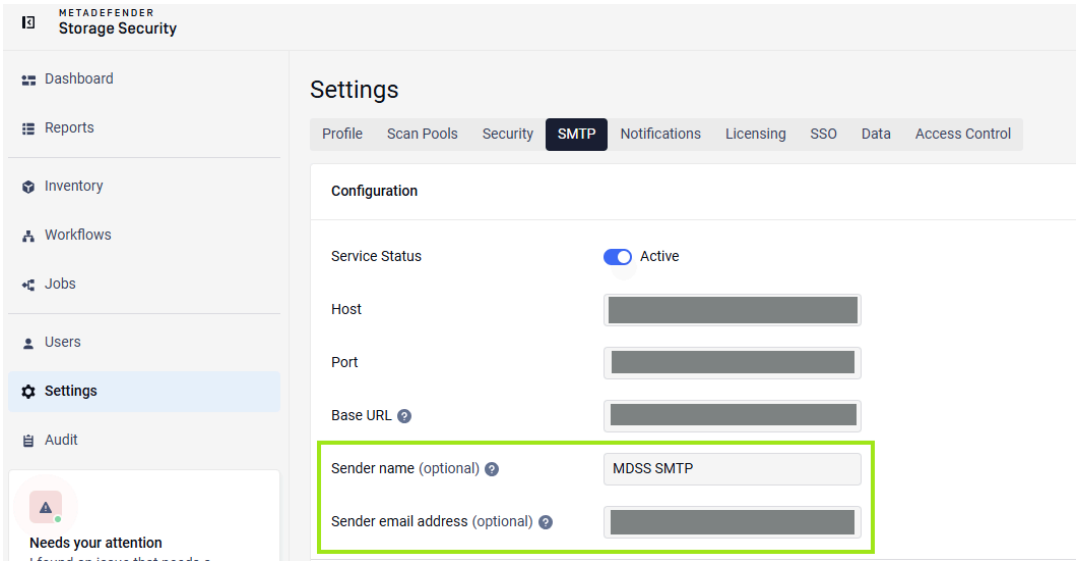
These registration requests need to be verified and approve by MDSS System Administrators.



To enable notifications for user registration requests, please select the **User Registration Requests** option from the **Email Notifications** section:

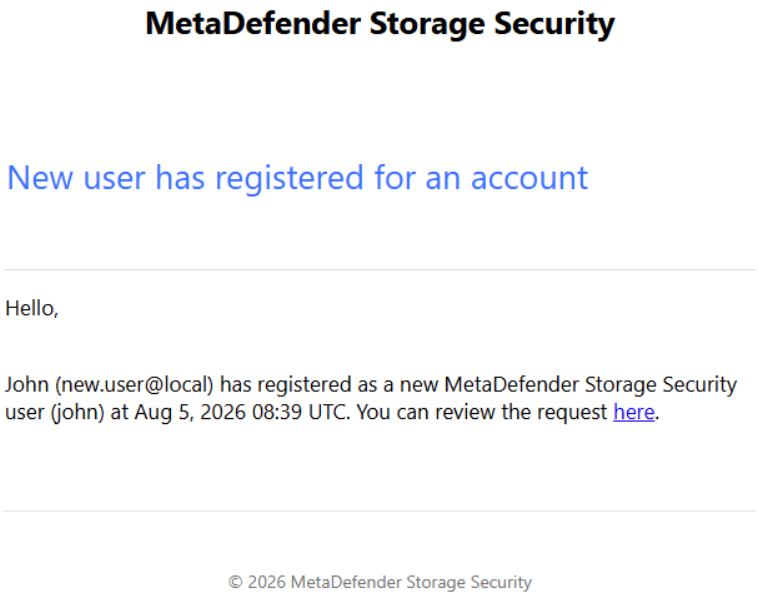


> Always triple check the sender of the email before trusting ANY links.



The email contains the following information:

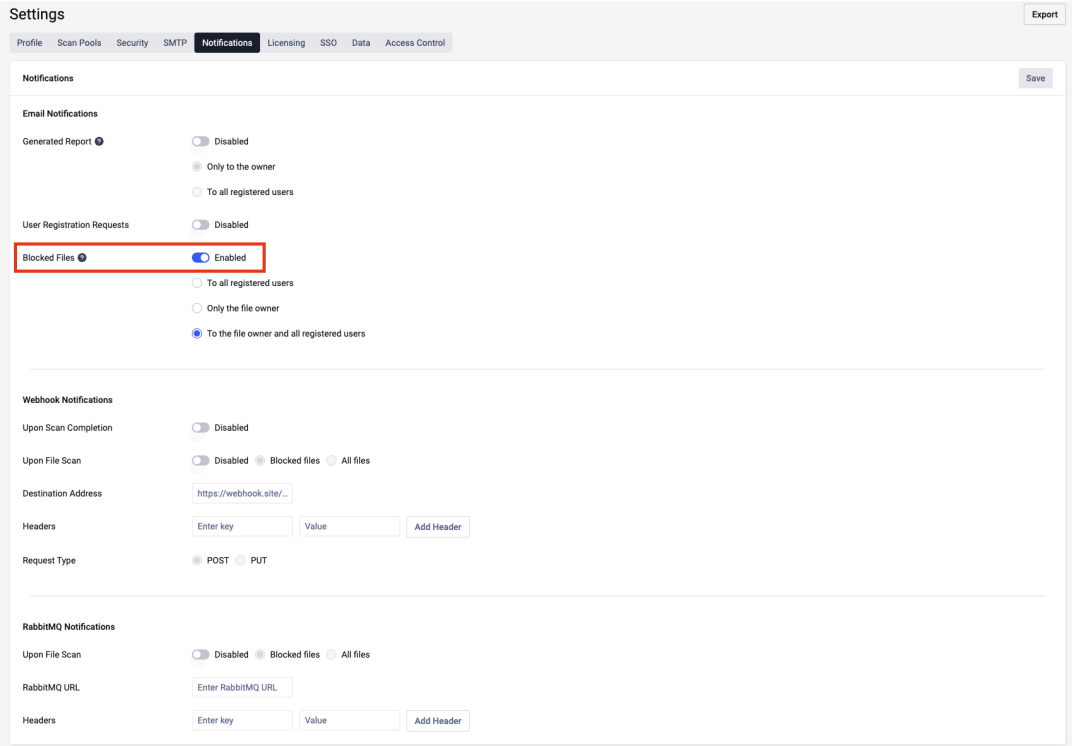
- Full Name of of registered user
- Email address of registered user
- Date of registration
- Link to the MDSS instance's Users tab



Blocked file notifications

Blocked files notifications are sent for each blocked file that is found during an Instant, Scheduled or Real-Time processing.

To enable notifications for blocked file, select the option to **Blocked Files** from the **Email Notifications** section:



You can choose to send this email to:

- All MetaDefender Storage Security registered users
- Only to the file owner
- To the file owner and all MetaDefender Storage Security registered users

> **Info**
>
> Please note that the notification to the file owner could be sent only for OneDrive, Box and MFT integrations.

The email contains the following information:

- File name
- File path
- Storage unit type and name
- The reason why the file is blocked

- Remediation actions details (delete/move blocked files), when these are configured for the storage that is processed


OPSWAT.
MetaDefender for
Secure Storage

Hello,

The file [redacted] has been blocked while scanning the S3 Compatible storage (Storage Name [redacted]), for the following reason: Sensitive Data Found.

The file has been moved due to remediation configured for the S3 Compatible storage (Storage Name: [redacted]) to: Box storage (Storage Name: [redacted]).

Regards,
The OPSWAT Team

Scan completed notifications

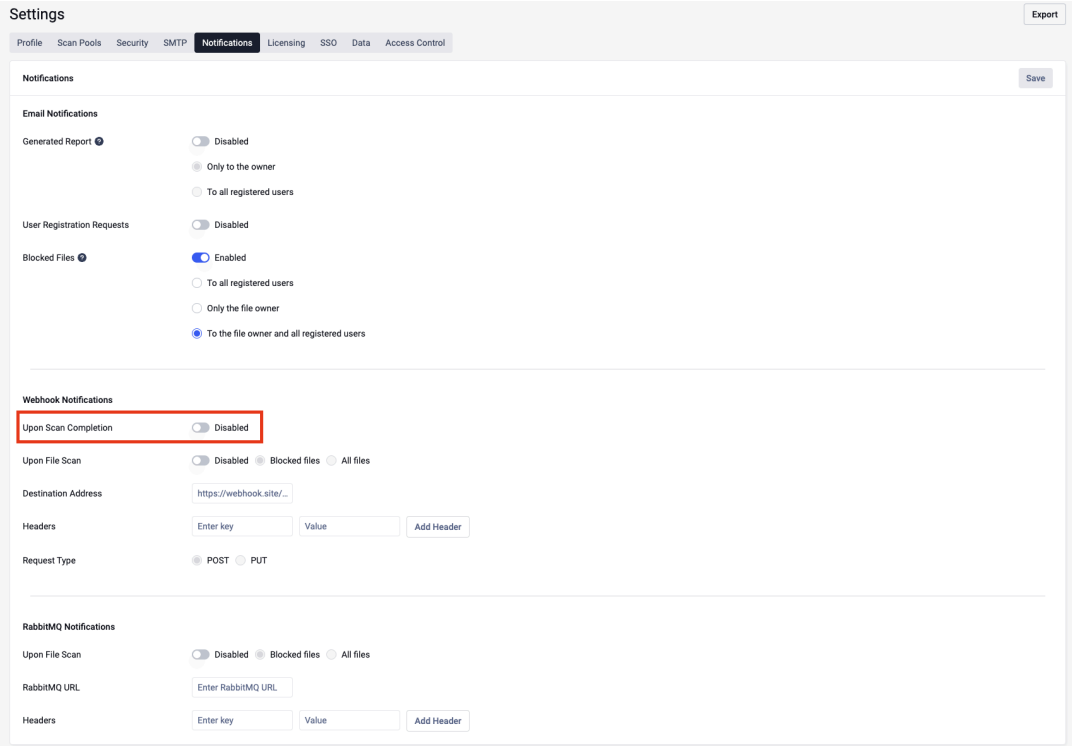
Overview

Scan Completed webhook notifications allow you to receive real-time alerts whenever an Instant or Scheduled scan has finished running in MetaDefender Storage Security. This feature is useful if you need to:

- Track the completion status of scans for auditing and compliance.
- Monitor whether scans succeed, fail, or are cancelled.
- Integrate scan lifecycle events into external systems (e.g., scheduling dashboards, reporting tools, SIEM).
- Trigger automated follow-up processes once a scan has completed.

Setup

1. Go to **Settings > Notifications** in the MetaDefender Storage Security console.
2. Select **Upon Scan Completion** under the **Webhook Notifications** section.
3. Enter the **Destination Address** (your endpoint).
4. Select the **Request Type** (**POST** or **PUT**).
5. (Optional) Add any custom **HTTP headers**.
6. Save the configuration.



Configuration Details

- **Destination address** – The URL of the receiving system (must accept HTTP requests).
- **Request type** – Either `POST` or `PUT`.
- **Headers** – Optional key-value pairs (e.g., `Authorization: Bearer <token>`).

Webhook Payload

When triggered, the webhook sends a JSON request body to your configured endpoint.

```
{
   "ScanId":"01987f90-870a-7826-9115-18d96a1a4ae0",
   "ScanName":"second-scan",
   "StartTime":"2025-08-06T13:27:14.953181Z",
   "StopTime":"2025-08-06T13:27:20.12684Z",
   "Status":"Completed",
   "WebhookNotificationType": "ScanCompleted"
}
```


Field Reference

Field	Type	Example Value	Description
`ScanId`	string(UUID)	`01987f90-870a-7826-9115-18d96a1a4ae0`	Unique identifier of the scan.
`ScanName`	string	`second-scan`	Name of the scan as configured.
`StartTime`	datetime (UTC)	`2025-08-06T13:27:14.953181Z`	Time the scan started.
`StopTime`	datetime (UTC)	`2025-08-06T13:27:20.12684Z`	Time the scan finished.
`Status`	string	`Completed` /`Cancelled` /`Failed`	Final status of the scan.
`WebhookNotificationType`	string	`ScanCompleted`	Identifies the notification type.

Delivery Behavior

- Webhooks are delivered as standard HTTP requests.
- Delivery is **best-effort** → if the destination is invalid, the request is logged in the **Audit page**, but no retries are performed.
- Payload size is small (JSON only, no scan contents).

Security Considerations

- Configure HTTPS endpoints to ensure data is transmitted securely.
- Use authentication headers (e.g., API tokens) to protect your webhook endpoint.

- Ensure your endpoint validates payloads before processing.

Troubleshooting

- **No webhook received** → Verify the endpoint URL and request type (POST/PUT).
- **Audit shows delivery failed** → Endpoint is invalid or unreachable. Fix and test again.
- **Authentication errors** → Confirm your endpoint accepts the headers you configured.

File scanned notifications

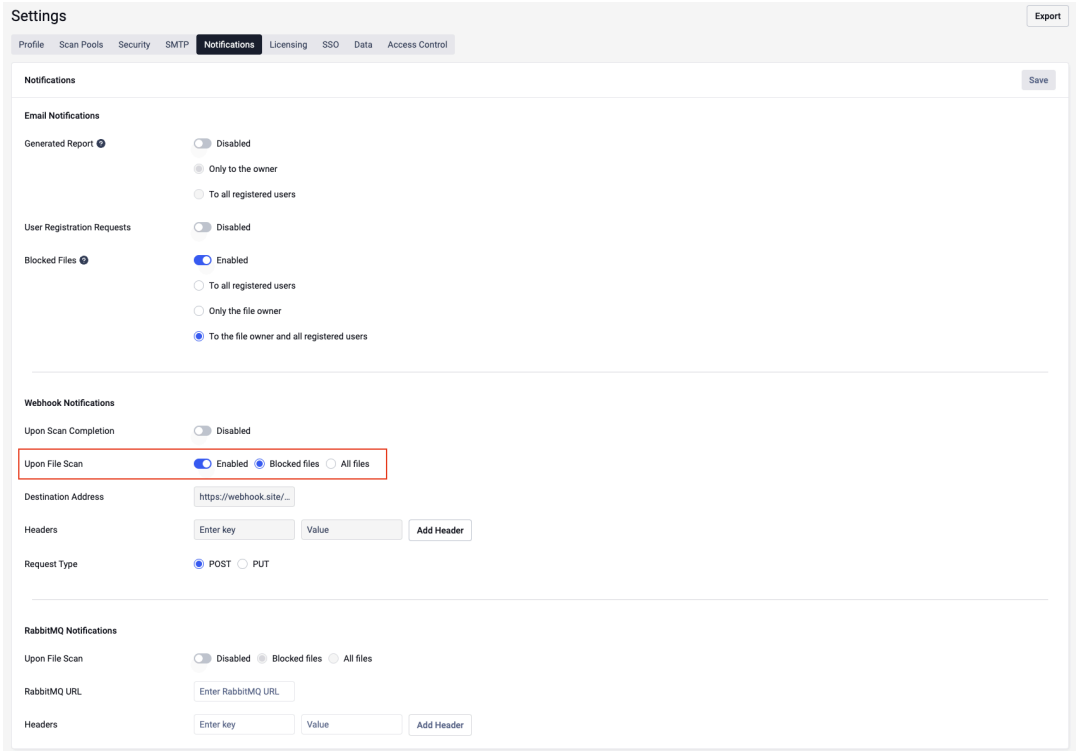
Overview

File Scanned webhook notifications allow you to receive real-time alerts whenever a file has been scanned by MetaDefender Storage Security. This feature is useful if you need to:

- Track the outcome of file scans for auditing or compliance.
- Integrate scan results into external systems (e.g., ticketing, alerting, SIEM).
- Trigger automated workflows based on file status (e.g., quarantine, user notification).

Setup

1. Go to **Settings > Notifications** in the MetaDefender Storage Security console.
2. Select **Upon File Scan** under the **Webhook Notifications** section
3. Choose whether you want to receive notifications for **all scanned files** or **only blocked files**.
4. Enter the **Destination Address** (your endpoint).
5. Select the **Request Type** (**POST** or **PUT**).
6. (Optional) Add any custom **HTTP headers**.
7. Save the configuration.



Configuration Details

- **Destination address** – The URL of the receiving system (must accept HTTP requests).
- **Request type** – Either `POST` or `PUT`.
- **Headers** – Optional key-value pairs (e.g., `Authorization: Bearer <token>`).
- **Notification scope** – Choose between:
  - **All scanned files** → You will receive a notification for every file scanned.
  - **Blocked only** → You will only receive notifications when a file is marked as blocked in MetaDefender Storage Security (see File Definition)

Webhook Payload

When triggered, the webhook sends a JSON request body to your configured endpoint.

```
{
  "FileId": "0198f02f-ed4d-7c0d-8668-1036d322209c",
  "FileName": "file.txt",
  "FilePath": "folder/file.txt",
  "FileStatus": "Allowed",
  "CoreScanResult": "No Threat Detected",
  "FileState": "Available",
  "RemediationResult": [
    "AllowedKept: Applied"
  ],
  "ScanCompletedTimeStamp": "2025-08-28T10:20:24.974998Z",
  "WorkflowName": "Default Workflow",
  "Rules": [
    "MetaDefender Storage Security"
  ],
  "ExceptionDetails": "",
  "RemediationErrors": [],
  "WebhookNotificationType": "FileScanned"
}
```

Field Reference

Field	Type	Example Value	Description
`FileId`	string (UUID)	`0198f02f-ed4d-7c0d-8668-1036d322209c`	Unique identifier of the file.
`FileName`	string	`file.txt`	Original file name.
`FilePath`	string	`folder/file.txt`	File's relative path.
`FileStatus`	string	`Allowed/Blocked`	File status based on MDSS file definitions
`CoreScanResult`	string	`No Threat Detected`	Result from Scan instance.
`FileState`	string	`Failed/Canceled/FailedRemediation/Available`	File status in MDSS.
`RemediationResult`	array[string]	`["AllowedKept:Applied"]`	Array of remediations applied and their result
`ScanCompletedTimeStamp`	datetime (UTC)	`2025-08-28T10:20:24.974998Z`	Time when scan completed.
`WorkflowName`	string	`Default Workflow`	Name of the workflow used.
`Rules`	array[string]	`["MetaDefender Storage Security"]`	Workflow rules applied.
`ExceptionDetails`	string	""	Details of any exceptions during scan.
`RemediationErrors`	array[string]	`[]`	Errors during remediation, if any.
`WebhookNotificationType`	string	`FileScanned`	Identifies the notification type.

Delivery Behavior

- Webhooks are delivered as standard HTTP requests.
- Delivery is **best-effort** → if the destination is invalid, the request is logged in the **Audit page**, but **no retries** are performed.
- Payload size is typically small (JSON only, no file contents).

Security Considerations

- Configure HTTPS endpoints to ensure data is transmitted securely.
- Use authentication headers (e.g., API tokens) to protect your webhook endpoint.
- Ensure your endpoint validates payloads before processing.

Troubleshooting

- **No webhook received** → Verify the endpoint URL and request type (POST/PUT).
- **Audit shows delivery failed** → Endpoint is invalid. Fix and test again.
- **Authentication errors** → Confirm your endpoint accepts the headers you configured.

File scanned RabbitMQ notifications

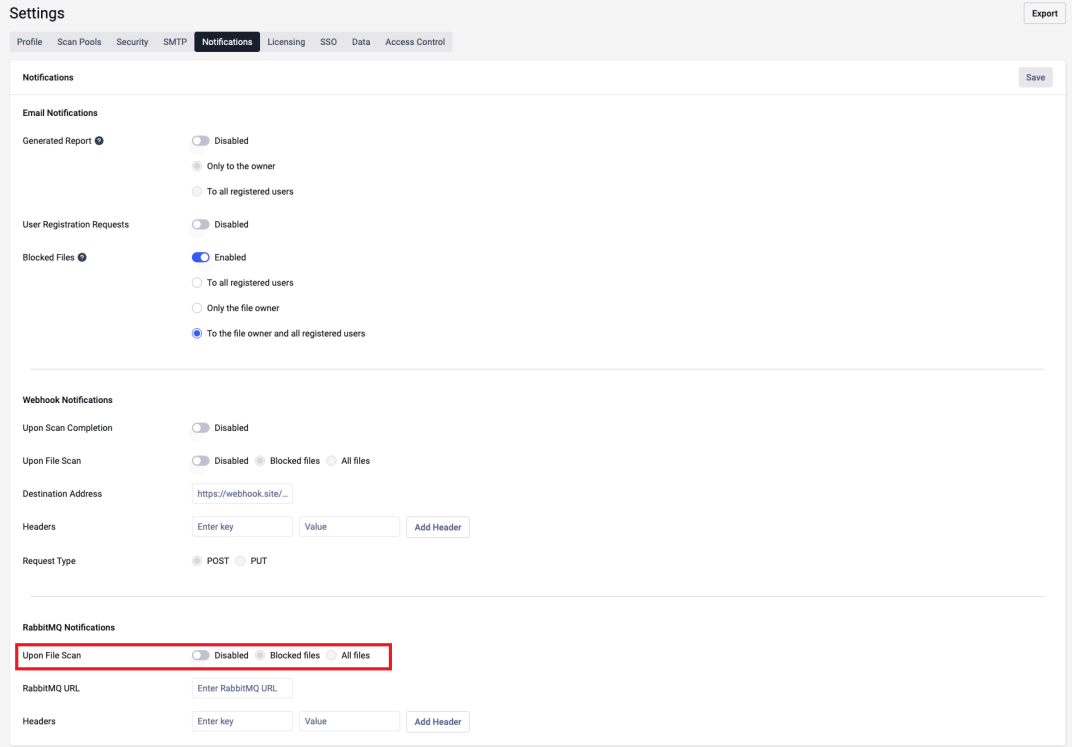
Overview

File Scanned RabbitMQ notifications allow you to receive real-time alerts whenever a file has been scanned by MetaDefender Storage Security. This feature is useful if you need to:

- Enable cross-MDSS instance scanning where one MDSS instance publishes scan results and another automatically rescans the file

Setup

1. Go to **Settings > Notifications** in the MetaDefender Storage Security console.
2. Select **Upon File Scan** from the **RabbitMQ Notifications **section.
3. Choose whether you want to receive notifications for **all scanned files** or **only blocked files**.
4. Enter the **RabbitMQ URL** (your endpoint).
5. (Optional) Add any custom **headers**.
6. Save the configuration.



Configuration Details

- **RabbitMQ URL** – The URL of the receiving system (ex: amqp://username:passowrd@localhost:27049).
- **Headers** – Optional key-value pairs (e.g., `StorageClientId: string; ScanId: string`).
  - StorageClientId - destination storage client Id
  - ScanId - destination scanId
- **Notification scope** – Choose between:
  - **All scanned files** → You will receive a notification for every file scanned.
  - **Blocked only** → You will only receive notifications when a file is marked as blocked in MetaDefender Storage Security (see File Definition)

RabbitMQ Payload

When triggered, the RabbitMQ sends a JSON request body to your configured endpoint.

```
{
  "isPublicWebhook": false,
  "metadata": {
    "serializedEventMetadata": ""
  },
  "metadataToBuild": {
    "serializedEventMetadata": "{\"FileId\":\"019a0b68-1c71-7b7b-ac39-
8357658c3220\",\"FileName\":\"text1.txt\",\"FilePath\":\"folder1/text1.
txt\",\"FileStatus\":\"Allowed\",\"CoreScanResult\":\"No Threat
Detected\",\"FileState\":\"Available\",\"RemediationResult\":
[],\"ScanCompletedTimeStamp\":\"2025-10-
22T10:12:49.647929Z\",\"WorkflowName\":\"Default Workflow\",\"Rules\":
[\"MetaDefender Storage
Security\"],\"ExceptionDetails\":\"System.ObjectDisposedException: The
CancellationTokenSource has been disposed.\\r\\n   at
System.Threading.CancellationTokenSource.get_Token()\\r\\n   at
opswat.mdcs.scanning.service.ScanningService.HandleScanResults(IPipelin
e`2 getScanResultsQueryPipeline, IMediator mediator,
ObjectReadyForScanEvent message, ScanningData scanningData,
CancellationTokenSource cts, InitiateScanCommandResponse
initiateScanCommandResponse) in C:\\\\Projects\\\\md-cloud-
services\\\\Backend\\\\src\\\\opswat.mdcs.scanning.service\\\\Services\
\\\Classes\\\\ScanningService.cs:line 427\\r\\n   at
opswat.mdcs.scanning.service.ScanningService.ObjectReadyForScanEventHan
dler(ObjectReadyForScanEvent message, CancellationToken
cancellationToken) in C:\\\\Projects\\\\md-cloud-
services\\\\Backend\\\\src\\\\opswat.mdcs.scanning.service\\\\Services\
\\\Classes\\\\ScanningService.cs:line 357\",\"RemediationErrors\":
[],\"WebhookNotificationType\":\"FileScanned\"}"
  },
  "scanId": "01985f77-2bf6-7895-bf6f-ceced4c95a87",
  "storageClientId": "01985f77-2b01-764a-bc6e-6cc3e38f3761"
}
```

```
{
  "FileId": "0198f02f-ed4d-7c0d-8668-1036d322209c",
  "FileName": "file.txt",
  "FilePath": "folder/file.txt",
  "FileStatus": "Allowed",
  "CoreScanResult": "No Threat Detected",
  "FileState": "Available",
  "RemediationResult": [
    "AllowedKept: Applied"
  ],
  "ScanCompletedTimeStamp": "2025-08-28T10:20:24.974998Z",
  "WorkflowName": "Default Workflow",
  "Rules": [
    "MetaDefender Storage Security"
  ],
  "ExceptionDetails": "",
  "RemediationErrors": [],
  "WebhookNotificationType": "FileScanned"
}
```


Field Reference

Field	Type	Example Value	Description
`ScanId`	string (UUID)	`01985f77-2b01-764a-bc6e-6cc3e38f3761`	Unique identifier of the destination scan.
`StorageClientId`	string	`01985f77-2b01-764a-bc6e-6cc3e38f3761`	Unique identifier of the destination storage client.
`IsPublicWebhook`	bool	`false`	Used only for webhook
`Metadata`	WebHookMetadataRtp	“”	Empty for RabbitMQ notification
`MetadataToBuild`	WebHookMetadataRtp	`Serialized FileScannedNotificationDetails`	Populated with FileScannedNotificationDetails

Delivery Behavior

- Cross-MDSS Instance Scanning: When MDSS Instance A completes a scan and publishes to RabbitMQ, Instance B consumes the notification and automatically triggers an Event-Based RTP scan for the same file.
- Delivery is **best-effort** → if the destination is invalid, the request is logged in the **Audit page**, but **no retries** are performed.

Security Considerations

- The RabbitMQ URL is stored encrypted

Troubleshooting

- **No RabbitMQ message received** → Verify the endpoint URL and headers you configured.
- **Audit shows delivery failed** → Endpoint is invalid. Fix and test again.

External Loggers Configuration

This feature allows you to forward event logs from MetaDefender Storage Security to an external system.

This is essential for centralized monitoring, security analysis, and integrating with a SIEM (Security Information and Event Management) platform, such as Splunk.

Supported Platforms and Formats

- **Syslog** uses the UDP protocol and requires the RFC5424 format
- **Kafka** connects directly to an Apache Kafkla cluster.

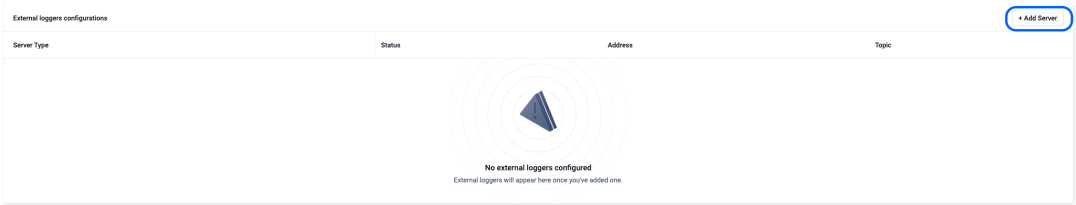
Before You Begin

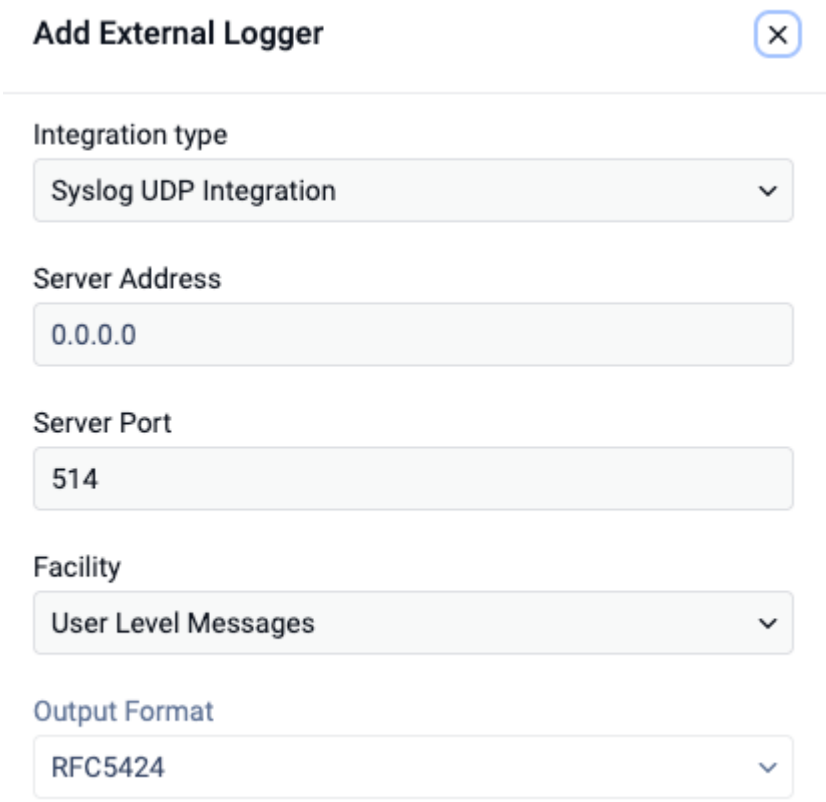
Before adding a configuration, ensure that your external logging server is already deployed and accessible from your MetaDefender Storage Security instance. You will need:

- A Syslog server ready to accept UDP traffic.
- Or, an Apache Kafka cluster with a designated topic.

How to Configure an External Logger

1. In the MetaDefender Storage Security interface, go to **Settings** > **External Loggers**.
2. Click the **+ Add new server** button to open the configuration panel.
3. Fill in the required server details, such as the server type, network address (IP or hostname), and port.
4. Enable the configuration using the toggle switch and save your changes.





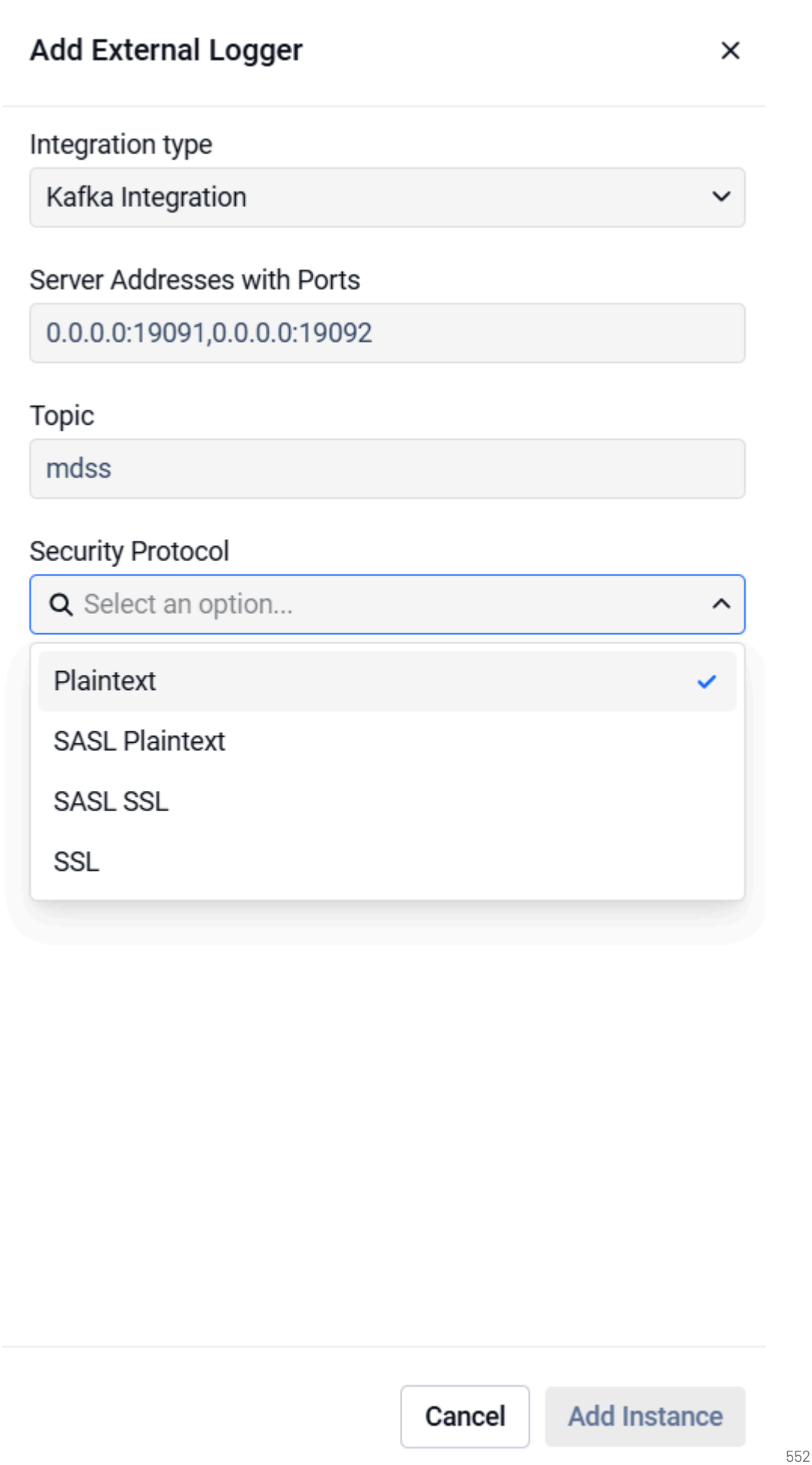
Kafka External Logger

For adding a new Kafka external logger configuration, you will need to enter the following details on the **Add External Logger window:**

- **Integration type**: Select **Kafka Integration**
- **Server addresses with ports** - Ipv4, Ipv6 and Host Name formats are supported with the port specified for each of them (multiple addresses can be separated with a comma)
- **Topic**: The Kafka topic where the logs will be sent
- **Security Protocol**: Select the desired protocol from Plaintext, SASL Plaintext, SASL SSL or SSL

For SASL Plaintext and SASL SSL security protocols, the following fields are also required:

- **SASL mechanism**: Select the desired mechanism from Plain, SCRAM SHA 256 and SCRAM SHA 512
- **SASL username**: The SASL username
- **SASL password**: The SASL password


Add External Logger
Integration type
Kafka Integration
Server Addresses with Ports
0.0.0.0:19091,0.0.0.0:19092
Topic
mdss
Security Protocol
Select an option...
Plaintext
SASL Plaintext
SASL SSL
SSL
Cancel
Add Instance

Kafka message format:

The log messages sent are sent as JSON, and contain information such as the timestamp, log type, level and the log message.


```
{
        "Timestamp": "2024-11-18T09:13:43.4018931+00:00",
        "Level": "Information",
        "MessageTemplate": "{User} logged in.",
        "RenderedMessage": "\"User (John Doe)\" logged in.",
        "Properties": {
                "User": "User(John Doe)",
                "Namespace": "ExternalLogger",
                "EventTimestamp": "2024-11-18 09:13:43",
                "UserName": "User",
                "UserId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxx",
                "Category": 3,
                "LogType": 600
        }
}
```


Logging non-audit MDSS messages to Kafka:

For logging other log messages produced by MDSS the appsettings.json can be modified to include a configuration to write to Kafka such as:

```
{
  "Serilog": {
    "MinimumLevel": {
      "Default": "Information",
      "Override": {
        "Microsoft": "Warning",
        "System": "Warning",
        "Microsoft.Hosting.Lifetime": "Information"

      }
    },
    "Enrich": [ "FromLogContext", "WithMachineName", "WithProcessId",
"WithProcessName", "WithThreadId" ],
    "WriteTo": [
      {
        "Name": "Kafka",
        "Args": {
          "bootstrapServers":
"kafkaServerAddress1:kafkaport1,kafkaServerAddress2:kafkaport2",
          "securityProtocol": "Plaintext",
          "topic": "mdss",
          "messageKey": "EventTimestamp"
        }
      }
    ]
  }
}
```

For "**securityProtocol"** values can be Plaintext, SaslSsl, SaslPlaintext or Ssl. For using SASL, the "**saslMechanism"** can be specified with values such as Plain, ScramSha512 or ScramSha256. For example, to use SASL with a Plain mechanism with username and password, the following values can be added to the Args key in the appsettings Kafka configuration.

```
{
          "saslMechanism": "Plain",
          "saslUsername": "username",
          "saslPassword": "password",
}
```

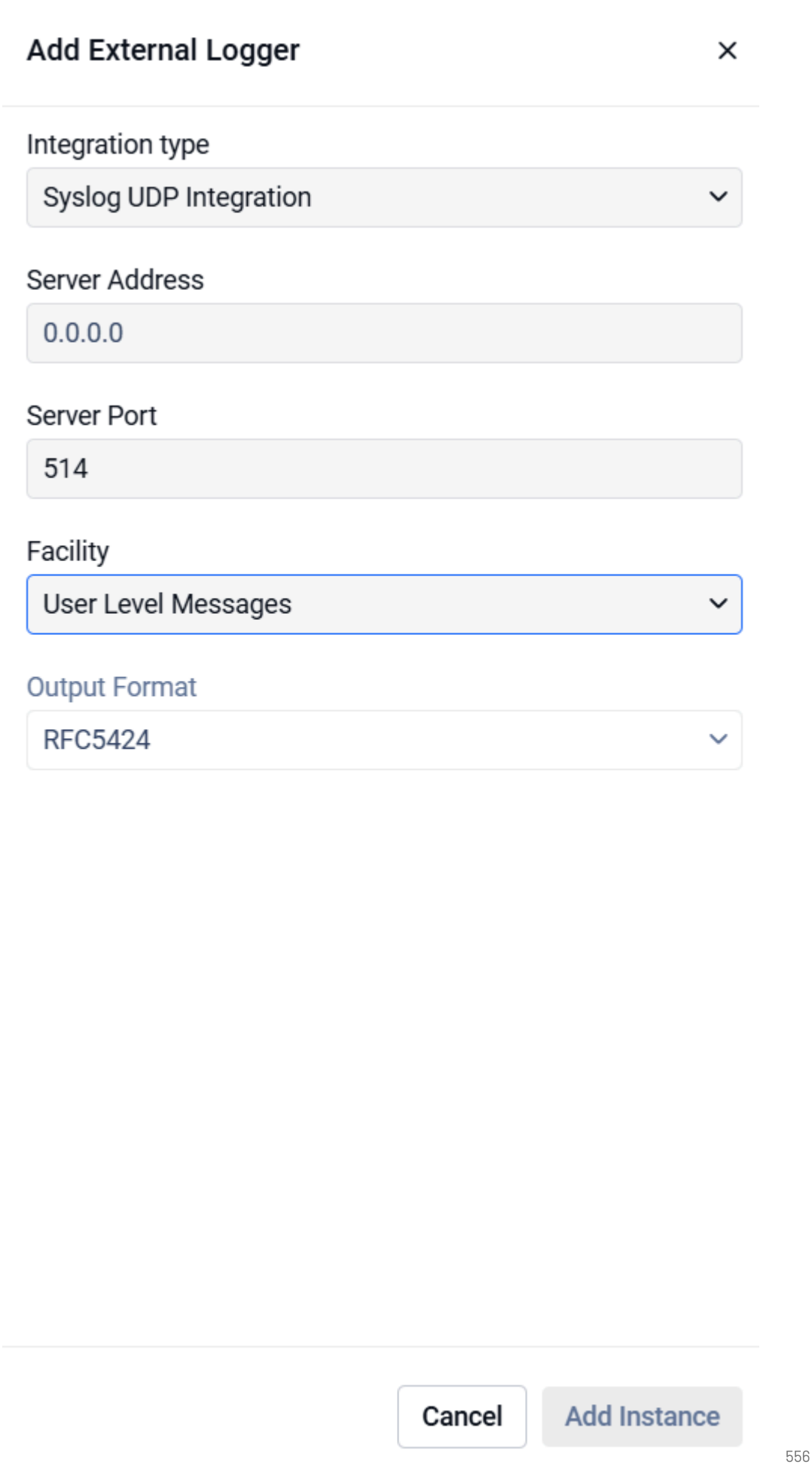
Syslog UDP External Logger

> **Info**
>
> Currently we support Syslog (UDP) using **only** the **RFC 5424** Output format as external logger configuration.

In order to visualize MetaDefender Storage Security audit logs in a centralized dashboard, integration with a Syslog (UDP) server is possible.

For adding a new Syslog configuration, you will need to enter the following details on the **Add External Logger window:**

- Server address - Ipv4, Ipv6 and Host Name formats are supported
- Port
- Syslog Facility
- Output format - RFC5424 format is supported


Add External Logger
Integration type
Syslog UDP Integration
Server Address
0.0.0.0
Server Port
514
Facility
User Level Messages
Output Format
RFC5424
Cancel
Add Instance

> **Info**
>
> Up to 5 different external loggers configurations of each type are supported.

Syslog message format

We support format **RFC5424**. You can find more details about this format here.

```
<PRI>VERSION TIMESTAMP HOSTNAME APP-NAME PROCID MSGID STRUCTURED-DATA
MSG
```

Prefix Field (click for more info)	Description
PRI	Priority value
VERSION	The version of the syslog protocol specification
TIMESTAMP	A formalized timestamp
HOSTNAME	The machine that originally sent the syslog message
APP-NAME	The device or application that originated the message
PROCID	Process name or ID associated with a syslog system
MSGID	Type of the message
STRUCTURED-DATA	A mechanism to express information in an easily parsable and interpretable data format
MSG	Free-form message that provides information about the event

```
<110>1 2024-03-26T12:27:58.276724+02:00 LP10-D8569
MetaDefenderStorageSecurity 36800 - [meta UserName="jdavis"
UserId="e69e8c2d-4dcc-4489-8f40-0df84199de52"
EventTimestamp="03/26/2024 10:27:58" Category="3" LogType="600"] John
Davis (jdavis) logged in.
```

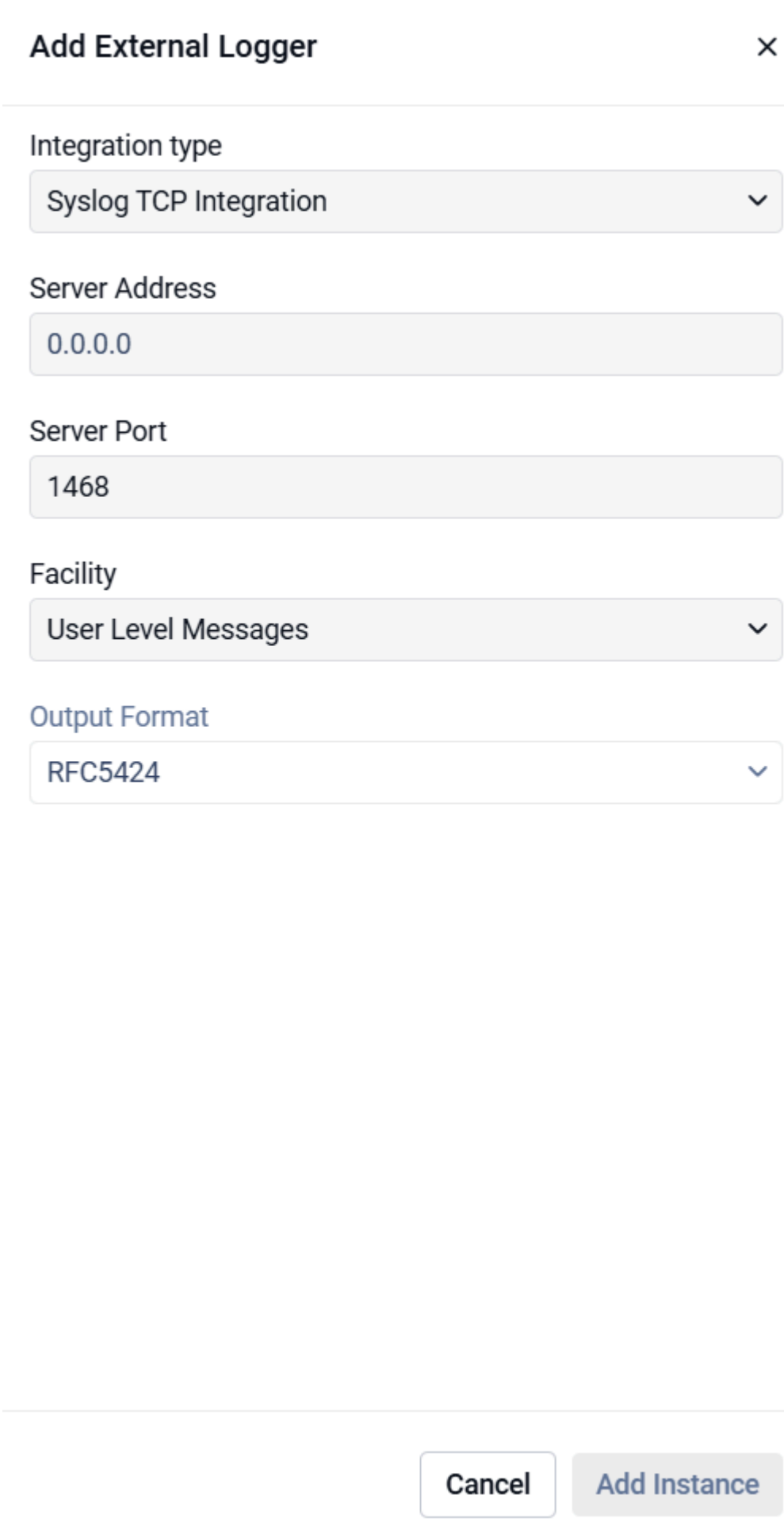
Syslog TCP External Logger

> **Info**
>
> Currently we support Syslog (TCP) using **only** the **RFC 5424** Output format as external logger configuration.

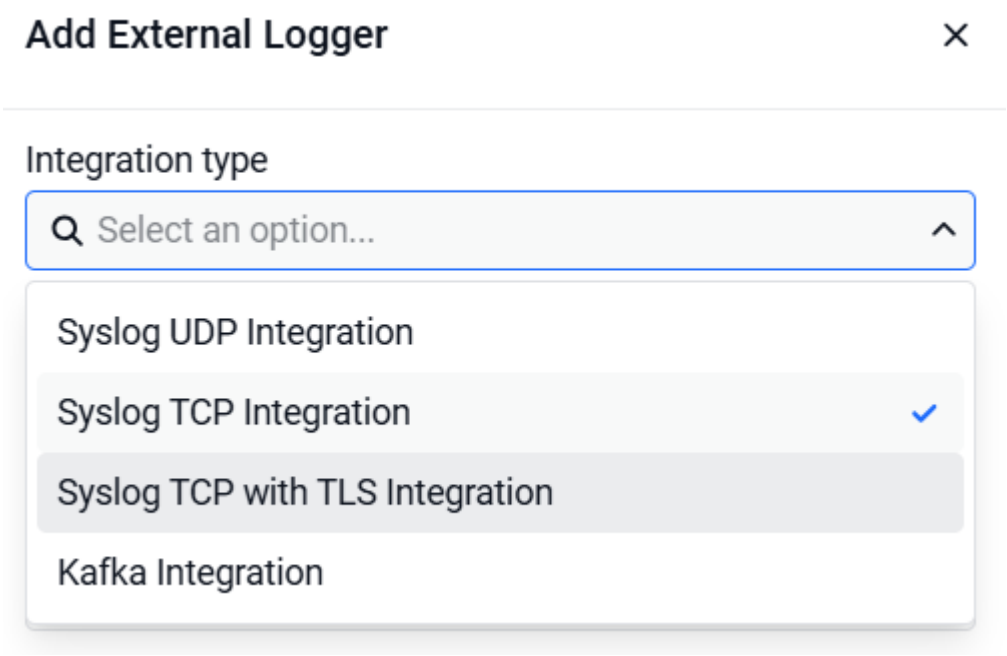
In order to visualize MetaDefender Storage Security audit logs in a centralized dashboard, integration with a Syslog (TCP) server is possible.

For adding a new Syslog configuration, you will need to enter the following details on the **Add External Logger window:**

- Server address - Ipv4, Ipv6 and Host Name formats are supported
- Port
- Syslog Facility
- Output format - RFC5424 format is supported


Add External Logger
Integration type
Syslog TCP Integration
Server Address
0.0.0.0
Server Port
1468
Facility
User Level Messages
Output Format
RFC5424
Cancel
Add Instance

The Syslog TCP integration also supports TLS certificates. MDSS might request CA certificates for the Syslog server.



> **Info**
>
> Up to 5 different external loggers configurations of each type are supported.

Syslog message format

We support format **RFC5424**. You can find more details about this format here.

```
<PRI>VERSION TIMESTAMP HOSTNAME APP-NAME PROCID MSGID STRUCTURED-DATA
MSG
```

Prefix Field (click for more info)	Description
PRI	Priority value
VERSION	The version of the syslog protocol specification
TIMESTAMP	A formalized timestamp
HOSTNAME	The machine that originally sent the syslog message
APP-NAME	The device or application that originated the message
PROCID	Process name or ID associated with a syslog system
MSGID	Type of the message
STRUCTURED-DATA	A mechanism to express information in an easily parsable and interpretable data format
MSG	Free-form message that provides information about the event

```
<110>1 2024-03-26T12:27:58.276724+02:00 LP10-D8569
MetaDefenderStorageSecurity 36800 - [meta UserName="jdavis"
UserId="e69e8c2d-4dcc-4489-8f40-0df84199de52"
EventTimestamp="03/26/2024 10:27:58" Category="3" LogType="600"] John
Davis (jdavis) logged in.
```

AWS CloudWatch External Logger

To use the AWS CloudWatch logging service, the following conditions must be met:

- AWS SSM agent installed, up to date and active on the EC2 instance (more info here)
- These IAM permissions allocated to the instance
  - AmazonSSMManagedInstanceCore
  - CloudWatchAgentServerPolicy
- AWS CloudWatch agent installed, up to date and active on the EC2 instance (more info here)

If the above conditions are met, the CloudWatch agent can be configured either through editing its configuration file or from the EC2 web UI for your instance.

When configuring the log collection section, you can set specific paths and filenames or, alternatively, you can use wildcard characters (*) to cover multiple folders and/or files. If you want to collect logs from exact paths or specific files, one separate entry needs to be created for each folder or file.

You also have the option to group logs from multiple locations into one or several log groups and streams which will then be visible in AWS CloudWatch. A log group can contain one or multiple log streams, and each log stream can be set to collect logs from one or multiple log files.

> **Info**
>
> If you group logs from multiple files into the same log stream they will be combined and displayed in chronological order, there will be no separation based on the source log files.

An example configuration to collect CPU, RAM and disk storage metrics from a Linux installation every 5 seconds as well as to pull all MetaDefender Storage Security logs using wildcards would look like this:

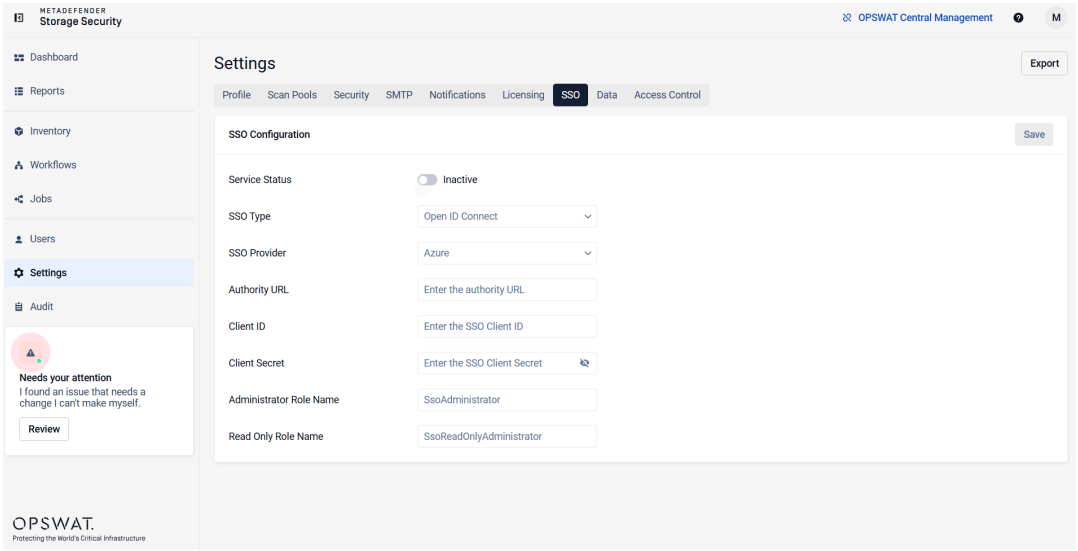
> **Note**
>
> The example below will collect **all logs** from MetaDefender Storage Security.
>
> If you would like to customize which logs are collected, please configure the paths according to the documented locations here.

```
{
    "agent": {
        "metrics_collection_interval": 60
    },
    "metrics": {
        "namespace": "CWAgent",
        "append_dimensions": {
            "InstanceId": "${aws:InstanceId}"
        },
        "metrics_collected": {
            "mem": {
                "measurement": [
                    "used_percent"
                ]
            },
            "cpu": {
                "measurement": [
                    "usage_active"
                ]
            },
            "disk": {
                "measurement": [
                    "used_percent"
                ]
            }
        }
    },
    "logs": {
        "logs_collected": {
            "files": {
                "collect_list": [
                    {
                        "file_path": "/var/log/mdss/*.log",
                        "filters": [],
                        "log_group_name": "mdss-logs",
                        "log_stream_name": "applogs"
                    },
                    {
                        "file_path": "/etc/mdss/services_logs/*/*.txt",
                        "filters": [],
                        "log_group_name": "mdss-logs",
                        "log_stream_name": "applogs"
                    }
                ]
            }
        },
        "log_stream_name": "mdss-logging"
    }
}
```

Single Sign On

In order to configure Single Sign On:

1. Navigate to **Settings -> Single Sign On**



2. **Enable** the Single Sign On toggle
3. Select the desired **Single Sign On Provider**
4. Select the desired **Single Sign On Type**
5. Enter the **Client ID**
6. Enter the **Client Secret**
7. Enter the **Authority URL**

> **Info**
>
> For the purpose of secure authentication, when configuring any SSO provider, either HTTPS need to be enabled for MDSS, or the web interface needs to be accessed through **localhost** instead of the deployment machine's IP.

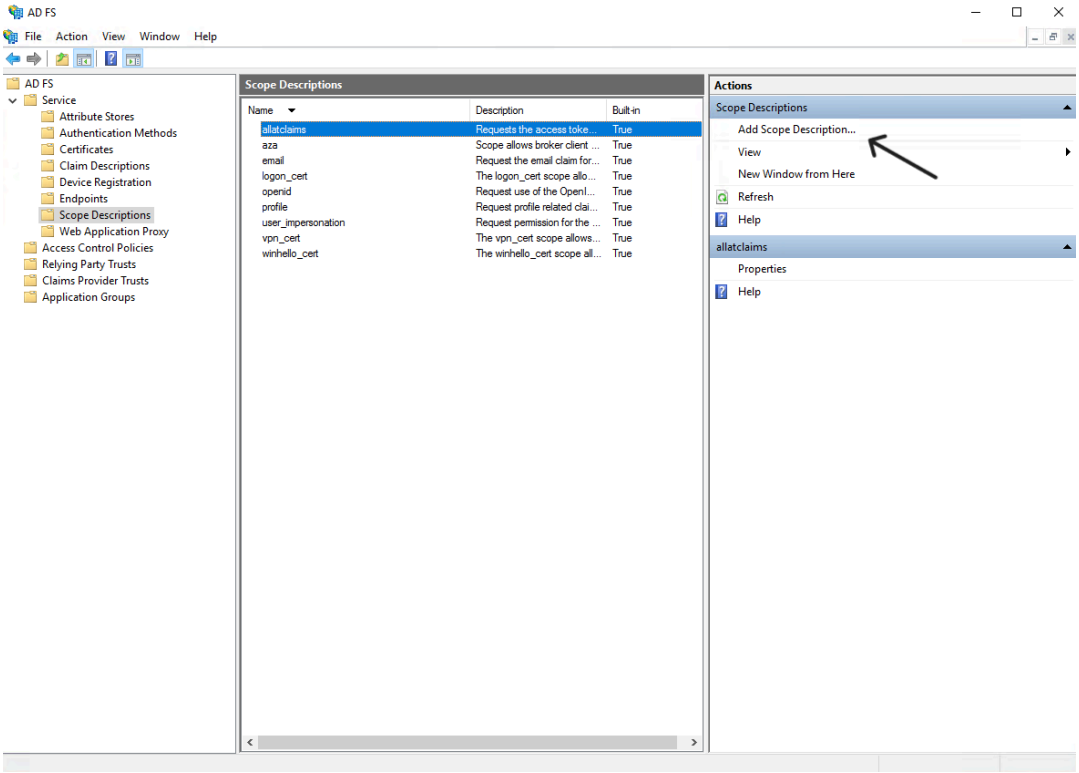
AD FS Specific Configuration

> **Info**
>
> To properly access MDSS, the following AD **Security Groups** are required:
>
> - **SsoAdministrator** - the equivalent of local MetaDefender Storage Security administrator role

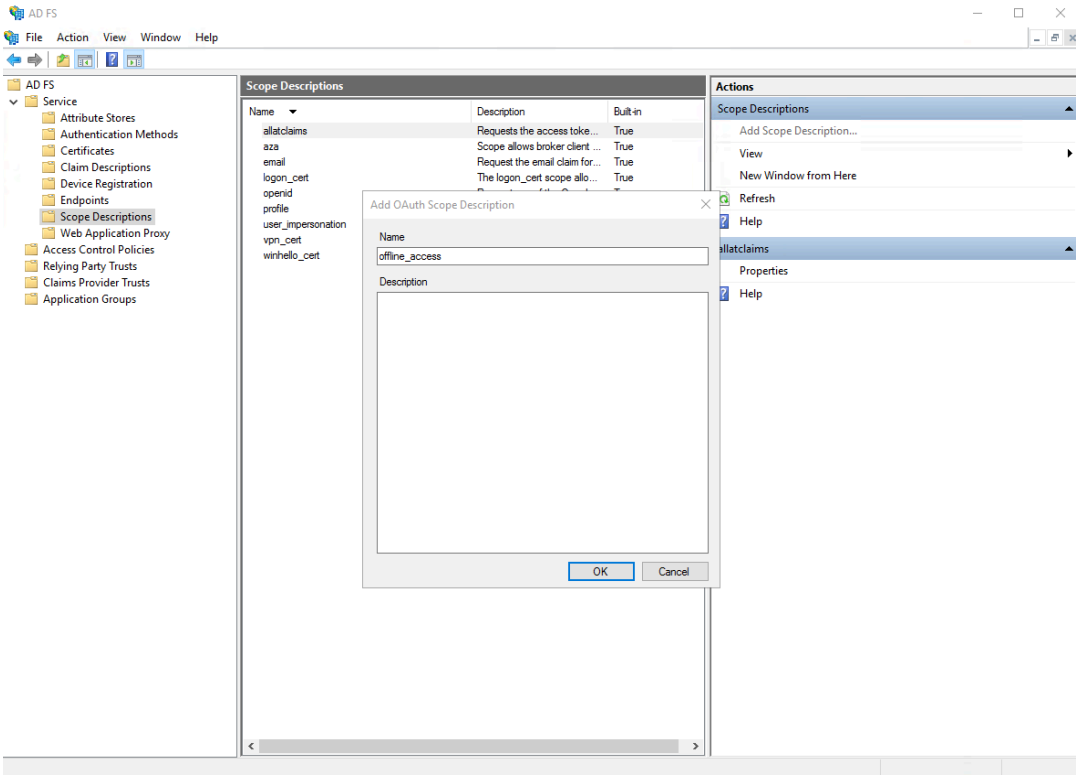
> It is also necessary for the AD user to have an **email address** set up, so that it is set as the MetaDefender Storage Security's user email

To set up AD FS SSO for MDSS, follow these steps:

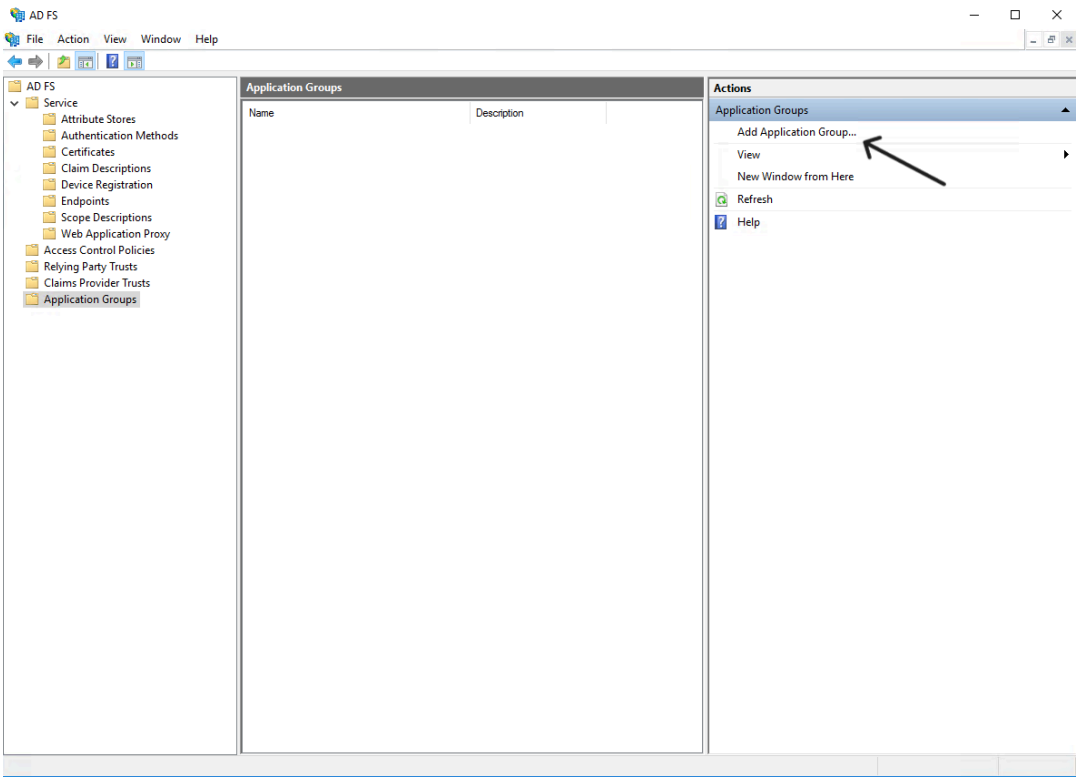
1. Within the AD FS settings, navigate to the "**Scope Descriptions**" directory and click "**Add Scope Description**"



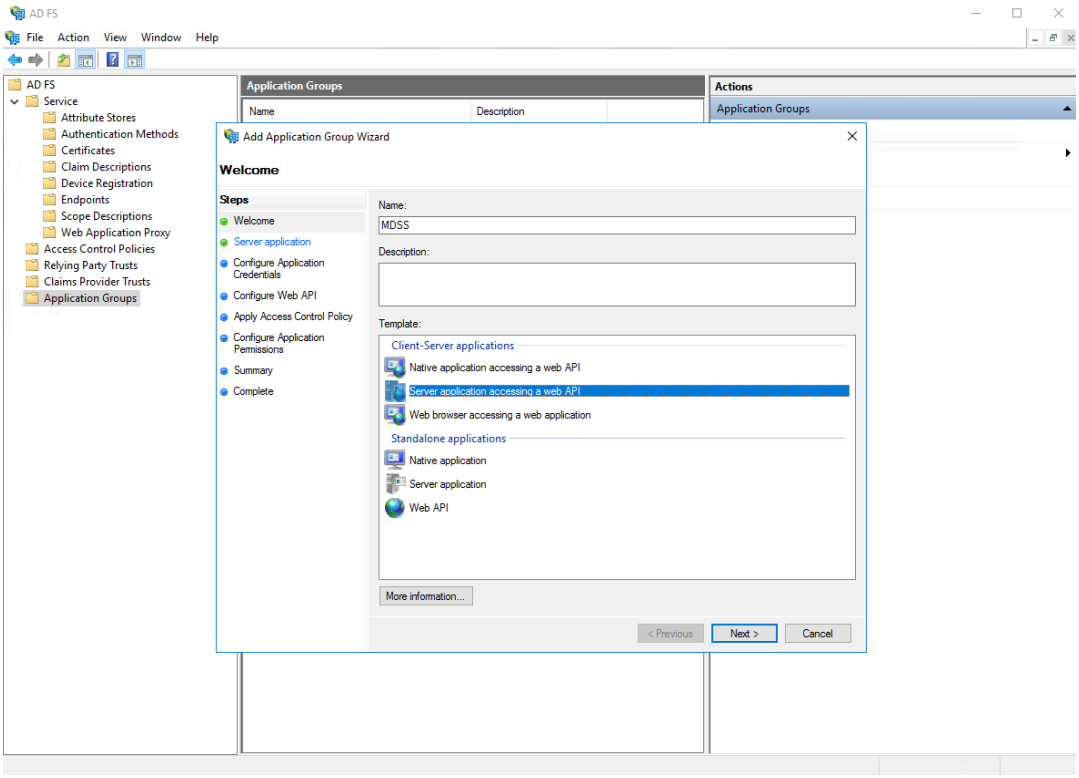
2. Create a scope with the name "**offline_access**"



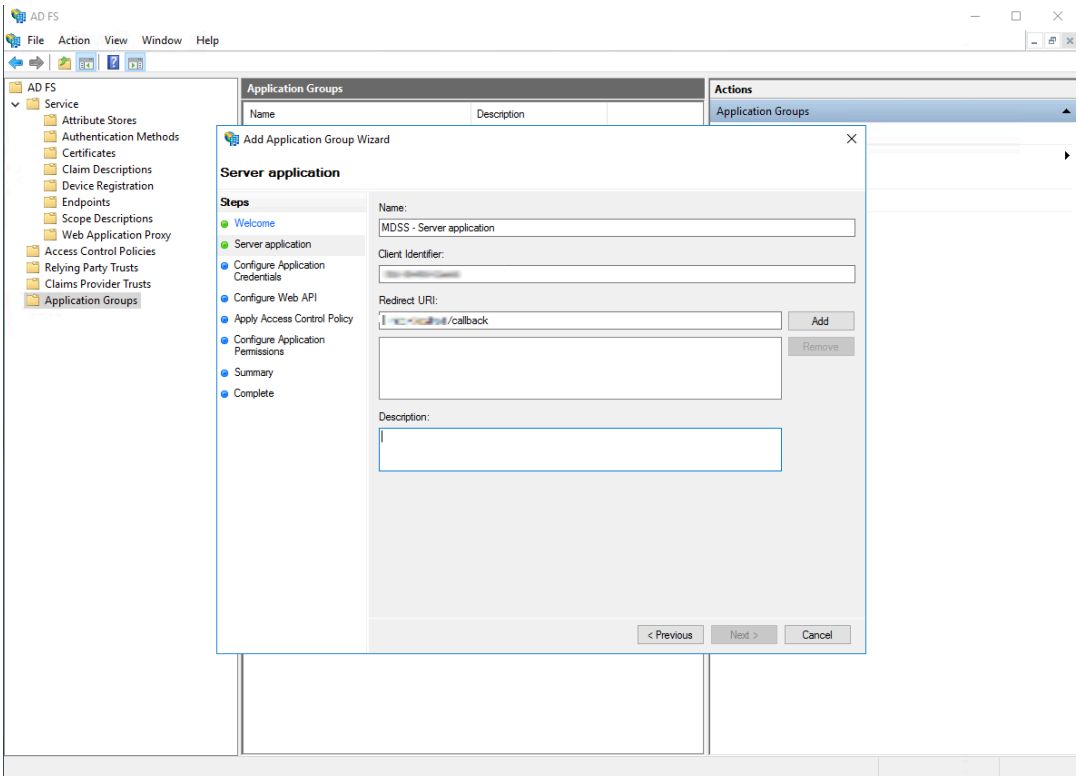
3. Navigate to the "**Application groups**" directory and click "**Add Application Group**"



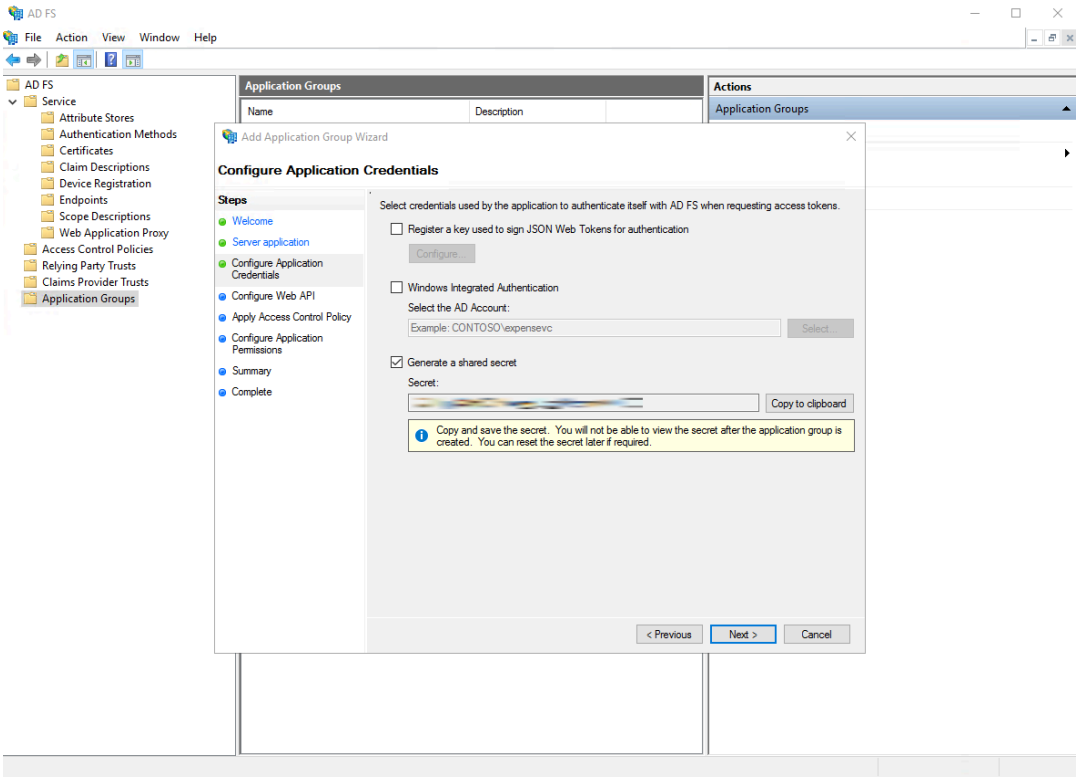
4. Give your application any name you like and for the template, select "**Server application accessing a web API**"



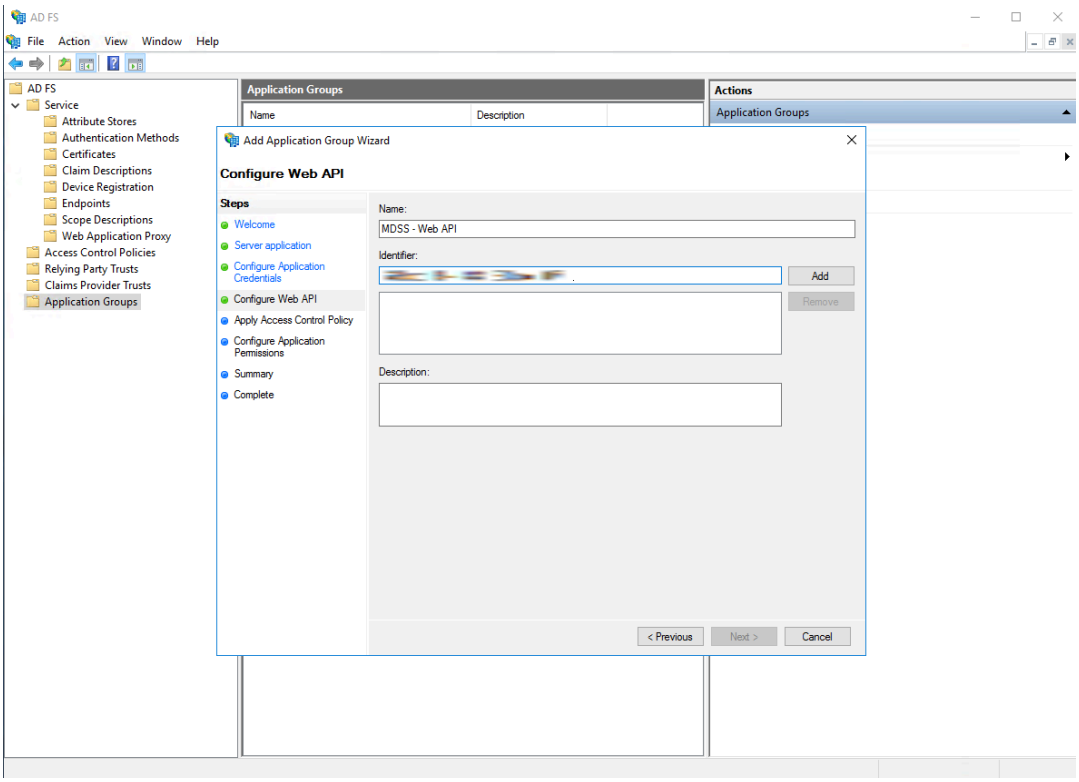
5. Give the server app a name, save the generated **Client Identifier** somewhere, and as the "**Redirect URI**", write the the base URL of the MDSS deployment, followed by the "**/callback**" suffix.



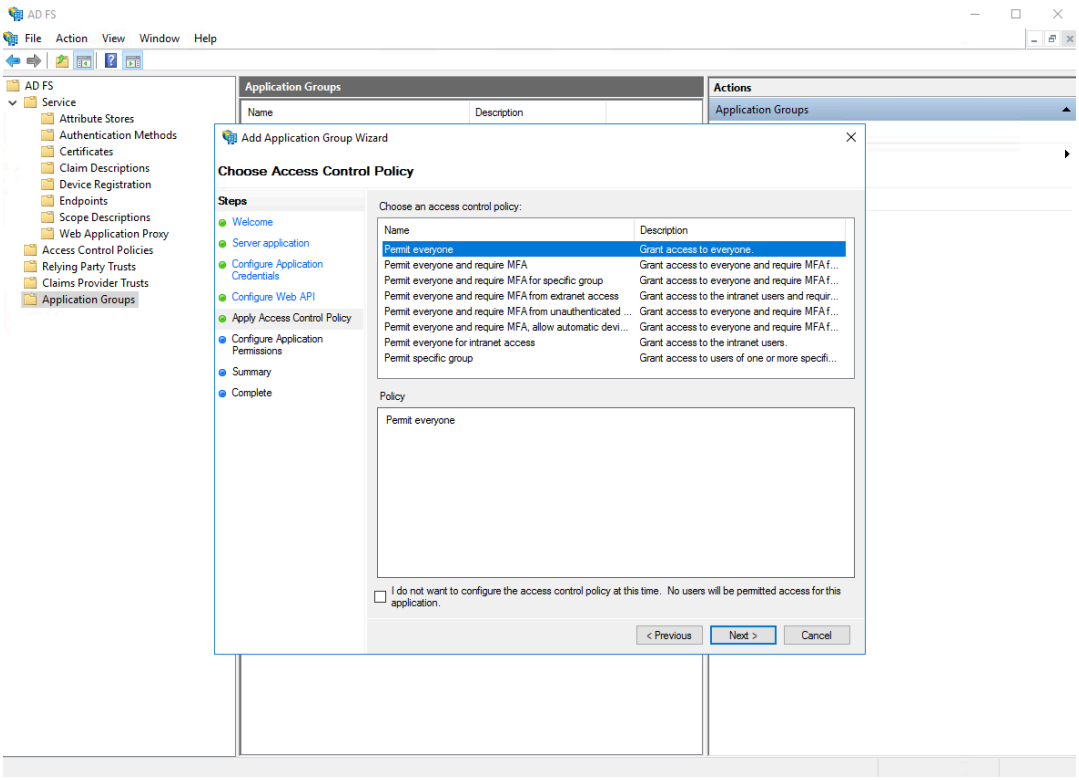
6. Select "**Generate a shared secret**" and store the secret somewhere



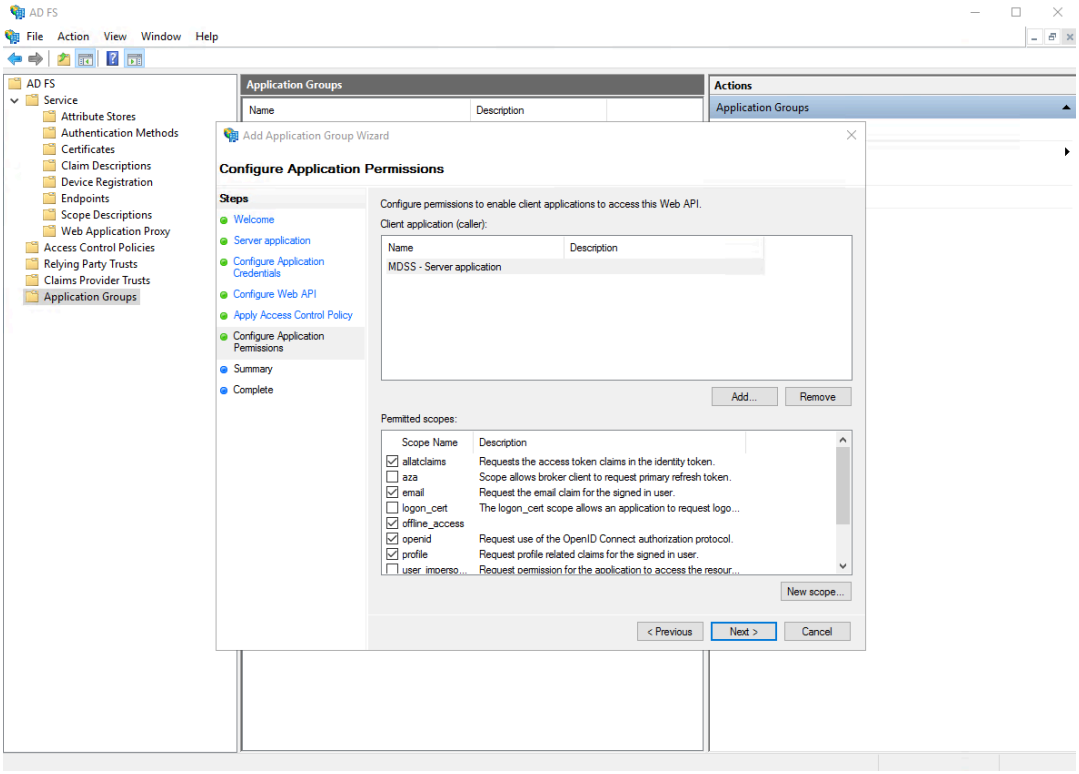
7. For the "**Identifier**" field, put the previously generated **Client Identifier** that you saved



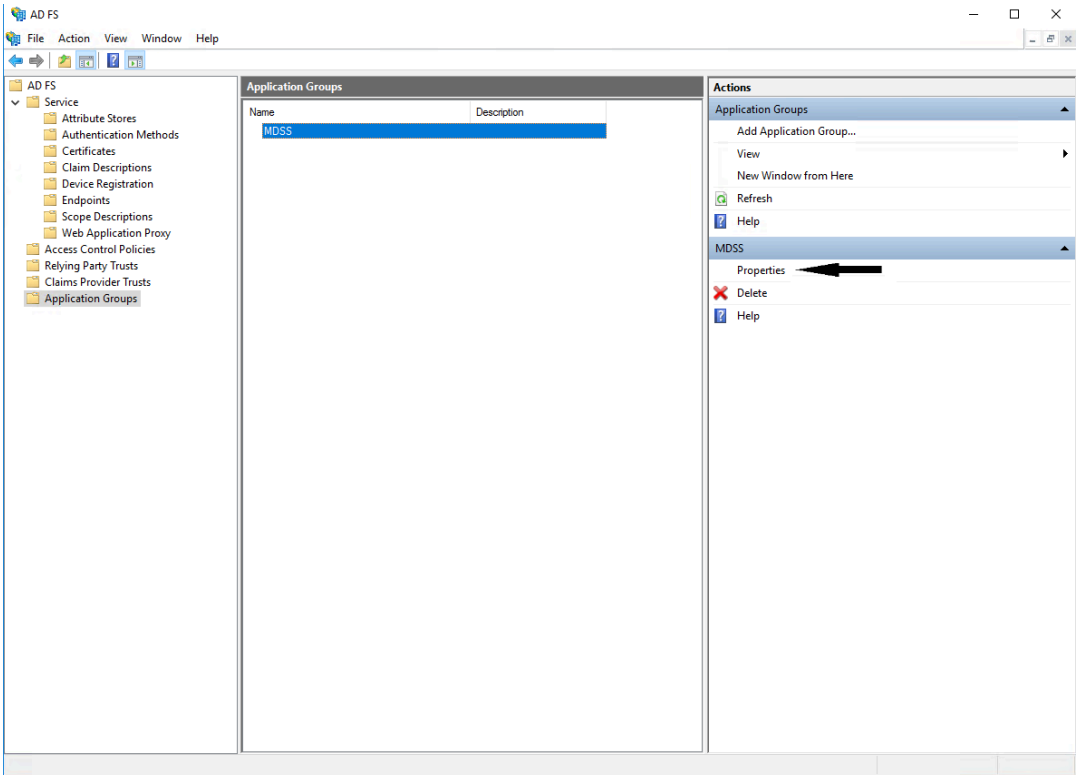
8. For the access **control policy**, select "**Permit everyone**"



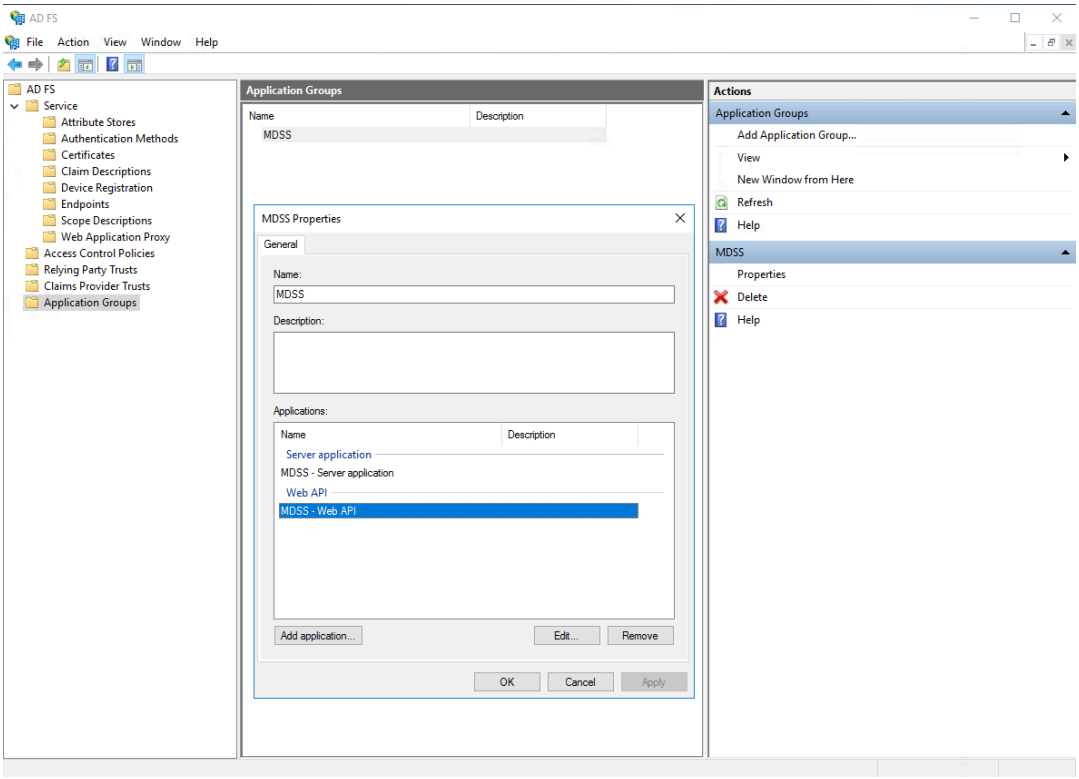
9. For the **permitted scopes**, select the following from the list: **allatclaims**, **email**, **offline_access**, **openid**, **profile**



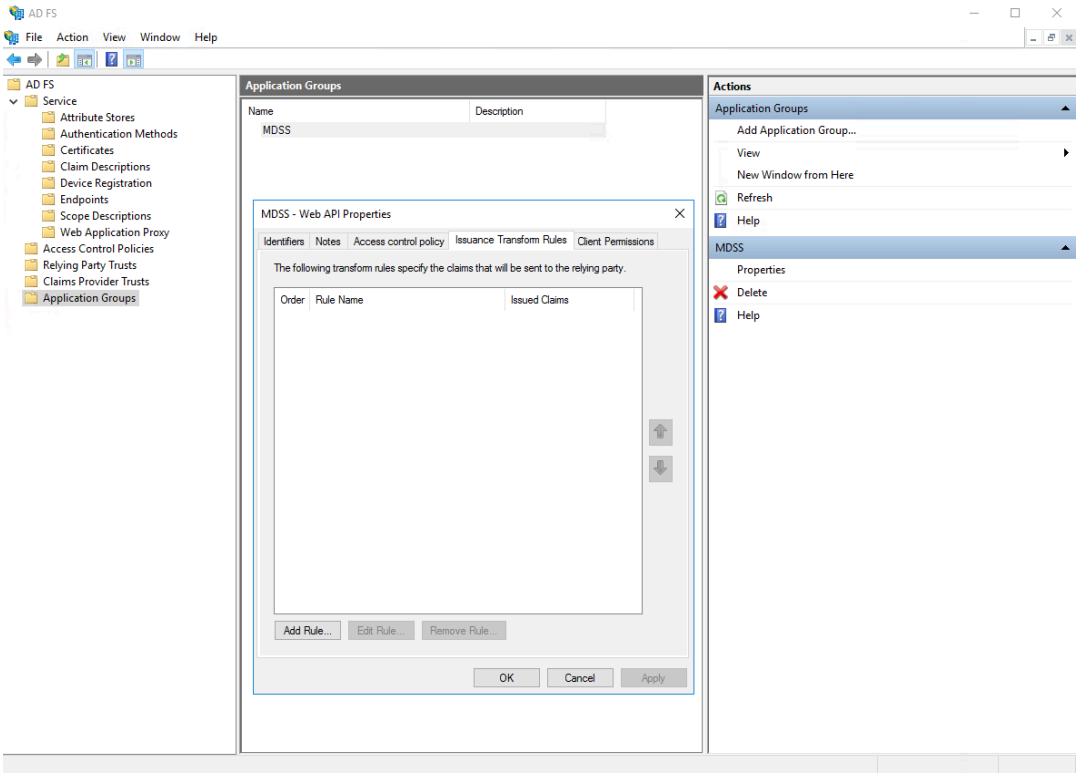
10. After saving the created application group, select it and click on **Properties**



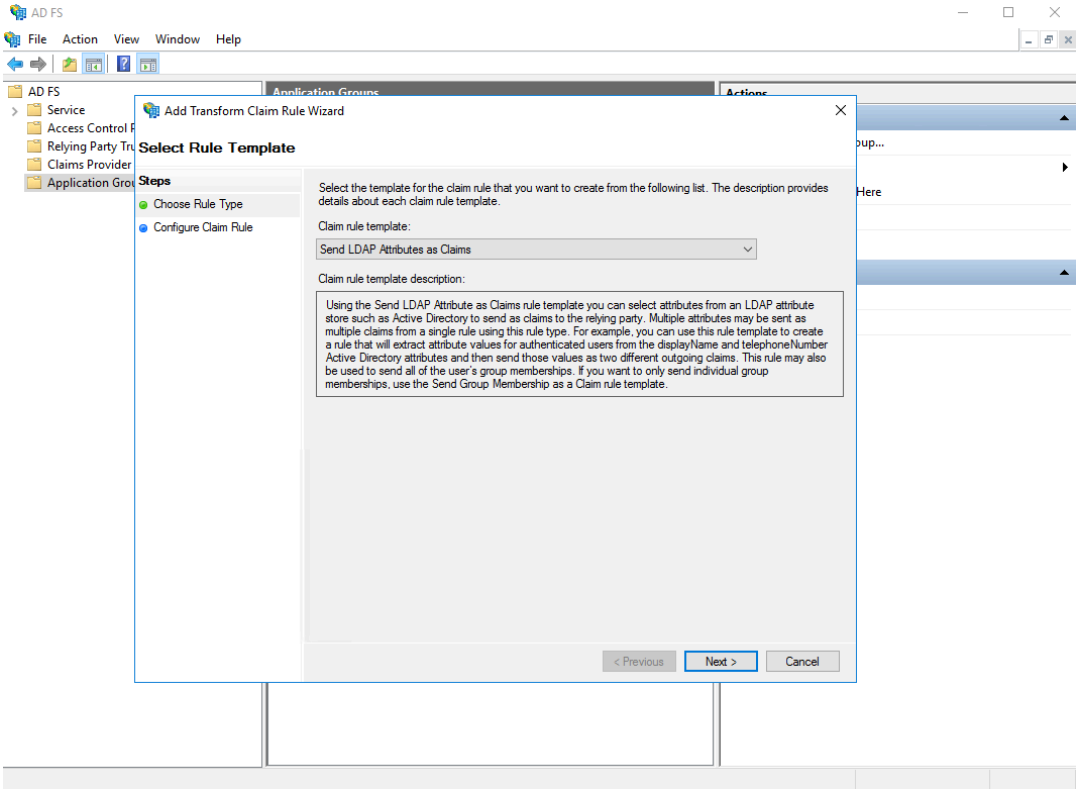
11. Select the configured **Web API** and click on **Edit**



12. Go to "**Issuance Transform Rules**" and click "**Add Rule**"

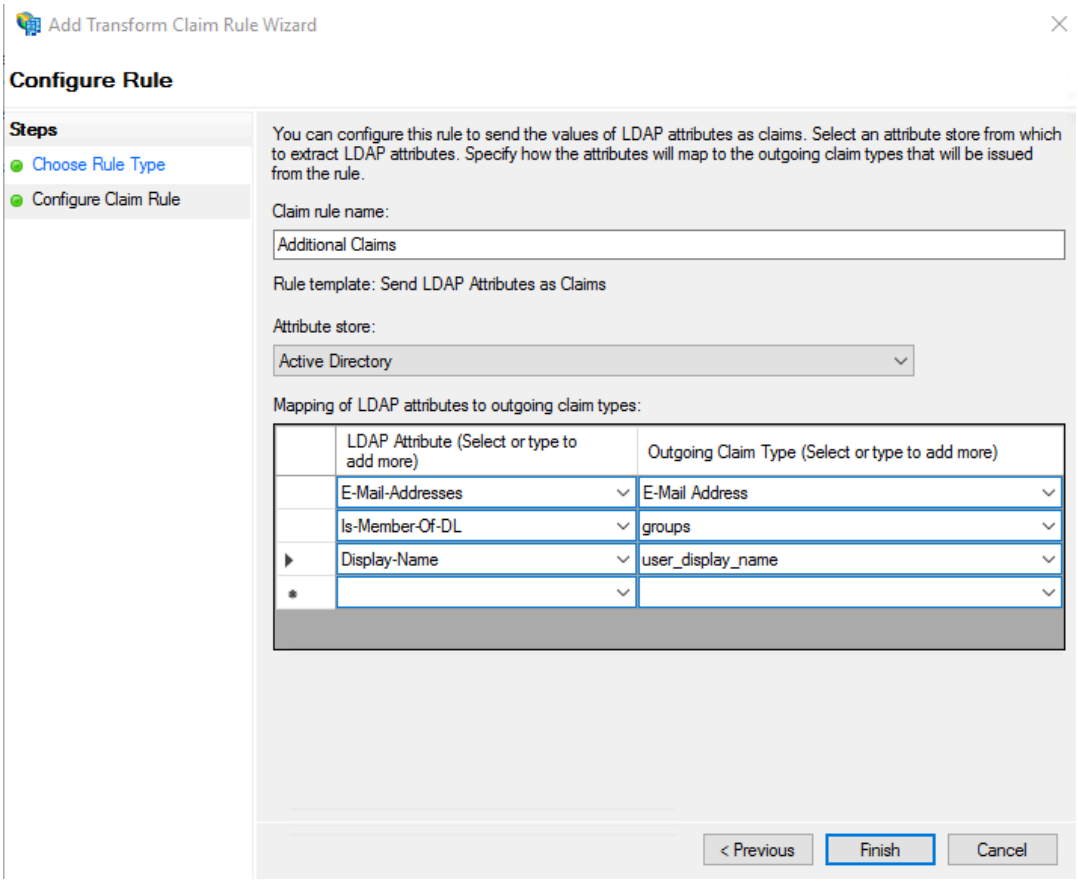


13. Select "**Send LDAP Attributes as Claims**" as the **Claim Rule Template**



14. Give the rule a name, select Active Directory as the attribute store and configure the following:

- **E-Mail Address** as **E-mail Address**
- **Is-Member-Of-DL** as **" groups*****"***
- **Display-Name** as ***"user_display_name"***



Okta Specific Configuration

Info

To properly access MDSS, the following Okta **Groups** are required

Solution:

Create a Perquisite section on providing specific instructions on how to add the Specific Group and and assign users to them.

For Example:

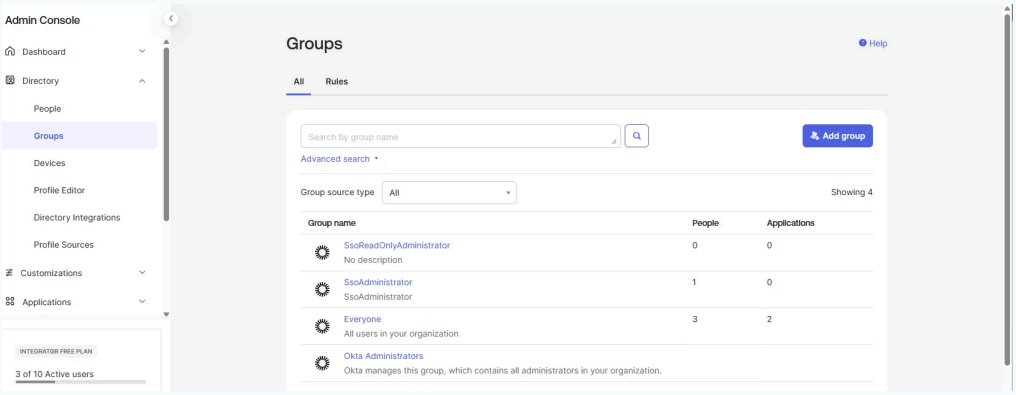
Prerequisite:

The following specific groups needs to be added into OKTA:

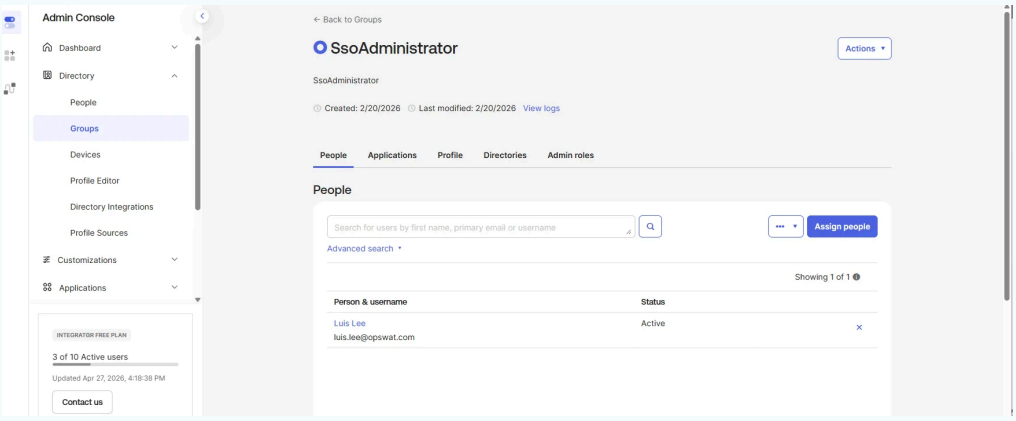
SsoAdministrator - the equivalent of local MetaDefender Storage Security administrator role.

SsoReadOnlyAdministrator - the equivalent of local MetaDefender Storage Security read-only role.

To add the two required groups in Okta, navigate to **Directory > Groups**, then select **Add Group**.



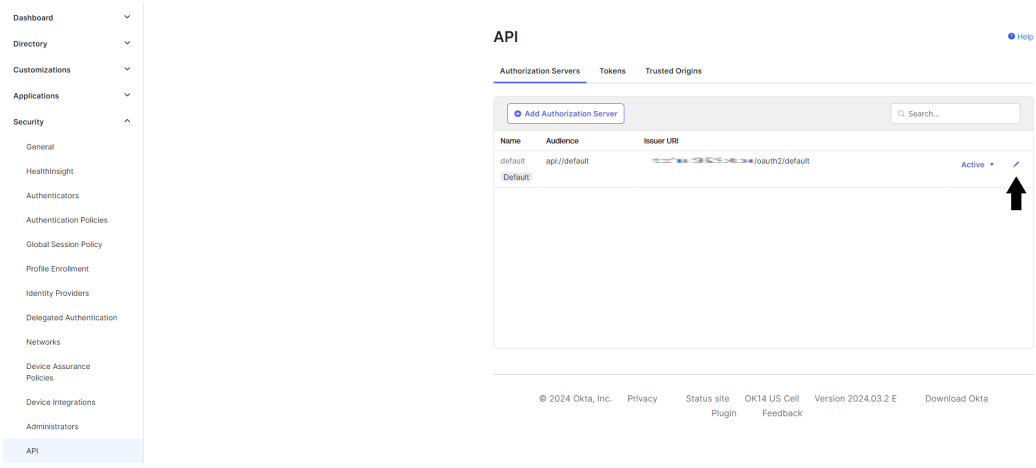
Within the specific group, ensure that users are assigned by selecting the **Assign People** option.



This will help ensure that Okta administrators remember to create the necessary groups and properly assign users to them.

To set up Okta SSO for MDSS, follow these steps:

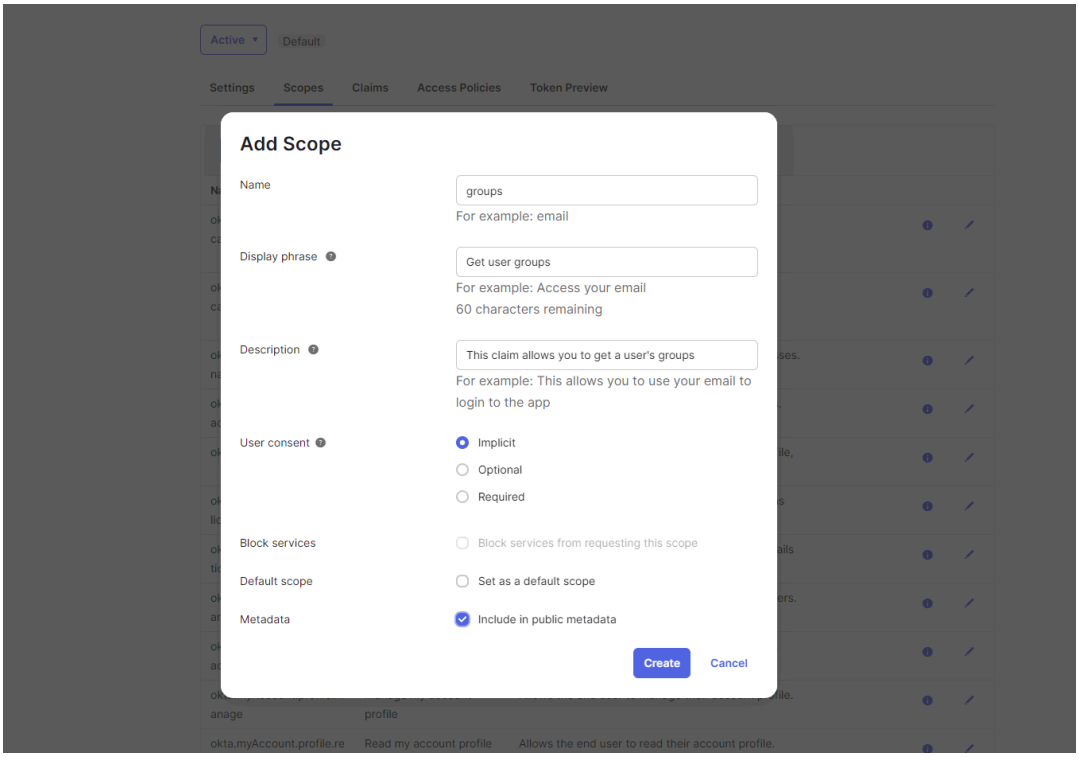
1. Under the **Security** tab, go to the **API** section, locate your **Authorization Server** and click on the pencil icon to edit



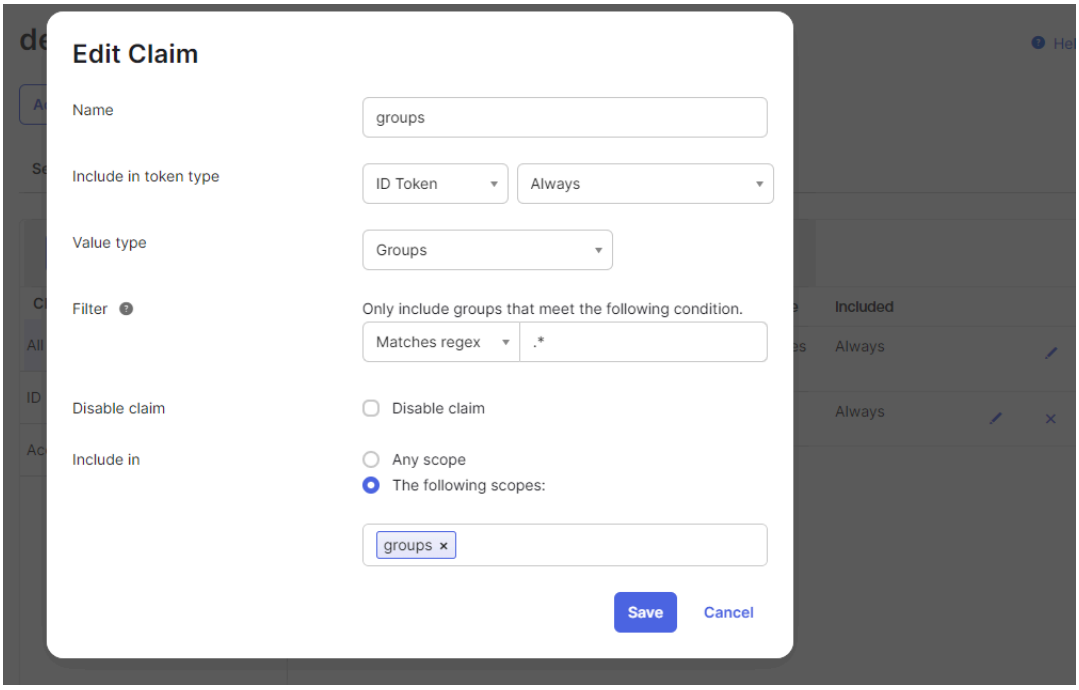
Info

The **Issuer URI** from the **Authorization Server** should be inputted as the **Authority URL** of the MDSS SSO Configuration.

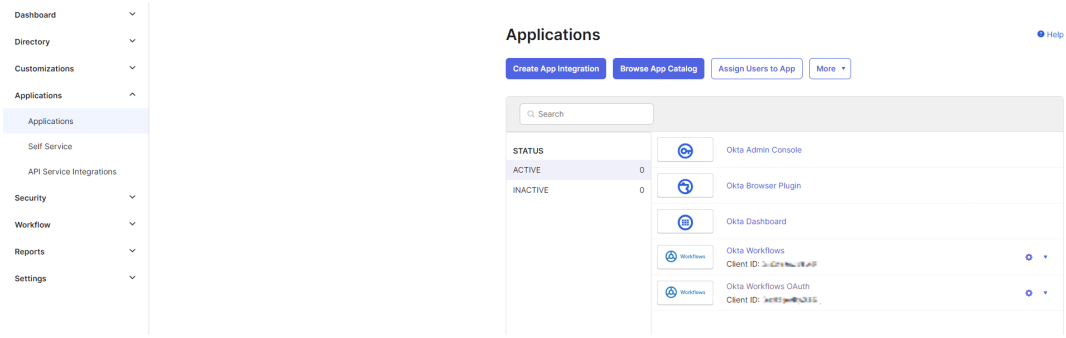
2. Within the **Scopes** tab, add a new scope. Name it "***groups***", select **Implicit** for the **User Consent**, and check the **Include** in public metadata checkbox



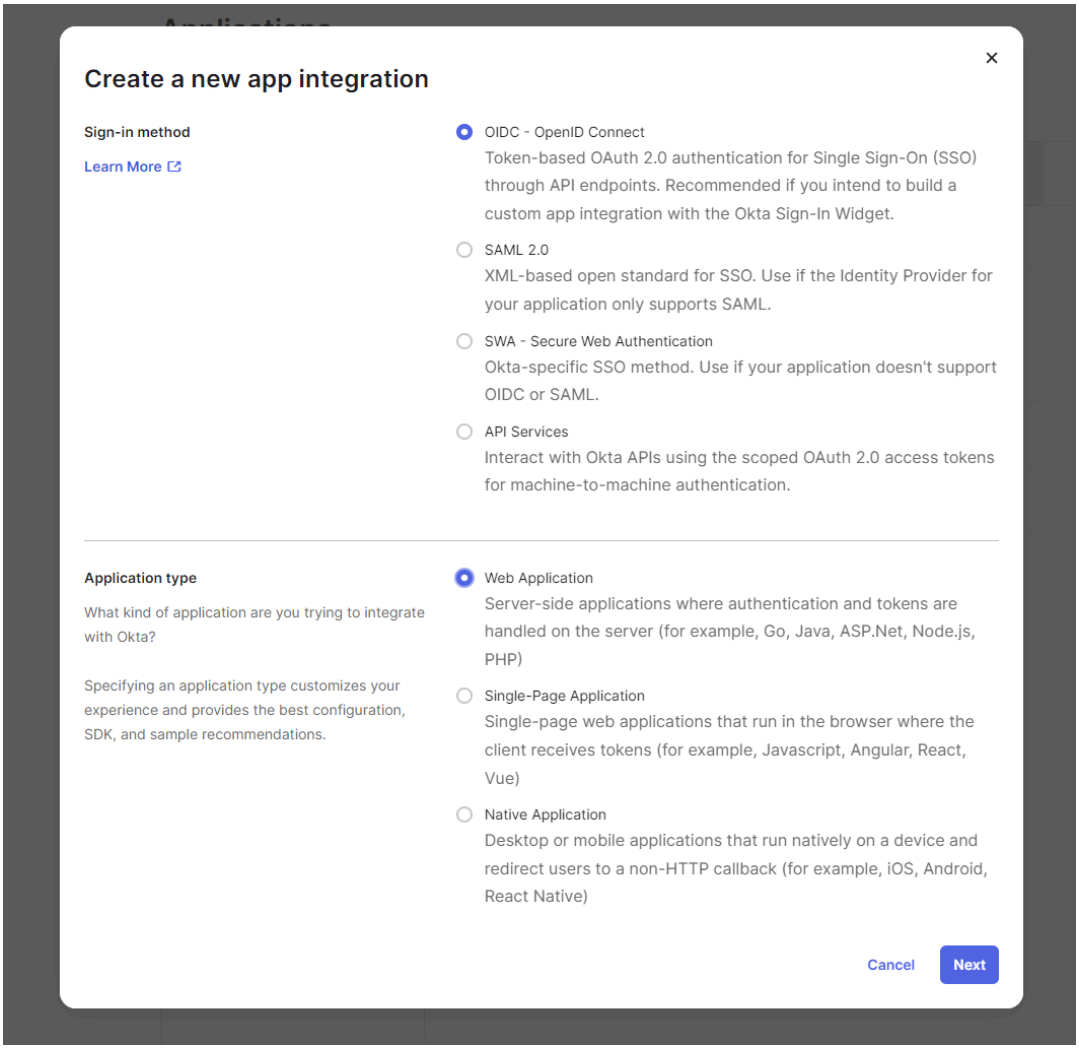
3. Within the **Claims** tab, add a new claim. Name it "***groups***", select **Id Token** and **Always** from the **Include in token type** dropdowns, select **Groups** as the **Value Type**, input "**.***" next to **Matches regex** to get all groups, and include it in the previously created "***groups***" scope



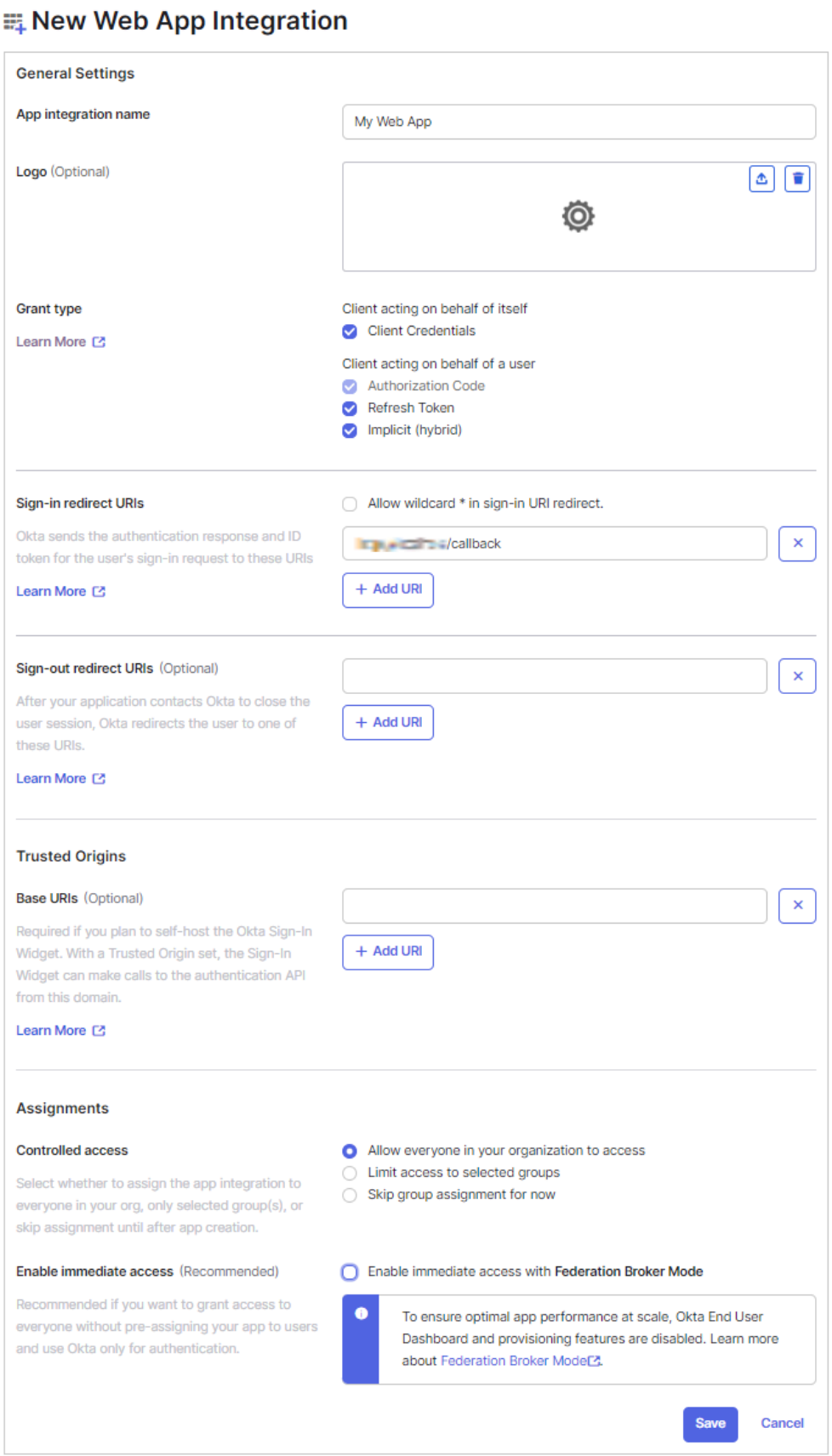
4. Navigate to the **Applications** tab and select the **Applications** section. Afterwards, click "**Create App Integration**"



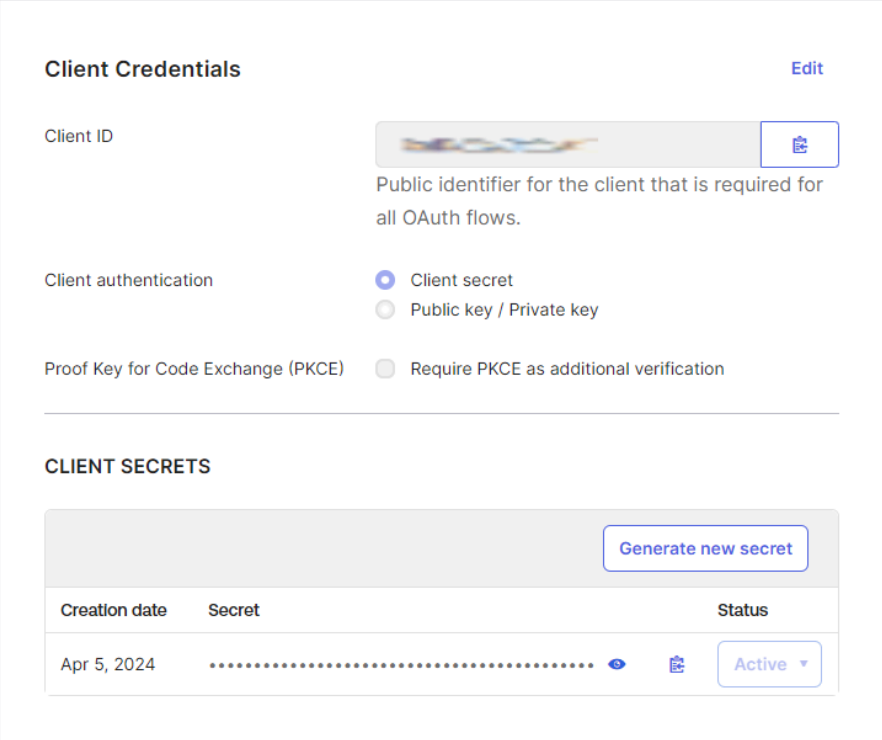
5. For the **Sign-in method**, select **OIDC - OpenID Connect**, and for the **Application Type**, select **Web Application**



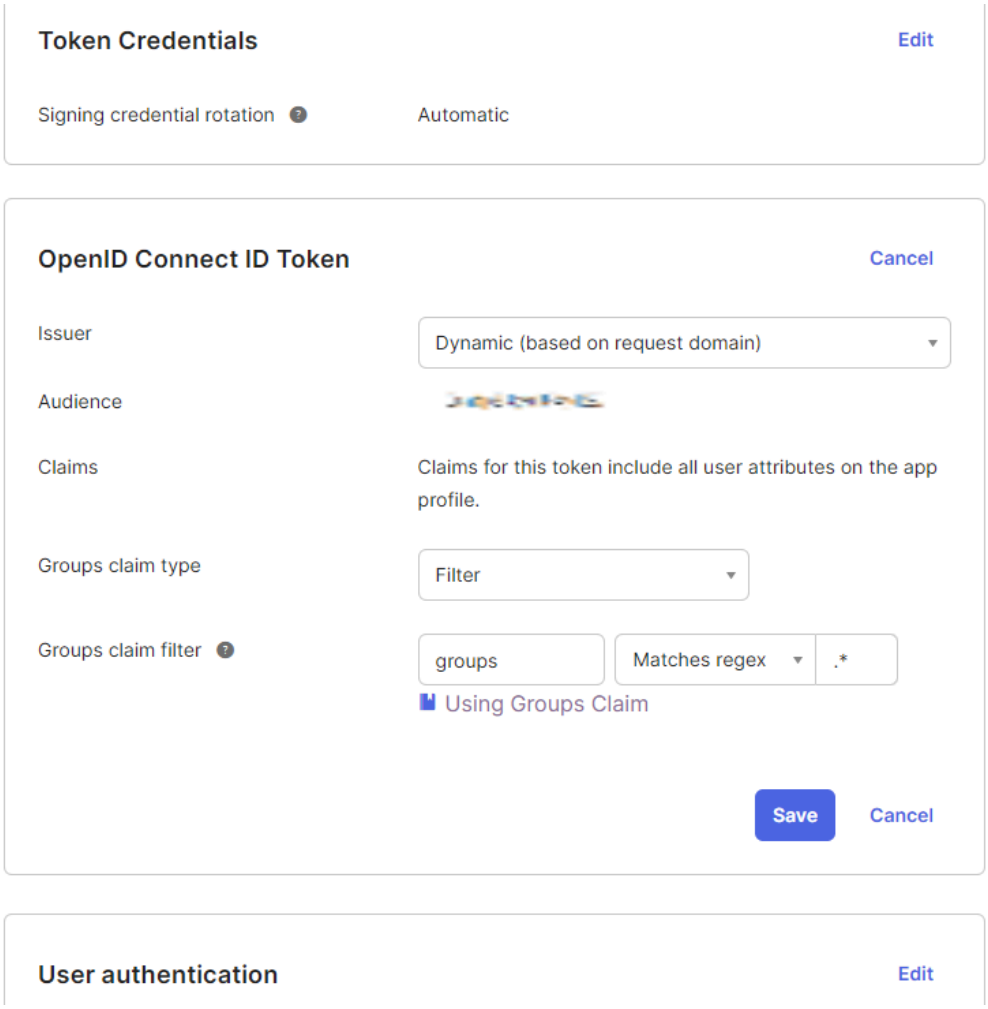
6. Give your app integration a name, for the **Grant Type,** tick the checkboxes. Under the **Sign-in redirect URIs**, write the the base URL of the MDSS deployment, followed by the "**/callback**" suffix. For the **Controlled access**, select **Allow everyone in your organization to access.**


New Web App Integration
General Settings
App integration name
My Web App
Logo (Optional)
Grant type
Learn More
Client acting on behalf of itself
Client Credentials
Client acting on behalf of a user
Authorization Code
Refresh Token
Implicit (hybrid)
Sign-in redirect URIs
Allow wildcard * in sign-in URI redirect.
Okta sends the authentication response and ID token for the user's sign-in request to these URIs
/callback
Learn More
+ Add URI
Sign-out redirect URIs (Optional)
After your application contacts Okta to close the user session, Okta redirects the user to one of these URIs.
+ Add URI
Learn More
Trusted Origins
Base URIs (Optional)
Required if you plan to self-host the Okta Sign-In Widget. With a Trusted Origin set, the Sign-In Widget can make calls to the authentication API from this domain.
+ Add URI
Learn More
Assignments
Controlled access
Select whether to assign the app integration to everyone in your org, only selected group(s), or skip assignment until after app creation.
Allow everyone in your organization to access
Limit access to selected groups
Skip group assignment for now
Enable immediate access (Recommended)
Recommended if you want to grant access to everyone without pre-assigning your app to users and use Okta only for authentication.
Enable immediate access with Federation Broker Mode
To ensure optimal app performance at scale, Okta End User Dashboard and provisioning features are disabled. Learn more about Federation Broker Mode.
Save
Cancel

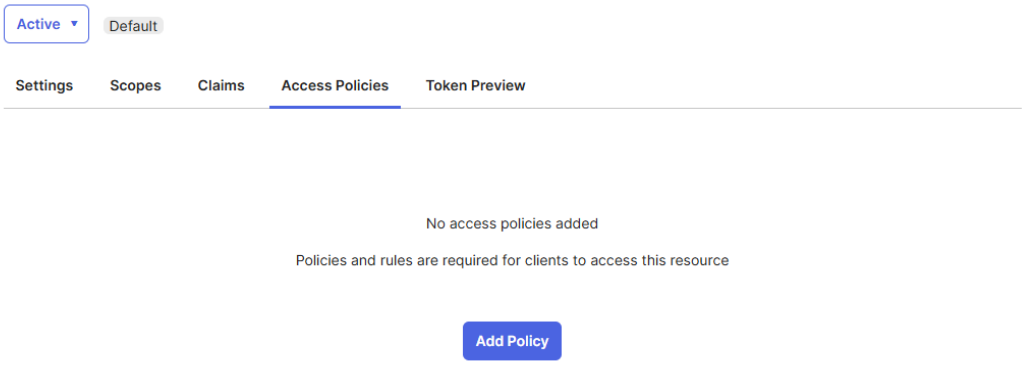
7. Save the **Client ID**, select **Client Secret** for the Client authentication, generate a new secret and save it.



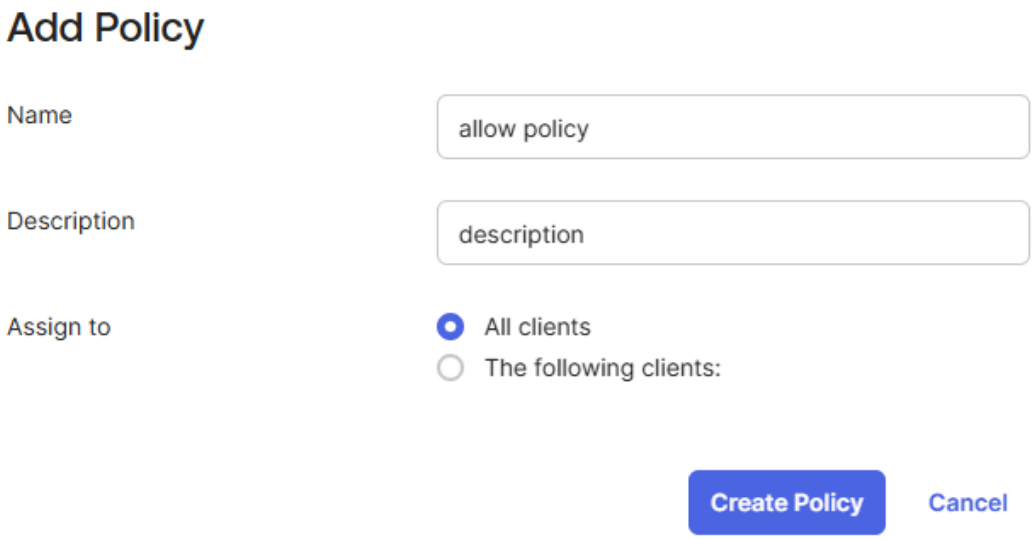
8. Within your **Application**, navigate to the **Sign On** tab, and scroll down to **OpenID Connect ID Token**. There, click the edit button and, for the **Groups Claim Type**, select **Filter**, and for the **Groups Claim Filter**, select **groups**. Select **Matches regex** from the dropdown and input "**.***"



9. Under the **Security** tab, go to the **API** section, locate **Access Policies** and add a new one.



10. Add policy and assign to **All clients** or **your desired clients.**



11. Create a rule, and make sure that on **Advanced** label you have checked **Implicit(hybrid).**

Add Rule

Rule Name

allow rule

IF Grant type is

Client acting on behalf of itself

- [x] Client Credentials

Core grants

- [x] Authorization Code
- [x] Device Authorization

Advanced

Non-interactive grants

- [ ] SAML 2.0 Assertion
- [ ] Token Exchange

Okta direct auth API grants

- [ ] OTP
- [ ] OOB
- [ ] MFA OTP
- [ ] MFA OOB

Other grants

- [x] Implicit (hybrid)
- [ ] Resource Owner Password

AND User is

- (•) Any user assigned the app
- ( ) Assigned the app and a member of one of the following:

AND Scopes requested

- (•) Any scopes
- ( ) The following scopes:

THEN Use this inline hook

None (disabled)

AND Access token lifetime is

1 Hours

AND Refresh token lifetime is

90 Days

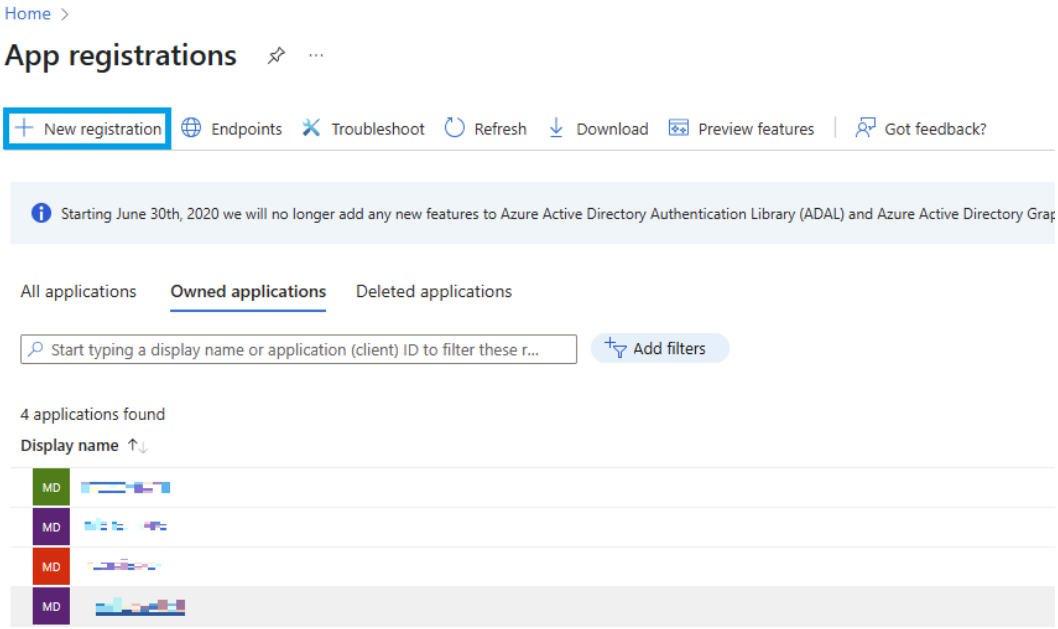
but will expire if not used every 7 Days

Create rule Cancel

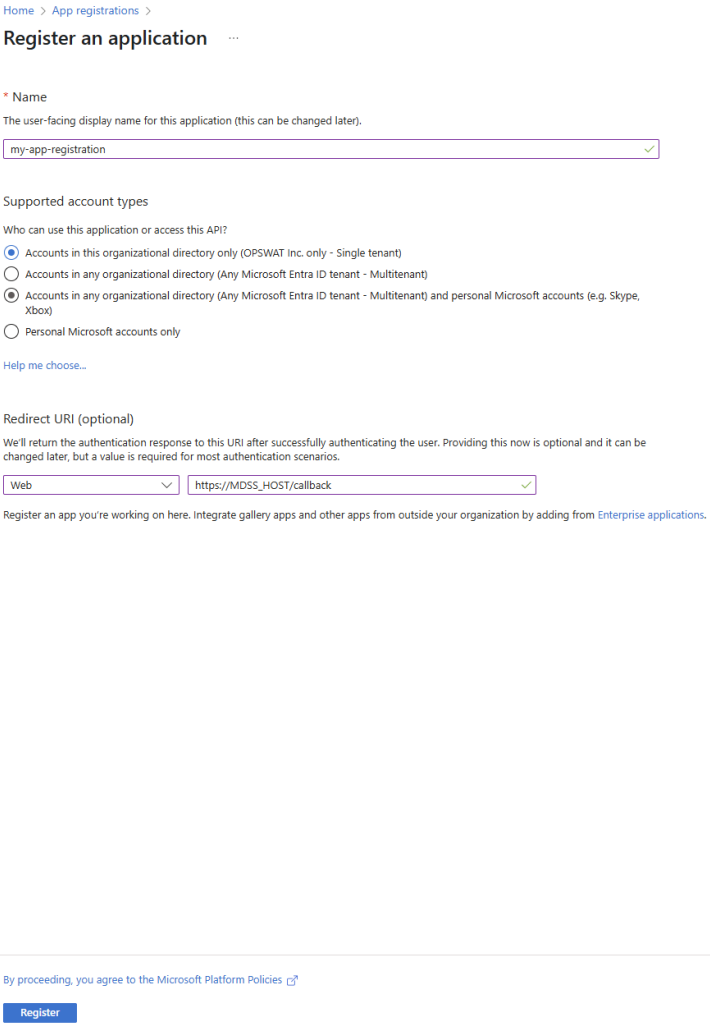
Azure AD Specific Configuration

To set up Azure AD SSO for MDSS, follow these steps:

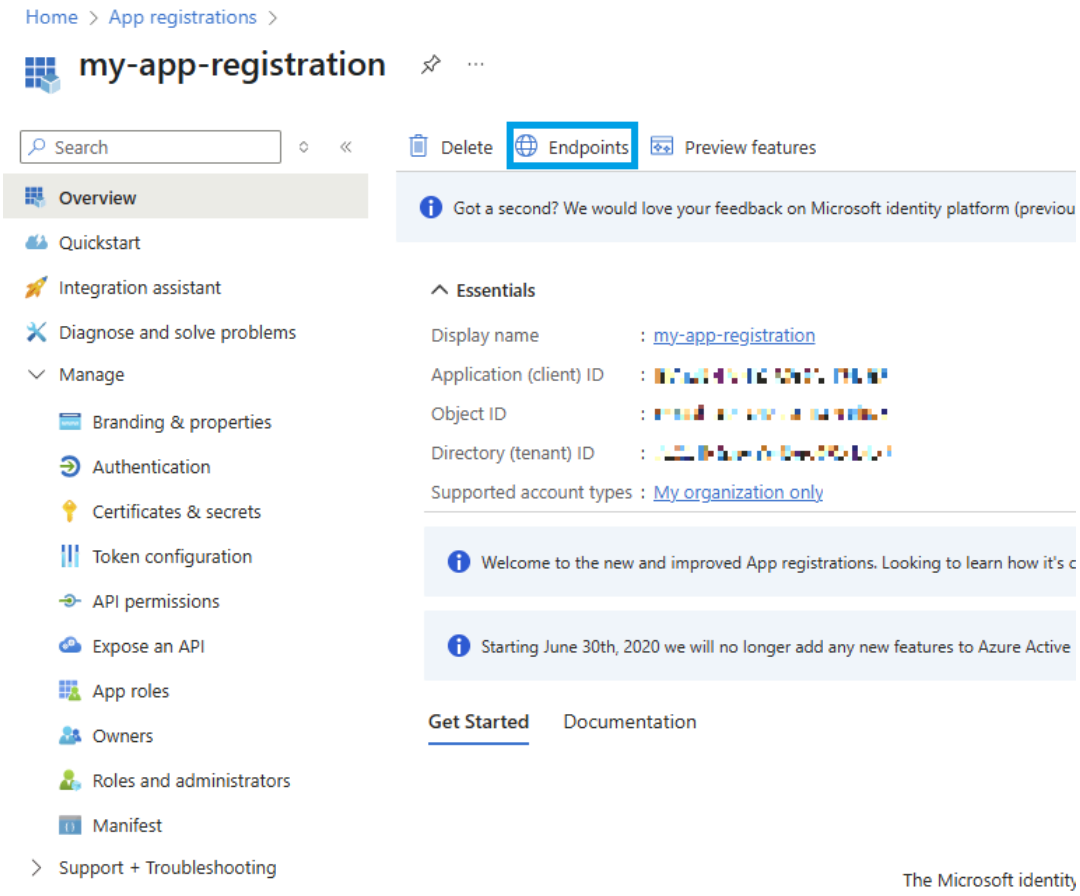
1. In the Microsoft Azure portal, navigate to **App registrations** and click on **New registration**



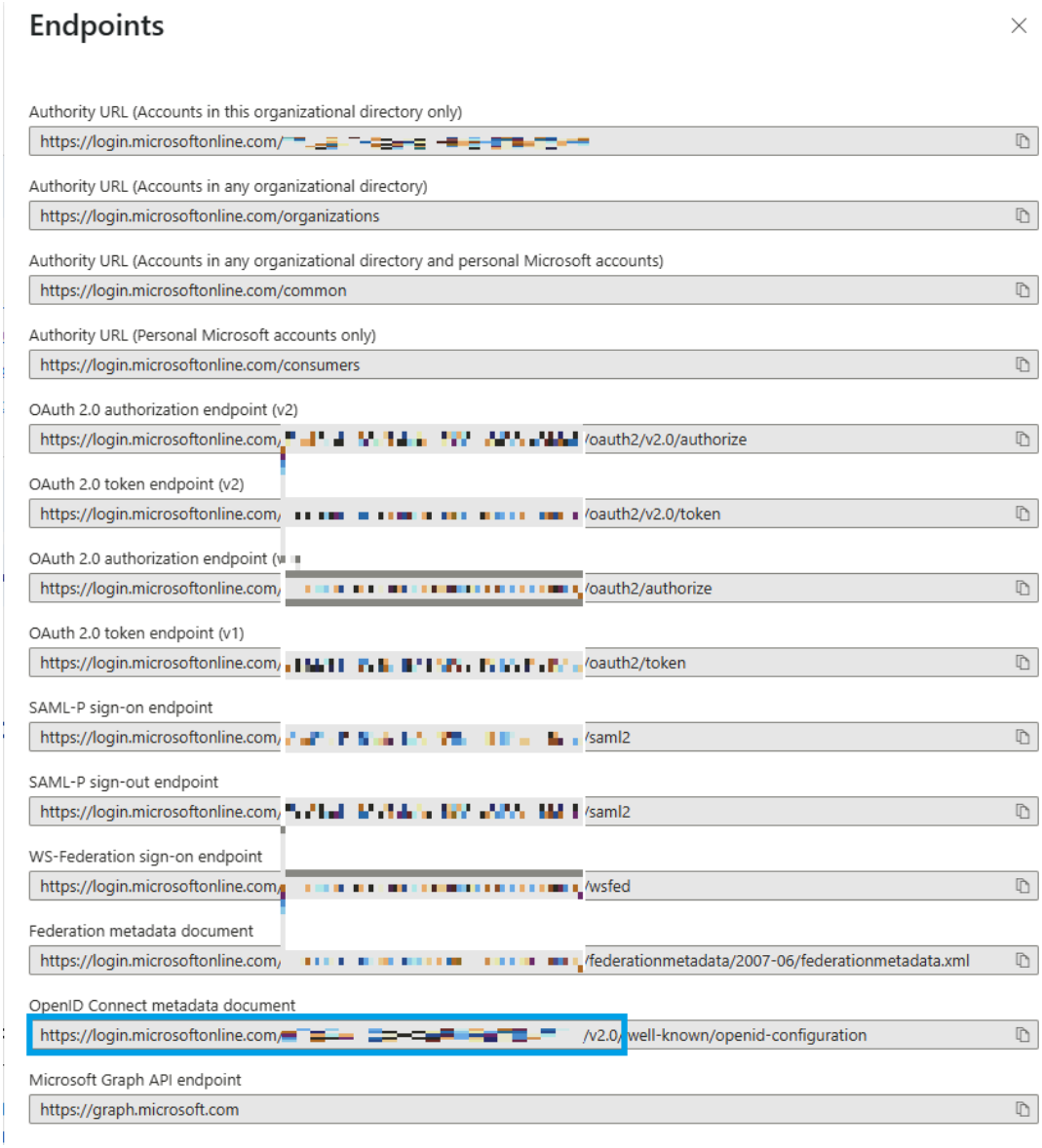
2. In the **Register an application** window, enter a name for your application, select your preferred configuration for **Supported account types** (recommended is Single tenant), and configure the **Redirect URI**. The redirect URI should be composed of the address where MDSS is deployed, along with the **/callback** suffix added to the path.



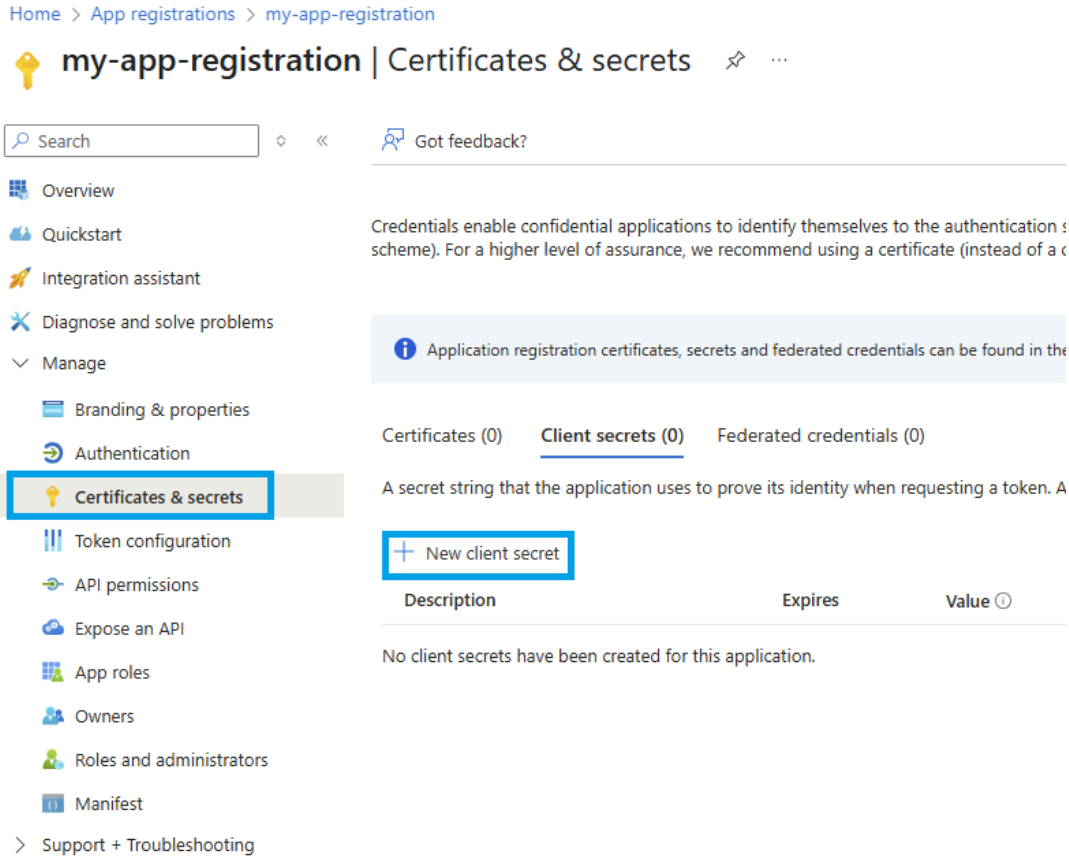
3. Afterwards, navigate to the **Overview** tab. There, to successfully configure the SSO Configuration in MDSS afterwards, store the **Application (Client) ID** and the **Direcory (Tenant) ID** somewhere safe on your computer. Then, click on the **Endpoints** button.



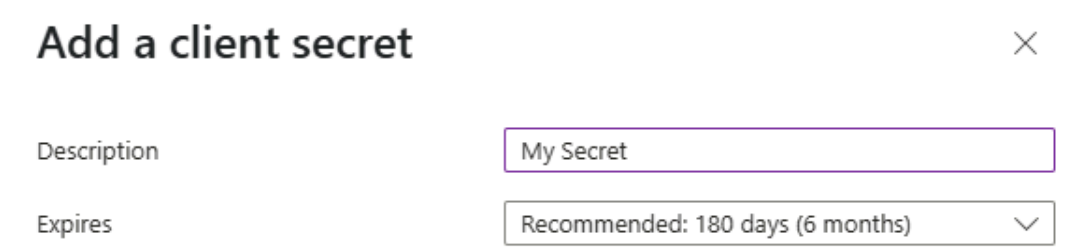
4. There, the authority URL is located in the right-hand side menu and is the following:
   "https: //login.microsoftonline.com//v2.0"



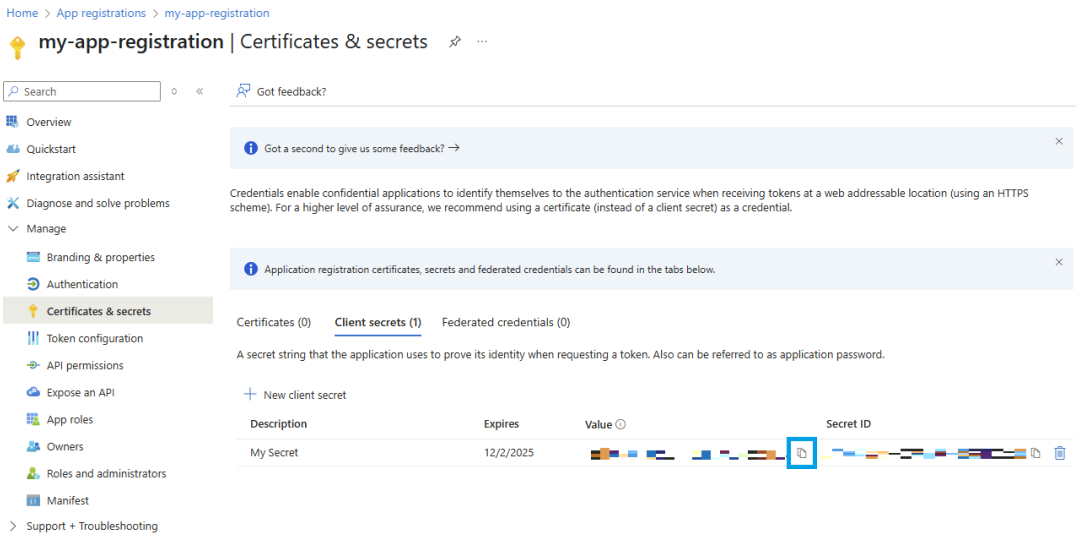
5. Navigate to the **Certificates & secrets** tab and click on **New client secret**



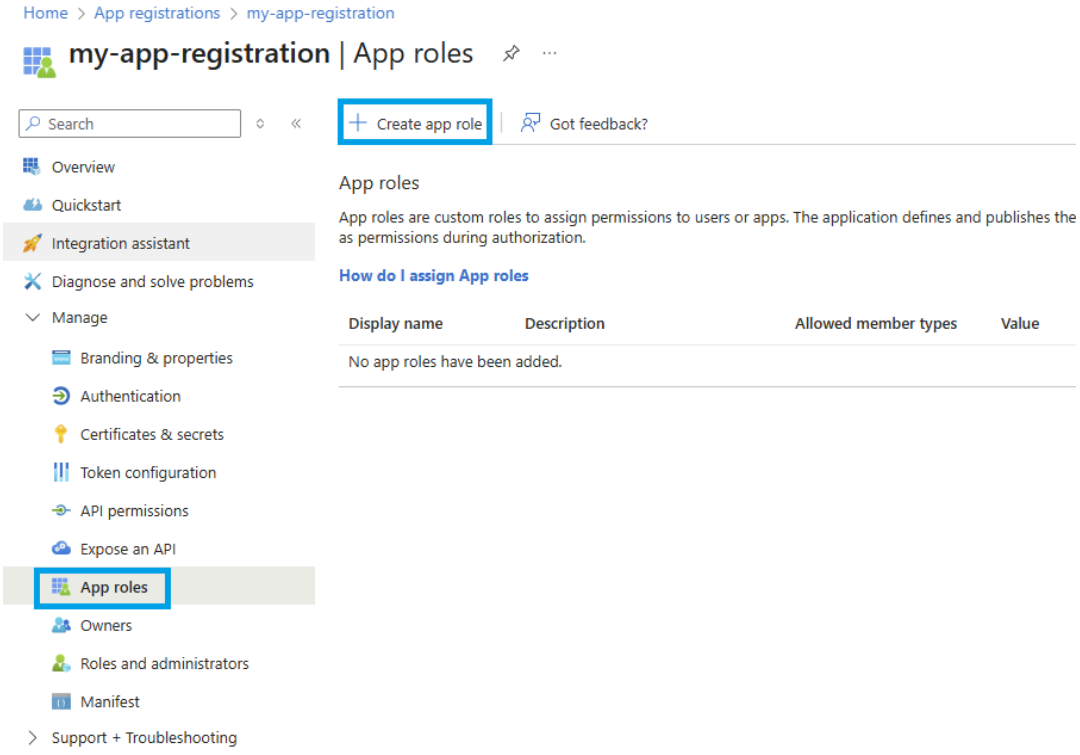
6. Give a description to your client secret, configure the expiration to your liking, and add it



7. Copy your secret value and store it somewhere safe on your computer (you will only be able to copy it at this stage)

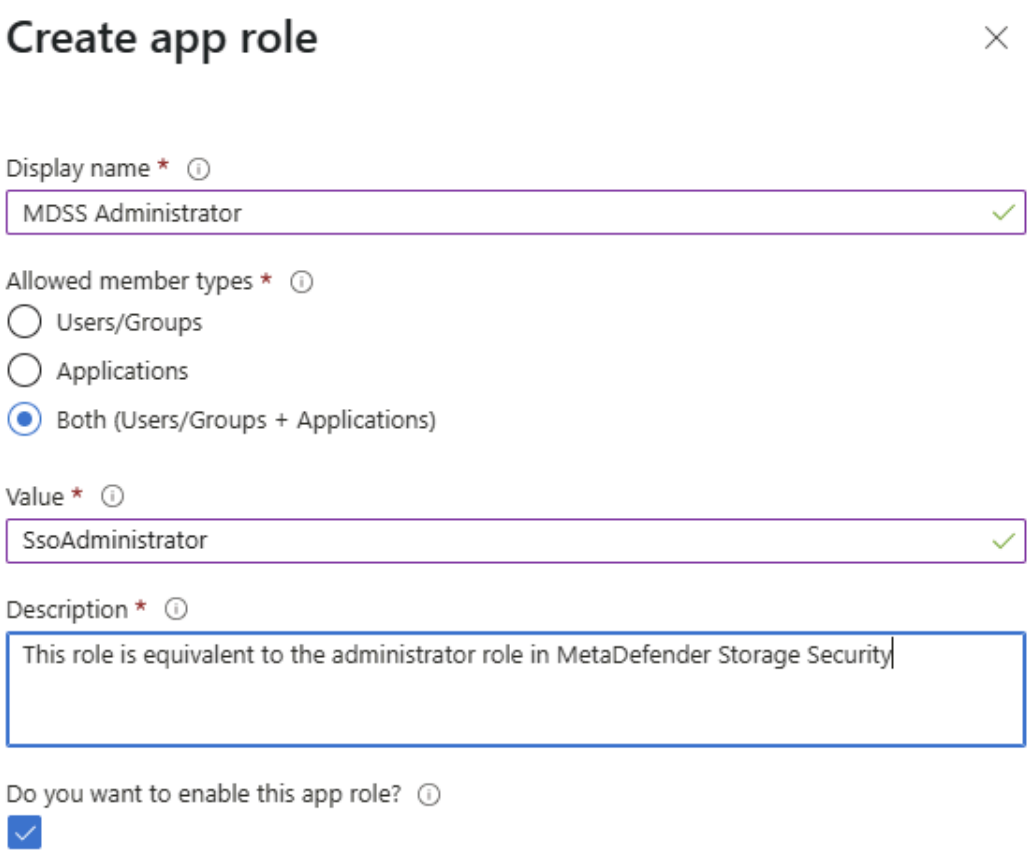


8. Navigate to **App Roles** and click on **Create app role**

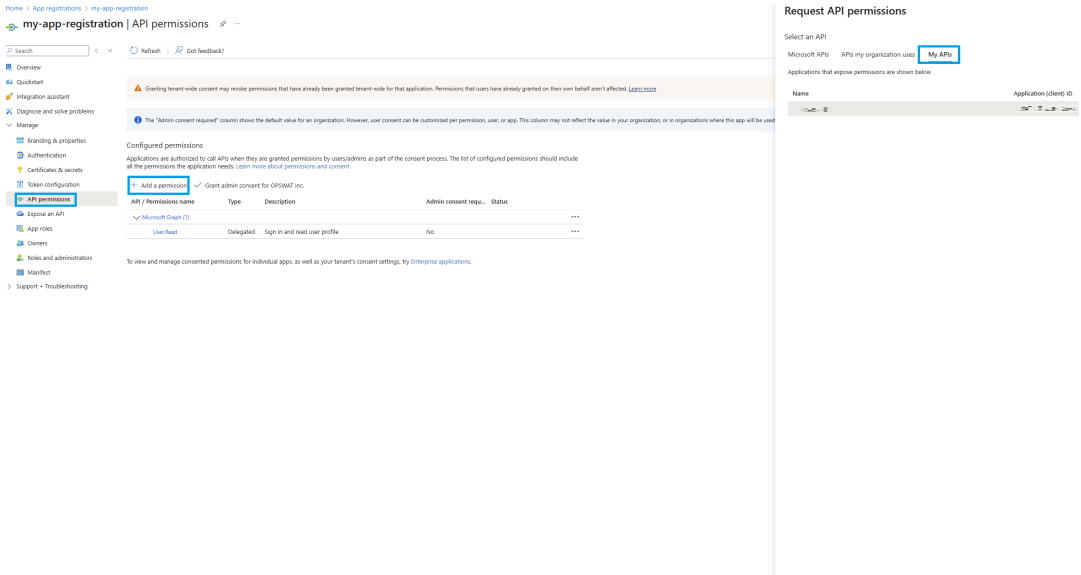


9. Give your role a Display name, configure the **Allowed member types** to be set to **Both (Users/Groups + Applications)**. Then, for the value, give one of the two values to determine the role that this user will have within MDSS:

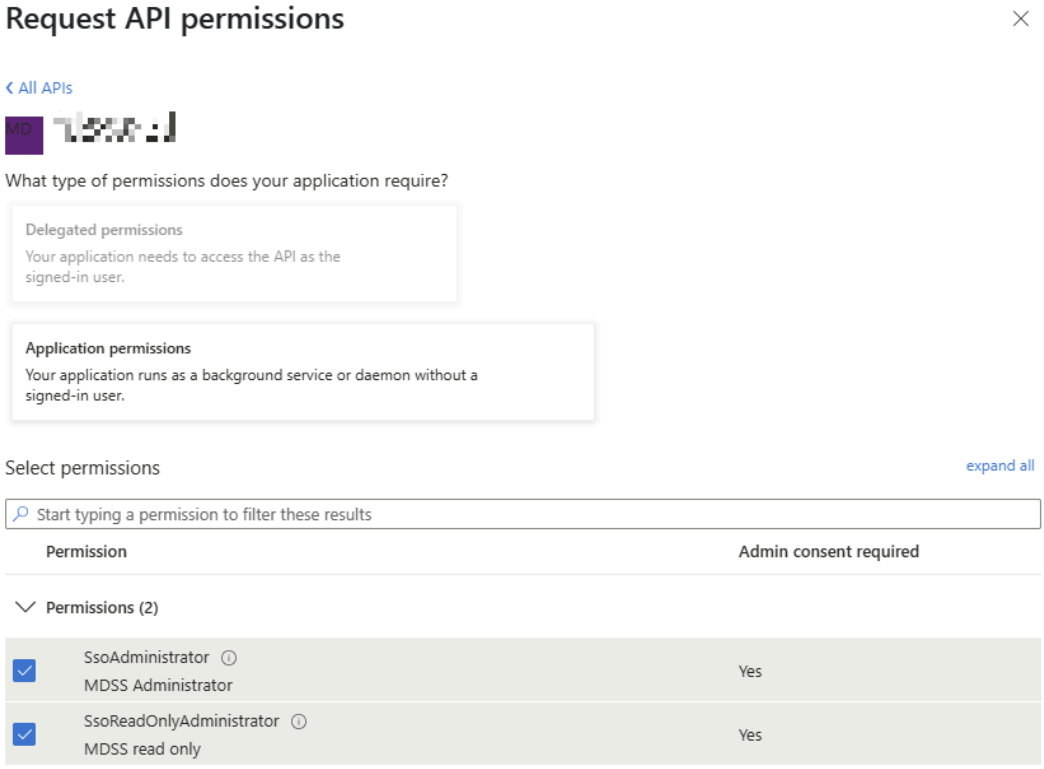
- **SsoAdministrator** - equivalent to the administrator role in MDSS
- **SsoReadOnlyAdministrator** - equivalent to the read-only admin role in MDSS



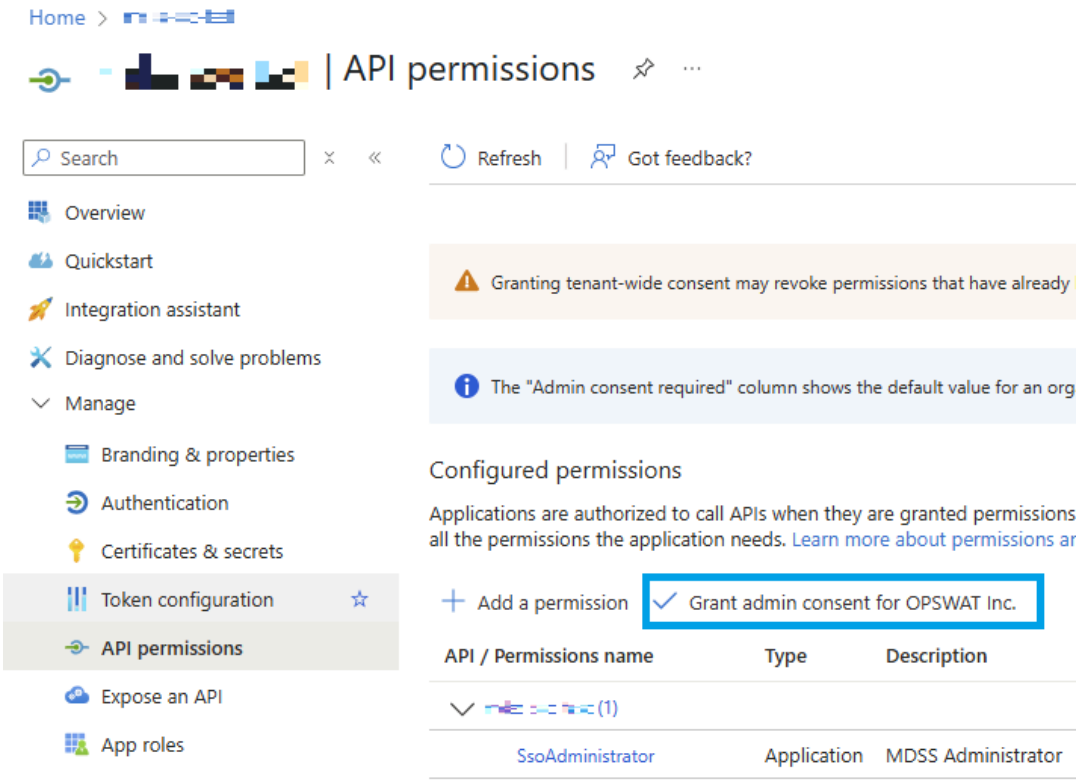
10. Navigate to the **API permissions** and click on **Add a permission**. Then, select the **My APIs** tab and find your app



11. Click on your app and select the app role you have configured.

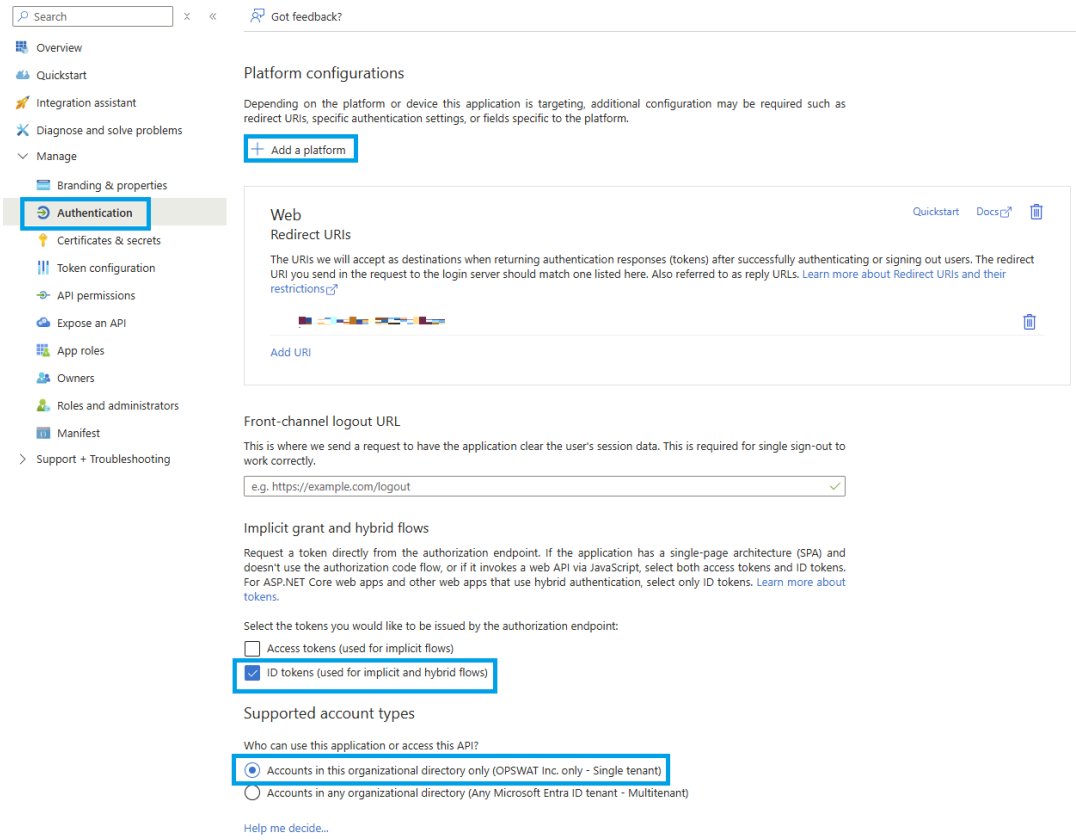


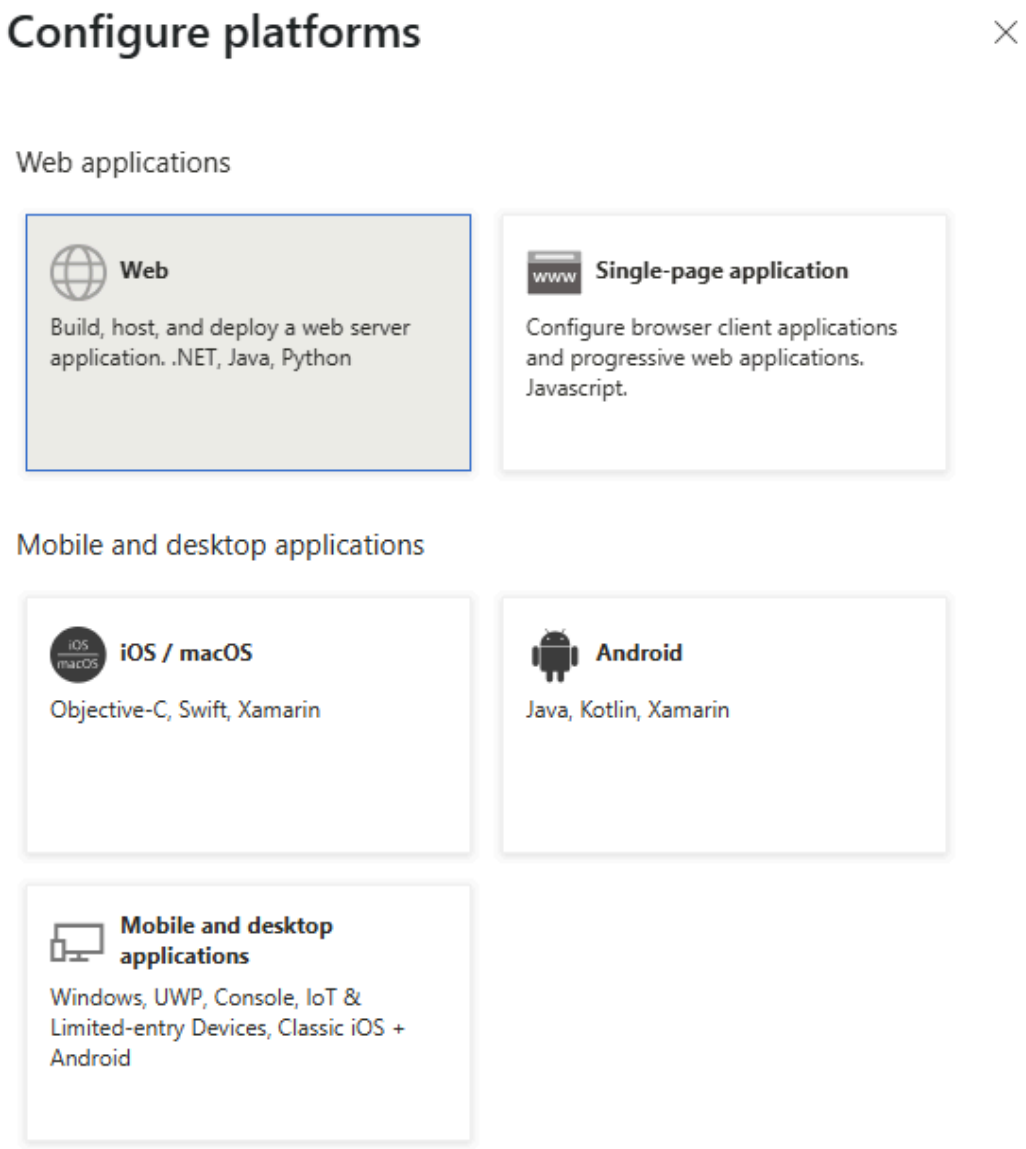
12. Next, click on **Grant admin consent**



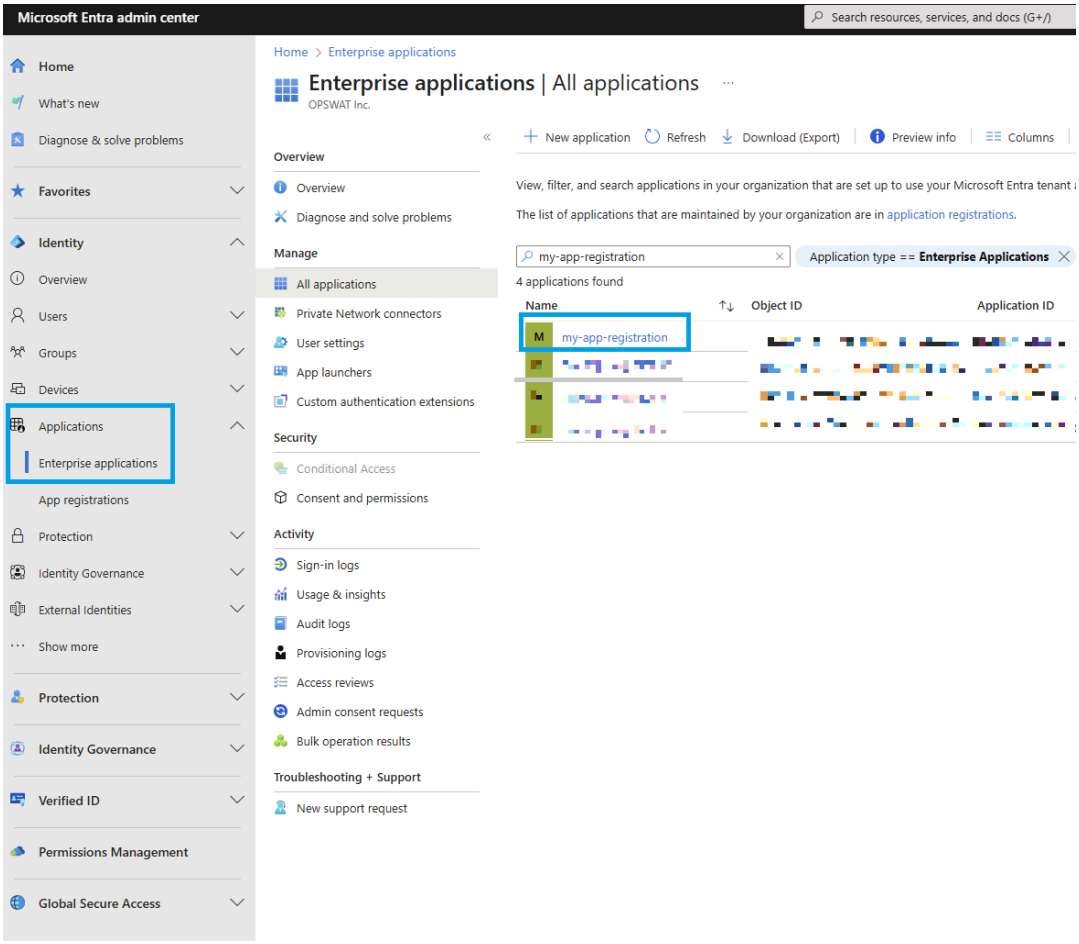
13. Go to the **Authentication** tab. Here, click **Add a platform** and select the **Web** option. Then, click on the checkbox for **ID tokens** and also for **Supported account types**,

select whatever you prefer (recommended is Single tenant).

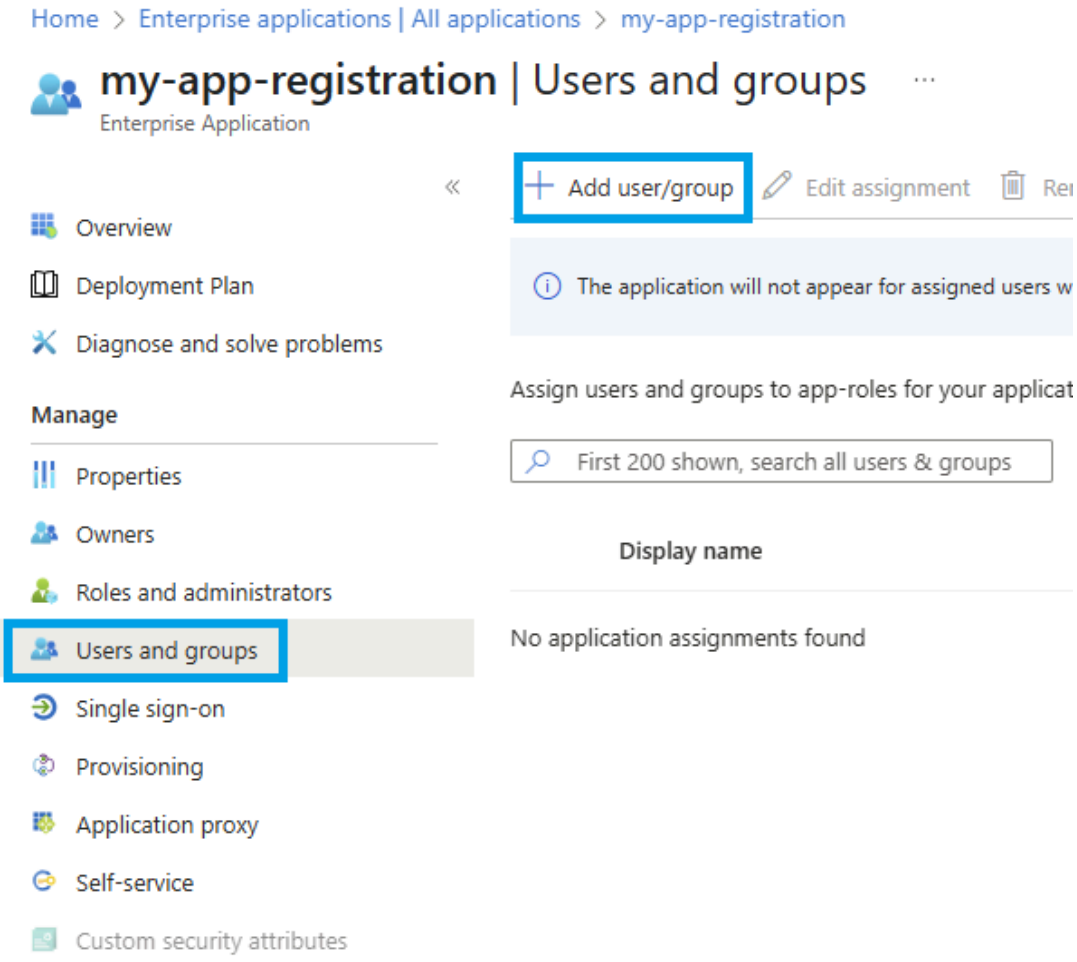




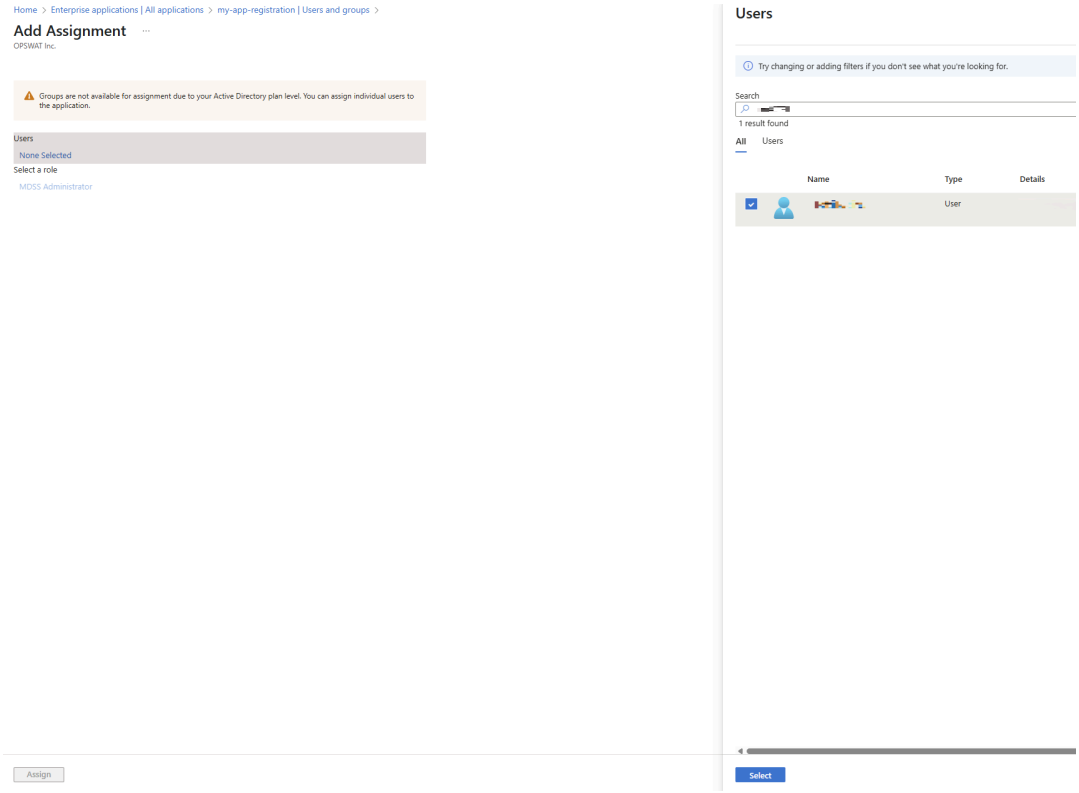
14. Now, leave the **Azure Portal** and navigate to the **Microsoft Entra admin center**. Here, find the **Enterprise applications** window under the **Applications** tab on the left. Search for your configured app.



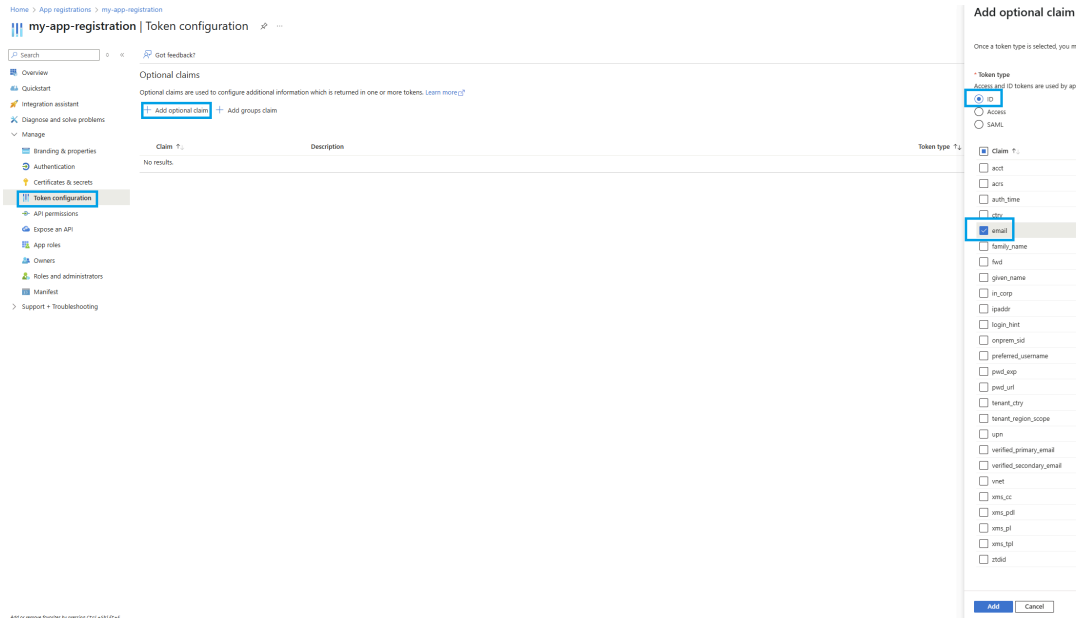
15. Navigate to the **Users and groups** tab and select **Add user/group**



16. Find your user and assign the configured App Role to it.



17. Lastly, navigate to **App registrations**, find your app and click on the **Token configuration**. There, click on **Add optional claim**, select **ID**, and add the **email** claim.



PingFederate Specific Configuration

> **Info**
>
> To properly access MDSS, the following AD **Security Groups** are required:

- **SsoAdministrator** - the equivalent of local MetaDefender Storage Security administrator role
- **SsoReadOnlyAdministrator** - the equivalent of local MetaDefender Storage Security read-only role

It is also necessary for the AD user to have an **email address** set up, so that it is set as the MetaDefender Storage Security's user email

This documentation was tested on PingFederate 12.1 with an Active Directory datastore.

Create the scopes

1. Go to System > OAuth Settings > Scope Management.
2. On the **Common Scopes** tab, add the following **Scope Values** along with a description. Click Add after entering each value and description.
   - openid
   - profile
   - email
   - offline_access
3. Skip the Exclusive Scopes tab.
4. On the Default Scope tab, enter a description for Default Scope. A description is required. If Default Scope Description is empty, PingFederate will log the following error: The requested scope is invalid, unknown, malformed, or exceeds that which the client is permitted to request.
5. Click Save.

Create a configuration on PingFederate

Create the Access Token Manager

1. Go to Applications> OAuth > Access Token Management.
2. Click Create New Instance
3. On the Type tab:

- Instance Name: Enter an instance name. For example, MDSS Access Token Manager.
- Instance ID: Enter the instance ID. For example, MDSS.
- Type: Select JSON Web Tokens.
- Parent Instance: Leave the default, None.

4. On the Instance Configuration tab:

- Use Centralized Signing Key: Select the checkbox.Leaving this checkbox unselected causes PingFederate to expect “Active Signing Certificate Key ID” to be configured.
- JWS Algorithm: Select an algorithm. For example, RSA using SHA-256.
- At the bottom of the screen, click Show Advanced Fields.
  - JWT ID Claim Length: Add a number greater than zero (0). For example, 24. If you do not enter a value, the JTI claim is omitted in the access token.

5. Click Next.
6. On the Access Token Attribute Contract tab:

- In the Extend the Contract text box, add the following claims to be generated in the Ping access token. Click Add after entering each claim.
  - userName
- Subject Attribute Name: Select the userName claim to be used for auditing purposes.

7. Click Next twice to skip the Resource URIs and Access Control tabs.
8. Click Save.

Add the objectGUID Attribute

1. Go to System > Data Stores > Your Data Store > LDAP Configuration.
2. On the LDAP Configuration tab, click Advanced at the bottom.
3. On the LDAP Binary Attributes tab, in the Binary Attribute name field, use objectGUID and click Add.
4. Click Save

Create the OpenID Connect Policy.

1. Go to Applications> OAuth > OpenID Connect Policy Management.
2. Click Add Policy.
3. On the Manage Policy tab:

- Policy ID: Enter a policy ID. For example, OIDC.
- Name: Enter a policy name. For example, OIDC Policy.
- Access Token Manager: Select the access token manager that you previously created. MDSS Access Token Manager.
- Tick **Include User Info in ID Token**

4. Click Next.
5. On the Attribute Contract tab:
   - Click **Delete** to remove all the attributes except **sub**.
   - Add the following attributes:

- Displayname
- Email
- Groups
- upn
- userName

6. Click Next, then click Next again to skip the Attributes Scope tab.
7. On the Attribute Sources & User Lookup tab, click Add Attribute Source.
8. After entering the information on each tab that follows, click Next to advance.

Data Store:

- Attribute Source ID: Enter an attribute source ID. For example, MDSSLDAP.
- Attribute Source Description: Enter a description. For example: MDSSLDAP
- Active Data Store: Select your Active Directory domain name from the drop-down.

LDAP Directory Search:

- Base DN: Enter your base DN to find your users and groups.
- Search Scope: Leave the default, Subtree.
- Attributes to return from search: Select and select the following.Click Add Attribute.
  - displayName
  - memberOf
  - objectGUID
  - userPrincipalName
  - mail
- LDAP Binary Attribute Encoding Types:
  - ObjectGUID: Select Hex for the Attribute Encoding Type.
- LDAP Filter:
  - Filter: Enter a filter. For example, `userPrincipalName=${userName}`.

9. On the Summary page, click Done.
10. Click Next to advance, and on the Contract Fulfillment tab, map the Attribute Contract for the ID token:

Attribute Contract	Source	Value
displayname	MDSSLDAP	displayName
email	MDSSLDAP	mail
groups	MDSSLDAP	memberof
sub	MDSSLDAP	objectGUID
upn	MDSSLDAP	userPrincipalName
userName	No Mapping	

11. Click Next, then click Next again to skip the Insurance Criteria tab.
12. Click Save

Create the OAuth Client Application

1. Go to Applications > OAuth > Clients.
2. Click Add Client.
3. On the Clients | Client page:
    1. Client ID: Enter the client ID. For example, MDSS
    2. Name: Enter a name. For example, MDSS.
    3. Client Authentication: Select Client Secret.
    - Client Secret: You can input your own client secret or generate a secret. After you leave this page, you cannot view the secret again. You only have the option to change the secret
    - Redirection URIs: write the the base URL of the MDSS deployment, followed by the "**/callback**" suffix.
    - Allowed Grant Types: Check Authorization Code, Implicit, Refresh Token, Client Credentials.
    - Default Access Token Manager: Select the access token manager you previously created. MDSS Access Token Manager.
    - OpenID Connect: For Policy, select the one you previously created. For example, the one used in this documentation is OIDC
4. Click Save.

Create the Password Grant Flow Configuration

1. Create the Password Credential Validator.
    1. Go to System > Data & Credential Stores > Password Credential Validators.

2. Click Create New Instance.
3. In the Password Credential Validators | Create New Instance page, enter the information as follows for each tab, then click Next to advance.
   - On the Type tab:
     - Instance Name: Enter the instance name. For example, MDSS Validator.
     - Instance ID: Enter the instance ID. For example, MDSS.
     - Type: Select LDAP Username Password Credential Validator.
   - On the Instance Configuration tab:
     - LDAP Datastore: Select the data store you are using.
     - Search Base: Enter your base DN to find your users and groups.
     - Search Filter: Enter a filter. For example, userPrincipalName=${userName}.
     - Scope of Search: Select Subtree.
   - On the Extended Contract tab:
     - By default, the following are added:
       - DN
       - email
       - givenName
       - username
4. Click Next then click Save.

2. Map the Validator in the Authorization Server Settings.
   1. Go to System > OAuth Settings > Authorization Server Settings.
   2. In Password Credentials Validator, select the one you previously created. For example, this documentation uses MDSS Validator.
   3. Click Save.
3. Create the Resource Owner Credentials Grant Mapping.
   1. Go to Authentication > OAuth > Resource Owner Credentials Mapping.
   2. In the Resource Owner Credentials Grant Mapping window:
      - Source Password Validator Instance: Select the one you previously created and click Add Mapping.
   3. In the Resource Owner Credentials Grant Mapping | Resource Owner Credentials Mapping page, click Next to skip the Attribute Sources & User Lookup tab.
   4. On the Contract Fulfillment tab:
      - For USER_KEY, select Password Credentials Validator, and for Value, select username.
   5. Click Next to skip the Insurance Criteria tab then click Save.

4. Create the Access Token Mapping - Map the Password Credentials Validator to the Access Token Manager.This mapping is needed for the Password Grant workflow. If the mapping does not exist, PingFederate logs the following error: There are no access token managers available for the selected client and authentication context.
    1. Go to Applications > Access Token Mappings.
        - Context: Select the one you previously created. For example, this documentation uses MDSS Validator.
        - Access Token Manager: Select the one previously created. For example, this documentation uses MDSS Access Token Manager.
    2. Click Add Mapping.
    3. Click Next to skip the Attribute Sources & User Lookup tab.
    4. On the Contract Fulfilment tab select userName
    5. Click Next to skip the Insurance Criteria tab then click Save.

Create the Authorization Code Flow Configuration

1. Create the IdP Adapter.
    1. Go to Authentication > Integration > IdP Adapters.
    2. Create Create New Instance.
    3. On the Type tab:
        - Instance Name: Enter a name, for example, HTML Form Auth Adapter.
        - Instance ID: Enter an ID, for example, HTMLFormAuthAdapter.
        - Type: Select HTML Form IdP Adapter.
        - Parent Instance: Select None.
    4. Click Next.
    5. On the IdP Adapter tab:Under Password Credential Validator Instance, click Add a new row to 'Credential Validators.' then select a validator (in this documentation, MDSS Validator is used), and click Update.
    6. Click Next.
    7. Click Next to skip the Extended Contract tab.
    8. On the Adapter Attributes tab:
        - Unique User Key Attribute: Select username and check Pseudonym.
    9. Click Next to skip the Adapter Contract Mapping tab then click Save.
2. Create IdP Adapter Grant Mapping.
    1. Go to Authentication > OAuth > IdP Adapter Grant Mapping.
    2. Source Adapter Instance: Select the Adapter instance you just created and click Add Mapping.

3. On the Attribute Sources & User Lookup page, click Add Attribute Source.
4. Enter the information as follows for each tab, then click Next to advance.
   - On the Data Store tab:
     - Attribute Source ID: Enter an ID with alphanumeric values.
     - Attribute Source Description. Enter a description.
     - Active Data Store: Select the active directory in use.
   - On the LDAP Directory Search tab:
     - Base DN: Enter your base DN to find your users and groups.
     - Search Scope: Use the default, Subtree.
     - Attributes to return from search: Select , then, once loaded, from the attribute list, select userPrincipalName.
5. Click Add Attribute then click Next.
6. On the LDAP Filter tab:
   - Filter: Enter the filter. For example, `userPrincipalName=${userName}`.
7. Click Next then click Save.
8. In the IdP Adapter Grant Mapping | IdP Adapter Mapping page, finish creating the IdP Grant Mapping.On the Contract Fulfillment Lookup tab, use the following table.

Contract	Source	Value
USER_KEY	Select the source previously created.	Subject DN
USER_NAME	Select the source previously created.	userPrincipalName

l. Click Next then click Save

3. Map the IdP Adapter to Access Token Manager.
4. Go to Applications > OAuth > Access Token Mappings.
   - Context: Select IdP Adapter:Adapter Name.
   - Access Token Manager: Select the access token manager instance previously created. For example, in this documentation, it is MDSS Access Token Manager.
5. Click Add Mapping.If you do not perform this mapping, PingFederate generates the following log file message: There are no mapped authentication sources to choose from. Please map an IdP Adapter or IdP connection first.
6. Skip the Attribute Sources & User Lookup tab, and on the Contract Fulfillment tab, add the following:

Contract	Source	Value
userName	Adapter	username

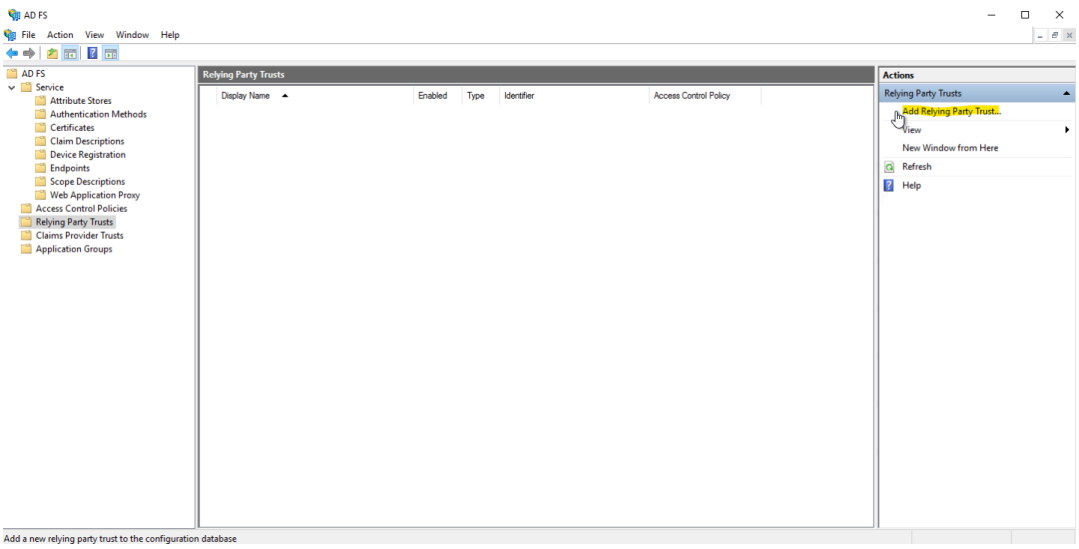
- Click Next to skip the Insurance Criteria tab then click Save.

AD FS SAML2 Specific Configuration

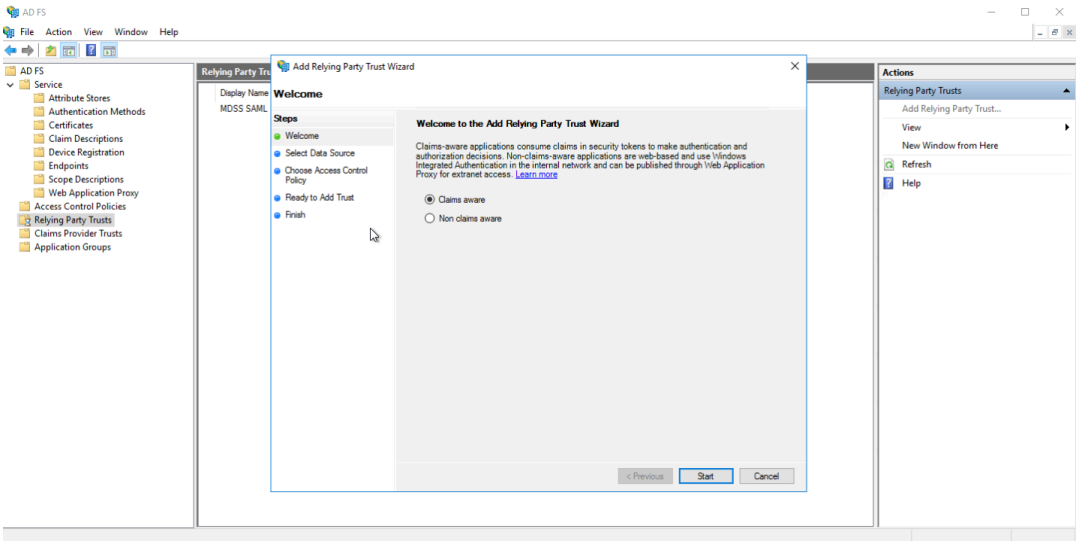
> **Info**
>
> To properly access MDSS, the following AD **Security Groups** are required:
>
> - **SsoAdministrator** - the equivalent of local MetaDefender Storage Security administrator role
> - **SsoReadOnlyAdministrator** - the equivalent of local MetaDefender Storage Security read-only role
>
> It is also necessary for the AD user to have an **email address** set up, so that it is set as the MetaDefender Storage Security's user email.
>
> The only supported secure hash algorithm for signing SAML requests is **SHA-256**.

To set up AD FS SSO with SAML2 for **MDSS**, follow these steps:

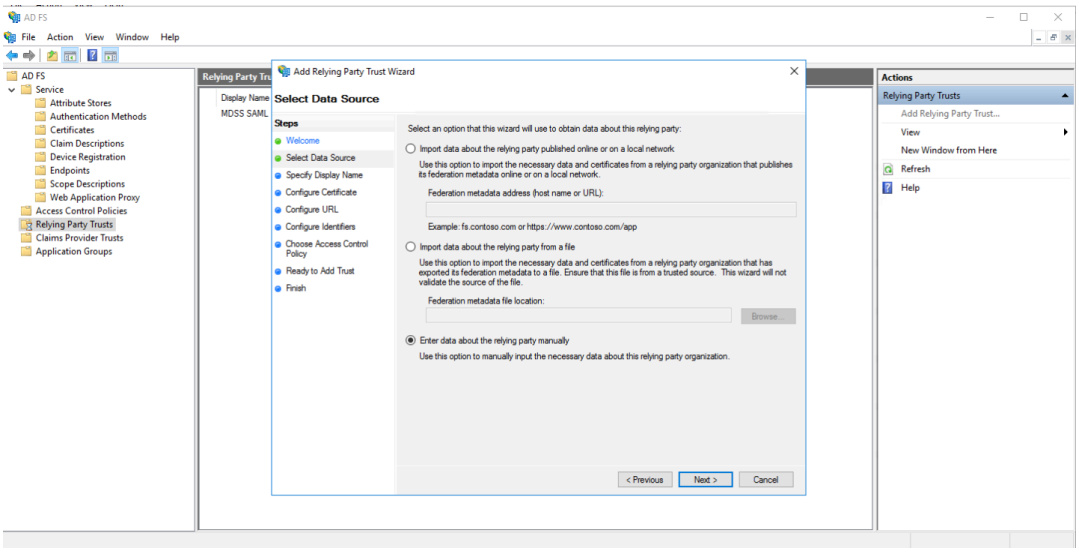
1. Within the AD FS settings, navigate to the "**Relaying Party Trusts**" directory and click "**Add Relaying Party Trust**"



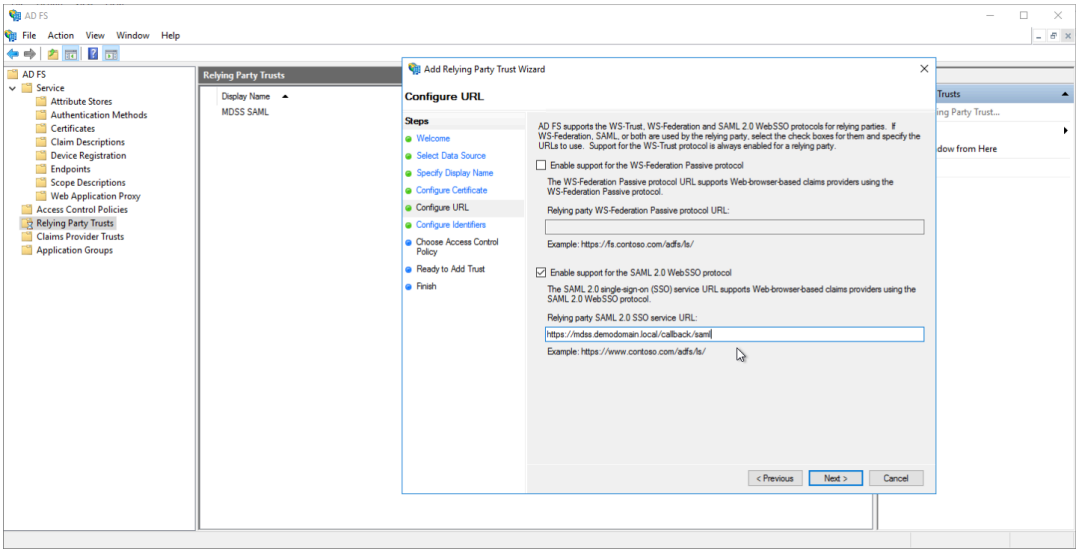
2. Check **Claims Aware** for the first step



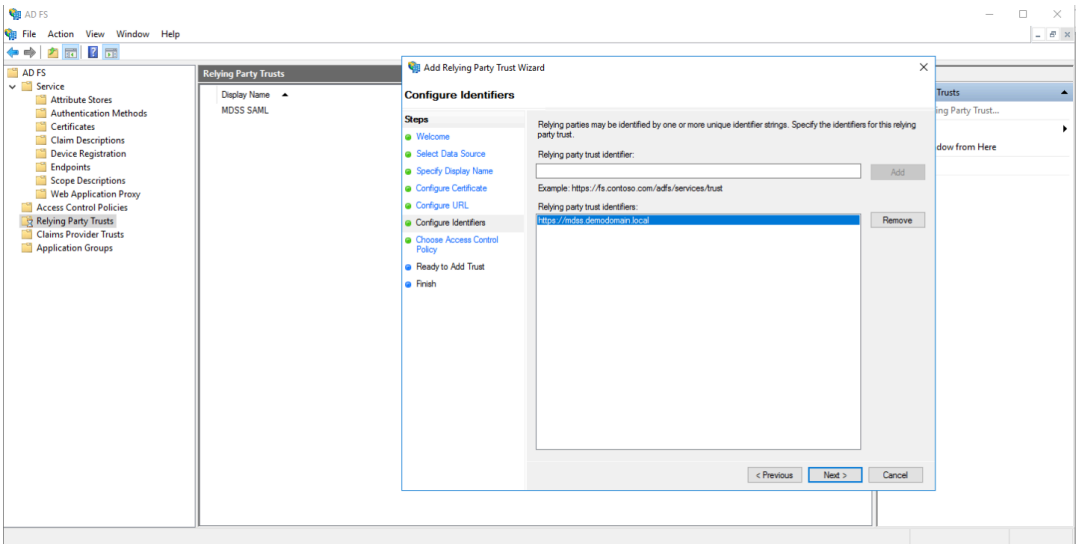
3. Select Enter data about the relaying party manually



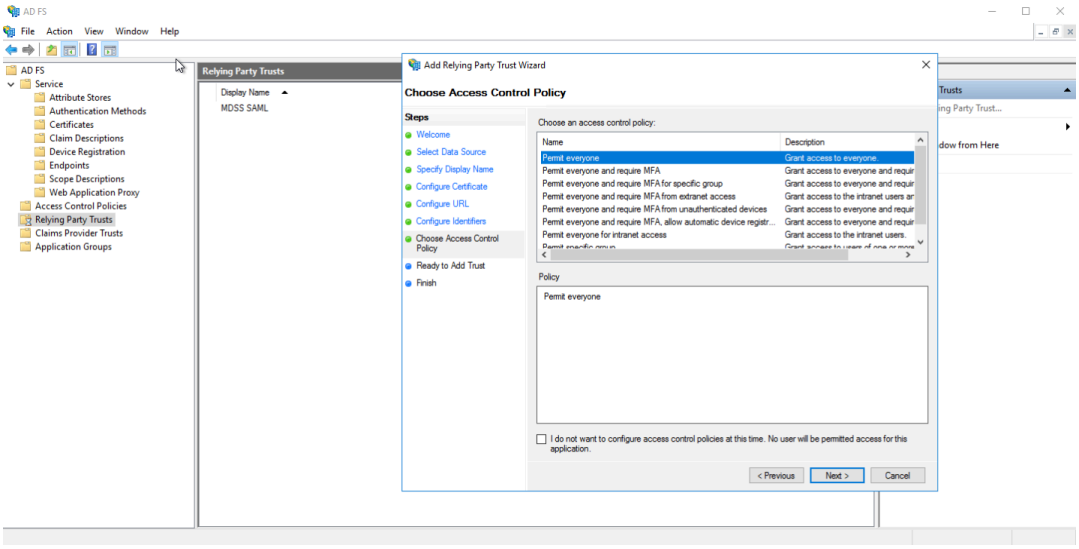
4. Give your Relaying party any name you like, and at the configure certificate step, do not enter any encryption certificate and go to the next step (MDSS does not currently support claim encryption)
5. Check Enable support for the **SAML 2.0 WebSSO** protocol, and enter the service URL of **MDSS** (this URI is of form *https:///callback/saml*)



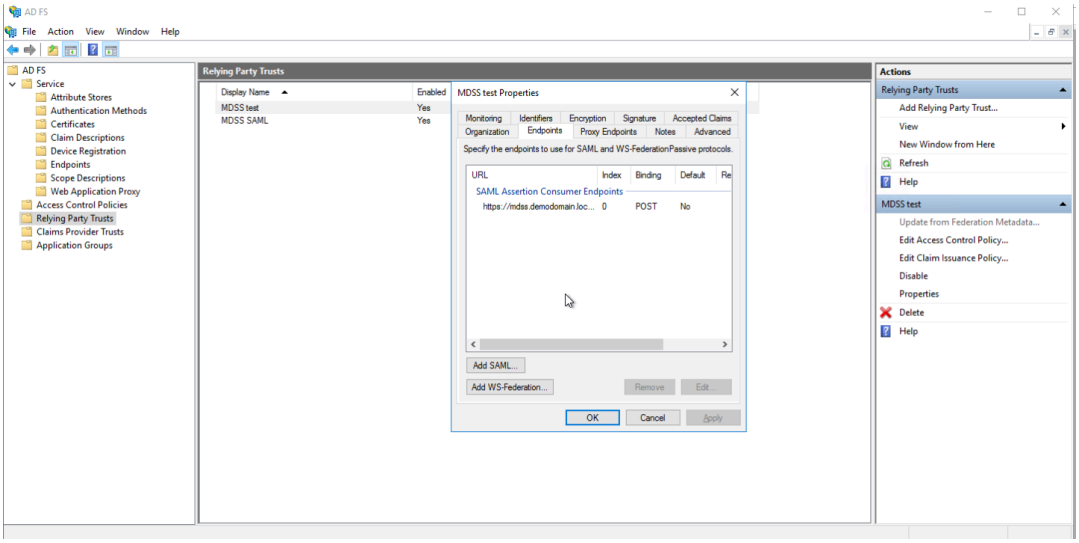
6. Enter the URI of the **MDSS** instance as the relaying party identifier (this URI is of form *https://*)



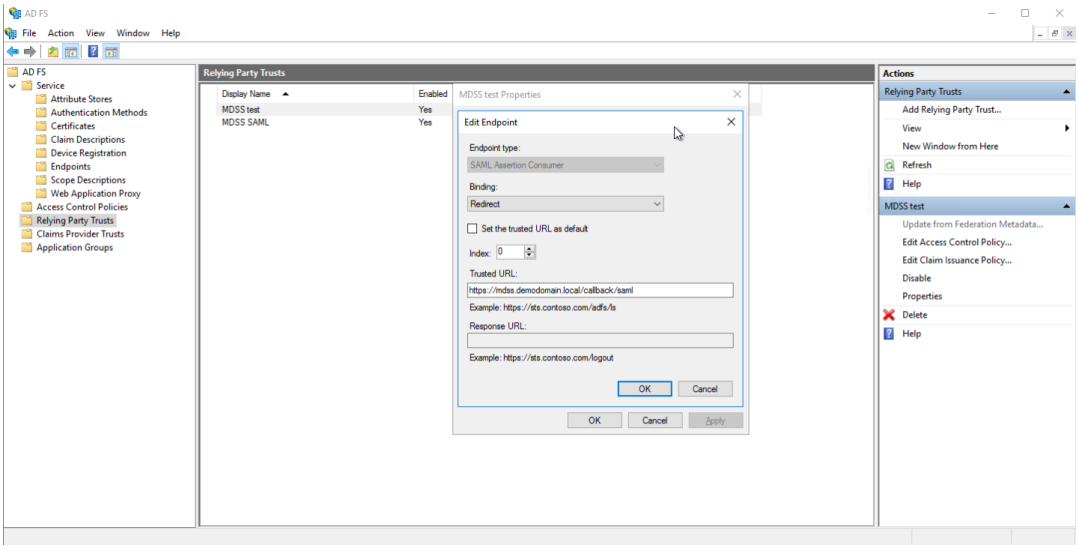
7. For the access **control policy**, select "**Permit everyone**"



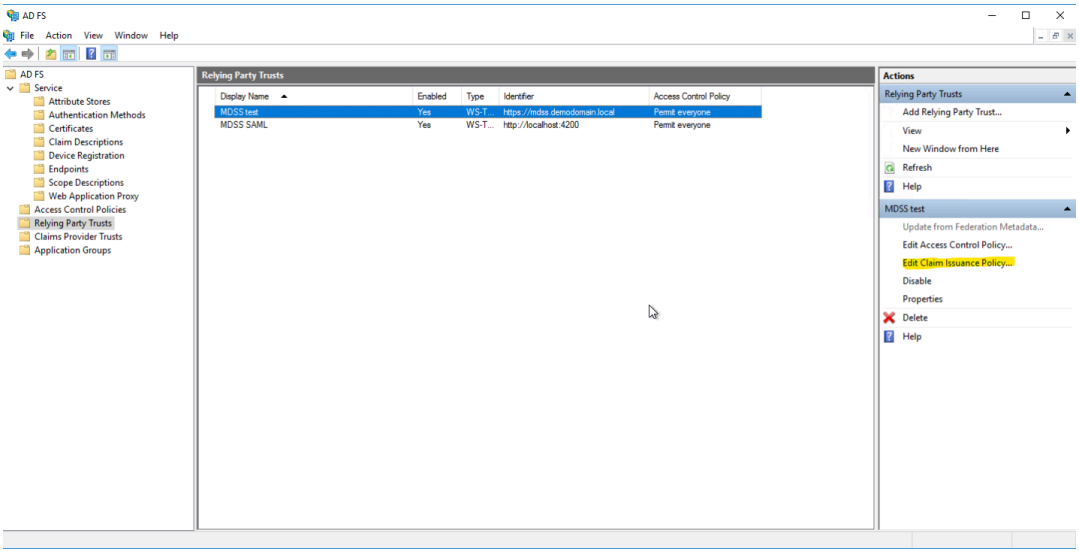
8. Confirm the settings added so far and finish.
9. Select the created Relaying Party Trust, click on **Edit Access Control Policy**, and go to the Endpoints tab.



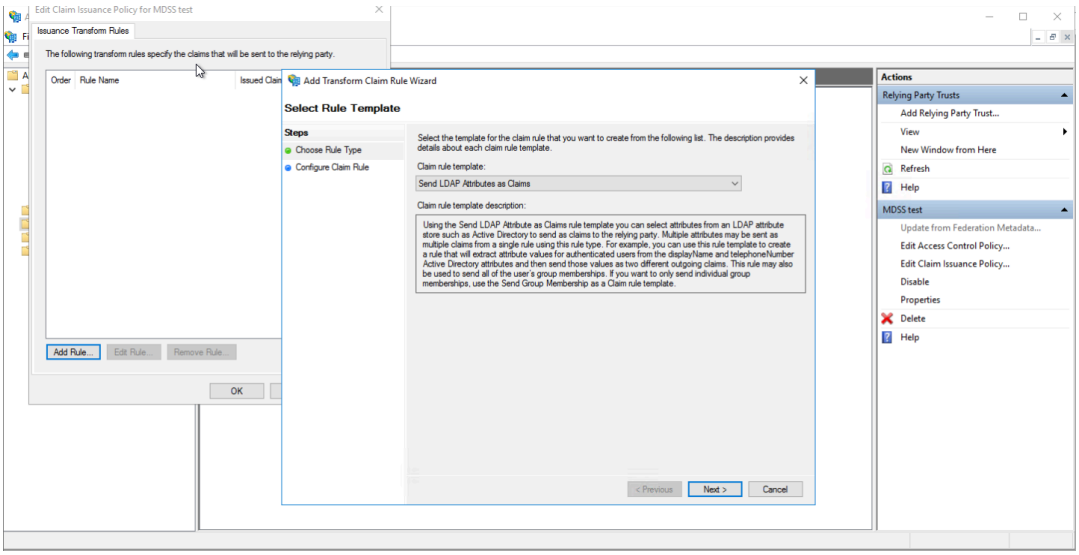
10. Select the configured endpoint, change the binding type to **Redirect**, and save it.



11. Select the created Relaying Party Trust and click on **Edit Claim Issuance Policy**

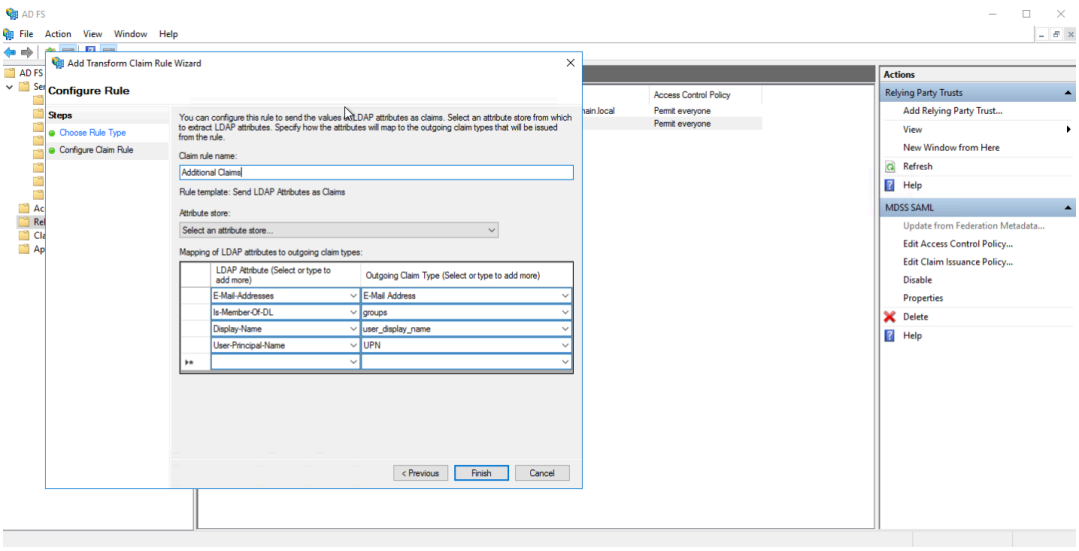


12. Add a rule and select "**Send LDAP Attributes as Claims**" as the **Claim Rule Template**

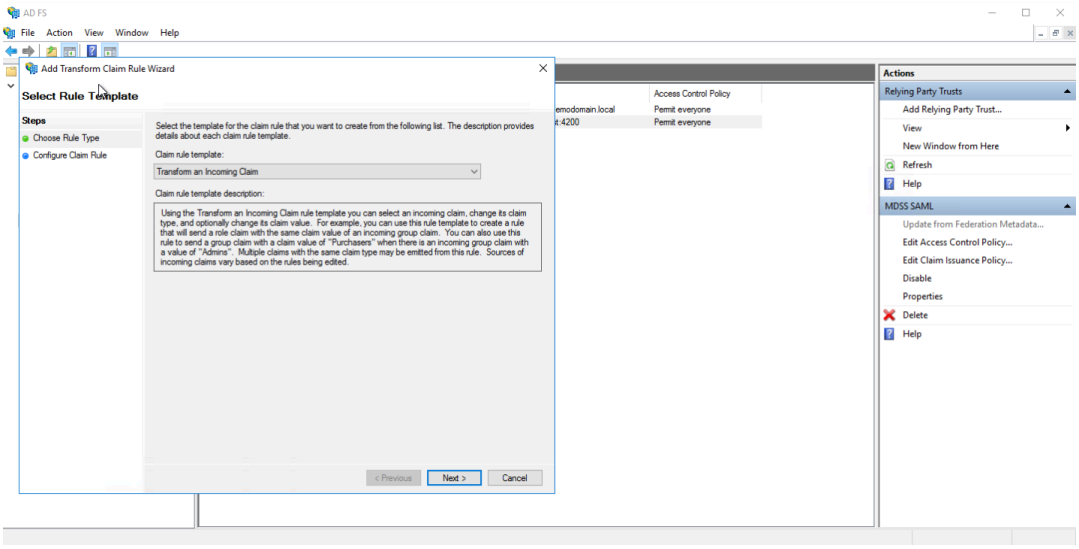


13. Give the rule a name, select **Active Directory** as the attribute store and configure the following:

- **E-Mail Address** as **E-mail Address**
- **Is-Member-Of-DL** as **" groups*****"***
- **Display-Name** as ***"user_display_name"***
- **User-Principal-Name** as ***"UPN"***

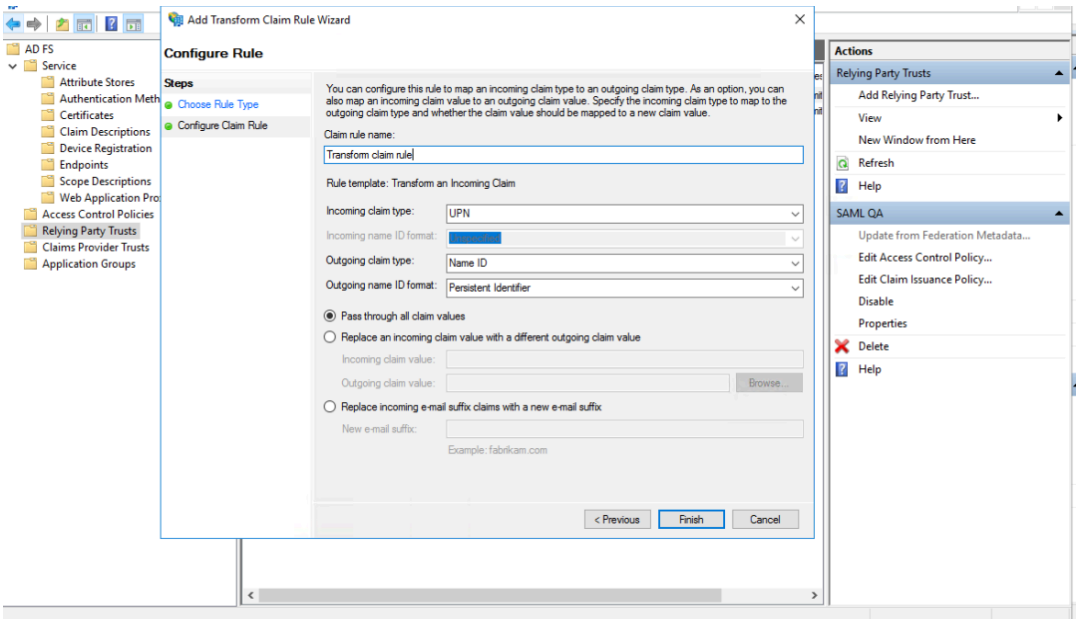


14. Add a rule and select "**Transform an Incoming Claim**" as the Claim Rule Template

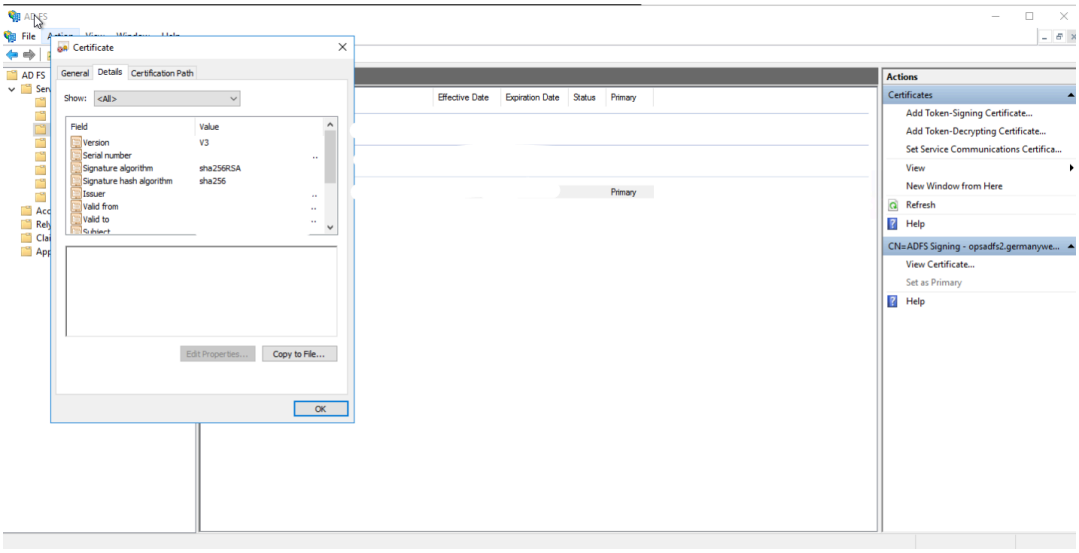


15. Give the rule a name, select **Active Directory** as the attribute store and configure the following:

- **“UPN”** at **Incoming claim type**
- **"Name ID"** at **Outgoing claim type**
- **“Persistent Identifier”** at ***Outgoing name ID format*** __ Next, within the



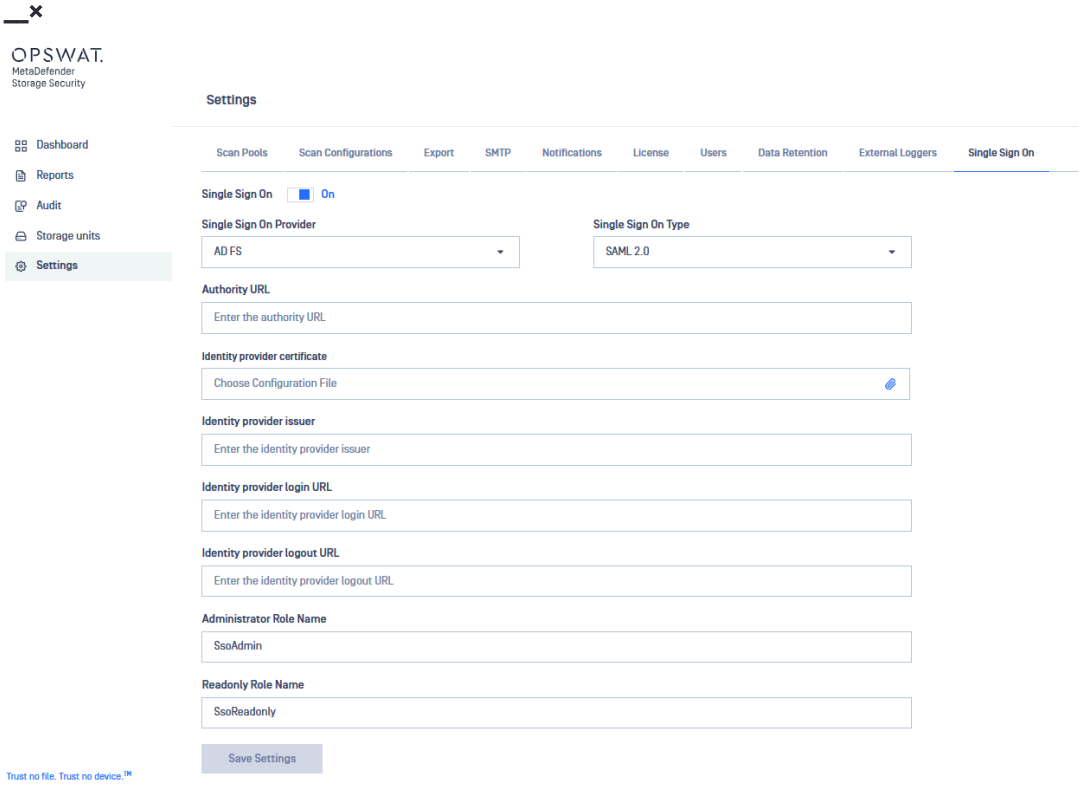
16. Next, within the **AD FS settings**, navigate to the “**Service/Certificates**” directory and double click on your token signing certificate. On the Details tab click Copy to File to save the certificate and use it when configuring SSO on MDSS.



17. On **MDSS**, under the **Single Sign On page** from Settings, configure your provider with the following settings:

- **Authority URL**: the base URL of your AD FS server used for retrieving Federation Metadata (eg. https:/adfs..com/)
- **Identity provider certificate**: the certificate obtained earlier from AD FS
- (Optional) **Identity provider issuer**: the claim issuer identifier used to sign the incoming claims (eg. https://adfs..com/adfs/services/trust)
- (Optional) **Identity provider login URL**: the login URL of the AD FS instance (eg. https://adfs..com/adfs/ls/)
- (Optional) **Identity provider logout URL**: the logout URL of the AD FS instance (eg. https://adfs..com/adfs/ls/)
- **Administrator account role name**: if you use custom group names assigned to your SSO users, pick a different role from the default
- **ReadOnly account role name**: if you use custom group names assigned to your SSO users, pick a different role from the default

> **Info**
>
> If your AD FS instance supports **Federation metadata** and has it enabled and configured on the */FederationMetadata/2007-06/FederationMetadata.xml* URL, the Issuer, login and logout URL fields are not required, and will be retrieved from the received federation metadata. Otherwise, they must be provided.
>
> If the fields are populated, they will override any value retrieved from the federation metadata, and cause the **Authority URL** to not be validated.


OPSWAT.
MetaDefender
Storage Security
Dashboard
Reports
Audit
Storage units
Settings
Trust no file. Trust no device.™
Settings
Scan Pools
Scan Configurations
Export
SMTP
Notifications
License
Users
Data Retention
External Loggers
Single Sign On
Single Sign On
On
Single Sign On Provider
AD FS
Single Sign On Type
SAML 2.0
Authority URL
Enter the authority URL
Identity provider certificate
Choose Configuration File
Identity provider issuer
Enter the identity provider issuer
Identity provider login URL
Enter the identity provider login URL
Identity provider logout URL
Enter the identity provider logout URL
Administrator Role Name
SsoAdmin
Readonly Role Name
SsoReadonly
Save Settings

Database procedures

> **Warning**
>
> Whether you use the internal PostgreSQL or an external one, the installed version should be 17.

MetaDefender Storage Security includes an internal PostgreSQL instance by default, currently running on version 17.

For environments with high loads or where high availability is required, the internal PostgreSQL instance can be replaced with a cloud-managed alternative. Instructions for making this switch are provided here: AWS RDS with PostgreSQL

AWS RDS with PostgreSQL

Support AWS Services with PostgreSQL Engines

MetaDefender Storage Security natively supports AWS RDS and Aurora with PostgreSQL engine [Supported PostgreSQL versions]**.** MetaDefender Storage Security does not require any specific pre-installation steps to work with Amazon RDS - all required extensions are performed internally during installation of the product.

Regarding Aurora support, as it generates 2 endpoints, **Cluster endpoint** for administrate, insert and read. And the **reader endpoint** that is only for reading using the LB that get the information from any instance of the cluster. MetaDefender Storage Security only supports connections to 1 endpoint, the cluster endpoint, that allow the application do all the actions needed. Info from AWS

You can create and configure AWS RDS and PostgreSQL by following the guide here.

Database Configuration

High Availability with AWS RDS

AWS RDS supports various multi-az deployment that can be configured following this documentation. You can compare the different configuration options available in this AWS documentation. For having MetaDefender Storage Security installed using any of this approaches we recommend to follow the instructions indicated by AWS in case of having MDSS installed in a EC2 instance. For EKS Cluster deployment OPSWAT provide a terraform code to deploy it together with the K8S cluster, see Deployment on EKS

Database Sizing

It depends on how many requests or traffic is sent to MetaDefender Storage Security, and how complex the files are. As average the minimum database size desired **for each 1,000,000 analysis reports** added is **6 GB.**

The recommended AWS RDS Instance Type is **db.r6g.xlarge** and the storage type recommended is General Purpose SSD. For gp2 storage type **the minimum size is 20 GiB** so this is marking as the minimum storage needed for working with MetaDefender Storage Security.

Security Group Configuration

MetaDefender Storage Security does not require that the RDS database is publicly accessible. For enhanced security, OPSWAT recommends the following:

- Do NOT deploy the database in a public subnet.
- Do NOT include ingress security group rules with 0.0.0.0/0 sources
- Do NOT set the Amazon RDS Publicly Accessible parameter to true in terraform project for Kubernetes deployment

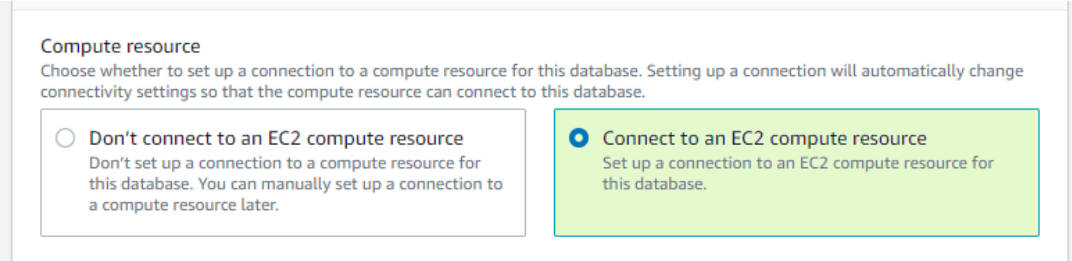
Database Connection

For general information of how to connect to the RDS depending on the scenarios it is recommended to read this docs from AWS.

Connect to RDS in Private Subnet

To connect MetaDefender Storage Security with RDS hosted in a private subnet, the configuration will depend on the type of deployment.

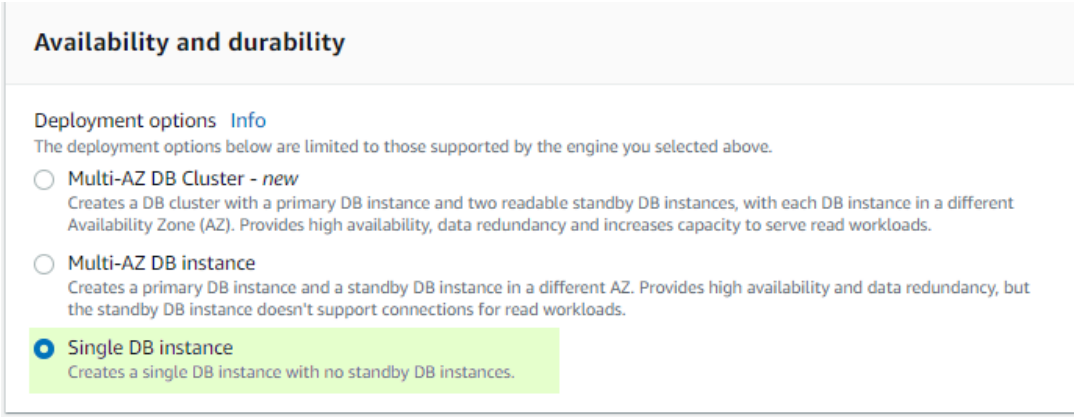
For Single EC2 deployment, it is needed to indicate that the RDS will be only available from the EC2 instance where MetaDefender Storage Security is running, configuring it from the Connectivity section when creating the RDS instance as following.



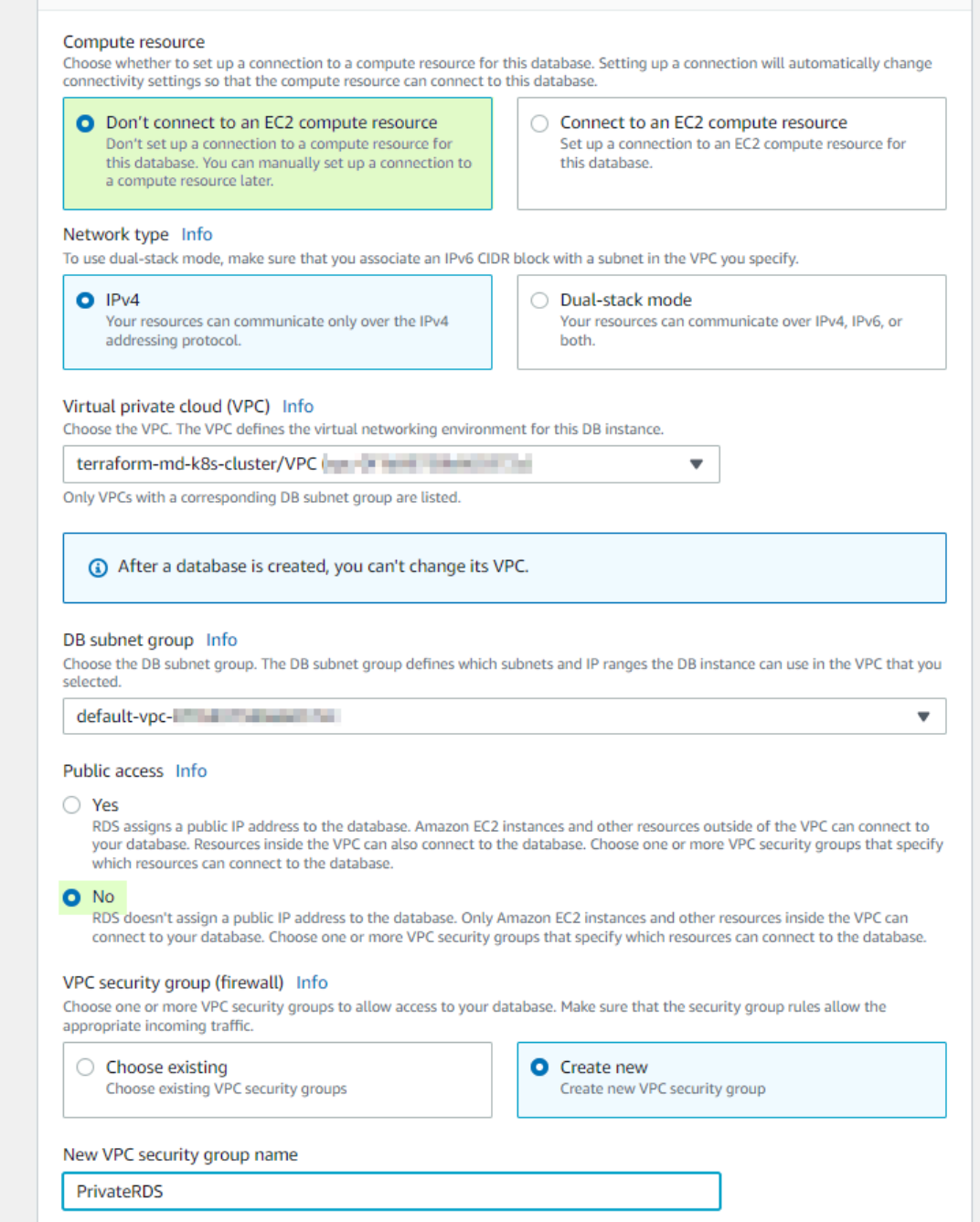
For EKS deployment, using the terraform project provided by OPSWAT it is deployed in a private subnet only allowing access from VPC CIDR where the EKS cluster is deployed. In

case of creating the RDS instance from the console follow these steps

1. Indicate if you want to deploy it in Multi-AZ or Single DB



2. Indicate not to connect to an EC2 and the VPC public access “No”. For multiple VPC, deploy it in a different VPC than EKS. This example is using the same VPC



Multiple VPC for MetaDefender Storage Security and AWS RDS

When working with multiple VPCs where we have deployed MetaDefender Storage Security either in a Single EC2 or with EKS in one VPC and the RDS instance in a different VPC, it will be needed to configure a VPC peering between both VPC for MetaDefender Storage Security to use the database.

Steps to configure it

1. Create a Peering Connection between both VPCs

2. Modify the route table to route the traffic to the peering for the requests from both VPCs
3. Modify the security group to allow connections from both VPCs

Database Endpoint Management

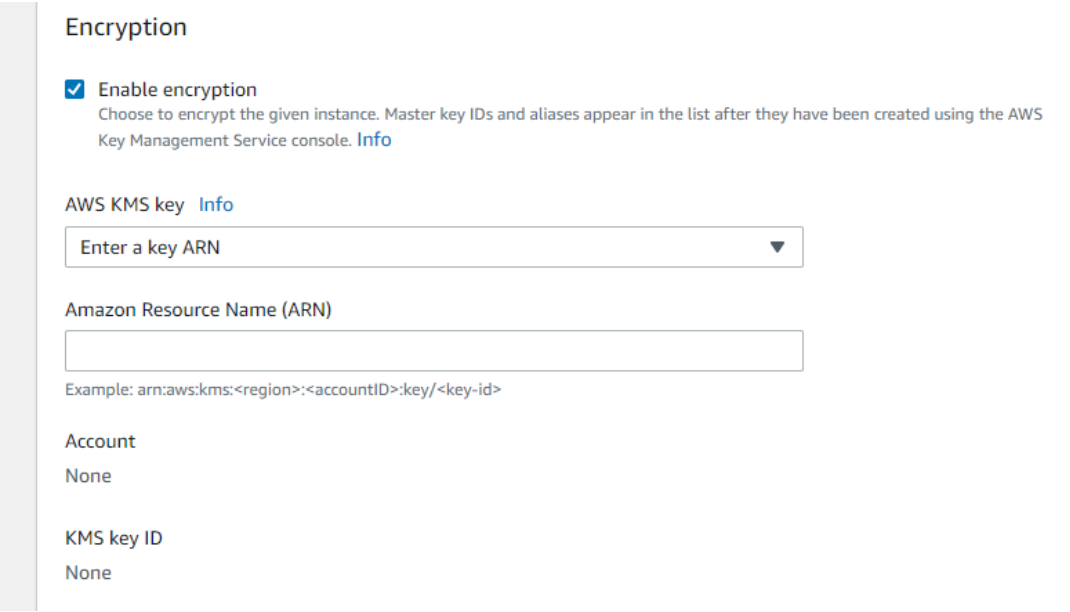
The database connection endpoint is generated by AWS once the RDS instance is created you can access to this information on the **Connectivity & security** tab. Once you have this information it should be set up in the `ConfigMap` , configuring the hostname of the database and the port to match the new RDS instance.

More detailed information from AWS here.

Database Encryption

MetaDefender Storage Security supports both native OS encryption or EBS encrypted volumes. The database is secured via credentials and can be configured to encrypt everything at rest. In general, OPSWAT recommends the use of a 3rd party database layer, e.g. Amazon Aurora or RDS, which should be encrypted. It is best practice to not share the database layer on the same hosts/VM/EC2 instance.

For encrypting RDS this is the recommended configuration



Exporting database

We offer the option to export your database if you need to use it for migration or debugging purposes.

```
PS> .\mdss.ps1 -u export_db
```

```
sudo mdss -u export_db
```

Upgrade Guide to PostgreSQL (MDSS v4.0.0)

MDSS v4.0.0 Upgrade Guide: MongoDB to PostgreSQL Migration

Overview

Starting with MDSS version 4.0.0, PostgreSQL replaces MongoDB as the primary database backend. This guide covers the upgrade process, migration scenarios, and best practices for a seamless transition.

Key Changes in v4.0.0

- **Database Backend**: PostgreSQL replaces MongoDB
- **Automatic Migration**: Data is migrated by default during upgrade
- **External Database Support**: Improved support for external PostgreSQL instances

Pre-Upgrade Considerations

When Data Migration Occurs

Data migration happens automatically when upgrading from any MDSS version < 4.0.0 and any of the following conditions are met:

- Previous version uses internal/bundled MongoDB
- New version is configured to use external PostgreSQL
- Both previous and new versions are configured with external databases

> **Warning**
>
> Data migration from MongoDB to PostgreSQL takes about **2-3 hours / 1M files** in the original database. In this time MDSS is not available for operation. Other upgrade options are available below.

When Data Migration is Skipped

Data migration is **automatically skipped** when all the following conditions are met:

- Previous version uses external MongoDB
- New version is configured with bundled PostgreSQL deployment

> **Info**
>
> ⚠️ Important: This prevents unintentional migration of cloud/external data to local deployments.

Data migration is also skipped after a successful installation or upgrade of MDSS v4.0.0 on the same machine so that **data is migrated only once**. This is done in order to ensure data in MDSS v4.0.0 is not overwritten by an unintentional migration from an older version.

Upgrade Options

Option 1: Standard Upgrade with Migration

Recommended for: Small to medium deployments where historical data is required.

1. **Backup your current deployment**
2. **Upgrade to v4.0.0** using your standard deployment process
3. **Monitor the migration** - duration depends on data volume

Configuration: No additional configuration needed for default migration.

Option 2: Skip Data Migration

Recommended for: Large deployments or when historical data is not critical.

Add the following to your `customer.env` file before upgrading:

```
SKIP_MONGO_TO_PG=yes
```

> **Info**
>
> 💡 Time Savings: For large datasets, migration can take several hours. Skipping migration provides immediate availability of the new version.

Option 3: Blue-Green Deployment

Recommended for: Production environments requiring zero downtime and maximum safety.

Process:

1. **Deploy v4.0.0 in parallel** to your existing environment **without starting any scans**
2. **Configure the new deployment** with the same settings as the old one
3. **Test thoroughly** without activating storage processing
4. **Stop storage processing** on the old version (while keeping it running for rollback capability)
5. **Activate storage processing** on the new v4.0.0 deployment
6. **Monitor the new deployment** for proper functionality
7. **Decommission old environment** after validation period

Benefits:

- Controlled transition of storage processing
- Easy rollback by reversing the process
- No data migration time
- Fresh start with optimized database
- Prevents duplicate storage scanning

External PostgreSQL Configuration

Recommended Providers

For production deployments, consider these managed PostgreSQL services for better performance, reliability and high availability options:

- **AWS RDS PostgreSQL**
- **Google Cloud SQL for PostgreSQL**
- **Azure Database for PostgreSQL**

Connection Configuration

To use an external PostgreSQL database, add the following to your `customer.env` file:

```
POSTGRESQL_URL=Host={your_pg_host};Port=5432;Username=
{your_pg_user};Password={your_pg_password};Database=MDSS
```

Example configurations:

```
# AWS RDS
POSTGRESQL_URL=Host=mdss-prod.cluster-xyz.us-east-
1.rds.amazonaws.com;Port=5432;Username=mdss;Password=SecurePassword123;
Database=MDSS

# Google Cloud SQL
POSTGRESQL_URL=Host=10.1.2.3;Port=5432;Username=mdss;Password=SecurePas
sword123;Database=MDSS

# Self-hosted with HA
POSTGRESQL_URL=Host=postgres-
cluster.internal;Port=5432;Username=mdss;Password=SecurePassword123;Dat
abase=MDSS
```

Database Configuration Recommendations

Connection Limits

Ensure your PostgreSQL instance has adequate connection limits:

```
-- Recommended minimum for production
ALTER SYSTEM SET max_connections = 1000;
SELECT pg_reload_conf();
```

Performance Tuning

Consider these PostgreSQL settings for optimal MDSS performance:

```
-- Memory settings (adjust based on your server capacity)
ALTER SYSTEM SET shared_buffers = '8GB'; -- for best performance set to
about 1/4 of total available memory
ALTER SYSTEM SET effective_cache_size = '16GB'; -- for best performance
set to about 1/2 of total available memory
ALTER SYSTEM SET work_mem = '256MB';
ALTER SYSTEM SET temp_buffers = '64MB';
ALTER SYSTEM SET maintenance_work_mem = '2GB';

-- WAL (Write-Ahead Logging) configuration
ALTER SYSTEM SET wal_level = 'minimal';
ALTER SYSTEM SET max_wal_senders = 0;
ALTER SYSTEM SET max_replication_slots = 0;
ALTER SYSTEM SET max_wal_size = '2GB';
ALTER SYSTEM SET min_wal_size = '160MB';

-- Connection and logging
ALTER SYSTEM SET log_statement = 'none';
ALTER SYSTEM SET log_min_duration_statement = 1000;
```

More details on configuring allocated resources on a PostgreSQL database can be found in their official documentation here.

What Gets Migrated

- User accounts and permissions
- Configuration settings
- Historical data and logs
- Custom configurations
- Dashboards and reports
- Storages

> **Warning**
>
> **Special attention** on Real Time Processing, Storage Remediations and Scan Configurations migration.
>
> The idea of Workflow is the result of merging Scan Configuration and Storage remediations. To not lose any remediations from storage, a Workflow shall be created starting from Default Scan Configuration and appending the Storage Remediations.
>
> Remediation of DeepCdr with encrypt original will be migrated to Keep Allowed and Keep Blocked for the original file
>
> The existent Scan Configuration will be migrated with only Keep Allowed and Keep Blocked remediations which are the defaults.
>
> Real Time Processing shall be migrated using the new Workflow created from that specific storage

Troubleshooting

Connection Issues

```
# Verify PostgreSQL connectivity
psql -h your-host -p 5432 -U mdss -d MDSS -c "SELECT version();"
```

Rollback Procedure on Linux

If you need to rollback after upgrade:

1. **Set** in `customer.env`: `ENV_IMAGE_TAG=stable3.6.1`
2. **Run the update command: ****sudo mdss -u update**

3. **Verify connectivity** to your original MongoDB database
4. **Confirm functionality** and data integrity

In order to perform the rollback on an offline deployment you need to [re]deploy the docker offline toolkit for version 3.6.1.

Rollback Procedure on Windows

If you need to rollback on a Windows deployment:

1. Stop MDSS
2. Backup the MongoDB data folder: `C:\Program Files\OPSWAT\MetaDefender Storage Security\data\mongo` - preferably before upgrading to v4.0.0 but this folder should be kept even after an upgrade
3. Uninstall MDSS 4.0.0
4. Install MDSS 3.6.1
5. Stop MDSS [after the installation is complete it start automatically]
6. Replace the MongoDB data folder with the previously created backup.
7. Start MDSS

Running the data migration manually

In order to migrate the data from an older MongoDB database and overwrite any existing data in the new PostgreSQL database:

1. Clear the **SKIP_MONGO_TO_PG** variable from the `.env` and `customer.env` files and from the system environment variables [Windows only]
2. Delete the PostgreSQL data docker volume `mdss_postgres_data` [Linux only]: `docker volume rm`mdss_postgres_data``
3. Restart MDSS v4.0.0

Info

💡 Notes:

- The migration process doesn't modify your original MongoDB database, so rollback is straightforward by reverting to the previous MDSS version.
- Modifications performed after the upgrade will not be kept after the rollback.
- Rollback is only supported to version 3.6.1.

Warning

> The only backup of the original MongoDB database is in the original docker volume called **mdss_****mongo*_db. Uninstalling MDSS or performing any cleanup operations will delete this volume permanetly.__

Database Vacuum

Over time, the MDSS database accumulates index bloat and heap fragmentation as a result of deleted rows — for example, after a retention policy run or a large scan cleanup job. The `mdss -u vacuum` command provides two maintenance modes to address these issues.

Normal mode -

```
sudo mdss -u vacuum
```

Rebuilds bloated indexes with no service interruption. All MDSS services remain running throughout.

What it does:

- Runs `VACUUM ANALYZE` on all tables — clears dead rows and updates query planner statistics
- Runs `REINDEX TABLE CONCURRENTLY` on `files`, `object_identity`, and `scan_results` — rebuilds indexes from scratch without locking reads or writes

What you get: Query performance is restored. Index sizes typically shrink by 55–70% after a large deletion.

What you don't get: Heap disk space is not reclaimed. Table file sizes on disk remain unchanged — this is expected PostgreSQL behavior. Only `VACUUM FULL` can physically rewrite and shrink the heap file.

Full mode -

```
sudo mdss -u vacuum -f
```

> **Warning**
>
> The full vacuum requires a service outage

What it does:

1. Stops all MDSS services (`mdss stop`)

2. Runs `VACUUM FULL ANALYZE` — rewrites every table into a compact heap file; all dead-row pages are fully eliminated
3. Runs `REINDEX DATABASE` — rebuilds every index from the freshly compacted heap
4. Restarts all MDSS services (`mdss start`)

> **External PostgreSQL:** For deployments using an external PostgreSQL server, no temporary container is used. Services are still stopped before and restarted after the operation.

What you get: Both index bloat and heap fragmentation are resolved. Disk space is returned to the OS. Table sizes drop in proportion to the number of live rows remaining.

What you don't get: There is no way to run this without a service interruption. If downtime is not acceptable, run normal mode instead — query performance will be restored, but disk space will not be reclaimed until a maintenance window is available.

Planning downtime for full vacuum

The duration depends primarily on the size of the PostgreSQL database on disk. Because services are stopped before the operation runs, there is no lock contention and the work proceeds at full I/O speed.

DB size	Vacuum full	Service stop/start	Total downtime estimate
<1 GB	~1 - 2 min	~2 - 5 min	~3 - 7 min
~5 GB	~2 - 5 min	~2 - 5 min	~4 - 10 min
~20 GB	~5 - 10 min	~2 - 5 min	~7 - 15 min
~100 GB	~30 - 40 min	~2 - 5 min	~35 - 45 min
500 GB+	> 3 h	~2 - 5 min	> 3 h

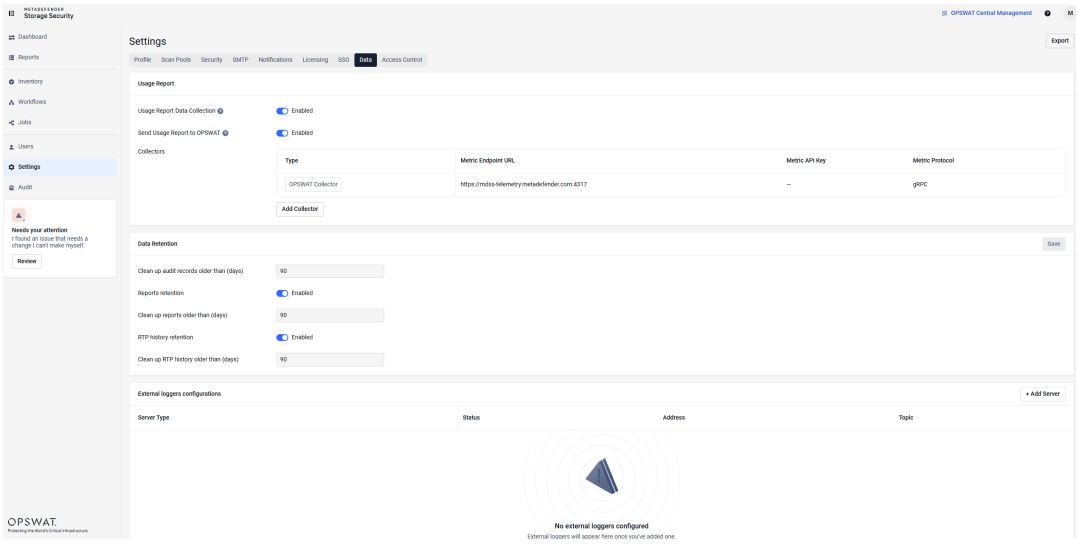
> **Info**
>
> On HDD storage, network-attached storage, or a heavily loaded host, the VACUUM FULL + REINDEX duration will be significantly longer.

Database Retention

To effectively manage data volume, MDSS provides the ability to clean up specific types of information after a defined period.

MDSS focuses on certain types of information that tend to grow significantly over time.

- Audit records can be automatically cleaned after a configurable number of days.
- Report records can be automatically cleaned after a configurable number of days.
- Real-time processing history can also be cleaned after a configurable number of days.



File Discovery Debounce

The MDSS file discovery system uses a debounce mechanism to determine when files are stable and ready for processing. Several key environment variables control this behavior, working together to create a consistent file discovery experience while accommodating the unique characteristics of different storage types.

Environment Variables

Global Debounce Interval

`FILE_DISCOVERY_DEBOUNCE_INTERVAL=0`

This variable applies to real-time processing only. It defines the minimum time that must elapse before a previously processed file is considered "new" again after being edited. The maximum is 3600 seconds.

- **Default value**: 0 seconds (files are considered new after any edit)
- **Purpose**: Prevents duplicate processing of the same file when multiple rapid edits occur
- **Global setting**: Applies to all storage types in real-time processing, and caps the storage-specific values below

Storage-Specific Debounce Times

`SFTP_DISCOVERY_DEBOUNCE_TIME_SECONDS=5`

`NFS_DISCOVERY_DEBOUNCE_TIME_SECONDS=5`

`SMBSERVICE_DISCOVERY_DEBOUNCE_TIME=5`

These variables define how long the system waits to confirm a file is fully uploaded and stable before processing it. The system checks if the file has been modified within this time period. If no modifications are detected during this interval, the file is considered ready for processing.

- **Default value**: 5 seconds
- **Purpose**: Prevents processing of partially uploaded files

Interaction Between Settings

The interaction between these settings follows this rule:

Storage-specific debounce values must be lower than or equal to the global `FILE_DISCOVERY_DEBOUNCE_INTERVAL`. If a storage-specific value is set to be bigger than

the global interval, the system will automatically use the global `FILE_DISCOVERY_DEBOUNCE_INTERVAL` value instead. This keeps every storage-specific value at or below the global interval. However, if the global `FILE_DISCOVERY_DEBOUNCE_INTERVAL` is set to 0, the system will honor the storage-specific values, allowing for customized debounce intervals per storage type.

For SMB units, the default SMB service skips a file that was modified less than `SMBSERVICE_DISCOVERY_DEBOUNCE_TIME` seconds ago and picks it up at the next discovery. On Windows deployments it also skips a file that another client still holds open for writing. On Linux deployments MDSS cannot reliably detect such a file, so the time window is the main protection against scanning a partially copied file: increase it when files are copied to the share by slow clients, or set it to `0` to process files as soon as they are found.

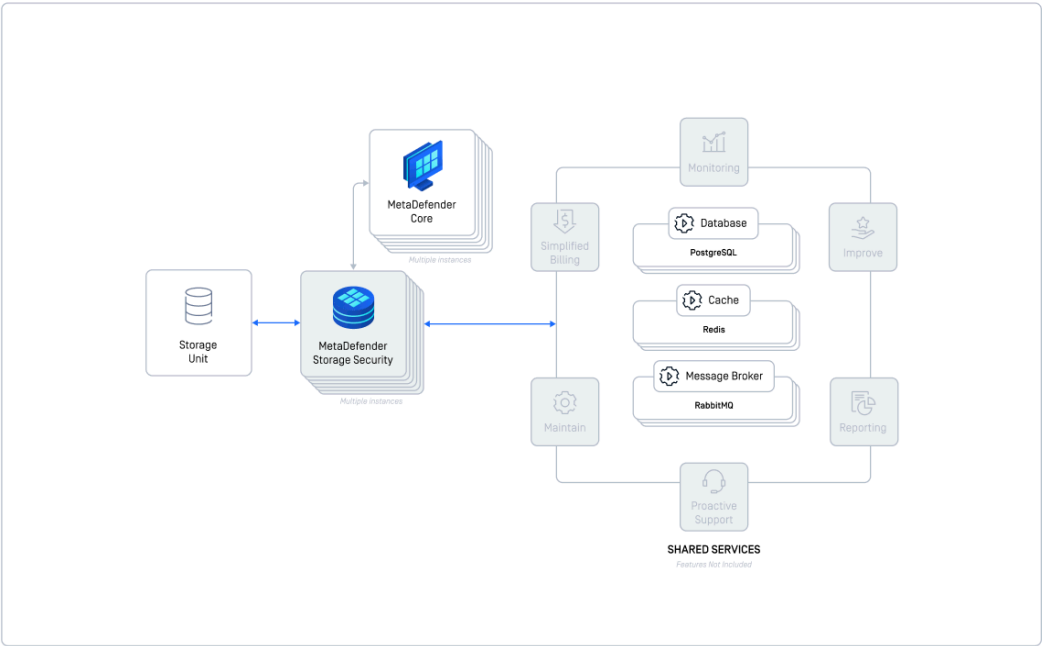
The legacy SMB service (the `smb_legacy` module on Linux, or `USE_OPSWAT_SMB_SERVICE=false` on Windows) works differently: by default it skips a file that another client still holds open, and uses `SMBSERVICE_DISCOVERY_DEBOUNCE_TIME` only when `SMBSERVICE_FILE_READY_CHECK_MODE=0` is set.

Active-Active configuration

This guide explains how to configure an active-active deployment of MDSS where multiple instances can use the same shared external services for improved performance and scalability.

Overview

In an active-active deployment, multiple MDSS instances work together using shared external components (RabbitMQ, PostgreSql, and Redis) to distribute workload, add redundancy and improve throughput. This configuration is particularly useful when handling large volumes of files.



Active-Active Configuration Behavior

In an active-active configuration, file processing operations are distributed dynamically across available instances:

- **File Discovery** - any available MDSS instance can detect and identify files requiring processing
- **File Scanning** - any available MDSS instance can perform the security scanning operation
- **Remediation** - any available MDSS instance can execute the remediation actions

Key Points

- Operations may be handled by different instances (distributed processing) or by the same instance (consolidated processing)
- The actual distribution depends on which instance processes the message first
- This design ensures optimal load balancing and high availability across your deployment

This approach maximizes system efficiency while maintaining consistent security coverage regardless of which specific instances handle each operation.

External Services Deployment Options

External services are required for this type of deployment, a deployment and configuration guide is available here: External Services Configuration

Alternative: Active-Passive Failover

Active-active is not the only way to add redundancy. If you would rather keep a single instance processing at a time and switch over only when it becomes unavailable, an active-passive setup can be automated with the following reference script:

OPSWAT/mdss-active-passive-failover

The script continuously monitors the active instance's web interface health and, after a configurable number of consecutive failed health checks, automatically triggers a scan on the passive instance. It re-arms itself once the active instance recovers again.

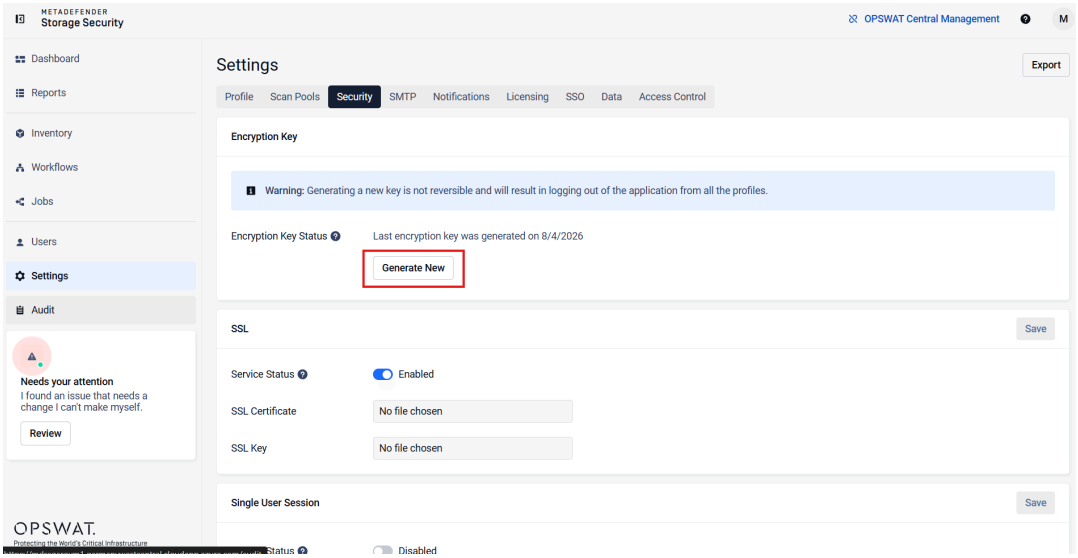
All settings are supplied as `PASSIVE_MODE_*` environment variables. Required: the active and passive URLs and ports, an MDSS API key, the ID of the storage to scan (one storage), the workflow ID and a name for the scan it starts. Optional: the scan priority, the request timeout, and the health check interval and failure threshold. Prerequisites, configuration and usage are documented in the repository's README.

> **Note:**
>
> This script is provided as a reference implementation to be adapted to your own environment and is not covered by OPSWAT product support.

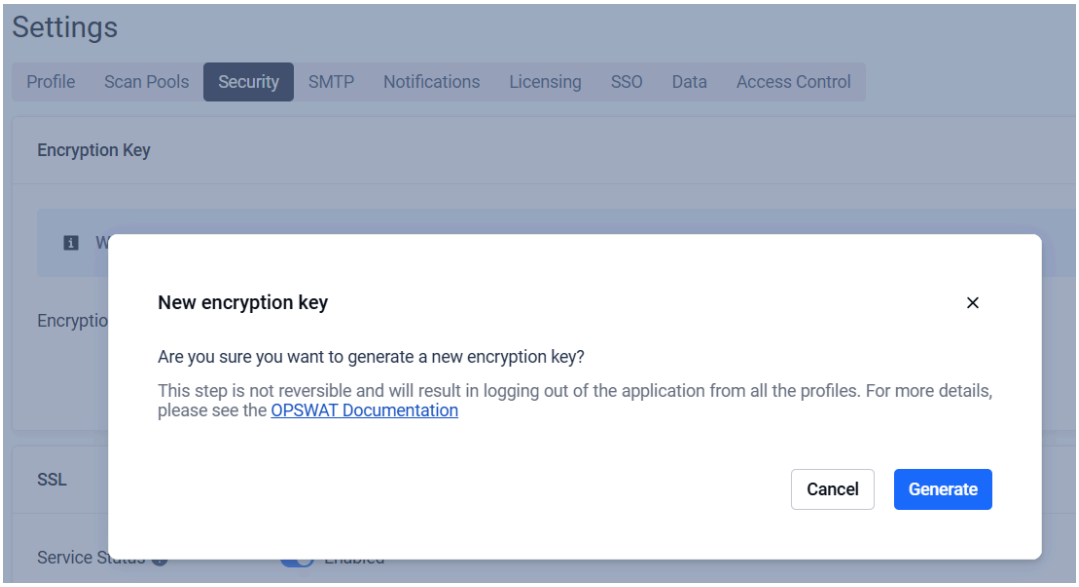
Security & Encryption Keys

To enhance your system's security, you can generate a new master encryption key. This action will re-encrypt all sensitive data—such as storage credentials, API keys, and others—and securely retire the previous key.

You can start this process by navigating to **Settings** and selecting the **Security** tab. Here, click the **Generate New **button from the **Encryption Key** section.



A confirmation dialog will appear to ensure you understand the consequences. Please note that this action is irreversible and will immediately log out all active users from the application. To proceed, click **Generate**.



While the key rotation is in progress, the system prioritizes data integrity. To prevent data corruption, any operations that access or modify encrypted information will be temporarily

unavailable. For example, you will not be able to add or edit storage configurations or user accounts during this time. The process is designed to be as fast as possible, but the total duration will depend on the amount of data stored in your system. We recommend performing this critical security operation during a planned maintenance window or a period of low system activity to minimize disruption for your users.

My OPSWAT Central Management

Central Management is a centralized console that lets you manage multiple MetaDefender products & instances of the same product from a single platform. Customers get improved data flows visibility and provide better reports for each product and entire account. It also provides critical information like product instance health, license, and connection status, report history, license, and audit trails.

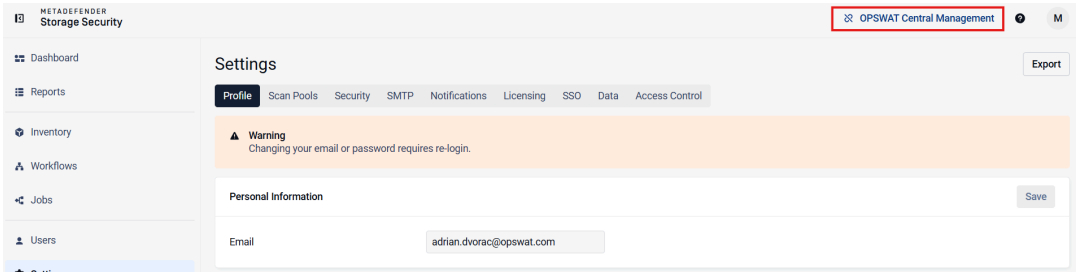
This scope of this topic is to describe how to enroll into MetaDefender Storage Security into My OPSWAT Central Management.

> **Info**
>
> MetaDefender Storage Security currently supports integration with Central Management v8

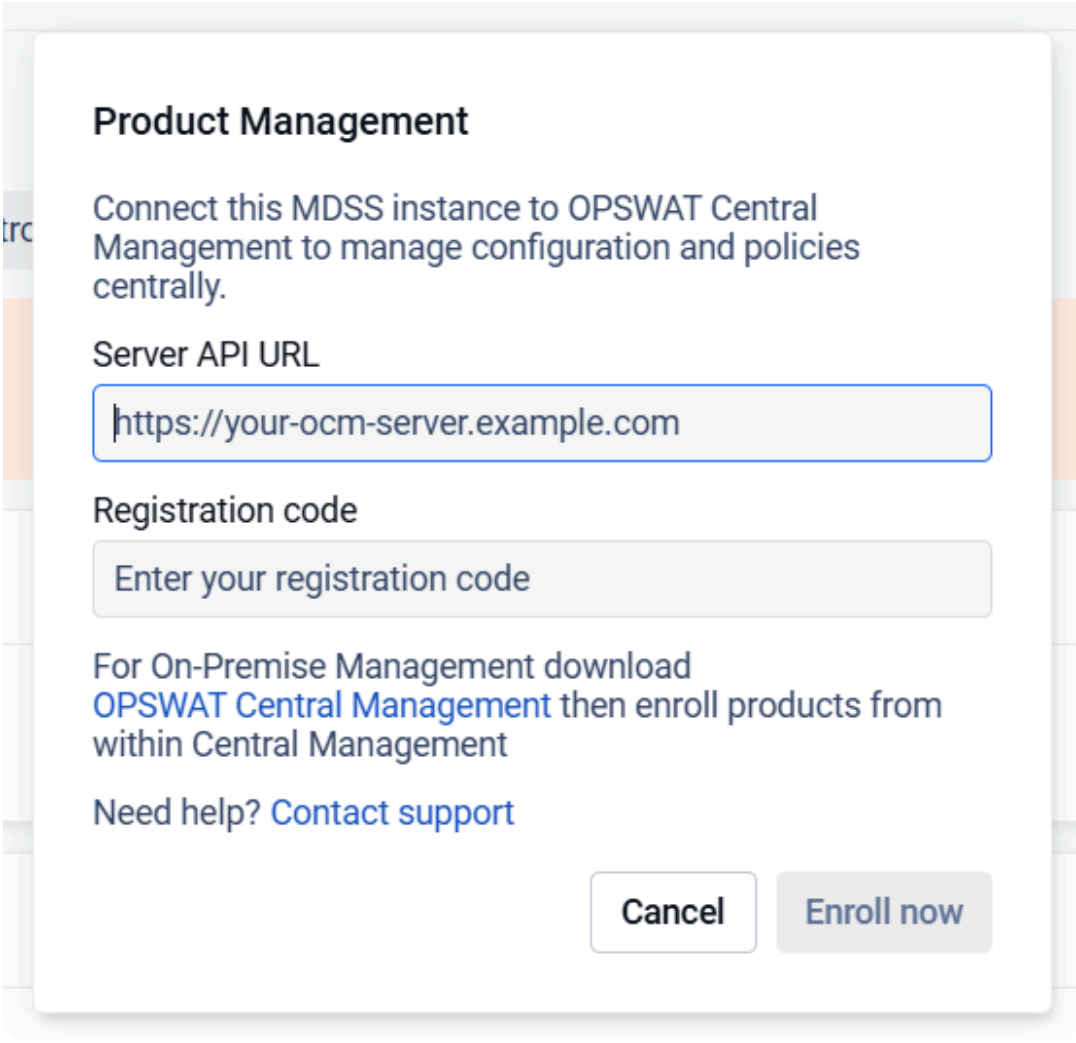
Enrolling the product to Central Management

To enroll MetaDefender Storage Security into My OPSWAT Central Management, these steps should be followed:

1. In the top-right corner, next to the username, click the **OPSWAT Central Management** button. The button keeps this label after enrollment; only its icon changes to show that the product is connected. Users without permission to change the enrollment see the label but cannot click it.

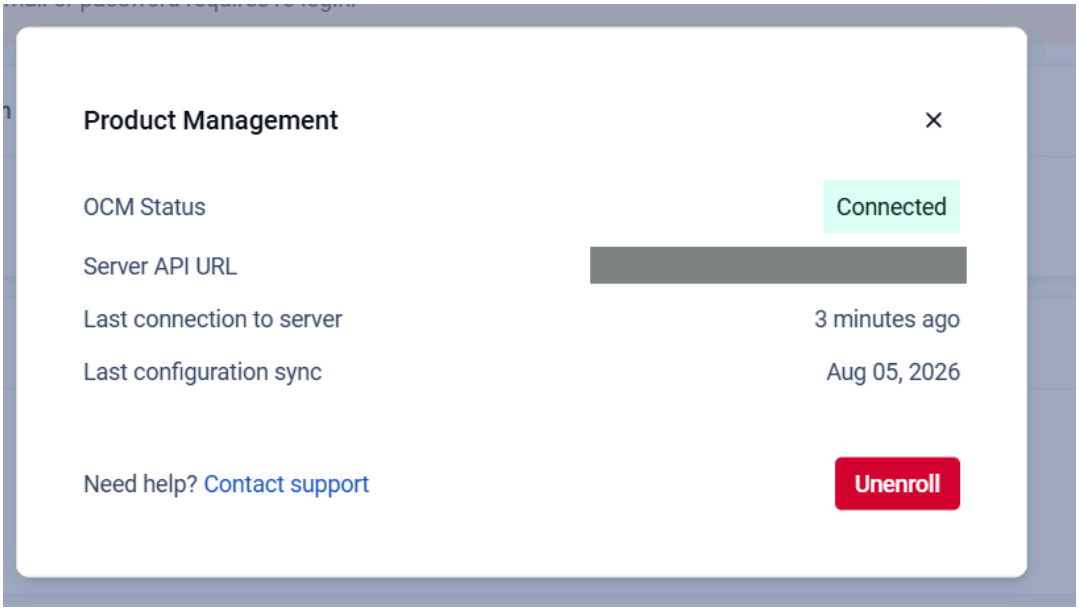


2. Clicking the button will open a form containing the necessary fields to enroll. These are **Server URL**, which is the address of the Central Management server to which you want to enroll the product, and **Registration Code**, which can be found in the Central Management UI by navigating to **Settings -> Global**. The registration code will be present on that page.

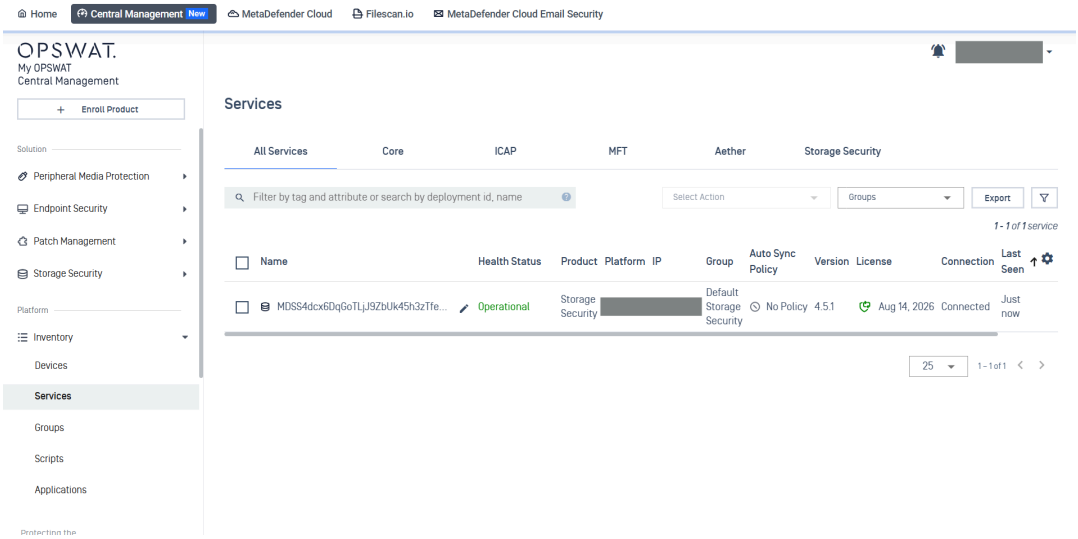


3. After completing the form, click **Enroll now**. If the information is correct, MDSS should display a message saying that enrollment was successful.

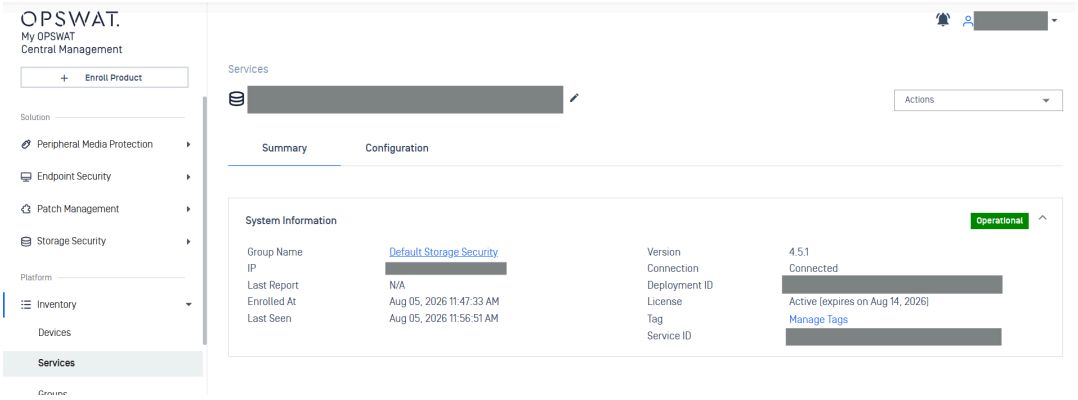
Once you have successfully enrolled your MetaDefender Storage Security instance, the button **OPSWAT Central Management** will reveal **Product Management** information about the connection to the **OCM** platform.

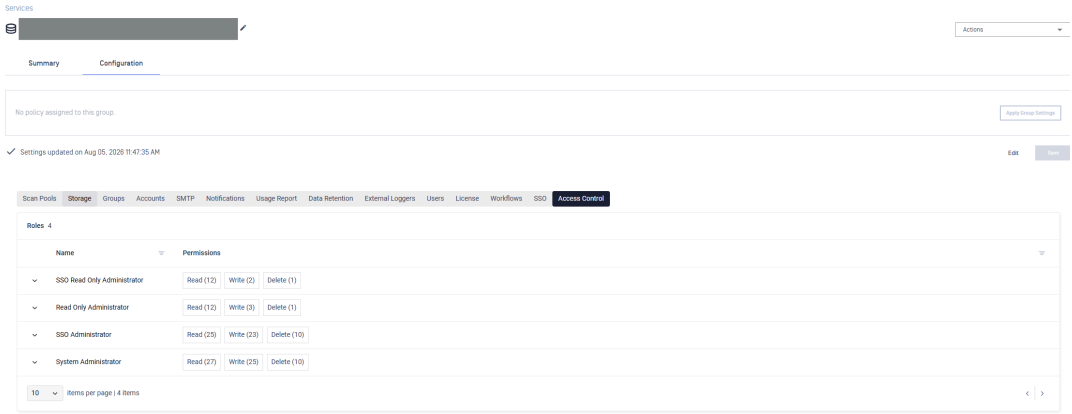


Additionally, navigating to the Central Management server, you should be able to see your enrolled instances under the **Inventory > Services** tab



Furthermore, clicking on the card above will take you to a page where more information about each instance can be seen:

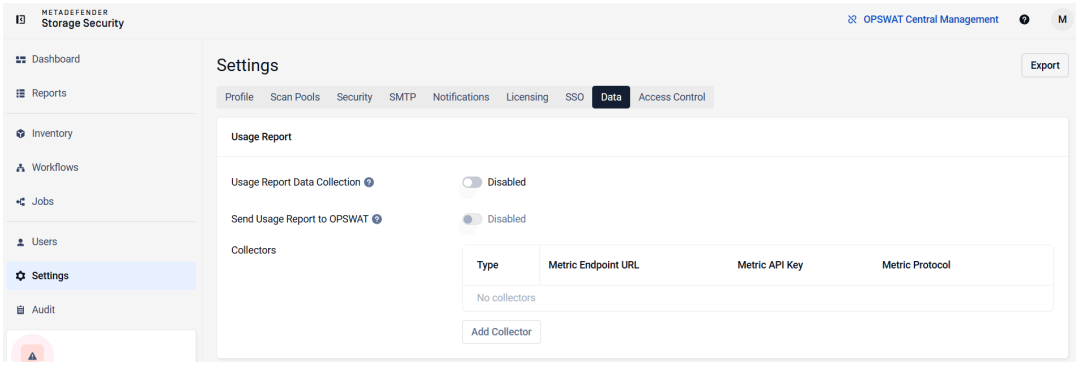



Services
[redacted]
Actions
Summary
Configuration
No policy assigned to this group.
Apply Group Settings
Settings updated on Aug 05, 2026 11:47:35 AM
Edit
Save
Scan Pools
Storage
Groups
Accounts
SMTP
Notifications
Usage Report
Data Retention
External Loggers
Users
License
Workflows
SSO
Access Control
Roles 4
Name
Permissions
SSO Read Only Administrator
Read (12)
Write (2)
Delete (1)
Read Only Administrator
Read (12)
Write (3)
Delete (1)
SSO Administrator
Read (25)
Write (23)
Delete (10)
System Administrator
Read (27)
Write (25)
Delete (10)
10
items per page | 4 items

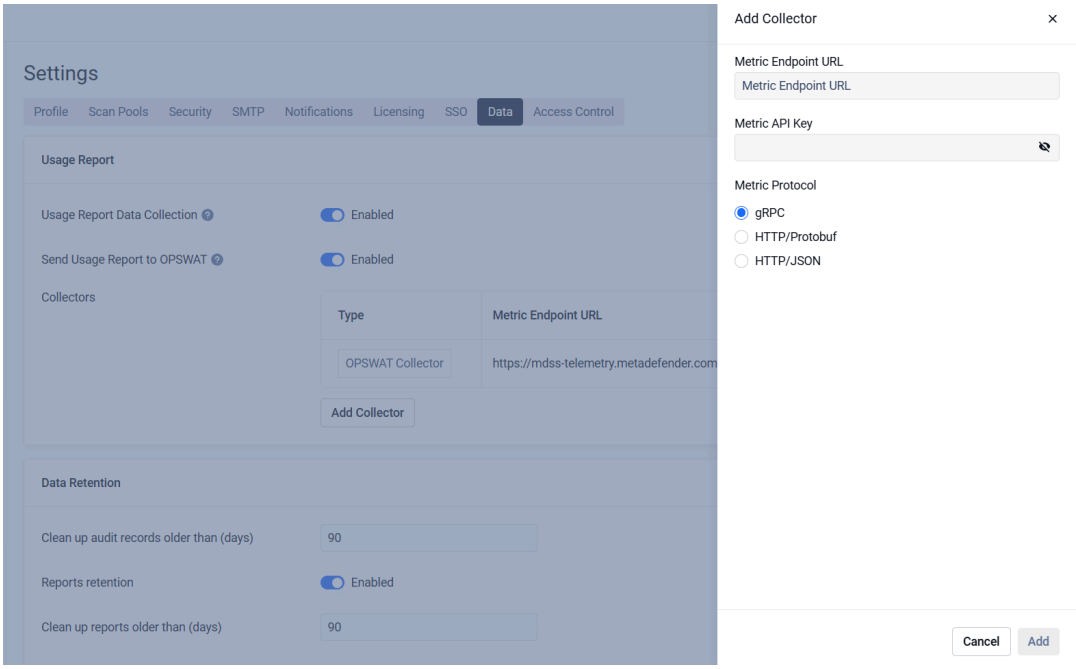
Telemetry

To configure the telemetry output and its collectors in MetaDefender Storage Security, navigate to the **Settings > Data **tab and switch the toggle **Usage Report Data Collection**.

Once enabled, the system will begin collecting operational data. At this time, only metrics are collected. Logs and traces are not yet included.



MetaDefender Storage Security supports adding custom collectors when the telemetry is active by providing the metrics URL and the protocol (HTTP(s) / gRPC)



Automatic Scaling (Autoscaler)

MetaDefender Storage Security (MDSS) continuously monitors its own workload and automatically adjusts how much data it processes at once, so scans keep moving during busy periods and resources are freed up when activity is low. This happens on its own, with no manual setup.

This page explains what automatic scaling does, the optional setting you can tune, what you can see on the built-in **System** dashboard, the recommendations it may show you, and how to turn it off if you need to.

Overview

The amount of work flowing through MDSS rises and falls throughout the day as files are discovered, scanned, and remediated. Automatic scaling (the "autoscaler") keeps performance steady across those changes.

Its goal is to process your files as fast as possible while keeping the system stable, by using the resources available to it as fully and efficiently as it can. Instead of running at a fixed pace, MDSS watches how busy the system is and adjusts how much work it processes in parallel. When there is work to do and capacity to spare, it raises throughput to get the most out of your hardware. When a resource approaches its limit, it settles at the highest rate that resource can sustain. When the backlog is cleared, it releases the resources it no longer needs.

Automatic scaling makes the resources you already have work as efficiently as possible. It does not add or remove servers on its own. When it detects that more capacity is needed, it tells you what to add through on-screen recommendations.

Automatic scaling is enabled by default and works out of the box. No setup is required.

How it works

Automatic scaling checks the system every few seconds and adjusts how much work MDSS takes on, so the available resources are used as fully as they can be without putting the system at risk.

MDSS pays attention to indicators such as:

- how busy the processors (CPU) and memory are,
- how large the backlog of waiting work is,
- how quickly scans are being completed,
- the health and responsiveness of connected MetaDefender Core scan instances, and
- available disk space.

Based on these, its behavior is deliberately smooth and cautious:

- **It raises throughput** when there is a backlog and spare capacity, using more of the available resources to clear work faster.
- **It settles at a sustainable level** when a resource comes under sustained pressure, running it at the highest rate it can sustain instead of overcommitting it and risking instability.
- **It throttles to protect the system** when a resource reaches a critical limit. This can happen when CPU or memory runs out of headroom, when the output pipeline cannot drain as fast as new work arrives, or when a connected MetaDefender Core scan instance is overloaded (its scan queue is nearly full) or is returning scan errors. In these situations MDSS reduces how much it takes on so that processing stays reliable and errors are avoided, then ramps back up once the pressure clears. An instance that only turns requests away because it is busy makes MDSS ease back gradually rather than throttle.
- **It rests when idle**, returning to a light baseline once the backlog is cleared so resources are free for other work.

Built-in safety guardrails

Separately from performance tuning, MDSS includes a safety guardrail that protects your data and the system. If free disk space becomes critically low, MDSS automatically **pauses** processing so the system does not run out of room, and **resumes** automatically once space is recovered. When this happens, it shows a message explaining what to do.

Desired load for MetaDefender Core instances

Every MetaDefender Core scan instance has a queue with a maximum size. **Desired load** is the share of that queue MDSS aims to keep filled on that instance.

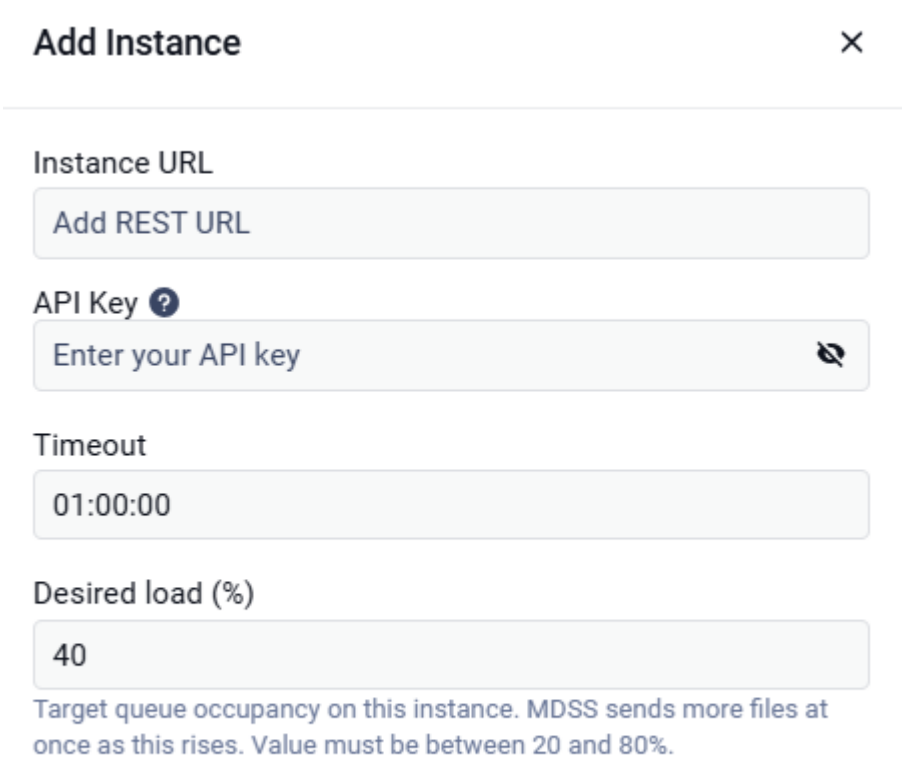
You do not need to set it. Automatic scaling works without it and the default suits most deployments. It exists so that a deployment whose MetaDefender Core instances differ in size can give each one an appropriate amount of work, instead of applying a single value everywhere.

Where to find it

Go to **Settings > Scan Pools**, open a pool that holds MetaDefender Core instances, then add or edit an instance. The field is **Desired load (%)**. The current value also appears in the **Desired load (%)** column of the pool's instance list.

It appears for MetaDefender Core instances only. MetaDefender Cloud and MetaDefender Core Cluster instances do not expose a queue maximum to aim at, so the setting does not

apply to them.



Accepted values

The value must be between **20 and 80**. The default is **40**, the value MDSS is tested against. The higher you set it, the more files MDSS sends to that instance at the same time.

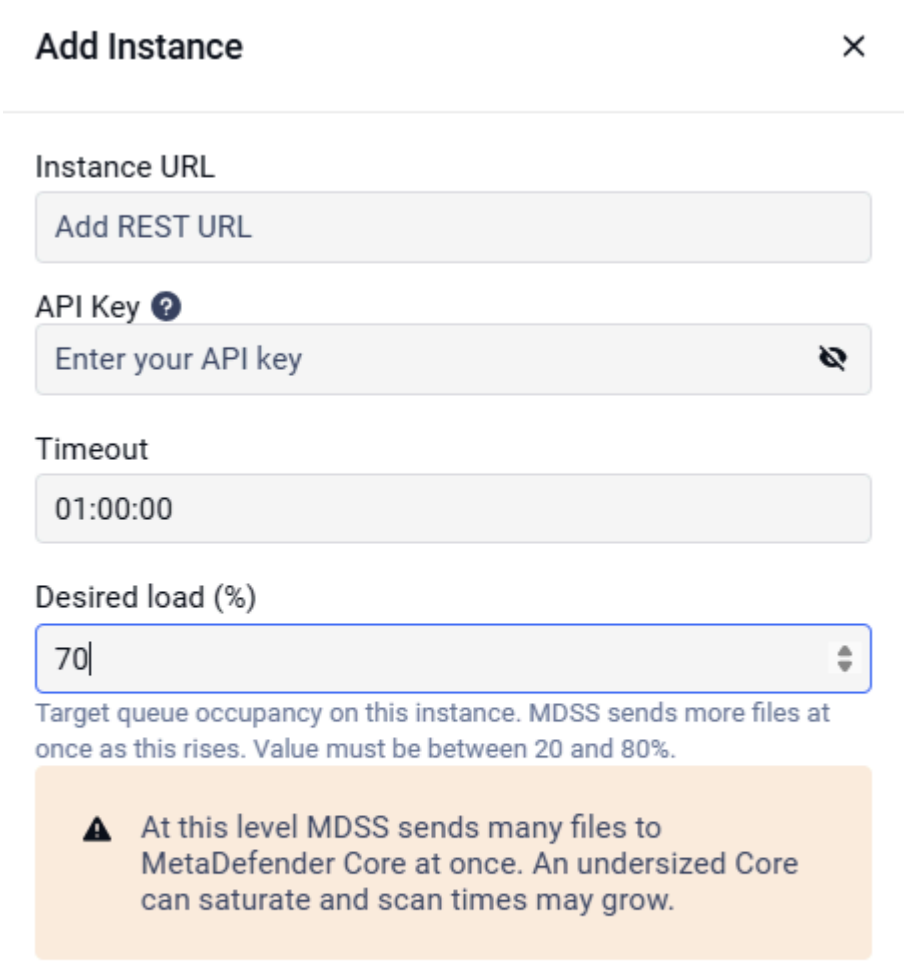
MDSS increases the rate at which it sends files only while each instance's queue is below its desired load by more than a small margin, 10 percentage points by default. The minimum of 20 keeps an idle instance outside that margin, so it never stops MDSS from ramping up. The maximum of 80 leaves room below 95% of the queue, the point at which MDSS treats the instance as critically loaded, so MDSS can ease off gradually instead of braking sharply.

A change takes effect within seconds, with no restart.

Choosing a value

Leave it at 40 unless you have a reason to change it.

Raise it when the instance has capacity to spare. Above 60 the product shows a caution: "At this level MDSS sends many files to MetaDefender Core at once. An undersized Core can saturate and scan times may grow."



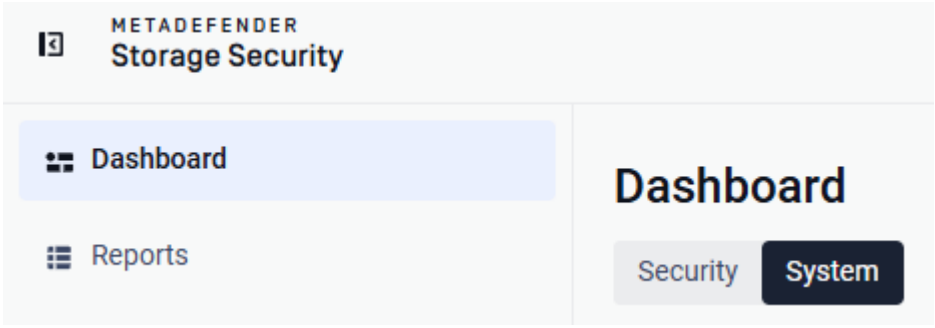
Lower it when the instance is struggling. MDSS uses the desired load to decide when to stop sending more, so a lower value makes it stop increasing sooner. If MDSS decides the desired load is too high for that instance, it says so on the **System** dashboard and recommends returning it to 40; see Recommendations you may see.



The System dashboard

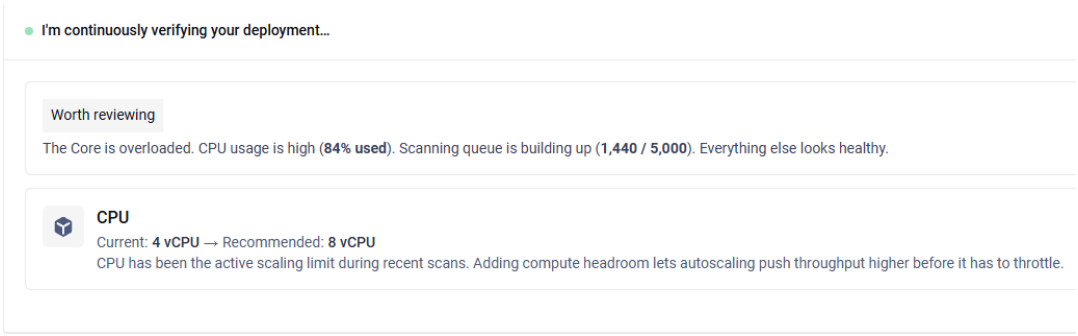
MDSS includes a **System** dashboard that shows, in real time, what automatic scaling is doing and why. It refreshes automatically every few seconds. The dashboard is for monitoring only; the system tunes itself, so there is nothing to configure here.

The dashboard appears as a **System** tab alongside the main **Dashboard** view, for users with the Administrator or SSO Administrator role.



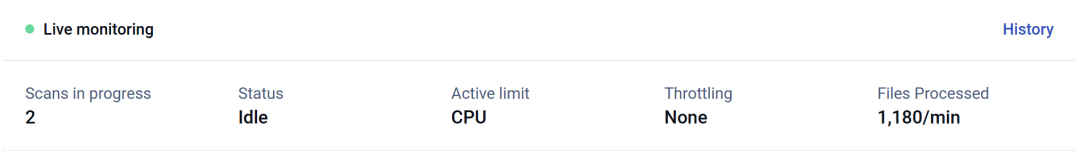
Autoscaling insight

The top card summarizes, in a single sentence, what MDSS is doing right now. Examples include *"I'm scaling up to keep scanning moving while you review this,"* and *"I'm continuously verifying your deployment."* When there is something to address, it explains the cause in plain language and marks how urgent it is **[Action required**, **Recommended**, or **Worth reviewing**]. When everything is healthy, it confirms that no action is needed.



Live status

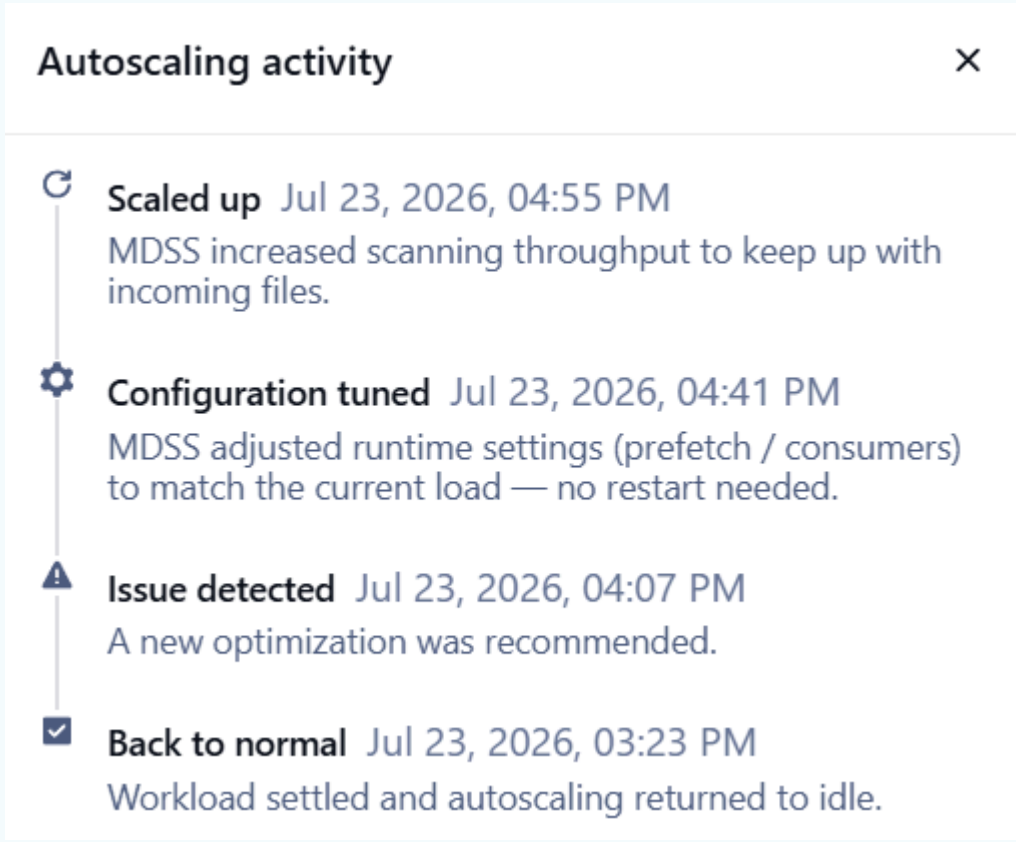
A strip of key figures gives a quick summary: how many scans are in progress, the current status [such as **Scaling up** or **Idle**], what is currently limiting performance [if anything], and how many files are being processed per minute. A **History** button opens a timeline of recent activity.



Autoscaling activity

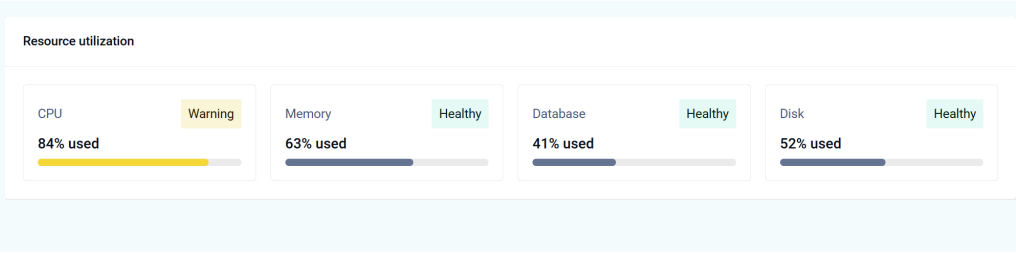
A running timeline of notable events, such as speeding up, slowing down, or pausing. Each event is described in plain language, for example *"MDSS increased scanning*

throughput to keep up with incoming files."



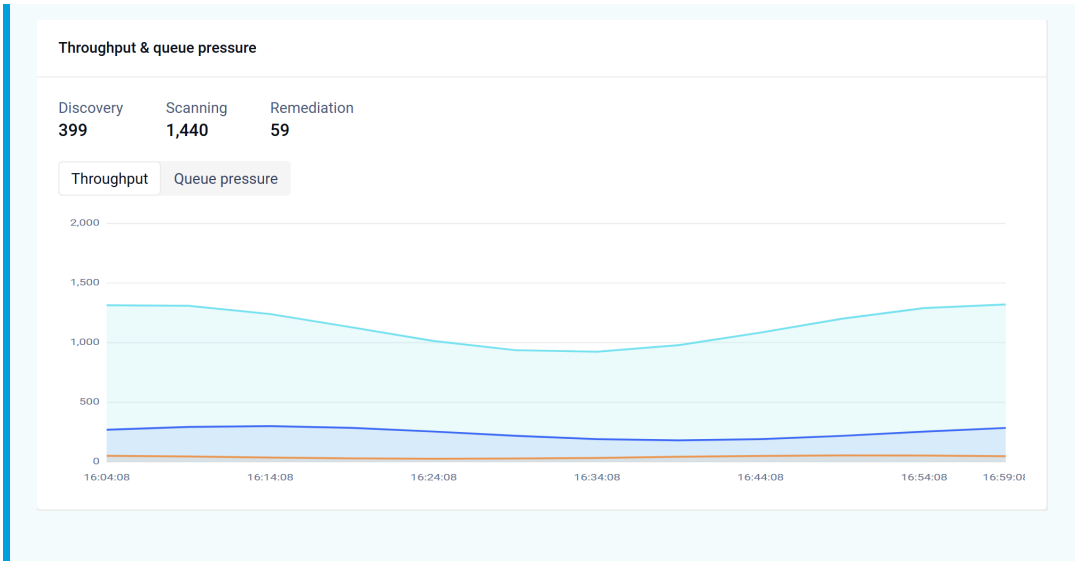
Resource utilization

Cards show current usage of CPU, memory, disk, database, and cache, each with a healthy, warning, or critical indicator.



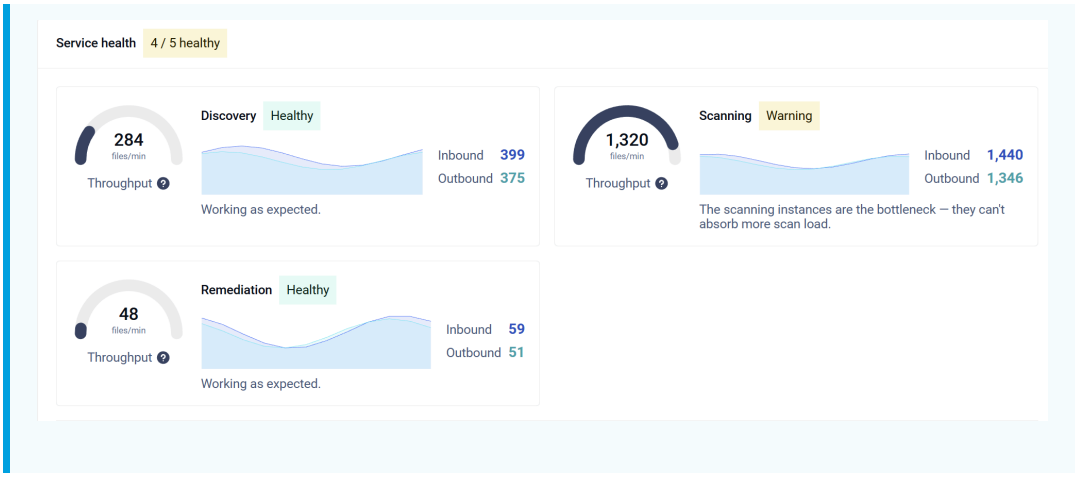
Throughput & queue pressure

Shows how much work is waiting at each stage and lets you switch between two live charts: **Throughput** (files processed per minute over time) and **Queue pressure** (backlog size over time). This makes it easy to see whether MDSS is keeping up with incoming work.



Service health

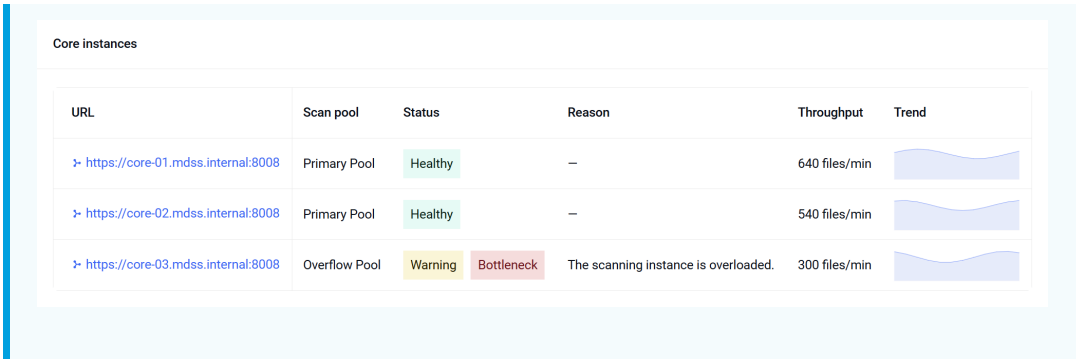
Shows the health of each MDSS service, with a summary of how many are healthy, each service's current processing rate, and a short explanation of any bottleneck.



MetaDefender Core instances

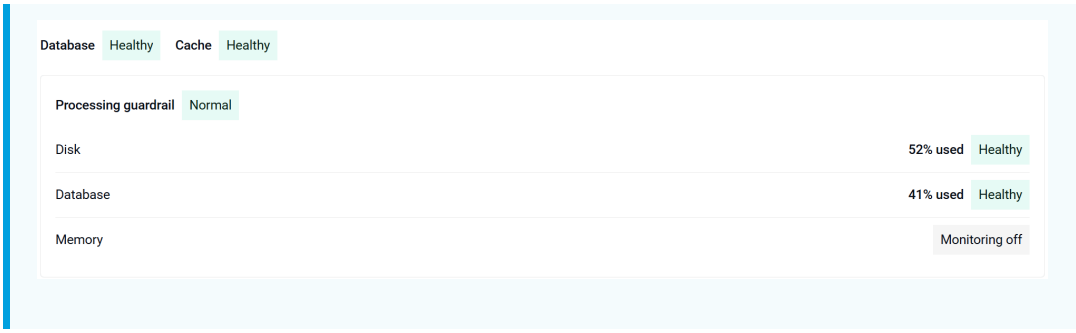
A table of the MetaDefender Core scan instances currently used by the active jobs, showing each instance's address, health status, and current processing rate. An instance that is holding back overall performance is flagged as a bottleneck.

A **Reason** column explains why: the instance is overloaded, is returning scan errors, or is rate-limited. When the reason is scan errors or rate limiting and the instance's desired load is above the default of 40, the **Autoscaling insight** card at the top of the dashboard also names the instance and its desired load.



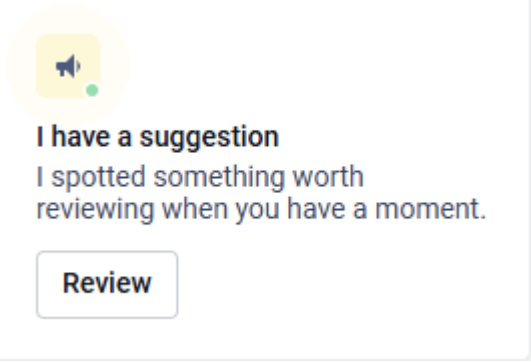
Processing status

When the safety guardrail is active, a card shows whether processing is **Normal** or **Paused**, along with the reason and the current disk space status. If processing is paused, MDSS also shows a banner and a recommended action to resolve it.



Status indicators elsewhere

Beyond the System dashboard, a compact status card appears in the sidebar. If performance becomes degraded, a banner appears across the application with a link to the recommended actions, so you are kept informed even when you are not viewing the dashboard.



Autoscaling states

The **Status** on the dashboard reflects what automatic scaling is doing at that moment:

State	Meaning
Idle	Little or no work to do. MDSS is running at a light baseline to conserve resources.
Scaling up	Demand is high and resources are available, so MDSS is increasing how much it processes at once.
Holding	Conditions are near a limit, so MDSS is holding steady rather than changing course.
Scaling down	A resource is under sustained pressure, so MDSS is easing back to relieve it.
Throttling	A resource has reached a critical limit, so MDSS is temporarily processing less at once to keep things reliable and avoid errors. It ramps back up automatically once the pressure clears. See the **How it works** section for the situations that trigger this.
Paused	The safety guardrail has paused processing to avoid running out of disk space. Processing resumes automatically once space is recovered.

Recommendations you may see

When MDSS determines that your hardware or a supporting component has become the limiting factor, it shows a single, prioritized recommendation. Recommendations are advisory. MDSS keeps running as efficiently as it can, and acting on a recommendation should improve performance or resolve a constraint.

Recommendation	Why you are seeing it	What to do
Increase specs	The MDSS host is below the minimum recommended hardware.	Bring the host up to the minimum recommended CPU and memory.
Increase CPU	Performance is being held back by available CPU.	Add CPU to the host, or add an additional MDSS instance to share the load.
Increase memory	Performance is being held back by available memory.	Add memory to the host, or add an additional MDSS instance.
Use an external database	The database is limiting throughput or growing large on the local host.	Move the database to a dedicated external host.
Use an external cache	The local cache is limiting throughput.	Move the cache to a dedicated external host.
Use an external MetaDefender Core	The local MetaDefender Core is limiting scan throughput.	Switch to an external MetaDefender Core so scanning can scale independently.
Increase MetaDefender Core performance	Your external MetaDefender Core is at capacity.	Increase its resources or add more Core nodes.
Lower the target load on this MetaDefender Core instance	A MetaDefender Core instance is returning scan errors or turning requests away, and its desired load is set above the default.	Set that instance's desired load back to 40.
Increase storage	Processing was paused because disk space or database capacity is critically low and the database runs on an external host. With a local database, MDSS shows **Use an external database** instead.	Free up or add disk space to resume processing.

> MDSS shows the single most important recommendation at any time rather than a long list, so you can focus on the change that will have the greatest impact.

Turning automatic scaling off

Automatic scaling is enabled by default, and we recommend leaving it on. If you need to turn it off, you can do so with an environment variable:

```
SCANNING_AUTOSCALER_ENABLED=false
```

With this set, MDSS stops adjusting how much it processes and runs at a steady baseline instead. It continues to monitor performance and show the dashboard. Restart MDSS for the change to take effect.

> **Note:** The safety guardrail that pauses processing when disk space is critically low stays active regardless of this setting, so the system stays protected even when automatic scaling is turned off.

Frequently asked questions

Does automatic scaling add or remove servers? No. It makes your existing deployment work as efficiently as possible. When more capacity is needed, it recommends what to add rather than provisioning it for you.

Does automatic scaling slow down my scans? No. It works to complete your scans as fast as possible by using the available system resources as fully and efficiently as it safely can. If a resource reaches a critical limit, MDSS temporarily processes less at once (throttling). This keeps scanning reliable and prevents the system from being overloaded, which would otherwise cause errors or instability. Once the pressure clears, MDSS ramps back up.

Do I need to configure anything to use it? No. It is enabled by default and self-tuning. There is one optional value, the desired load for each MetaDefender Core scan instance, which has a working default. See Desired load for MetaDefender Core instances.

Why does processing sometimes pause? The safety guardrail pauses processing when disk space becomes critically low, and resumes automatically once space is recovered. Follow the on-screen **Increase storage** recommendation to resolve it.

Why am I seeing a recommendation to add hardware or move a component externally? Because MDSS has determined that resource is currently limiting performance.

Automatic scaling settings

MetaDefender Storage Security (MDSS) adjusts how much it processes automatically, based on how busy the system is. This page lists the settings that control that behavior, with their default values and a short explanation of each one. For an overview of how automatic scaling works, see the Automatic Scaling page.

Altering these settings is optional. They are provided for cases where you want to adapt the behavior to a specific environment. Each setting is applied as an environment variable, in the same place as the other MDSS environment variables: the `.env` file for Docker Compose and Windows installations, or the Helm values for Kubernetes (see Deployment Using Helm). MDSS reads them when the service starts, so restart MDSS after making a change. The settings that control scaling apply to the scanning part of MDSS, which is why their names begin with `SCANNING_AUTOSCALER_`.

The desired load for a MetaDefender Core scan instance is not an environment variable. It is set per instance in the product; see Desired load for MetaDefender Core instances on the Automatic Scaling page.

---

1. General behavior

The prefetch settings apply per scanning service instance: the autoscaler keeps one prefetch value and applies it to every consumer of that service. On a single-node deployment the number of files in flight is therefore at most the current prefetch; with several scanning service instances it is at most the prefetch multiplied by their number.

Setting	Default	Unit	What it does
SCANNING_AUTOSCALER_ENABLED	true	true/false	Turns automatic scaling on or off. When off, MDSS still monitors performance and shows the dashboard, but it no longer adjusts how much it processes by itself.
SCANNING_AUTOSCALER_PROBE_INTERVAL	10	seconds	How often MDSS checks the system and decides whether to adjust.
SCANNING_AUTOSCALER_INITIAL_PREFETCH	20	items	How many items each scanning service instance starts with, and the level it returns to when there is no work.
SCANNING_AUTOSCALER_MIN_PREFETCH	5	items	The fewest items a scanning service instance will handle at once when scaling down.
SCANNING_AUTOSCALER_MAX_PREFETCH	200	items	The most items a scanning service instance will handle at once when scaling up. This is the hard ceiling on files in flight per instance. MDSS will not go above 200 by default, and never above an internal maximum of 400 even if you set a higher number.
SCANNING_AUTOSCALER_CONCURRENCY_LIMIT	unset	items	An optional number of files in flight on a scanning service instance at which MDSS stops increasing its

			throughput. It only blocks further scale-up and never reduces the load on its own. Leave it unset unless OPSWAT support asks for a value: with no value, MDSS relies on the resource limits below to decide when to stop increasing.
SCANNING_AUTOSCALER_IDLE_RESET_TICKS	30	checks	How many quiet checks in a row before MDSS returns to its starting level. At the default check interval this is about five minutes.
SCANNING_AUTOSCALER_THROTTLE_DURATION	30	seconds	After MDSS reaches a critical limit and halves its throughput, how long it waits before checking whether the pressure has cleared. If it has not, the wait starts again; if it has, MDSS restores part of the previous throughput and then ramps up normally.
SCANNING_AUTOSCALER_PROBES_COUNT_SCALEUP	5	checks	How many healthy checks in a row are needed before MDSS speeds up. A higher number makes it ramp up more cautiously.
SCANNING_AUTOSCALER_PROBES_COUNT_SCALEDOWN	2	checks	How many warning checks in a row before MDSS eases back. A lower

			number makes it back off sooner.
SCANNING_AUTOSCALER_LOCAL_HARDWARE_HYSTERESIS	5.0	percentage points	A small buffer around the CPU and memory limits that keeps MDSS from switching back and forth when usage sits right at a limit.
SCANNING_AUTOSCALER_OUTBOUND_HYSTERESIS	5.0	percentage points	The same kind of buffer for the queue of scanned results waiting for the next step.
SCANNING_AUTOSCALER_DOWNSTREAM_HYSTERESIS	10.0	percentage points	The same kind of buffer for how loaded the scan instances are.

2. Resource limits

These set the points at which a resource is considered under pressure. When a resource passes its warning level, MDSS starts to ease back gradually. When it passes its critical level, MDSS reduces its load right away.

Setting	Default	Unit	What it does
SCANNING_AUTOSCALER_CPU_WARNING_THRESHOLD	90	percent	CPU usage that MDSS treats as a warning.
SCANNING_AUTOSCALER_CPU_CRITICAL_THRESHOLD	95	percent	CPU usage that makes MDSS reduce its load immediately.
SCANNING_AUTOSCALER_MEMORY_WARNING_THRESHOLD	80	percent	Memory usage that MDSS treats as a warning.
SCANNING_AUTOSCALER_MEMORY_CRITICAL_THRESHOLD	90	percent	Memory usage that makes MDSS reduce its load immediately.
SCANNING_AUTOSCALER_OUTBOUND_WARNING_THRESHOLD	70	percent of the limit below	How full the results queue can get, as a percentage of the limit below, before MDSS treats it as a warning.
SCANNING_AUTOSCALER_OUTBOUND_CRITICAL_THRESHOLD	9000	items waiting	The number of scanned results waiting across the outbound queues that makes MDSS reduce its load immediately. The warning

			level above is a percentage of this number.

The queue of incoming work has no limit to set. Any waiting work simply tells MDSS there is something to process.

---

3. Scan instance settings

These control how MDSS reads the load and health of the connected MetaDefender Core scan instances and how much work it sends to each one.

Upgrade note: After upgrading to 4.6.0, every existing MetaDefender Core scan instance starts at the default desired load of `40`. If you had set `SCANNING_AUTOSCALER_DOWNSTREAM_DESIRED_LOAD` to another value, apply that value to each instance under **Settings > Scan Pools**.

Setting	Default	Unit	What it does
SCANNING_AUTOSCALER_DOWNSTREAM_DESIRED_LOAD	40	percent	Fallback desired load for a MetaDefender Core scan instance, as a percentage of what that instance can hold. Every instance carries its own **Desired load** value (default 40, accepts 20 to 80) and that is what MDSS uses; this variable is used only when the scanning service cannot read an instance's own value. It is not a way to configure instances in 4.6.0.
SCANNING_AUTOSCALER_DOWNSTREAM_WINDOW_SIZE	100	recent results	How many recent results MDSS looks at when judging how loaded a scan instance is.
SCANNING_AUTOSCALER_DOWNSTREAM_CRITICAL_ERROR_RATE_THRESHOLD	0.10	share (0 to 1)	The share of recent requests that fail before MDSS treats a scan instance as

			unhealthy and pulls work away from it. The default of `0.10` means 10 percent.
SCANNING_AUTOSCALER_DOWNSTREAM_CRITICAL_ERROR_RATE_WINDOW_SIZE	`100`	recent results	How many recent results are used for the failure-rate check above.
SCANNING_AUTOSCALER_DOWNSTREAM_WARNING_429_RATE_THRESHOLD	`0.10`	share (0 to 1)	The share of recent requests a scan instance turns away for being too busy before MDSS treats it as a warning. The default of `0.10` means 10 percent.
SCANNING_AUTOSCALER_DOWNSTREAM_WARNING_429_RATE_WINDOW_SIZE	`100`	recent results	How many recent results are used for the too-busy

			check above.

4. Disk space protection

MDSS includes a safety guardrail that pauses processing when disk space runs low and resumes it once space is recovered. This protects your data and keeps working even when automatic scaling is turned off.

Setting	Default	Unit	What it does
GUARDRAILS_CHECK_INTERVAL_SECONDS	30	seconds	How often MDSS checks available disk space. The lowest allowed value is 10.
GUARDRAILS_USED_CRITICAL_PERCENT	90	percent	How full the disk can get before MDSS pauses processing to avoid running out of space.
GUARDRAILS_USED_RESUME_PERCENT	80	percent	How far disk usage must drop before MDSS resumes processing. The gap below the pause level prevents rapid pausing and resuming.
GUARDRAILS_DISK_MONITOR_PATH	/var/log/mdss (Linux), install drive (Windows)	path	The disk location MDSS watches for free space.
SCAN_INSTANCE_DISK_CRITICAL_THRESHOLD_PERCENT	5	percent free	How little free disk space a

			scan instance can have before MDSS stops sending it work.
SCAN_INSTANCE_HEALTH_JOB_INTERVAL_SECONDS	30	seconds	How often MDSS checks the health of each scan instance.

Knowledge Base

The MDSS API Gateway - What It Is and Which Port It Uses

Applies to: MetaDefender Storage Security (MDSS) · All supported deployments (Docker on Linux, native services on Windows, Kubernetes)

MetaDefender Storage Security is built from a set of independent services — one that discovers files in storage, one that scans them, one that handles users and permissions, one that manages licensing, and so on. Customers never talk to those services individually. Every request from a browser, a script, or another product arrives at a single service called the **API Gateway**, which checks who is asking, whether they are allowed, and then routes the request to whichever service does the work.

This page explains what that means in practice, which ports a customer needs to open, and how to answer the questions that come up most often in a deployment conversation.

---

1. The short version

Question	Answer
What is the API Gateway?	The single front door into MDSS — for the web console and for all API traffic.
Which port does it use?	`8005` (HTTP) and `8006` (HTTPS) internally.
Which port does the customer connect to?	`80` (HTTP) or `443` (HTTPS) — the address of the MDSS console.
What does the customer open on the firewall?	Just `443` in a standard deployment.
Is the API included in the product?	Yes. Every action in the UI is available through the API, at no extra cost or module.

> **The one thing to remember**
>
> `8005` is the API Gateway's internal port. Customers do not connect to it. They connect to the MDSS console address on **443**, and everything — the UI and the API — is served from there.

---

2. What the API Gateway does

Think of the API Gateway as the reception desk of the MDSS deployment. Everyone entering the building goes through it, gets identified, and gets directed to the right department. Nobody walks into a department directly.

It has four jobs:

It is the only way in. All other MDSS services run on a private internal network with no exposure to the customer's network. This is the security argument that resonates most: MDSS may be a dozen services, but it presents **one** door to the outside world.

It authenticates every single request. Whether the caller is an administrator in a browser, an automated script using an API key, or a user arriving through the customer's SSO provider, the gateway verifies the identity before anything else happens. Nothing reaches a backend service unauthenticated.

It enforces what each user is allowed to do. MDSS distinguishes full administrators, tenant administrators, and read-only administrators. The gateway enforces those boundaries centrally, so the rules are applied consistently and cannot be bypassed by calling a service directly. In multi-tenant deployments, it also resolves which tenant a request belongs to and keeps tenants isolated from one another.

It powers everything customers see and automate. The web console is driven entirely by the gateway's API, including live scan progress that updates in the browser as files are processed. The same API is what customers use to integrate MDSS into their own tooling.

---

3. Ports

What the customer actually connects to

In every supported deployment, a web front end sits in front of the API Gateway and serves both the console and the API on the same address:

Port	Purpose	Notes
443	HTTPS — MDSS console and API	The standard, recommended option. Requires a certificate to be configured during setup.
80	HTTP — MDSS console and API	Enabled by default. Suitable for lab and evaluation environments only.

A customer browsing to `https://mdss.customer.com` gets the console. A script calling `https://mdss.customer.com/api/...` reaches the API. Same host, same port, same certificate.

The gateway's own ports

Port	Purpose
8005	API Gateway, HTTP
8006	API Gateway, HTTPS

These sit *behind* the web front end. In a Docker deployment they are not published to the host at all — the gateway is reachable only from inside the MDSS internal network. On a Windows native-services installation they are local ports on the MDSS server itself.

Both are configurable at install time if they conflict with something already running on the customer's server. The default of `8005` is what appears in configuration files, support packages, and log output, which is the main reason to know the number.

What to tell a customer about their firewall

For a standard deployment, one inbound rule:

- **Inbound TCP 80 and 443** to the MDSS server, from wherever administrators and API clients connect.

That is the complete list for reaching MDSS. Ports `8005` and `8006` do **not** need to be opened, and neither do the ports used by the internal services. MDSS still needs *outbound* access to the storage it scans, to the MetaDefender Core scanning engine, and to OPSWAT for licensing — but those are separate conversations, covered in the deployment documentation.

---

4. FAQ

"Do we need to open port 8005?" No. Customers connect on 443. `8005` is internal to the MDSS server.

"Can we change the port MDSS runs on?" Yes. Both the front-end port and the gateway port are set at installation and can be changed afterwards. Customers who already run something on 443 commonly move MDSS to another port.

"Is the API a separate license?" No. It is included, and it is the same API the MDSS console uses.

"Can we put MDSS behind our own load balancer or reverse proxy?" Yes — this is a normal deployment pattern. Terminate TLS at the customer's proxy and forward to MDSS. Bring in Sales Engineering to confirm the specifics for the customer's environment.

“Does the API support our SSO?” The gateway handles SSO sign-in for console users. Automated and scripted access uses API keys rather than SSO. Confirm the customer’s specific identity provider with Sales Engineering.

“How is the API secured?” Every request is authenticated before it reaches any backend service, using either a token, an API key, or an SSO session. Permissions are then checked against the caller’s role. Over HTTPS, all of it is encrypted in transit.

How to create an MDSS Scan Start API: Moving Files Between NFS Storages After Sanitization

Overview

This knowledge base documents how to initiate on-demand scans via the MetaDefender Storage Security (MDSS) API and have sanitized or allowed files automatically moved to a different storage unit: NFS → NFS. The move/copy/delete actions are executed by the workflow's **Remediations** configuration, not by the scan-start call itself.

> To move files post-scan, ensure your **Workflow** has the correct **Remediations** defined. The `/api/scan/start` endpoint only triggers discovery/scanning; it does not choose the destination. See "Configure the Workflow" below.

API Endpoint

> **POST**
>
> `/api/scan/start`

Starts an on-demand (instant) scan on one or more storage units. The destination of post-scan file movement is controlled by the Workflow's Remediations.

Request Body Schema

- `workflowId` (string, required) — Workflow to use
- `priority` (number, optional) — Queueing priority
- `storageIds` (array[string], required) — Source storage unit IDs
- `name` (string, optional) — Friendly scan name
- `ScanPartition` (string, optional) — JSON string specifying a subset (folder/prefix). See examples per storage type

> The property name is case-sensitive: use **ScanPartition**. The value is a *string* that contains JSON. Do not send an object for `ScanPartition`, or you will get a "could not be converted to System.String" validation error.

Minimal working examples (sanitized placeholders)

** Windows path (NFS/SMB mounted path) — folder subset**

Prefer forward slashes to avoid escaping:

```
POST http://localhost/api/scan/start Content-Type: application/json { "workflowId": "<workflow-id>", "priority": 1, "storageIds": ["<source-storage-id>"], "ScanPartition": "{\"Folder\":\"C:/mdss-demo/source/inbound\"}", "name": "PartitionScan_WindowsPath" }
```

Backslash version (double-escaped):

```
"ScanPartition": "{\"Folder\":\"C:\\\\mdss-demo\\\\source\\\\inbound\"}"
```

> If **Further Assistance** is required, please proceed to log a support case or chat with one of our support engineers.

Why Does MetaDefender Storage Security Intermittently Show "License Information Is Not Available" on RHEL 8?

> **Check Your Version:**
>
> This article applies to MetaDefender Storage Security deployments on **Red Hat Enterprise Linux (RHEL) 8** using Podman with the **CNI network backend**.

Issue

Users may intermittently see the following warning in the MetaDefender Storage Security dashboard:

> **Some functionality is disabled because there is a communication issue and license information is not available.**

The warning may disappear after refreshing the dashboard and reappear later.

When this issue occurs, MetaDefender Storage Security services may also intermittently lose communication with the local PostgreSQL database.

For example, the licensing service log may contain an error similar to:

```
An error occurred using the connection to database 'MDSS'
on server 'tcp://postgres:5432'.
```

Similar database communication or name resolution errors may also appear in other MDSS service logs.

Main Cause

MetaDefender Storage Security consists of multiple containerized services that communicate with each other using internal hostnames.

On some RHEL 8 environments, Podman uses the legacy **CNI network backend** for container networking and DNS resolution. An intermittent DNS resolution issue may prevent an MDSS service from resolving another container hostname, such as `postgres`.

For example, if the licensing service temporarily cannot resolve the `postgres` hostname, communication with the database is interrupted. The licensing service is then unable to

retrieve the required license information, causing MDSS to display the warning in the dashboard.

Once DNS resolution is restored, communication resumes. As a result, the warning may disappear after refreshing the dashboard and reappear when another DNS resolution failure occurs.

This behavior is related to the container networking and DNS environment and does not necessarily indicate an issue with the MDSS license itself.

> **Note:** The warning message alone does not confirm this issue. Other conditions, such as database availability, resource exhaustion, or connectivity issues, may cause similar symptoms.

Confirm the Podman Network Backend

Run the following command:

```
sudo podman info --format '{{.Host.NetworkBackend}}'
```

If the output is:

```
cni
```

the system is using the CNI network backend.

You can also check the installed container networking and DNS packages:

```
rpm -qa | grep -E 'dnsname|netavark|aardvark'
```

If the system is using the CNI backend and the MDSS service logs show intermittent database communication or DNS resolution errors, proceed with the resolution below.

Resolution

Recommended Long-Term Solution: Upgrade to RHEL 9

For the most stable long-term solution, we recommend upgrading the MDSS host operating system from **RHEL 8 to RHEL 9**.

RHEL 9 uses Podman's newer **netavark** network backend together with **aardvark-dns** by default. This newer networking stack handles the container networking and DNS resolution required for communication between MDSS services and avoids the intermittent CNI DNS behavior described in this article.

Before upgrading, review the supported operating systems and system requirements for your MetaDefender Storage Security version and plan the migration according to your organization's change management requirements.

Workaround for RHEL 8

If upgrading to RHEL 9 is not currently possible, you can address this issue on the existing RHEL 8 environment by switching Podman's network backend from **CNI** to **netavark + aardvark-dns**.

For detailed instructions, please refer to the following article:

How to Fix 'Name Does Not Resolve' Error in RHEL / Rocky Linux 8 with Podman

Note: This procedure requires restarting MetaDefender Storage Security, which will cause a temporary service interruption. We recommend taking an appropriate system backup or snapshot before applying the workaround.

> **Support:**
>
> If further assistance is required, please create a support case or chat with our support engineer.

CONFIGURATION _ SETUP

Is there a limitation on importing configuration to MetaDefender Storage Security?

> **Check Your Version:**
>
> This article applies to all MetaDefender Storage Security releases deployed on Windows or Linux systems.

Yes. We encountered multiple compatibility issues when attempting to import configurations into an already configured MetaDefender Storage Security . These issues can lead to system instability and conflicts.

To address this, we have decided to offer the import functionality exclusively for fresh installations. This ensures a smooth migration of your configuration to a new deployment without the risk of compatibility problems.

> **Support:**
>
> If **Further Assistance** is required, please proceed to create a support case or chat with our support engineer.

[Linux] What are MDSS-related resources and how to perform disk cleanup?

This guide provides an overview of MDSS resources, including paths, containers, and volumes, along with actionable recommendations for managing disk space.

By following these guidelines, you can safely clean unnecessary files while maintaining the integrity of critical data.

Primary Cleanup Mechanism

The main method for managing disk space in MDSS is configuring the **Database Retention Policy**. This policy automatically removes outdated data based on retention settings, reducing manual cleanup requirements. For more information on setting up and using the retention policy, see the Database Retention Configuration Guide.

Filesystem Paths

MD Storage Security uses several paths on the host system for its operations. Below is a breakdown of their purposes and cleanup recommendations:

1. `/etc/mdss` the main installation directory containing Docker Compose files and the `mdss.sh` script.
2. `/etc/mdss/services_logs` stores MDSS container logs mounted from each running container.
   - **Cleanup Guidance**: Log cleanup is generally unnecessary if proper logging configurations are in place. To adjust logging levels or enable extra logging features, refer to How to Enable Extra Logging in the Configuration.
3. `/var/log/mdss` contains logs generated by MDSS scripts.
   - **Cleanup Guidance**: Can be safely cleaned if the logs are no longer needed.
4. `/var/lib/docker` houses all Docker container filesystems and volumes.
   - **Important Note**: Avoid making direct changes to this path. Use `docker` commands for all operations to prevent accidental data loss.

Docker Containers

MDSS services run inside Docker containers. Below are the key containers and their roles:

1. **Containers with Names Starting with** `mdss_*` **and** `mdcs_*` run MDSS services.
2. `mdcs_rabbitmq` provides RabbitMQ services for communication between MDSS components.
3. `mdss_rediscache` temporary data storage using Redis.
4. `mdss_potgresql` database container for persistent data, including settings, storage units, and processing history.

Docker Volumes

Volumes store persistent and temporary data required by MDSS. Here's an overview:

1. `mdss_mongo_db` stores the old MDSS database (persistent data).
   - **Important Note**: Do not delete this volume unless performing a clean reinstallation.
2. `mdss_mongo_backup` temporary old database files for backups.
   - **Cleanup Guidance**: Can be safely removed when MDSS is not running.
3. `mdss_rediscache` temporary data storage.

- **Cleanup Guidance**: Can be safely removed when MDSS is not running.

4. `mdss_postgres_data` stores the MDSS database (persistent data).
   - **Important Note**: Do not delete this volume unless performing a clean reinstallation.

Useful Commands for Disk Usage Management

When investigating disk usage issues, use the following commands:

1. `df -h` displays overall disk usage in a human-readable format.
2. `du -h <PATH>` checks the size of a specific path on the system.
3. `docker system df -v` provides a detailed report of disk usage for Docker containers, images, and volumes.
4. `docker system prune -a --volumes` removes all unused containers, networks, images, and volumes.
   - **Important Note**: Make sure all MDSS services are running before executing this command to avoid removing critical data.

Additional Resources

- For more detailed information, refer to the Sizing Guide.
- We also recommend configuring an external database to enhance storage capacity for MDSS operations. For detailed instructions, please refer to the following article: Using an External Database.

> **Support**
>
> If you have any difficulty performing disk cleanup, please follow these instructions on How To Create a Support Package, before creating a support case or chatting with our support engineer.

How to Enable Specific Modules in the Configuration?

Enabling Modules

`ENABLED_MODULES` in `customer.env` lists the storage types this deployment offers. It is the product-level feature gate: storage types of modules not listed still appear in the user interface, but they are disabled and cannot be selected when an account is added.

For instance, if your purpose is to scan Amazon S3 buckets only, keep just the `amazonsdk` module in `ENABLED_MODULES`:

```
ENABLED_MODULES=amazonsdk
```

Please follow Custom configuration to learn how to modify the `customer.env` file.

It's important to remember that turning modules on or off is a non-destructive action, allowing you the flexibility to activate additional modules later if you need to work with different types of storage.

By default, all modules are enabled, which means that `ENABLED_MODULES` will look like this right after the installation:

```
ENABLED_MODULES=azureblob,amazonsdk,googlecloud,alibabacloud,azurefiles
,smb,box,graph,sftp,nfs,mft,ftp,oraclesdk
```

Module	Storage types
`amazonsdk`	Amazon S3 and the S3-compatible storages (S3 Compatible, Wasabi, Oracle S3-Compatible, Cubbit, Scality RING)
`azureblob`	Azure Blob
`azurefiles`	Azure Files
`googlecloud`	Google Cloud Storage
`alibabacloud`	Alibaba Cloud OSS
`oraclesdk`	Oracle Cloud Storage
`box`	Box
`graph`	SharePoint Online and OneDrive (Microsoft Graph). Replaces the former `onedrive` module name
`sftp`	SFTP
`ftp`	FTP
`mft`	MetaDefender Managed File Transfer
`nfs`	NFS and NFS with Data Wrapping
`smb`	SMB / CIFS, including Dell Isilon and NetApp ONTAP shares

`smb_legacy` is an additional module for Linux deployments. It selects the previous SMB service for hosts that cannot run the SMB service's kernel mount; keep `smb` in the list and add `smb_legacy` at the end. See Add an SMB-Compatible unit in Linux deployments.

Use the module names exactly as listed above. Names are case-sensitive and must be separated by commas without spaces. Unknown module names are ignored, and the corresponding integrations are not available. Before upgrading, review your list as described in Upgrade MDSS.

Please change it accordingly but remember that in order for these changes to be applied you need to restart MetaDefender Storage Security.

> **Resource usage**
>
> Disabling a module other than `nfs` or `smb` does not reduce the number of containers. Every storage type except NFS and SMB runs in-process inside the central Storages, Discovery, Remediations and Scanning services, so no container is started or stopped when it is enabled or disabled. Only `nfs` and `smb` add a container each (`mdintegrations_nfsservice`, `mdintegrations_smbservice`); removing them from `ENABLED_MODULES` also removes those containers.
>
> An upgraded installation keeps the `ENABLED_MODULES` value it had before the upgrade.

Disabling Services

The following services can be disabled when they are managed externally, to prevent deploying their local container versions:

- MongoDB
- PostgreSQL
- RabbitMQ
- Redis

They can be disabled by adding them individually to the `DISABLED_SERVICES` variable in `customer.env`:

```
DISABLED_SERVICES=mongo,postgres,rabbitmq,redis
```

Which API endpoints from MD Core does MDSS utilize?

Endpoints	Permissions	Usage
`/file`	Processing result fetching - Self-Only/Anyone Download processed file - Self-Only/Anyone	Process files and retrieve results Download processed files
/readyz	-	System health
/stat	System Information - Full	System health
/status	-	System health
/admin/license	License - Read-Only/Full	License information
/version	-	Version information

Advanced File API Filtering and Sorting

Filtering and sorting on the File and PDF report APIs include support for a variety of filter and sort combinations.

Some filtering and all sorting parameters support sending a list of values separated by comma, which take part of a logical OR filter (the property can be any of the given values).

Filtering

Filtering supports the following query parameter fields:

Query parameter	Possible values	Description	Allows list of values
startDate	A valid DateTime		no
endDate	A valid DateTime		no
search	A string value		no
scanType	- 0 = Instant - 1 = Scheduled - 2 = RealTime		yes
storageId	A valid storage id		yes
scanId	A valid scan id		yes
scanWorkflowSnapshotId	A valid snapshot id		yes
processingState	- InProgress = 0 - Failed = 1 - Canceled = 2 - Available = 3	State of processing in MDSS	yes
result	- 0 = Allowed - 1 = Blocked - 2 = Processing	Result of MDSS	yes
scanAllResultI	- See core result codes	General result of the Scan Instance	yes
scanResultICount	An integer >= 0	The number of engine results that are not "No threat detected"	yes
vulnerabilityCount	An integer >= 0	The number of vulnerabilities detected	yes
dlpDetection	Boolean true or false	DLP detection	no
remediations	- 0 = Tagging - 1 = DeepCdrSanitization - 2 = DeepCdrDelete - 3 = DeepCdrEncrypt - 4 = DeepCdrKeep - 5 = AllowedMoved - 6 = BlockedMoved - 7 = BlockedDelete - 8 = SanitizedMoved - 9 = AllowedCopied - 10 = BlockedCopied - 11 = SanitizedCopied - 100 = SoftDelete	Remediations performed	yes

Sorting

Sorting supports multiple parameters, which are used if there is an equality between the values of the preceding sort. If multiple sorts are used, the property and order are matched in order between the two lists of values. The following query parameter fields are available:

Query parameter	Possible values
sortBy	- 0 = DiscoveryStarted - 1 = ScanId - 2 = ProcessingState - 3 = Result - 4 = ScanAllResultI - 5 = ScanResultICount
sortOrder	- 0 = Ascending - 1 = Descending

Is SNMP monitoring supported for MetaDefender Storage Security?

At the current time SNMP monitoring is not supported for our MetaDefender Storage Security product.

This limitation is affecting all versions of the product.

> **Support:**
>
> If **Further Assistance** is required, please proceed to create a support case or chat with our support engineer.

[Linux] How to download docker images from alternate location?

If Docker Hub is inaccessible during installation, you can proceed with either an Linux Offline Installation or by specifying an alternative container registry. To configure an alternate registry, add the necessary details to the `customer.env` file.

```
MDSS_REPO=public.ecr.aws/v2f5u9p9/opswat-mdss
POSTGRES_IMAGE=public.ecr.aws/v2f5u9p9/opswat-mdss/postgres
RABBITMQ_IMAGE=public.ecr.aws/v2f5u9p9/opswat-mdss/rabbitmq
REDIS_IMAGE=public.ecr.aws/v2f5u9p9/opswat-mdss/garnet-alpine
```

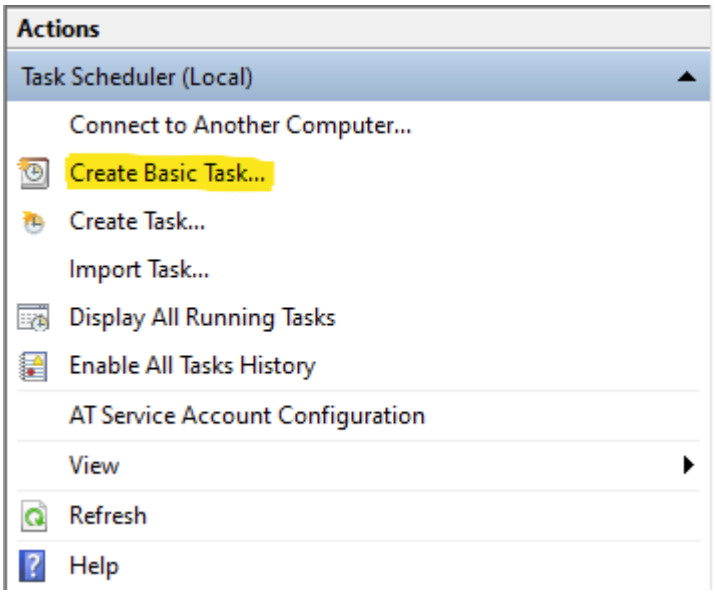
[Windows] How to create a persistent mount for SMB and NFS?

> **Info**

> All the below steps must be performed on the same Windows Server machine where MetaDefender Storage Security is installed.

Step 1

Navigate to **Task Scheduler** and click on **Create Basic Task** (on the right side)

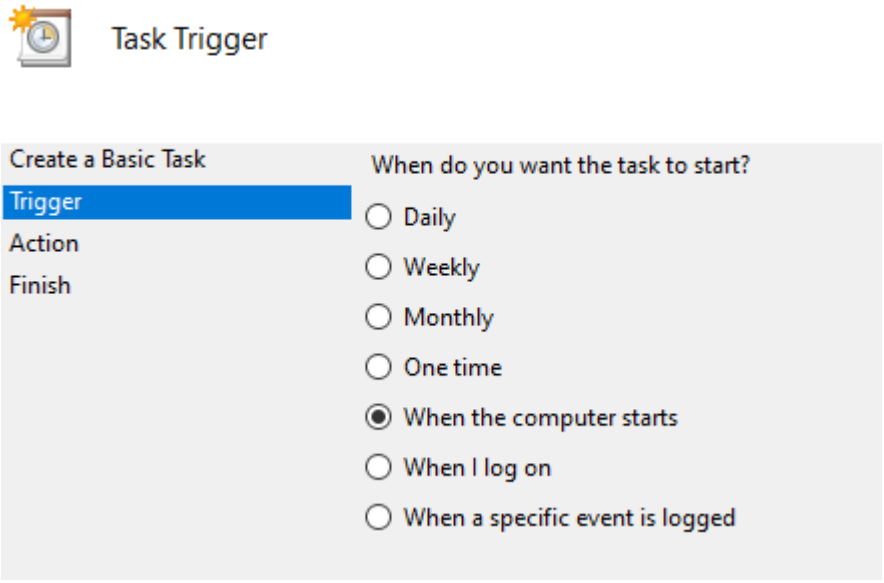


Step 2

Give your task a specific name and click **Next**

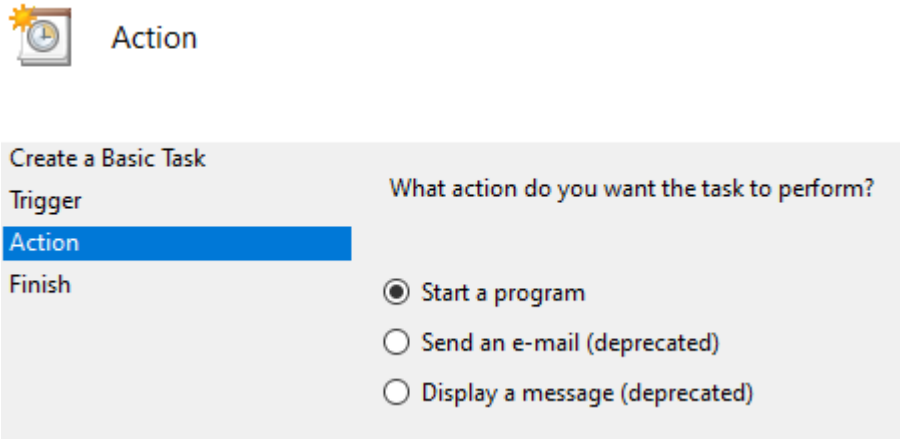
Step 3

Select **When the computer starts**



Step 4

Select **Start a program**



Step 5

Enter your mount command in the **Program/script** field.

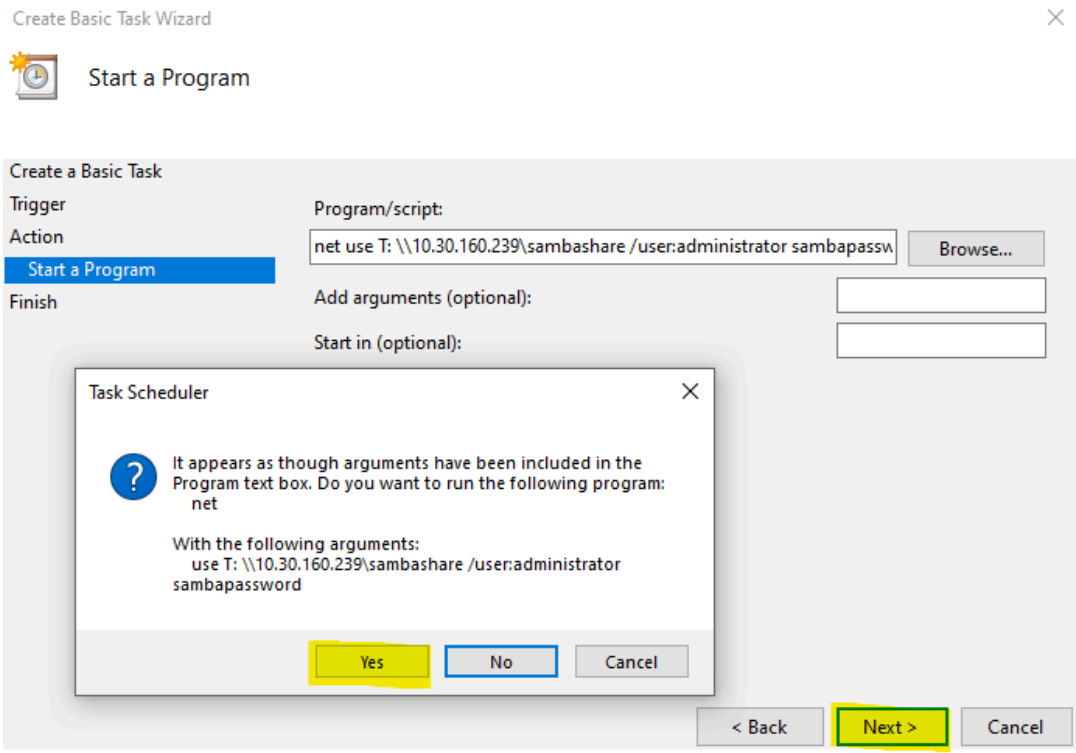
For SMB mount, below is an example: [the password must be provided here because at startup, Windows will not prompt you for it]

```
net use T: \\10.30.160.239\sambashare /user:sambauser sambapassword
```

For NFS mount, below is an example: [no password is required for NFS]:

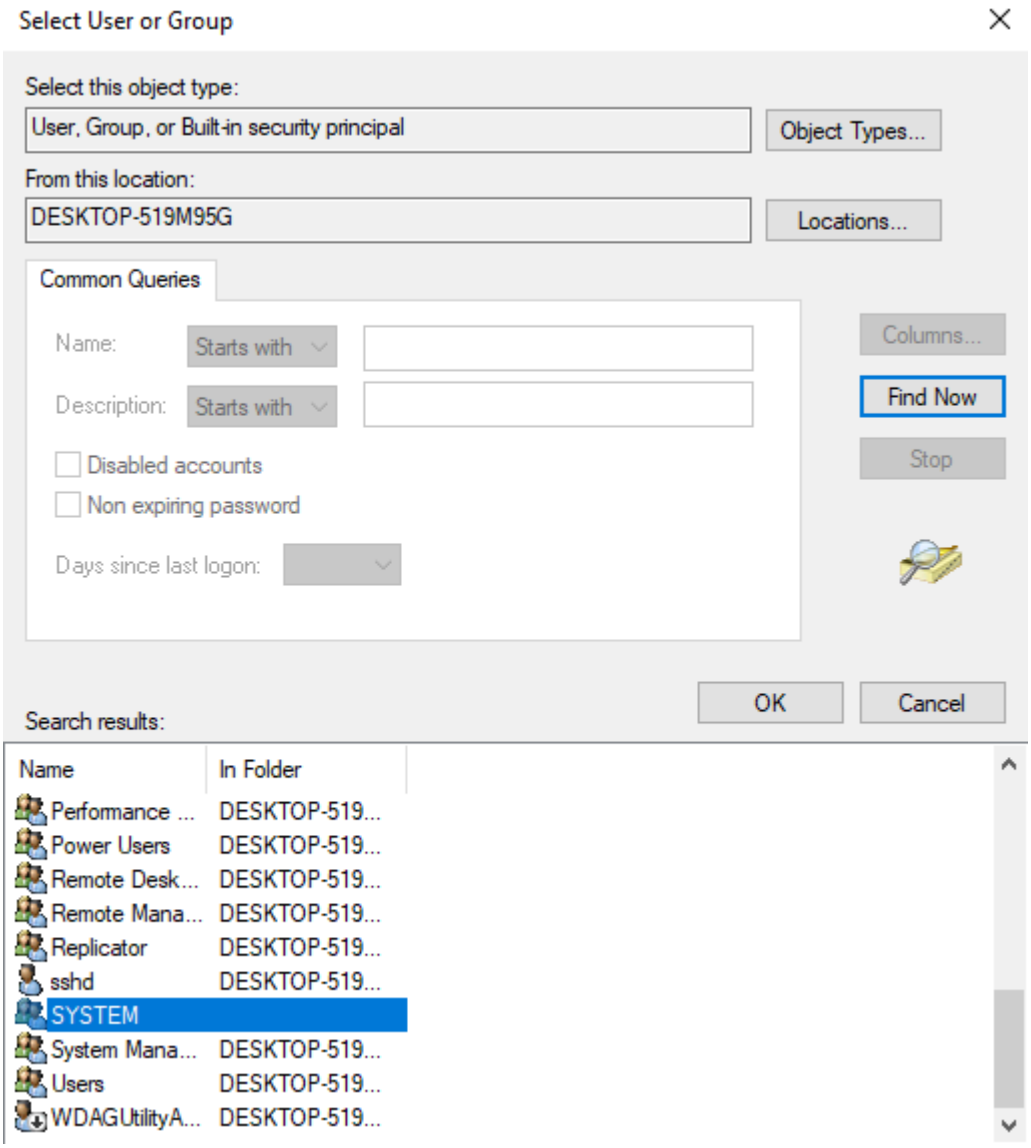
```
mount -o "nolock,sec=sys" 192.168.207.139:/srv/company-public-files/ Y:
```

Click **Next -> Yes -> Finish**



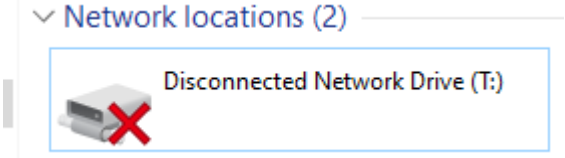
Step 6

From **Task Scheduler**, navigate to **Task Scheduler Library** -> Select your newly created task -> Right click -> Properties -> General -> Change User or Group -> Advanced -> Find Now -> Choose **SYSTEM** -> Click OK and that's it.



Warning

When logged in as a normal user or administrator, the mapped network drive will appear as disconnected. It is not disconnected, the user that is logged in does not have privileges to access it. If you open a terminal as a NT SYSTEM, you will be able to access the newly mapped network drive.



[K8s] How to enable Horizontal Pod Autoscaling based on RabbitMQ queue metrics

This guide explains how to scale the MDSS processing services (Scanning, Discovery and Remediations) with the number of messages waiting in RabbitMQ, so the number of pods grows when a backlog builds up and shrinks when the work is done.

CPU is a poor signal for these services: they spend most of their time waiting on the storages and on MetaDefender Core, so a large backlog can build up while CPU usage stays low. The services exchange work through RabbitMQ queues, which makes the queue length the direct measure of pending work.

Two ways to do it are described here:

Option	Best for	Effort
A. KEDA (recommended)	Any cluster where KEDA can be installed. Built into the MDSS Helm chart.	Install KEDA, then set `keda.enabled: true`. No Prometheus needed.
B. Prometheus + Prometheus Adapter	Clusters that already run Prometheus and cannot run KEDA.	Manual setup of scraping, adapter rules and an HPA per service.

> **Info**
>
> This documentation provides a general implementation approach that will need to be adapted to your specific environment. The exact commands, configurations, and values presented here should be reviewed and modified according to your organization's specific Kubernetes infrastructure, network architecture, and operational requirements.

> **Warning**
>
> Use one autoscaling mechanism per service. Two autoscalers on the same Deployment fight each other. When KEDA is enabled the chart skips the CPU based HPA (`hpa.enabled`) for the services KEDA manages, and a manually created HPA (Option B) must not target a service that KEDA manages.

Prerequisites

- Kubernetes cluster with HPA support (Kubernetes 1.23 or higher for `autoscaling/v2`)

- MDSS deployed with the Helm chart (the KEDA integration is available from chart version 4.6.0)
- The RabbitMQ instance used by MDSS (in-cluster or external) must be reachable from the cluster
- Option A (KEDA):
  - KEDA installed in the cluster (on OpenShift: the *Custom Metrics Autoscaler* operator from OperatorHub)
  - KEDA must be able to reach RabbitMQ over AMQP (port 5672). The RabbitMQ management plugin and the Prometheus plugin are **not** required
- Option B (Prometheus):
  - If using an external RabbitMQ instance: version 3.8.0 or higher, the RabbitMQ Prometheus plugin enabled, port 15692 accessible for metrics scraping and the management interface available
  - Basic understanding of Kubernetes and Prometheus

Option A: Autoscaling with KEDA

Step 1. Install KEDA (if not already deployed)

```
helm repo add kedacore https://kedacore.github.io/charts
helm repo update
helm install keda kedacore/keda --namespace keda --create-namespace
```

Check that the operator is running and the CRDs are registered:

```
kubectl get pods -n keda
kubectl get crd scaledobjects.keda.sh triggerauthentications.keda.sh
```

On OpenShift, install the *Custom Metrics Autoscaler* operator from OperatorHub and create its `KedaController` instance instead.

Step 2. Enable KEDA in the MDSS Helm values

Add the following to your values file (or pass the same keys with `--set`):

```
keda:
  enabled: true
  minReplicaCount: 1
  maxReplicaCount: 5
  queueLength: 100      # target number of ready messages per replica,
evaluated per queue
  cooldownPeriod: 300   # seconds to wait before scaling back to
minReplicaCount
  behavior: {}          # optional autoscaling/v2 behavior block for
the HPA KEDA creates
```

Then install or upgrade the release:

```
helm upgrade --install mdss <path-to-mdss-chart> --namespace
<namespace> --reuse-values -f keda-values.yaml
```

The chart creates, in the release namespace:

- one `ScaledObject` for each service that has queues configured (`scanningservice`, `discoveryservice`, `remediationsservice`),
- one `TriggerAuthentication` (`mdss-rabbitmq`),
- one Secret (`mdss-keda-rabbitmq`) with the AMQP connection string, built from `RABBITMQ_HOST`, `RABBITMQ_PORT`, `RABBITMQ_DEFAULT_USER` and `RABBITMQ_DEFAULT_PASS`.

To use your own Secret instead, create one containing the full `amqp://user:password@host:5672/` URI under the key `host` and set `keda.existingSecret` to its name. An external RabbitMQ needs no other change, as long as `RABBITMQ_HOST` points to it.

A service managed by KEDA is created without a fixed `replicas` value, so a later `helm upgrade` does not reset the replica count chosen by the autoscaler.

Step 3. Which queues are monitored

Every work queue exists in three priorities (`_High`, `_Medium`, `_Low`). Each one is a separate trigger, and the service is scaled to whatever the busiest queue requires.

Service	Queues (each with _High, _Medium, _Low)
scanningservice	scanning.object_ready_for_scan.v2_queue, object_ready_for_scan_queue, object_scan_instance_result_queue
discoveryservice	start_discovery_queue, request_object_download_queue, event_based_discovery_queue, folder_discovery_queue
remediationsservice	object_scanned_queue, object_not_scanned_queue

Step 4. Verify

```
kubectl get scaledobject
kubectl get hpa
kubectl describe scaledobject scanningservice
```

Each `ScaledObject` should report `READY=True`. KEDA creates and manages an HPA for it, named `keda-hpa-<service>` (for example `keda-hpa-scanningservice`).

To see the queue depth that drives the scaling:

```
kubectl exec deploy/rabbitmq -- rabbitmqctl list_queues name messages
consumers
```

Step 5. Tune the scaling

Every setting can be overridden per service, which is also how you change the thresholds for a single service:

```
mdss_components:
  scanningservice:
    keda:
      maxReplicaCount: 10
      queueLength: 200
```

- `queueLength` is the number of ready messages one replica is expected to handle, evaluated for each queue separately. A lower value scales up earlier.
- `minReplicaCount` / `maxReplicaCount` bound the number of pods.
- `cooldownPeriod` is how long the queues must stay empty before the service scales back to `minReplicaCount`. Scale-down between those limits follows the standard HPA stabilization window; use `behavior` to change it.

> **Info**

> Scanning throughput is ultimately bound by the MetaDefender Core instances behind it. Scaling the Scanning service beyond what Core can absorb only moves the backlog, so scale MetaDefender Core as well (see the MetaDefender Core Helm chart).

Scaling behavior with KEDA

- When the number of ready messages in any monitored queue exceeds `queueLength` per replica, KEDA scales the service up, up to `maxReplicaCount`.
- When the queues drain, the HPA scales the service down gradually; after `cooldownPeriod` seconds with empty queues it returns to `minReplicaCount`.

Troubleshooting KEDA

1. `kubectl describe scaledobject <service>` shows the conditions and events. A `READY=False` status usually means KEDA cannot reach RabbitMQ.
2. Check the KEDA operator logs: `kubectl logs -n keda -l app=keda-operator`
3. The AMQP connection string is stored in the `mdss-keda-rabbitmq` Secret (key `host`). KEDA runs in its own namespace, so a short in-cluster host such as `rabbitmq` is qualified automatically with the release namespace; use a fully qualified host name for an external broker.
4. If the service does not scale, compare `queueLength` with the real queue depth (`rabbitmqctl list_queues`).

Option B: Autoscaling with Prometheus and the Prometheus Adapter

Use this option when KEDA is not available. It scales the Scanning service on the total number of messages in its scan queues, using a manually created HPA. Set `hpa.enabled: false` for the service in the Helm values (it is disabled by default) so the chart does not create a CPU based HPA for the same Deployment, and do not enable `keda` for it.

Step 1. Install Prometheus (if not already deployed)

If Prometheus is not already deployed in your cluster, install it using Helm:

```
helm repo add prometheus-community https://prometheus-
community.github.io/helm-charts
helm install prometheus prometheus-community/prometheus --namespace
default
```

Step 2. Configure Prometheus to Scrape RabbitMQ Metrics

Add the following scrape configuration to your Prometheus ConfigMap to collect metrics from RabbitMQ:

```
- job_name: rabbitmq
  kubernetes_sd_configs:
  - role: service
    namespaces:
      names:
        - default
  relabel_configs:
  - action: keep
    source_labels: [__meta_kubernetes_service_name]
    regex: rabbitmq
  - action: keep
    source_labels: [__meta_kubernetes_service_port_name]
    regex: 15692
  - action: labelmap
    regex: __meta_kubernetes_service_label_(.+)
  - source_labels: [__meta_kubernetes_namespace]
    target_label: kubernetes_namespace
  - source_labels: [__meta_kubernetes_service_name]
    target_label: kubernetes_service_name
  metrics_path: /metrics
```

Apply the updated ConfigMap and restart Prometheus to apply the changes.

> **Info**
>
> **Note for External RabbitMQ**: If using an external RabbitMQ instance, you'll need to modify the scrape configuration to target your external instance instead of using Kubernetes service discovery.

Step 3. Install Prometheus Adapter (if not already deployed)

The Prometheus Adapter is required to expose Prometheus metrics to the Kubernetes metrics API:

> **Info**
>
> Please update `prometheus.url` and `prometheus.port` according to your environment

```
helm install adapter prometheus-community/prometheus-adapter \
  --set prometheus.url=http://prometheus-server.default.svc \
  --set prometheus.port=80
```

Step 4. Configure Prometheus Adapter

Update the Prometheus Adapter ConfigMap to expose the RabbitMQ queue metric:

```
data:
  config.yaml: |
    rules:
    - seriesQuery:
'{__name__="rabbitmq_queue_messages_ready",kubernetes_namespace!=""}'
      seriesFilters: []
      resources:
        overrides:
          kubernetes_namespace: {resource: "namespace"}
      name:
        matches: "rabbitmq_queue_messages_ready"
        as: "rabbitmq_scan_queue_messages"
      metricsQuery: sum(rabbitmq_queue_messages_ready{queue=~"
(scanning\\.object_ready_for_scan\\.v2_queue|object_ready_for_scan_queu
e)_(Low|Medium|High)"}) by (<<.GroupBy>>)
```

This configuration:

- Queries metrics with name “rabbitmq_queue_messages_ready”
- Associates metrics with Kubernetes namespaces
- Renames the metric to “rabbitmq_scan_queue_messages”
- Filters for the scan queues (both the `scanning.object_ready_for_scan.v2_queue` and `object_ready_for_scan_queue` queues) with Low, Medium, and High priorities
- Sums the total number of messages across these queues

After modifying the ConfigMap, restart the adapter:

```
kubectl rollout restart deployment adapter-prometheus-adapter
```

Step 5. Verify the Custom Metric

Check that the metric is properly exposed to the Kubernetes API:

```
kubectl get --raw /apis/custom.metrics.k8s.io/v1beta1/ | jq
```

You should see “rabbitmq_scan_queue_messages” listed in the output.

```
$ kubectl get --raw /apis/custom.metrics.k8s.io/v1beta1/ | jq
{
  "kind": "APIResourceList",
  "apiVersion": "v1",
  "groupVersion": "custom.metrics.k8s.io/v1beta1",
  "resources": [
    {
      "name": "namespaces/rabbitmq_scan_queue_messages",
      "singularName": "",
      "namespaced": false,
      "kind": "MetricValueList",
      "verbs": [
        "get"
      ]
    }
  ]
}
```

To get the metric value:

```
$ kubectl get --raw
/apis/custom.metrics.k8s.io/v1beta1/namespaces/default/metrics/rabbitmq
_scan_queue_messages | jq
```

You should see the actual value of the queue size:

```
$ kubectl get --raw
/apis/custom.metrics.k8s.io/v1beta1/namespaces/default/metrics/rabbitmq
_scan_queue_messages | jq
{
  "kind": "MetricValueList",
  "apiVersion": "custom.metrics.k8s.io/v1beta1",
  "metadata": {},
  "items": [
    {
      "describedObject": {
        "kind": "Namespace",
        "name": "default",
        "apiVersion": "/v1"
      },
      "metricName": "rabbitmq_scan_queue_messages",
      "timestamp": "2025-03-28T11:37:19Z",
      "value": "4994",
      "selector": null
    }
  ]
}
```

Step 6. Create a Horizontal Pod Autoscaler

Create an HPA resource that uses the custom metric to scale the Scanning service:

```
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: scanningservice-hpa
  namespace: default  # Change if your deployment is in a different
namespace
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: scanningservice
  minReplicas: 1  # Minimum number of pods
  maxReplicas: 10  # Maximum number of pods
  metrics:
  - type: Object
    object:
      metric:
        name: rabbitmq_scan_queue_messages
      describedObject:
        apiVersion: v1
        kind: Namespace
        name: default  # Or the namespace where your RabbitMQ is
running
      target:
        type: Value
        value: 5000  # Total ready messages that trigger scale-up; tune
it to your workload
```

Apply the HPA:

```
$ kubectl apply -f scanningservice-hpa.yaml
```

Step 7. Monitor the HPA

Check the status of your HPA:

```
kubectl get hpa scanningservice-hpa
kubectl describe hpa scanningservice-hpa
```

Scaling Behavior (Option B)

- When the total number of messages in the scan queues exceeds the configured target (5000 in the example), the HPA scales up the number of scanning pods, up to the

maximum of 10.

- When the number of messages decreases, the HPA will gradually scale down the number of pods.

Troubleshooting (Option B)

If the HPA isn't working as expected:

1. Verify Prometheus is collecting RabbitMQ metrics:

```
kubectl port-forward svc/prometheus-server 9090:80
```

Then visit http://localhost:9090 and query:

```
rabbitmq_queue_messages_ready{queue=~"
(scanning\\.object_ready_for_scan\\.v2_queue|object_ready_for_scan_queue)_(Low|Medium|High)"
```

2. Check that the Prometheus Adapter can access the metrics:

```
kubectl logs -l app.kubernetes.io/name=prometheus-adapter
```

3. Verify the custom metric is available:

```
kubectl get --raw
/apis/custom.metrics.k8s.io/v1beta1/namespaces/default/metrics/rabbitmq
_scan_queue_messages
```

4. Ensure the HPA is targeting the correct deployment and namespace.

How to Configure and Use Syslog and Serilog Sinks?

Syslog Configuration

Recommended Tools:

- Visual Syslog Server (recommended)
- Syslog Watcher Manager (Note: Has a limit of 5000 messages/hour)

Configuration Steps:

Currently, we support Syslog (UDP) using the RFC5424 Output format for external logger configuration.

To visualize MetaDefender Storage Security audit logs centrally, integrate a Syslog (UDP) server using these details:

- Server address: IPv4, IPv6, and Host Name formats supported.
- Port: Specified Syslog server port.
- Syslog Facility: As per configuration needs.
- Output format: RFC5424.

You can add up to 5 external logger configurations of each type.

Syslog Message Format

RFC5424 message format is supported:

VERSION TIMESTAMP HOSTNAME APP-NAME PROCID MSGID STRUCTURED-DATA MSG

- PRI: Priority value
- VERSION: Syslog protocol specification version
- TIMESTAMP: Formalized timestamp
- HOSTNAME: Sending machine
- APP-NAME: Application or device originating message
- PROCID: Process name or ID
- MSGID: Message type
- STRUCTURED-DATA: Parsable data format
- MSG: Informative free-form message

Example Syslog Message

```
<110>1 2024-03-26T12:27:58.276724+02:00 LP10-D8569
MetaDefenderStorageSecurity 36800 - [meta UserName="jdavis"
UserId="e69e8c2d-4dcc-4489-8f40-0df84199de52"
EventTimestamp="03/26/2024 10:27:58" Category="3" LogType="600"] John
Davis (jdavis) logged in.
```

Serilog Sinks

What is Serilog?

Serilog is a structured logging library for .NET applications, which allows logging to various destinations called sinks, including Syslog, files, HTTP endpoints, and console.

Commonly Used Serilog Sinks and Configurations:

1. Serilog.Sinks.Syslog

Logs events to remote syslog servers via UDP or TCP, supporting RFC3164 and RFC5424 formats.

```
"WriteTo": [
  {
    "Name": "TcpSyslog",
    "Args": {
      "host": "<ip_host>",
      "port": 514,
      "appName": "<your_app_name>",
      "format": "RFC5424",
      "facility": "<facility_name>",
      "outputTemplate": "[{Timestamp:u}] [{Level}] {SourceContext}:
{Message:lj}{NewLine}{Exception}"
    }
  }
]
```

2. Serilog.Sinks.Http

Sends log events via HTTP/S to remote endpoints.

```
"WriteTo": [
  {
    "Name": "Http",
    "Args": {
      "requestUri": "<your_request_uri>",
      "queueLimitBytes": null
    }
  }
]
```

3. Serilog.Sinks.Console

Writes log events to the console.

```
"WriteTo": [
  {
    "Name": "Console",
    "Args": {
      "outputTemplate": "[{Timestamp:u}] [{Level}] {Namespace}:
{Message:lj}{NewLine}{Exception}"
    }
  }
]
```

4. Serilog.Sinks.File

Writes logs to a file.

```
"WriteTo": [
  {
    "Name": "File",
    "Args": {
      "buffered": false,
      "fileSizeLimitBytes": 500000000,
      "outputTemplate": "[{Timestamp:u}] [{Level}] {Namespace}:
{Message:lj}{NewLine}{Exception}",
      "path": "logs/log.txt",
      "retainedFileCountLimit": "10",
      "rollingInterval": "Day",
      "rollOnFileSizeLimit": true
    }
  }
]
```

Testing & Validation:

Use tools like Visual Syslog Server or test HTTP endpoints to validate your logging configurations and ensure logs are captured correctly.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

Does MetaDefender Storage Security support symlinks scanning?

Yes, MDSS fully supports symbolic links (symlinks) **when using the NFS storage type**. This means that MDSS can seamlessly follow these links and process the files they point to.

All major types of symlinks are supported across both **Windows** and **Linux** deployments, including:

- **Soft links (symbolic links)**
- **Hard links**
- **Junction points** (Windows-specific)

Feature	Soft Link	Hard Link	Junction Point
What it is	File containing a path string	Directory entry for file data	Filesystem pointer
Points to	File or Directory	File only	Directory only
Cross Volume?	Yes	No	Yes (Local Volumes Only)
Target Deleted?	Link becomes broken	Other links remain valid	Link becomes broken
Platform	Linux/Unix, Windows	Linux/Unix, Windows	Windows only

---

Symlink scanning exclusivity

Symlink scanning is **exclusively applicable to NFS** and is not supported for other storage types. To learn more about integrating an NFS share, please refer to this page.

Broken link handling in MDSS

When MDSS encounters a broken link or a link pointing to a non-existent or inaccessible target, it will generate an error indicating that it was unable to locate the file at the specified path (the path to which the broken symlink was pointing).

Does a CVE affect MetaDefender Storage Security?

In today's rapidly evolving digital landscape, understanding the impact of security vulnerabilities is more crucial than ever. The Common Vulnerabilities and Exposures (CVE) list is an invaluable resource for identifying potential security risks associated with software products. This article provides a comprehensive overview of various CVEs, helping you determine whether MetaDefender Storage Security is affected. By staying informed about these vulnerabilities, you can take proactive measures to protect your systems and data from potential threats.

CVE	Summary
CVE-2024-27322	MetaDefender Storage Security doesn't use R programming language within the product, we are not affected by CVE-2024-27322.
CVE-2024-4603	This vulnerability affects the functions `EVP_PKEY_param_check()` and `EVP_PKEY_public_check()`. Since these are not used, MetaDefender Storage Security is not affected.
CVE-2024-0727	Upon thorough review and analysis, it has been determined that all versions of the License SDK integrated with the MDSS product do not utilize any of the mentioned vulnerable functions. Hence, MetaDefender Storage Security is not affected by CVE-2024-0727.
CVE-2023-46589	MetaDefender Storage Security does not use Apache so it's not being affect by CVE-2023-46589
CVE-2023-44487	MetaDefender Storage Security is not impacted by CVE-2023-44487.
CVE-2024-41110	We tested with versions of Docker up to `Docker version 27.1.1`, build 6312585 and can confirm that MetaDefender Storage Security is not impacted by CVE-2024-41110
CVE-2024-7592	MetaDefender Storage Security is container based, and the python version used by MetaDefender Storage Security internally is 3.12, hence MetaDefender Storage Security is not impacted by CVE-2024-7592. Python libraries present on the host machine where MDSS is installed can be removed.
CVE-2018-15133	MetaDefender Storage Security does not use PHP and/or Laravel Framework so it's not affected

Support

If **Further Assistance** is required, please proceed to log a support case or chat with our support engineer.

How do I Check Port 80 and 443 Usage Before Installing MDSS on Windows and Linux?

Introduction

Before installing MDSS, it is crucial to check the availability of ports 80 and 443. If these ports are in use by another application, the MDSS service will not start. This document provides step-by-step instructions to check port 80 and 443 usage on both Windows and Linux systems.

Windows

1. Checking Port 80 and 443 Using PowerShell

Steps:

1. Open PowerShell as an Administrator.
2. Copy and paste the following command:

```
Get-NetTCPConnection -LocalPort 80,443 | ForEach-Object {
    Get-CimInstance Win32_Process -Filter "ProcessId = '$($_.OwningProcess)'"
} | Select-Object ProcessId, Name, ExecutablePath
```

3. Interpret the Output:

- If no application is using the ports, you will see an error message like this:

```
Get-NetTCPConnection : No MSFT_NetTCPConnection objects found with property
'LocalPort' equal to '80'.
Get-NetTCPConnection : No MSFT_NetTCPConnection objects found with property
'LocalPort' equal to '443'.
```

This indicates that port 80 and 443 are available, and you can proceed with the MDSS installation.

- If another application is using the ports, the output will look like this:

```
ProcessId Name      ExecutablePath
--------- ----      --------------
    47752 httpd.exe c:\xampp\apache\bin\httpd.exe
```

This means that httpd.exe is currently using port 80 and 443.

---

2. Checking Port 80 and 443 Using Netstat

Steps:

1. Open Command Prompt (CMD) as an Administrator.
2. Run the following commands:

```
netstat -ano | findstr :80
netstat -ano | findstr :443
```

3. Interpret the Output:

- If the ports are in use, you will see an output like this:

```
TCP    0.0.0.0:80      0.0.0.0:0      LISTENING    47752
TCP    0.0.0.0:443     0.0.0.0:0      LISTENING    47752
```

- The last column (47752) is the Process ID (PID) of the application using the ports.

4. To identify the application using the PID, run:

```
tasklist | findstr 47752
```

Example Output:

```
httpd.exe                    47752 Console                    1     22,456 K
```

This confirms that httpd.exe is using port 80 and 443.

Linux

Checking Port 80 and 443 Using Netstat

Steps:

1. Open a Terminal.
2. Run the following command:

```
netstat -tulpn | grep -E ':80|:443'
```

3. Interpret the Output:

- If ports 80 and 443 are in use, you will see an output like this:

```
tcp    0    0 0.0.0.0:80     0.0.0.0:*     LISTEN     1234/nginx
tcp    0    0 0.0.0.0:443    0.0.0.0:*     LISTEN     5678/apache2
```

** Netstat Command Options Explained: **

- `-t` ➔ Show TCP connections
- `-u` ➔ Show UDP connections
- `-l` ➔ Show Listening ports
- `-p` ➔ Show the process name
- `-n` ➔ Show numeric addresses instead of resolving hostnames

---

Conclusion

- If ports 80 and 443 are available, you can proceed with the MDSS installation.
- If ports 80 and 443 are in use, you need to stop or reconfigure the conflicting application before installing MDSS.
- Use the provided commands on Windows and Linux to check and troubleshoot port usage effectively.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

What's the Best Practice for Managing Logs and Data Retention in MDSS?

> **Check Your Version:**
>
> This article applies to MetaDefender Storage Security versions up to 3.5.1

If you're experiencing storage issues on your MetaDefender Storage Security (MDSS) deployment, there are several strategies you can use to manage and free up disk space. This article outlines recommended practices for monitoring storage, adjusting configurations, and leveraging built-in features to maintain system performance and availability.

Recommended Actions

1.Configure Database Retention

To help manage storage usage, you can configure the system to automatically delete older records from the MDSS database. This helps prevent data buildup and keeps your storage usage under control. Refer to this support article: Database Retention

- **Enable automatic cleanup for the following record types:**
  - Audit records
  - Report records
  - Real-time processing history
- **Set retention periods** to define how many days these records are kept before being automatically deleted.
- **Lower retention thresholds** as needed to reduce overall database size and free up space.

2. Increase Disk Size for Key Directories

Especially on Linux-based installations, directories like `/var` often store temporary files related to file scans and remediation. If storage is running low:

- Extend the disk size allocated to MDSS.
- **Recommended formula:**
- *Temporary space ≥ 2 × [Size of the largest file in GB]**

3. Regularly Monitor Disk Space

- Regularly monitor disk space and identify large directories or files that may be consuming excessive storage.

4. Extend `/var` if Remediation Files Are Large

If remediation actions produce temporary files that fill up your `/var` directory:

- Consider resizing or extending the `/var` partition.
- Monitor this location closely, especially if you process large files.

5. Cloud Deployments: Use Native Monitoring Tools

If you've deployed MDSS in a cloud environment (e.g., AWS, Azure, GCP):

- Use built-in monitoring tools to track disk space.
- Set up automated alerts when usage reaches critical thresholds.

6. Upgrade to the Latest Version:

If you are using an older version of MDSS, upgrade to the latest version as it may include improvements for storage management. You can check and download the latest version by logging in to **My OPSWAT** and navigating to "Product Downloads".

Performance Limitations on Windows:

- Be aware that on Windows, each Docker container can only be assigned 2GB of RAM, which may impact performance. Consider moving to MDSS Linux for better resource management if possible

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

How to disable the Local PostgreSQL when using an external Database?

> **Check Your Version:**
>
> This article applies to MetaDefender Storage Security v4.x

Overview

If you prefer to connect the MetaDefender Storage Security (MDSS) application to your own remote PostgreSql database server, you can disable the built-in PostgreSql container so that it is not deployed. The bundled PostgreSQL container is tuned to use a share of the host memory (on Linux, shared buffers are set to 25% of RAM at installation), so disabling it frees those resources. Follow the steps below.

Instructions

1. Stop the MDSS Application

Before making any configuration changes, stop the MDSS services:

```
sudo mdss -c stop
```

---

2. Disable the Built-in PostgreSql Container

Open the file `/etc/mdss/customer.env` in a text editor and add `postgres` to the `DISABLED_SERVICES` variable (create the line if it does not exist):

```
DISABLED_SERVICES=postgres
```

`DISABLED_SERVICES` accepts a comma-separated list, so other externally managed services can be listed alongside it, for example `DISABLED_SERVICES=postgres,rabbitmq,redis`. See How to Enable Specific Modules in the Configuration? for details.

3. Start the MDSS Application

Once all modifications are complete, start the MDSS services again:

```
sudo mdss -c start
```

Verification

After restarting, verify that:

- MDSS is running without the internal PostgreSql container.
- The application is successfully connected to your external PostgreSql server.

You can check logs or test basic application functionality to ensure a proper connection.

> **Support:**
>
> If you have followed all of the steps above but are still experiencing issues, please follow these instructions on how to generate a Support Package before creating a support case or chatting with one of our support engineers.

Why remediation actions are not working?

> **Check Your Version:**
>
> This article applies to all MetaDefender Storage Security versions.

Remediation actions are executed after a successful file scan.

Please follow the steps below to analyze why remediation actions may not be working in your environment:

Step 1: Review correct setup

1. Make sure that the workflow has been setup correctly.
2. Ensure that MetaDefender Storage Security is connected to correct storage unit and there are not permission issues.

3. Check if MetaDefender Storage Security is connected to the expected MetaDefender Core Instance.

Step 2: Ensure that there are no issue scanning files

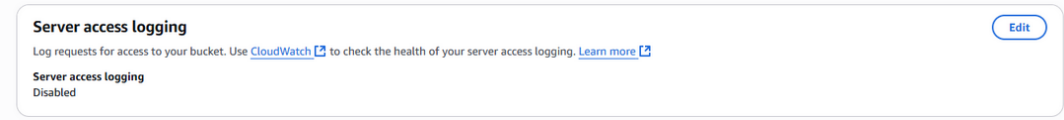
Remediation actions are executed after a successful file scan. If a file was not scanned, please check on MetaDefender Core the reasons why. You can review the different scan results codes in here.

Step 3: Transfer Speeds

Issues related to transfer speeds can cause file upload to take longer than expected which can cause issues performing remediation actions. For example in AWS S3 integration:

AWS

1. Low transfer speeds With big files, usually, there may be an extended period of time that it is used for file uploads. This can be caused due to a network infrastructure limitation. For checking the transfer speed in AWS for example, we recommend using “Server Access logging”.



2. Temporary credentials expire during file upload Depending on the transfer speeds, the temporary credentials provided by the IAM may expire mid-upload. This issue would be improved if transfer speeds improve. Alternatively, you can increase the temporary credential timeout period through AWS AIM Console settings as needed.



Step 4: Credentials and permission to access Storage units

Credentials and permissions are what allow access to MetaDefender Storage Security access storage units to apply remediation actions. In cases where these are not set correctly or lack enough priviledge, this function would run into issues. For example in SMB integrations:

SMB

1. **Permission** ** **restrictions preventing access to the SMB share.** To troubleshoot, please:
    1. Verify that the user account has appropriate permissions on the SMB share.

2. Check both share-level and NTFS permissions for read/write access.
3. Ensure the user account is not locked or disabled.

2. **Authentication** ** ****protocol error.** This issue could result from an authentication protocol mismatch between MDSS client and your SMB server. To troubleshoot:
   1. **If** ** ****your SMB server runs on Windows**: Configure the username from MDSS UI to be in UPN format (user@domain.com) to enable Kerberos authentication
   2. **If** ** ****your SMB server runs on Linux**: Use the format DOMAIN\username for NTLM authentication

> **Info:**
>
> Based on our analysis and extensive experience with similar connectivity challenges, we strongly recommend implementing **SMB Share via NFS Storage Unit Configuration**. This approach has consistently delivered superior results for customers experiencing SMB connectivity issues.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

What is the expected MetaDefender Storage Security Remediation Behavior Behind F5 Load Balance?

> **Check Your Version:**
>
> This article applies to all MetaDefender Storage Security versions.

Issue:

When F5 is configured with cookie-based persistence, MetaDefender Storage Security remediation could fail. This could occur if the XML provided was not well-formed or was not valid during initial communication. **cookie-based persistence on F5** causes inconsistent routing of related S3 API requests during remediation workflows.

Description

When persistence is enabled, the F5 load balancer modifies the headers cookies of API requests from MetaDefender Storage Security to the S3 compatible storage, causing API responses to get corrupted and not being processed by our MetaDefender Storage

Security. While switching F5 to source address persistence seems the least invasive there is no guarantee that it will not modify the headers for tracking.

Mitigation:

1. Disable persistence for the virtual server handling S3 traffic, OR
2. Create a separate virtual server with FastL4 profile for S3 connections

> **Info:**
>
> S3 API calls are stateless and don't require session affinity - persistence only adds processing overhead that corrupts the XML payloads.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

Why connection to SMB share cannot be established?

> **Check Your Version:**
>
> This error is reported by the legacy SMB service: releases before 4.5.0 on Windows, releases before 4.6.0 on Linux, Linux deployments that select the `smb_legacy` module, and Windows deployments with `USE_OPSWAT_SMB_SERVICE=false`. The SMB service shipped with 4.6.0 for Linux reports the messages listed in the troubleshooting table of Add an SMB-Compatible unit in Linux deployments.

Issue:

MetaDefender Storage Security connection with SMB fails with error:

```
smbprotocol.exceptions.SMBOSError: [Error 0] [NtStatus 0xc00000c9] Unknown NtStatus error returned 'STATUS_NETWORK_NAME_DELETED': '\\[IP_HOSTNAME]\ftp\CN'
```

Description

- `NtStatus 0xc00000c9` corresponds to `STATUS_NETWORK_NAME_DELETED`.
- This error typically occurs when the network share or resource you are trying to access has been removed, deleted, or is temporarily unavailable.

- It can also happen if the SMB session has become stale (e.g., the server closed the connection due to timeout, network issues, or a manual server-side operation).

Possible Causes:

- The network share `\\[IP_HOSTNAME]\ftp\CN` was deleted or unshared on the server. The SMB server might be up but the whole path might be unshared.
- The server at `[IP_HOSTNAME]` rebooted, crashed, or restarted its SMB service.
- There is a network connectivity issue between our SMB service and customer's SMB server.
- Permissions on the share changed, making it inaccessible.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

What permissions are needed for the user of the storage of Google Cloud?

Introduction:

This knowledge base article provides information on what roles the user that maps the Google Cloud Storage should be.

Permissions Overview

For MDSS to operate correctly when integrating with Google Cloud Storage, the user account must have sufficient privileges. The documentation initially recommends that the user be granted the Owner role. This role provides comprehensive permissions across the Google Cloud project, including the ability to:

- Create, modify, and delete buckets and objects
- Configure access controls and security settings
- Manage project resources and billing

Important Note: While granting the Owner role ensures all necessary permissions are available, it is generally best practice to adhere to the principle of least privilege. In cases where only storage operations are required, consider using more specific roles such as:

- Storage Admin (roles/storage.admin): Grants full control over Cloud Storage resources without broader project permissions.
- Storage Object Admin (roles/storage.objectAdmin): Provides full control over objects in the storage buckets while limiting administrative capabilities on bucket settings.

Select the role based on the required functionality and security guidelines of your organization.

Detailed Configuration Steps

1. Evaluating Requirements

- Assess MDSS Needs: Determine if MDSS requires full project management capabilities or just access to storage resources. For most scenarios where the sole purpose is mapping storage, the Storage Admin role may suffice.
- Review Security Policies: Consult your organization's IT and security policies to ensure compliance with role assignments and the principle of least privilege.

2. Assigning Permissions via the Google Cloud Console

- Log into the Google Cloud Console: Access your Google Cloud project.
- Navigate to IAM & Admin: Select "IAM & Admin" from the navigation menu.
- Locate the User Account: Find the user account that MDSS will use for mapping the storage.
- Assign the Appropriate Role:
  - For comprehensive permissions, assign the Owner role.
  - Alternatively, assign Storage Admin or Storage Object Admin if full project permissions are not required.
- Save Changes: Confirm and save the changes to update the permissions.

3. Verifying Permissions

- Test Storage Access: Use MDSS to perform operations such as creating, reading, updating, and deleting objects to verify that the assigned role meets the required functionality.
- Audit Logs: Regularly review the Cloud Audit Logs to ensure no unintended or unauthorized actions are performed.

Additional Considerations

- Security Best Practices: Always follow the principle of least privilege. Grant only the permissions that are necessary for the MDSS operations to minimize security risks.
- Periodic Review: Regularly review IAM roles and permissions in your Google Cloud project. Reassess whether the granted permissions still match the operational requirements of MDSS.
- Role Customization: If none of the predefined roles align with your needs, consider creating a custom role that aggregates only the required permissions for storage mapping. This approach can offer a more tailored security posture while meeting functional requirements.

Conclusion

Mapping Google Cloud Storage for MDSS requires careful consideration of the permissions assigned to the user account. While the initial recommendation is to use the Owner role, assessing the operational needs and following best security practices might suggest using more limited roles such as Storage Admin. By following the configuration steps and periodic reviews outlined in this article, you can ensure both functionality and security in your MDSS deployment.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

How to Integrate Cerberus SFTP with MDSS?

> **Check Your Version:**
>
> This article applies to all supported MetaDefender Storage Security releases deployed on Windows and Linux systems

> **Info:**
>
> This guide provides instructions to setup Cerberus SFTP shared folder configuration to integrate with MDSS using the basic credentials (user/pass)

What is Cerberus SFTP?

Cerberus SFTP Server enables organizations to become more efficient with fast, secure, compliant and easy file transfers. Homepage Cerberus FTP Server - The Secure and Compliant FTP Server | Cerberus FTP Server

Step-by-Step Setup for Cerberus SFTP Server

> **Info:**
>
> Only supported on a Windows Server, recommended to install Cerberus on a Windows Server 2022

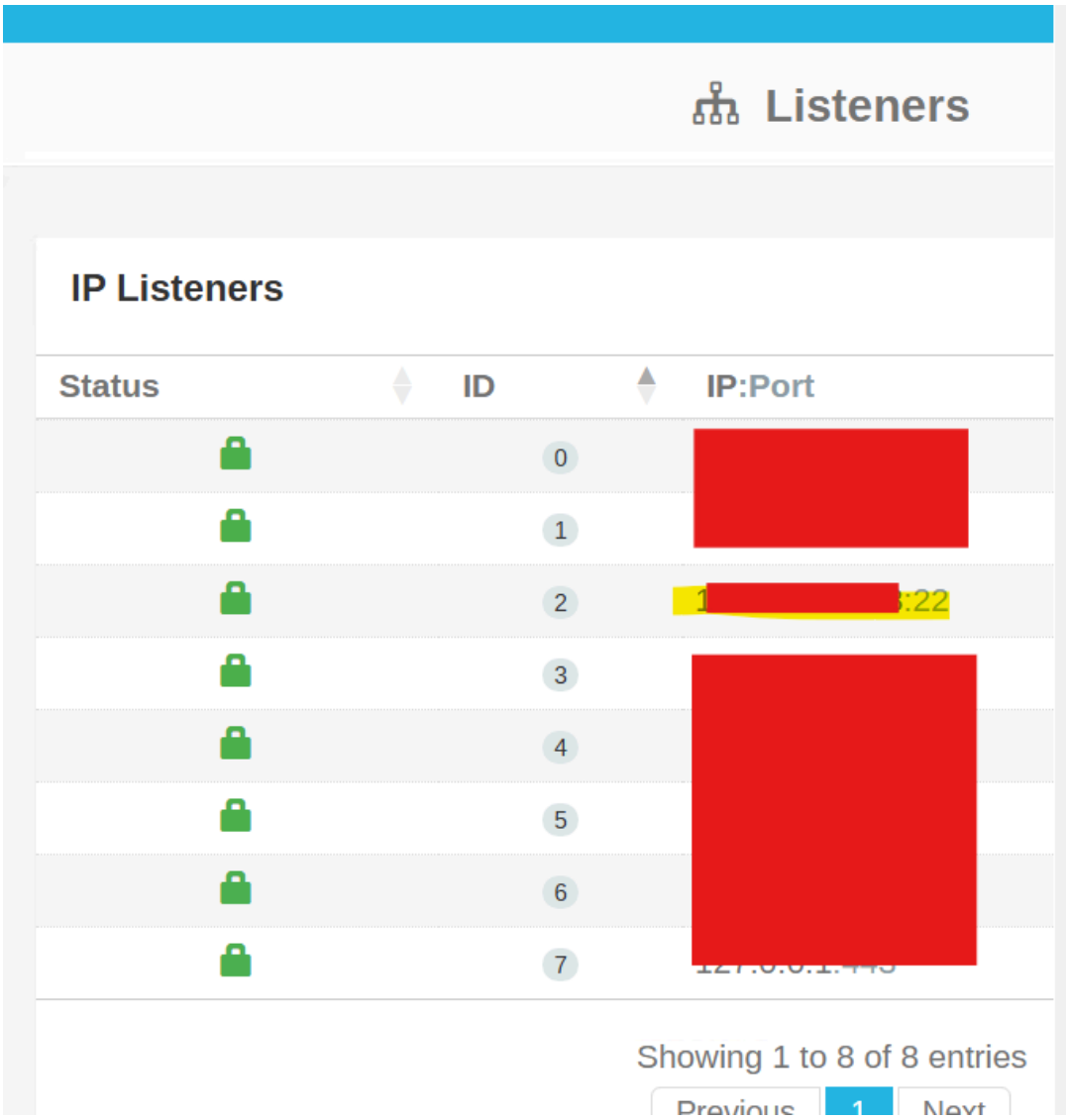
1. Download and Install Cerberus FTP Server

- Go to the Cerberus FTP Server website and download the installer. FTP Server Download | Cerberus FTP Server
- Run the installer as an Administrator.
- Enter first account

2. Enable SSH for SFTP

- SFTP requires SSH. Cerberus uses the built-in OpenSSH on Windows.
- Ensure OpenSSH is enabled on your system:
  - Go to **Settings > Apps > Optional Features**.
  - Install **OpenSSH Server** if it's not already installed.

3. Set Up an SFTP Listener

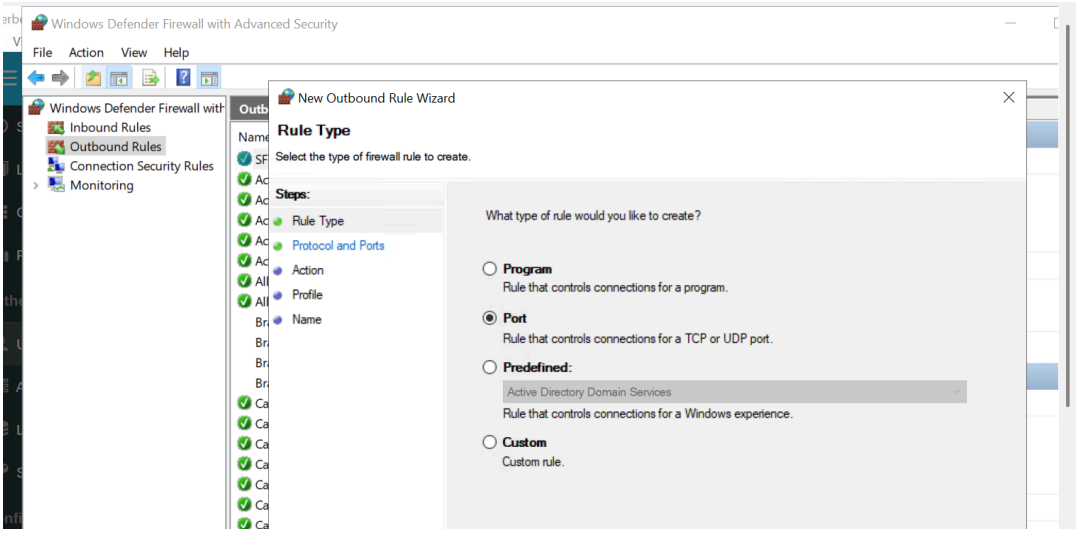


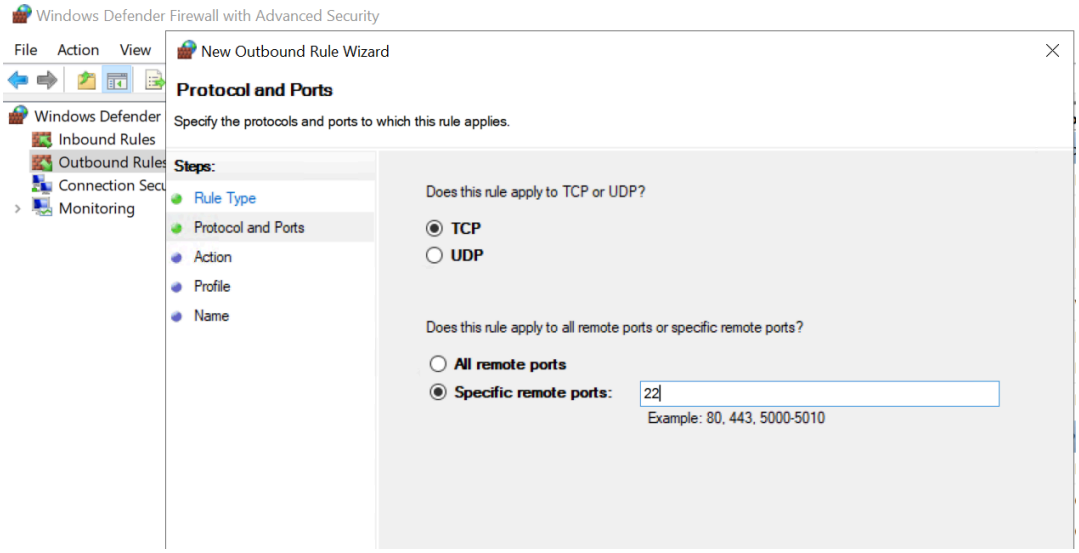
How to setup an SFTP Listener (in case it doesn't create by default):

- Open Cerberus FTP Server.
- Go to **Server Manager > Listeners**.
- Add a new listener:
  - **Protocol**: SFTP
  - **Port**: 22 (or another if needed)
  - **IP Address**: Choose the appropriate one (current server's IP) or use `0.0.0.0` for all.

4. Configure Firewall and Router

- Open port **22** (or your chosen port) on your firewall.

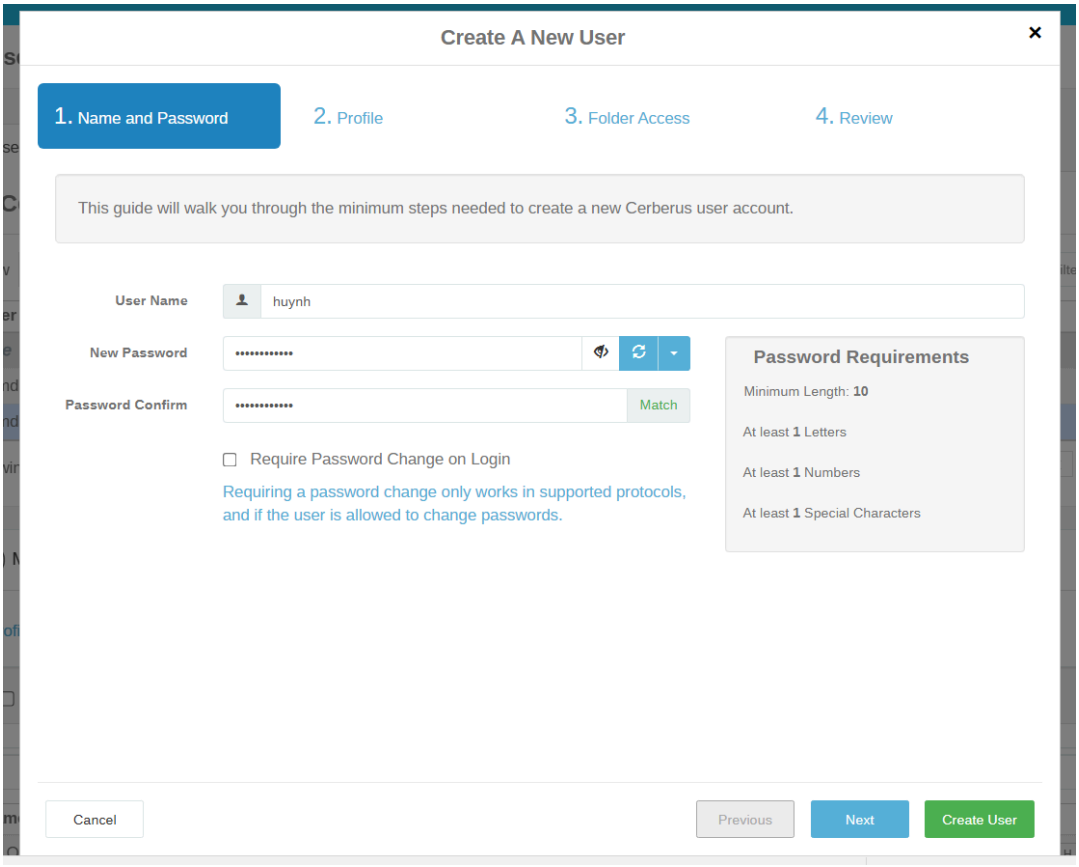




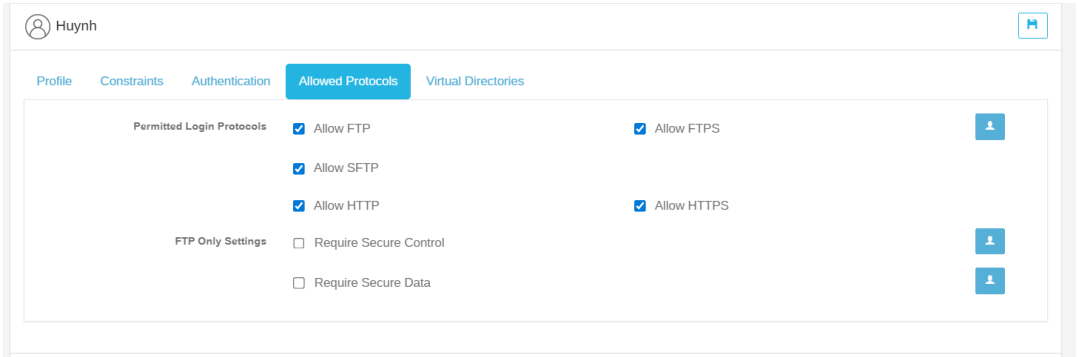
- Set up **port forwarding** on your router to direct traffic to your server's internal IP (if needed)

5. Mapping a folder with Cerberus local users

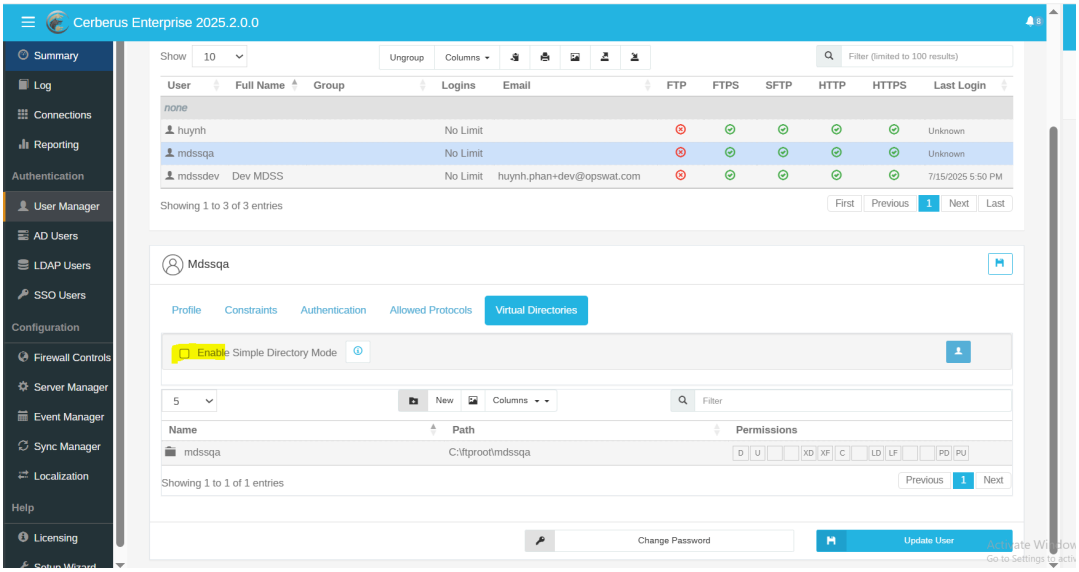
- Create a local user who has full permissions to the folder



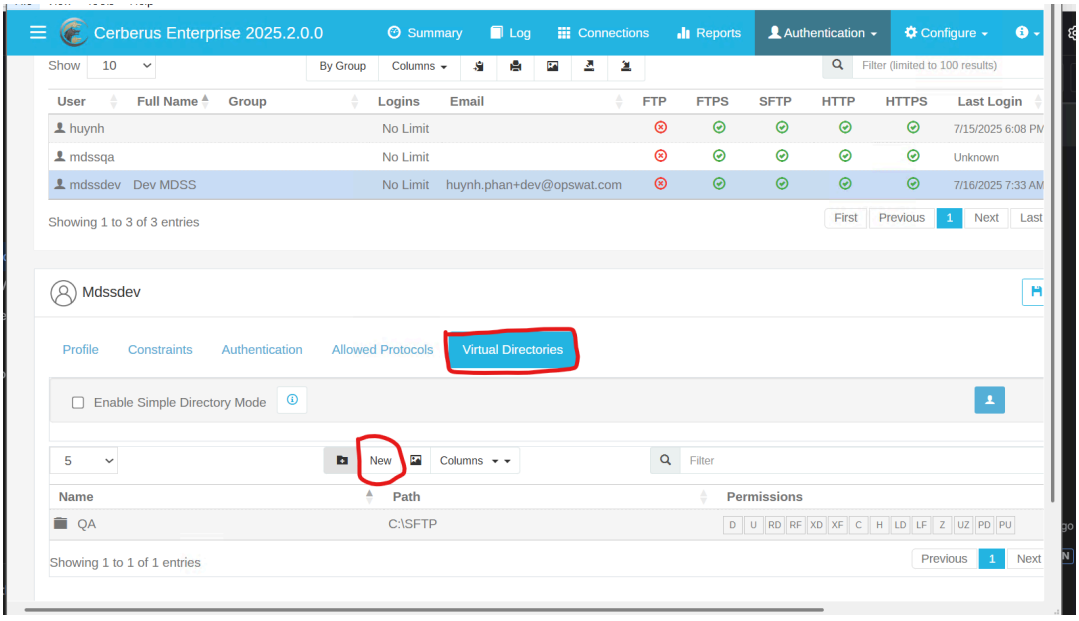
- Choose the user in the list
- Make sure the user is having the permission to login into SFTP protocol



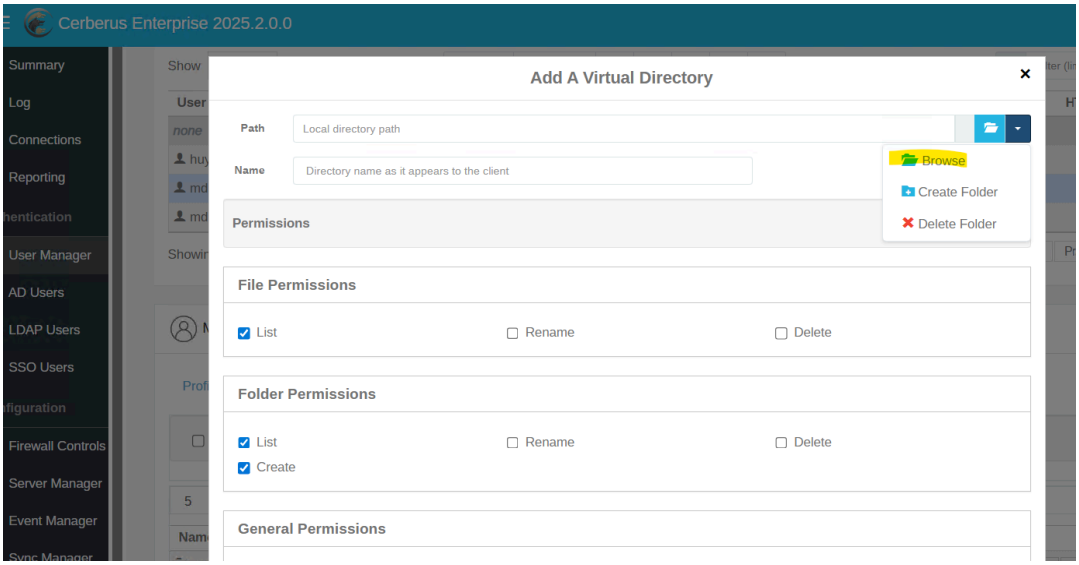
- Make sure Simple directory mode option is disable

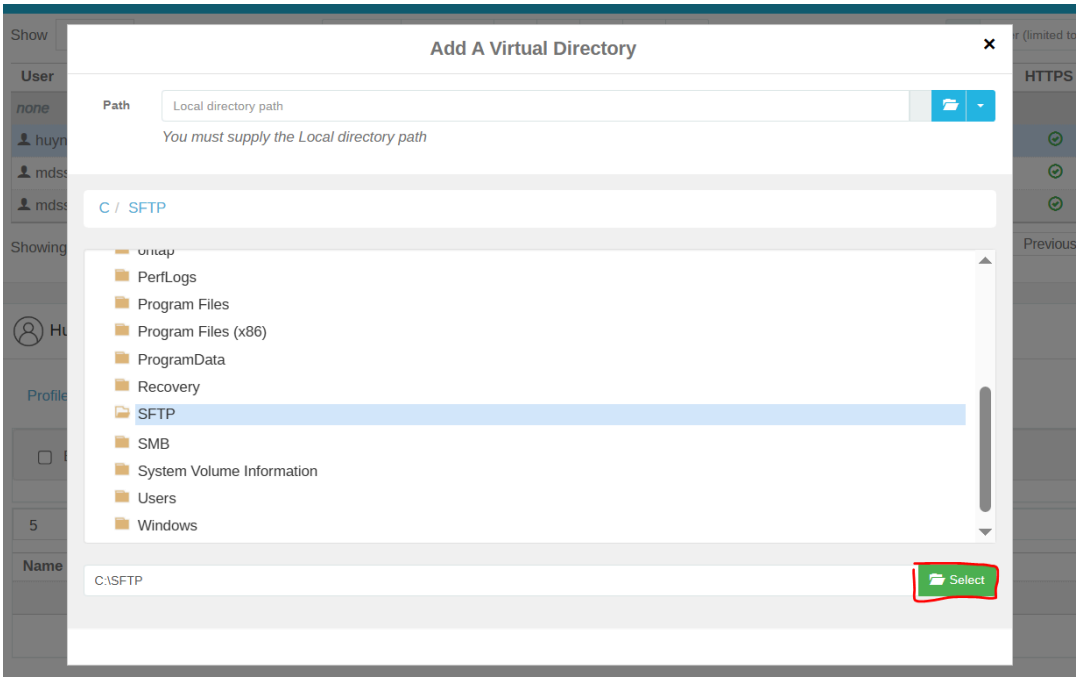


- Click New Virtual Directory

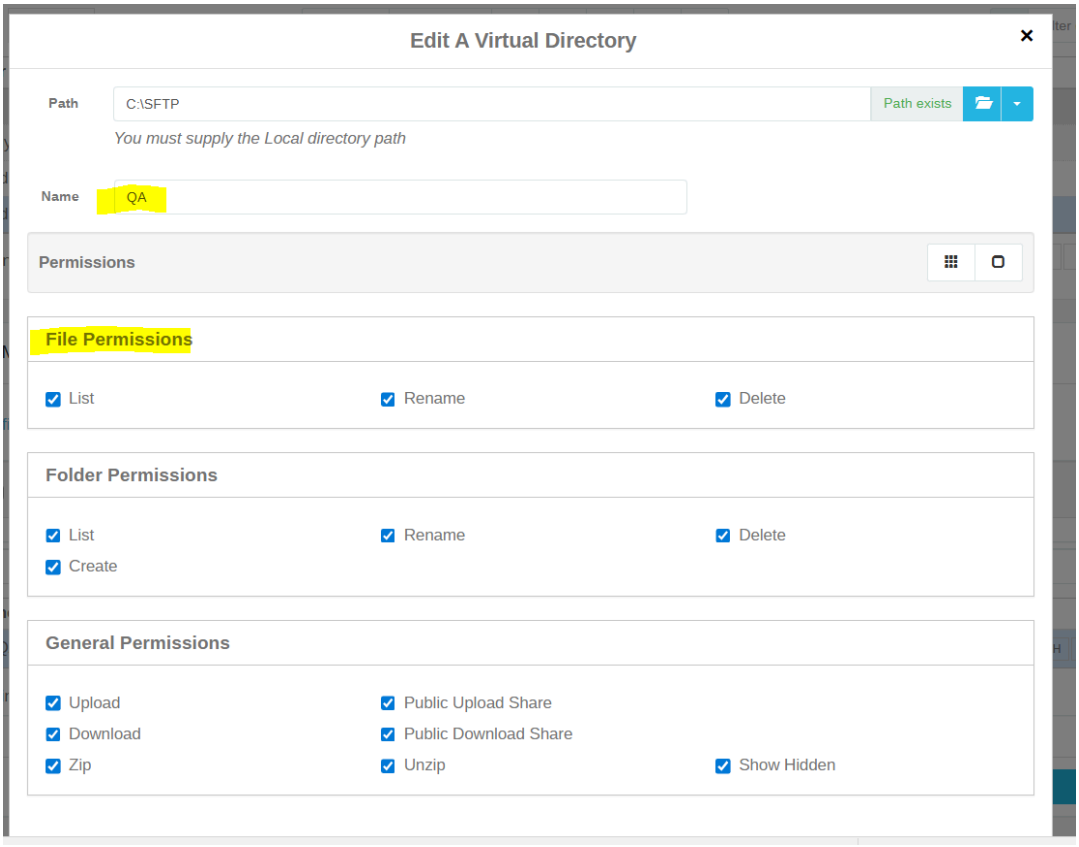


- Open “Browse” to choose the mapping directory from your Windows server or shared folder

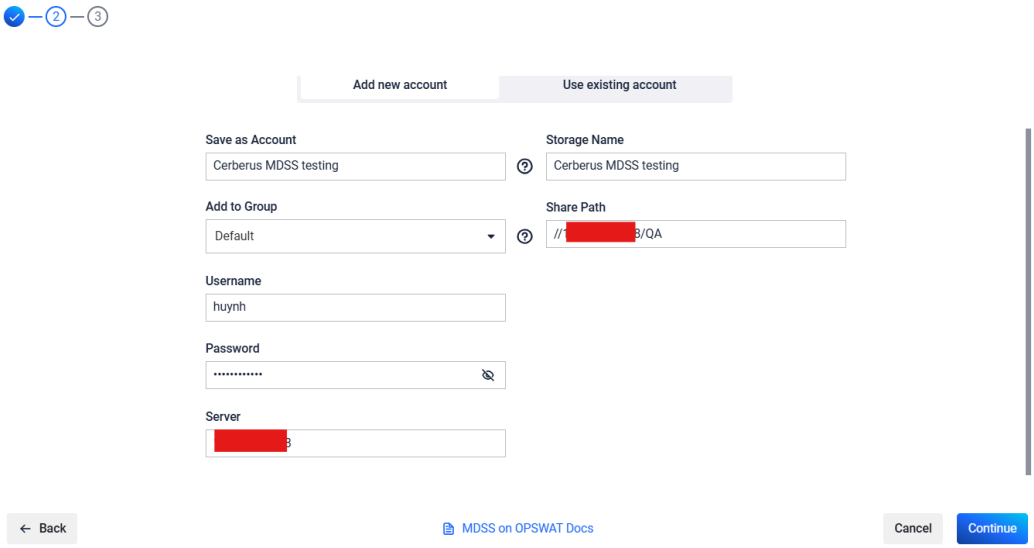




- Enter the shared name and give all permissions to the folder (save the shared name to configure later on from MDSS)



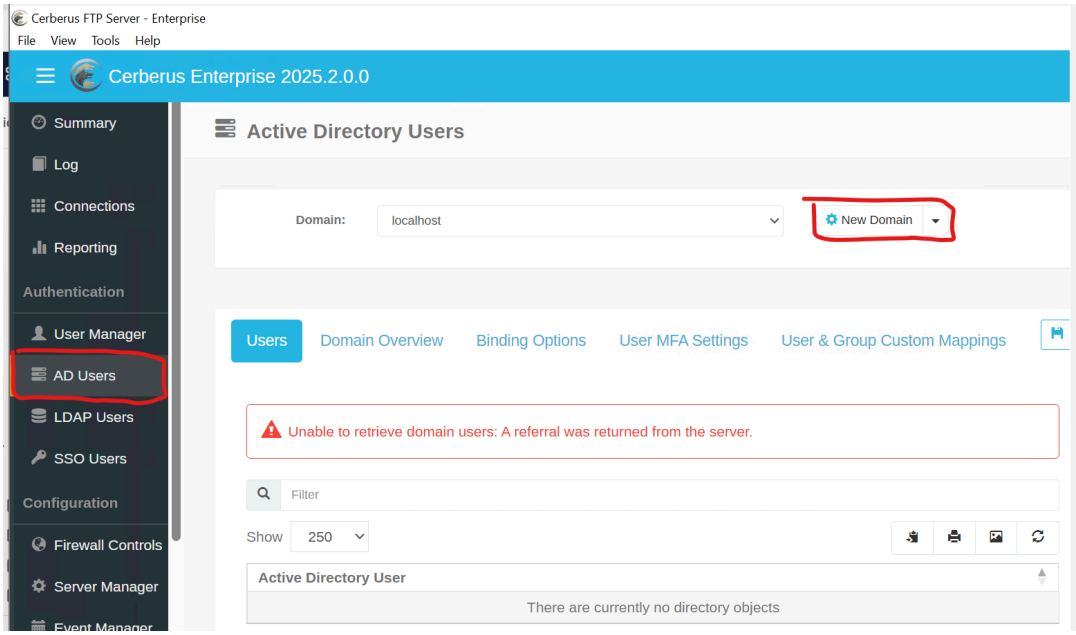
- Now the shared path is good to go integrating with MDSS, we could try with WinSCP first to see if it works, share path should come with the format as //CerberusIP/Name as configured, username and password are the credentials configured to the folder.



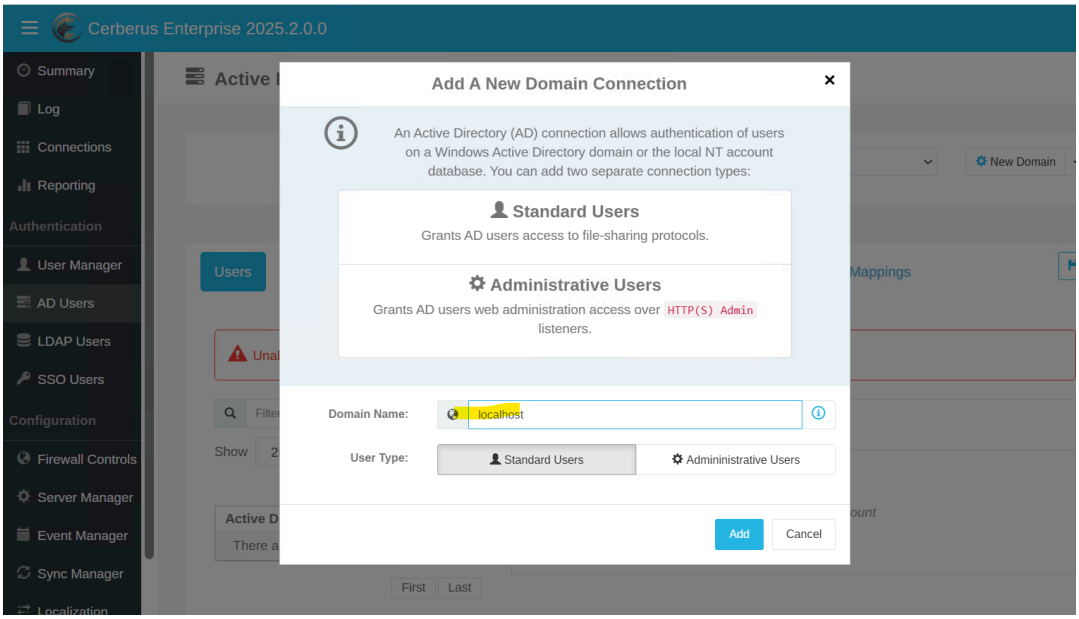
Mapping a shared folder with AD users via Cerberus group

As Cerberus is allowing AD integration, we could also integrate an AD or LDAP for users management

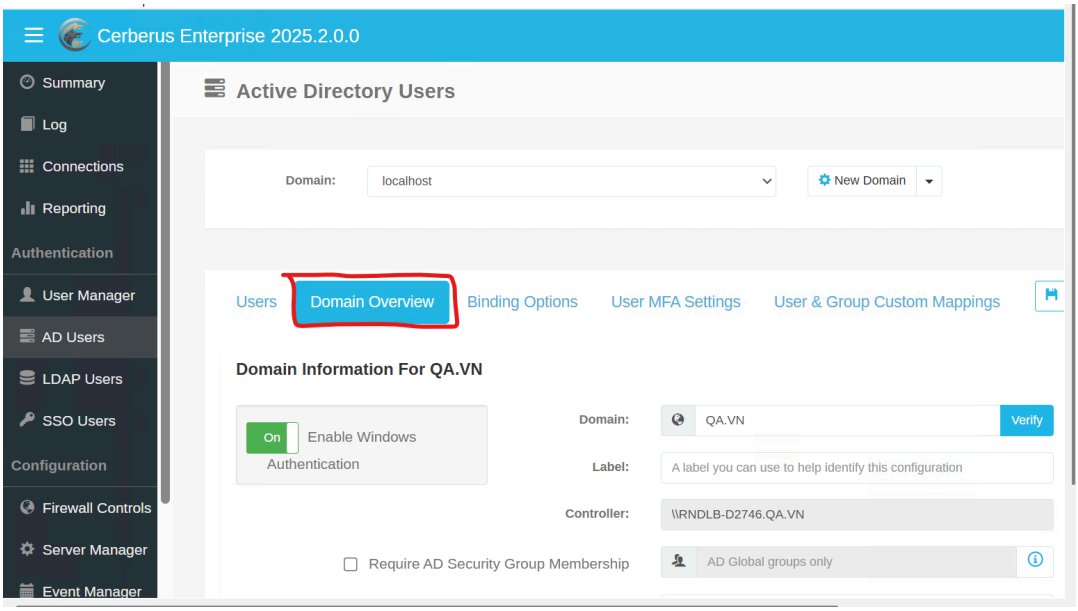
1. To add an AD, simply go to Authentication> AD users > Add New Domain



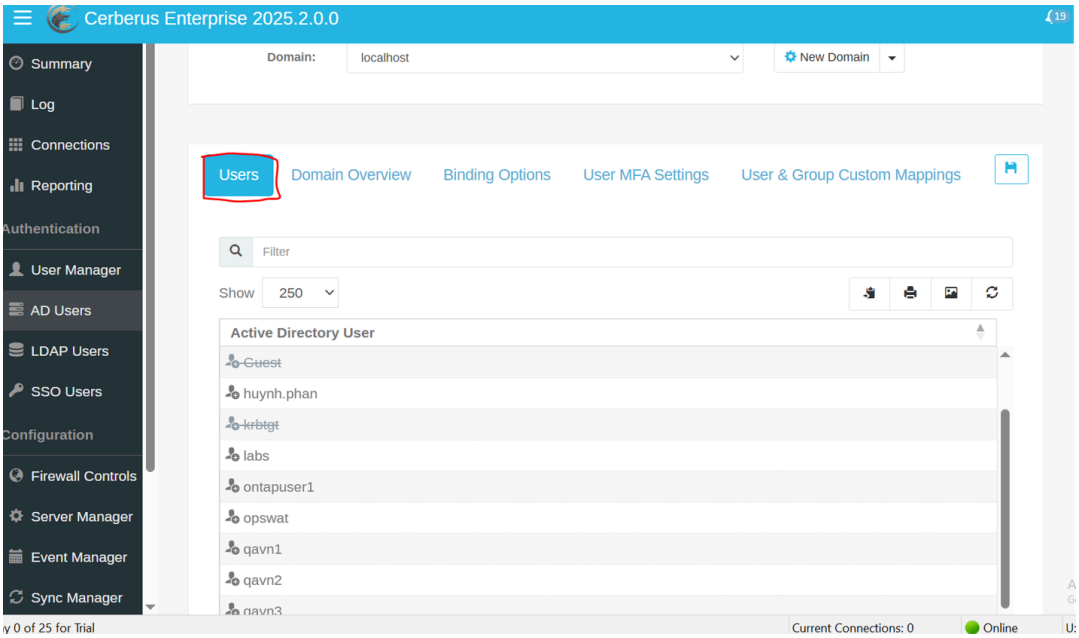
Fill in domain name



We could modify the domain later on after added

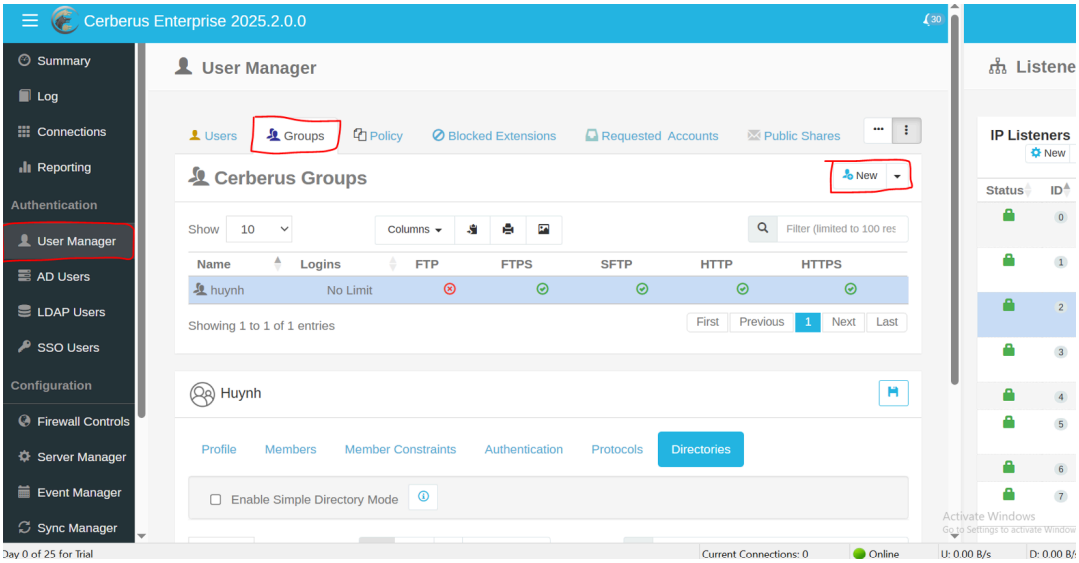


After connected successfully to the AD, the users list could be displayed at Users tab

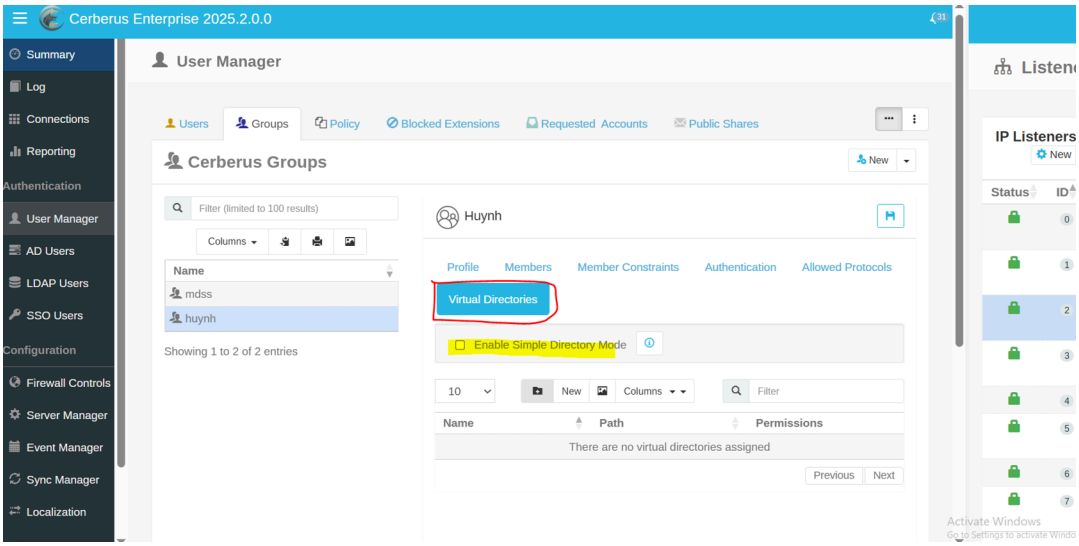


Synchronized AD users could be unable to map a Virtual Directory, to map a folder to multiple (AD) users, we could use a Cerberus group

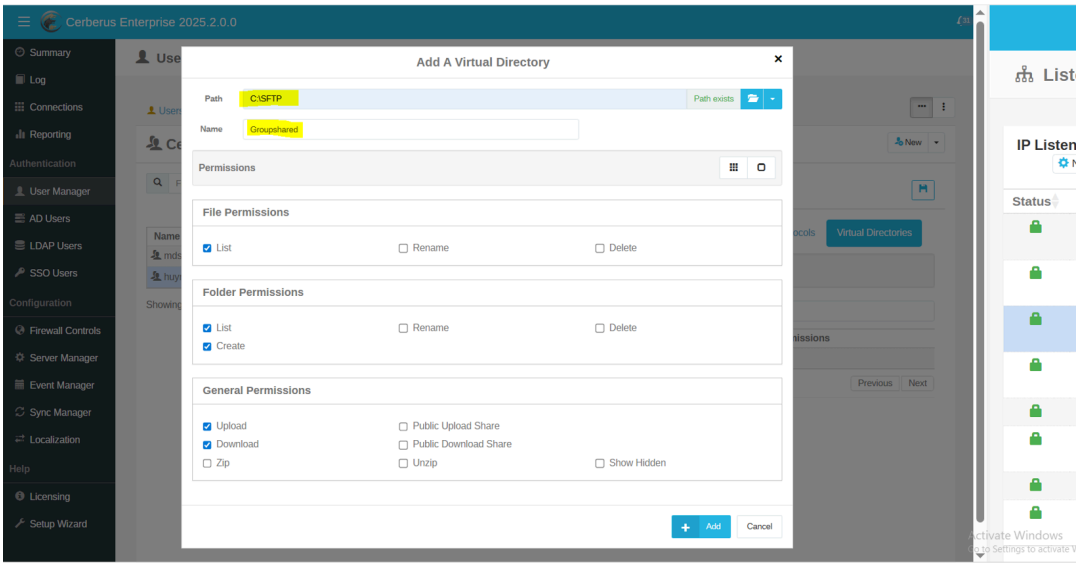
2. Add a group from User manager



3. Make sure Single mode is disabled at Virtual Directories tab for the group

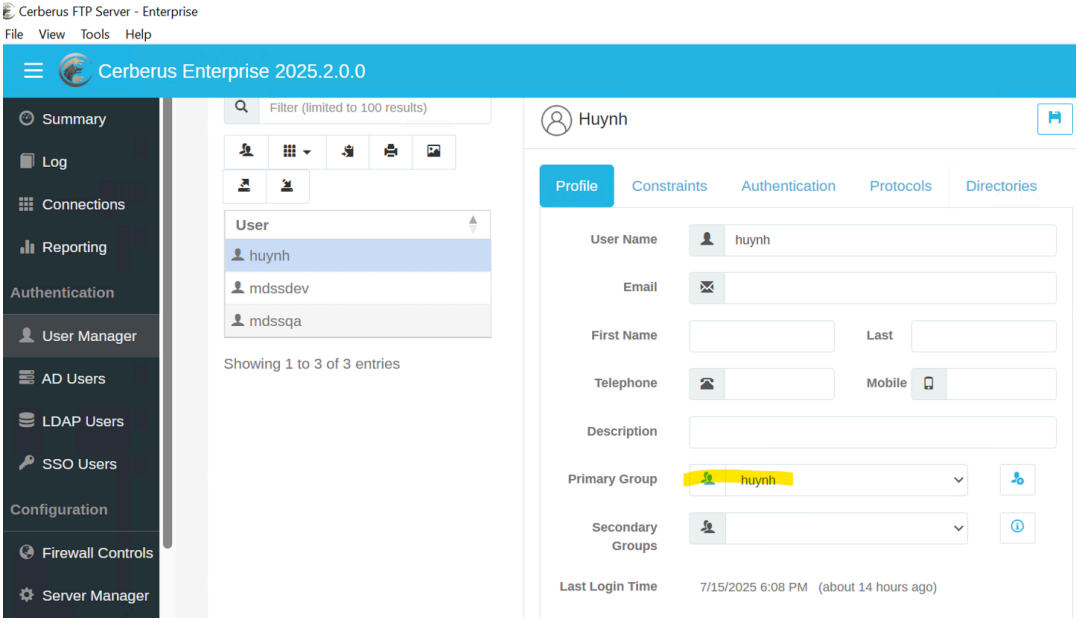


4. Create a New mapping, map a directory into the group

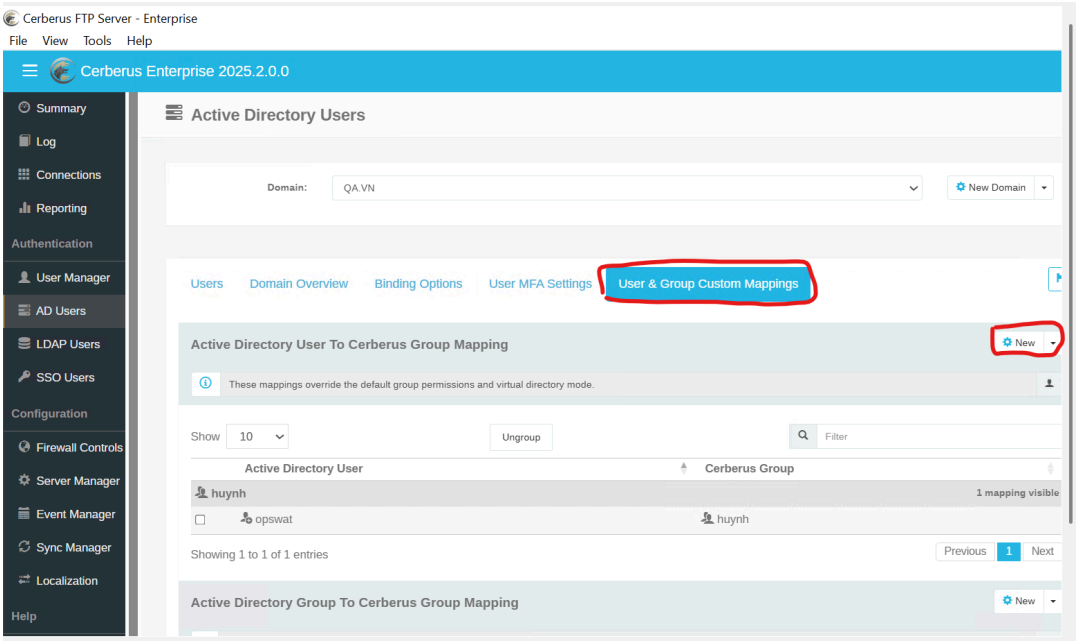


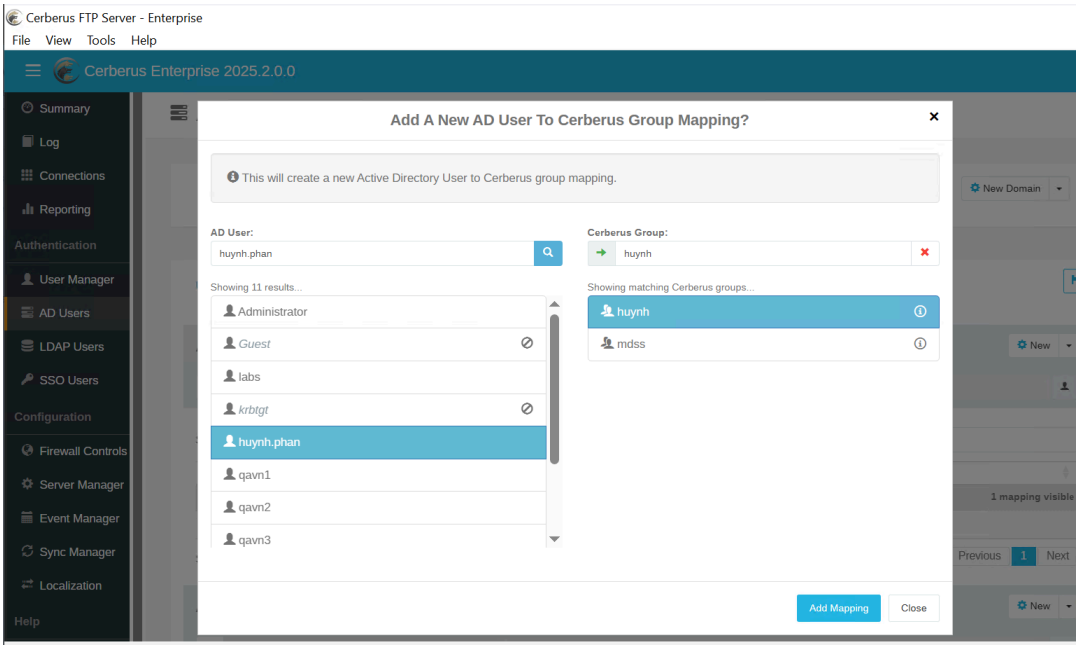
5. Assign the users into the created Cerberus group

a. Local users: go to User profile, and assign a primary group

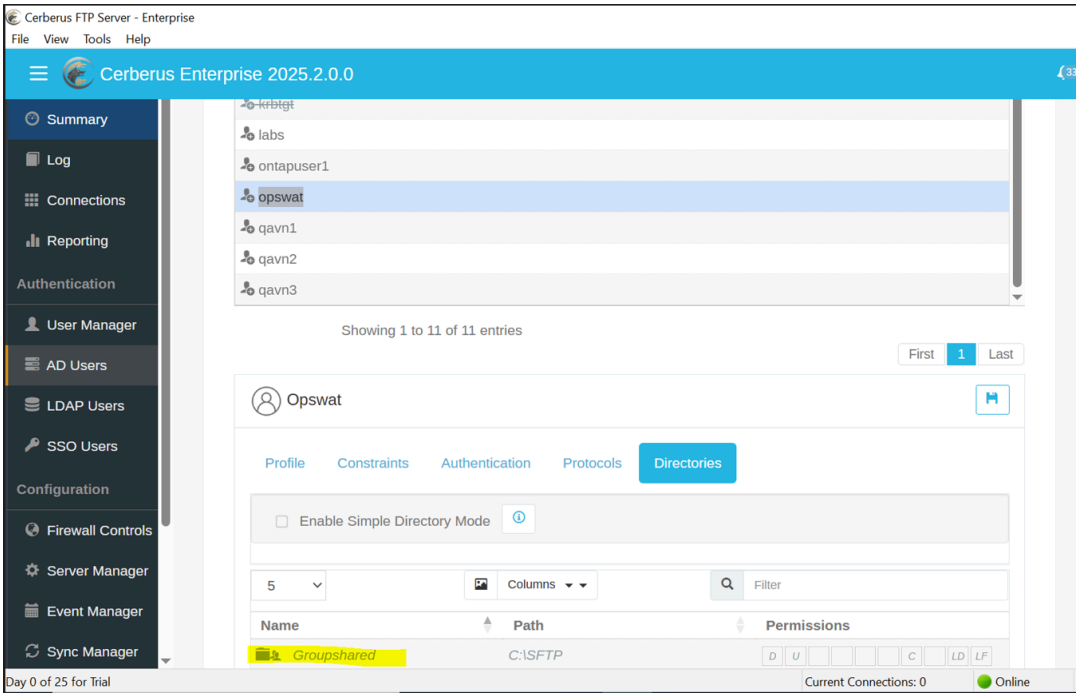


b. AD users





Head to the user's Directory, the group mapped folder should be displayed

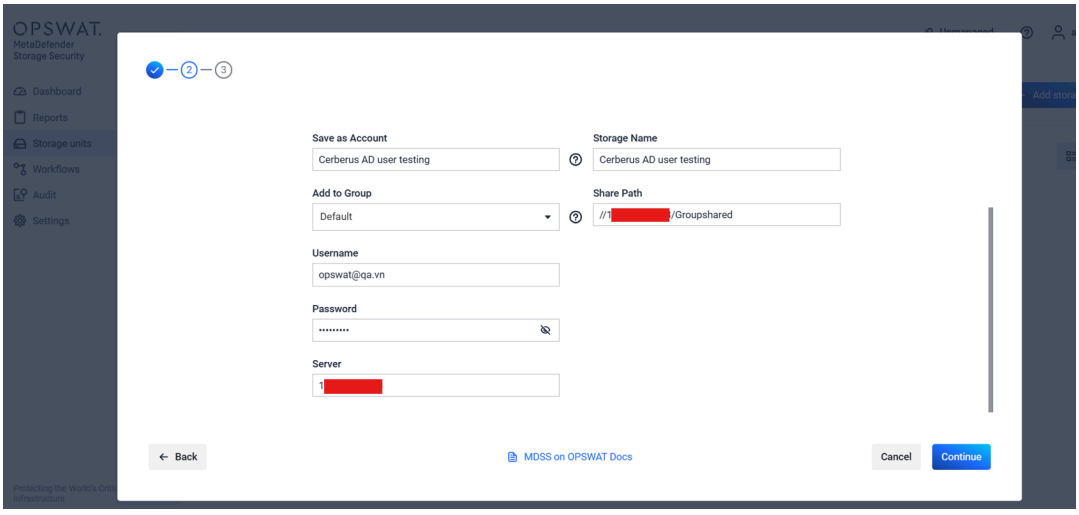


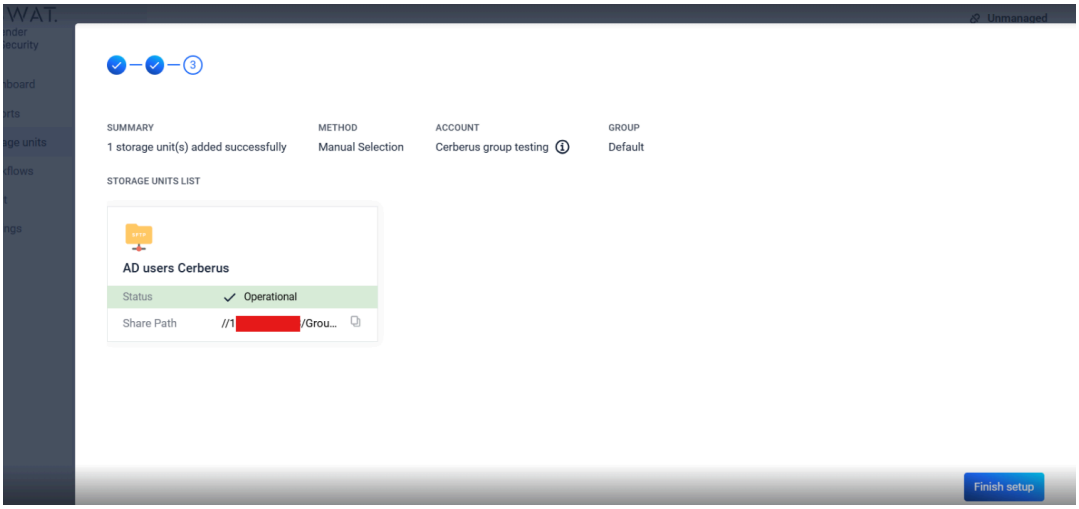
From then on, we could configure in MDSS the SFTP mapping Directory for the assigned users to the group with the share path format as //CerberusIP/Name

> **Info:**
>
> Format should be **username@domainame** for username

server:

Share Path ///Groupshared





Webhooks

We've prepared MDSS to be able to accept an API request sent by Cerberus

With the request, we need some information as well:

1. A header with an API key for auth
2. A payload containing a storageclient id and a metadata

Example:

```
{
"metadata": "{'Name': 'folder1/folder2/test-file.exe' }",
"storageClientId": '0c5454e3-d576-44c5-8ce4-cdc078011d0b'
}
```

> **Warning:**
>
> Scans with Deep CDR enabled have limited functionality, this is will be resolved in future releases.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

How to Handle SSL/TLS Issues with OPSWAT MetaDefender Core Webhook Callback?

Overview

This article describes how to resolve SSL/TLS certificate errors that can cause the webhook callback functionality to fail in OPSWAT MetaDefender Core version 5 on both Windows and Linux platforms.

Environment

OPSWAT MetaDefender Core version: 5

Supported Platforms: Windows, Linux

Problem Description

When the webhook callback functionality is invoked, it may fail due to SSL/TLS errors. The core.log file will contain an error similar to the following:

[core.webhook] SSL/TLS failed, issue='The issuer certificate of a locally looked up certificate could not be found, No certificates could be verified'

Root Cause

The webhook server's SSL certificate chain is not trusted by the operating system. This can occur if the intermediate or root certificate authorities are not installed in the local certificate store.

Resolution

Windows

1. Obtain the webhook server's SSL certificate chain (root CA, intermediate CA, and server certificate).
2. Open the Microsoft Management Console (mmc.exe).
3. Add the Certificates snap-in for the local computer.
4. Navigate to Trusted Root Certification Authorities > Certificates.
5. Import the root, intermediate and server certificates into the Trusted Root Certification Authorities store.
6. Restart the OPSWAT MetaDefender Core services:

- OSPWAT MetaDefender Core

After the service restart, the Trusted root certificates will be exported to a certificate located in the MetaDefender Core installation folder Data folder (by default C:\Program Files\OPSWAT\MetaDefender Core\data)

Linux

By default, MetaDefender Core loads root CAs from the following locations:

- /etc/ssl/certs/
- /usr/share/ssl/
- /etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem

Steps to import the certificates: __

Debian/Ubuntu

1. sudo cp -f <cert_file> /usr/local/share/ca-certificates/
2. sudo update-ca-certificates
3. sudo systemctl restart ometascan

CentOS/RHEL

1. sudo cp -f <cert_file> /etc/pki/ca-trust/source/anchors/
2. sudo update-ca-trust enable
3. sudo update-ca-trust extract
4. sudo systemctl restart ometascan

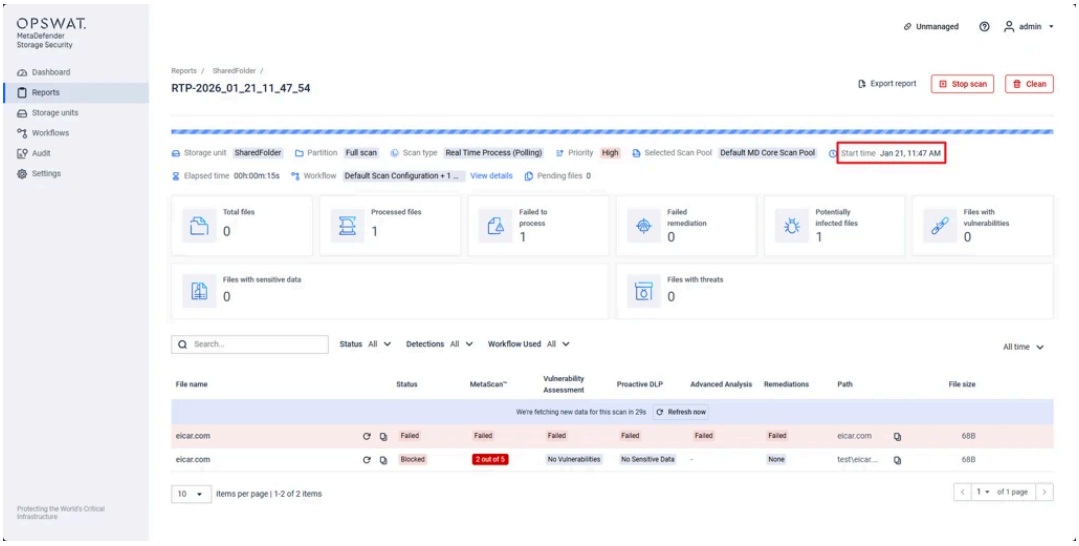
Validation

After completing the steps, trigger a webhook callback to confirm that the SSL/TLS error no longer occurs. Check the core.log for any remaining warnings or errors.

Why does MetaDefender Storage Security reset the RTP StartTime every day?

> **Check Your Version:**
>
> This article applies to all MetaDefender Storage Security v4.2.0 deployed on Windows and Linux systems.

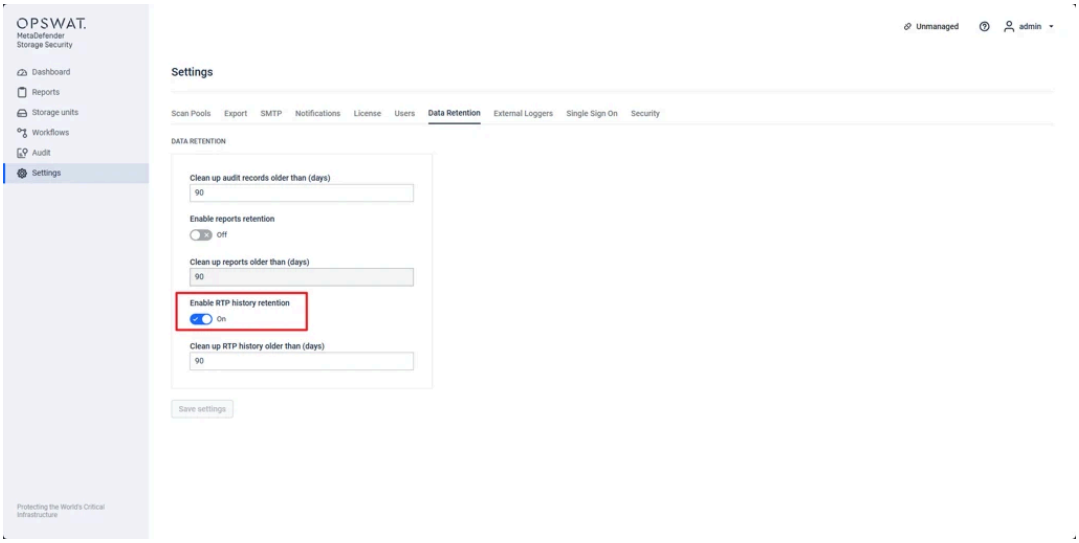
Issue: MetaDefender Storage Security resets the Real-Time Processing (RTP) Start Time every day.



Resolution: The reason for this behavior is enabling “Enable RTP history retention” under **Settings > Data Retention**.

This issue will be fixed in version 4.4.0.

Workaround: Currently, the workaround is to disable “Enable RTP history retention.”



> **Support:**
>
> If **Further Assistance** is required, please proceed to log a support case or chatting with our support engineer.

How to Exclude the OPSWAT Folder from Windows Defender?

> **Please Note**

> This article applies to MetaDefender Storage Security deployed on Windows systems.

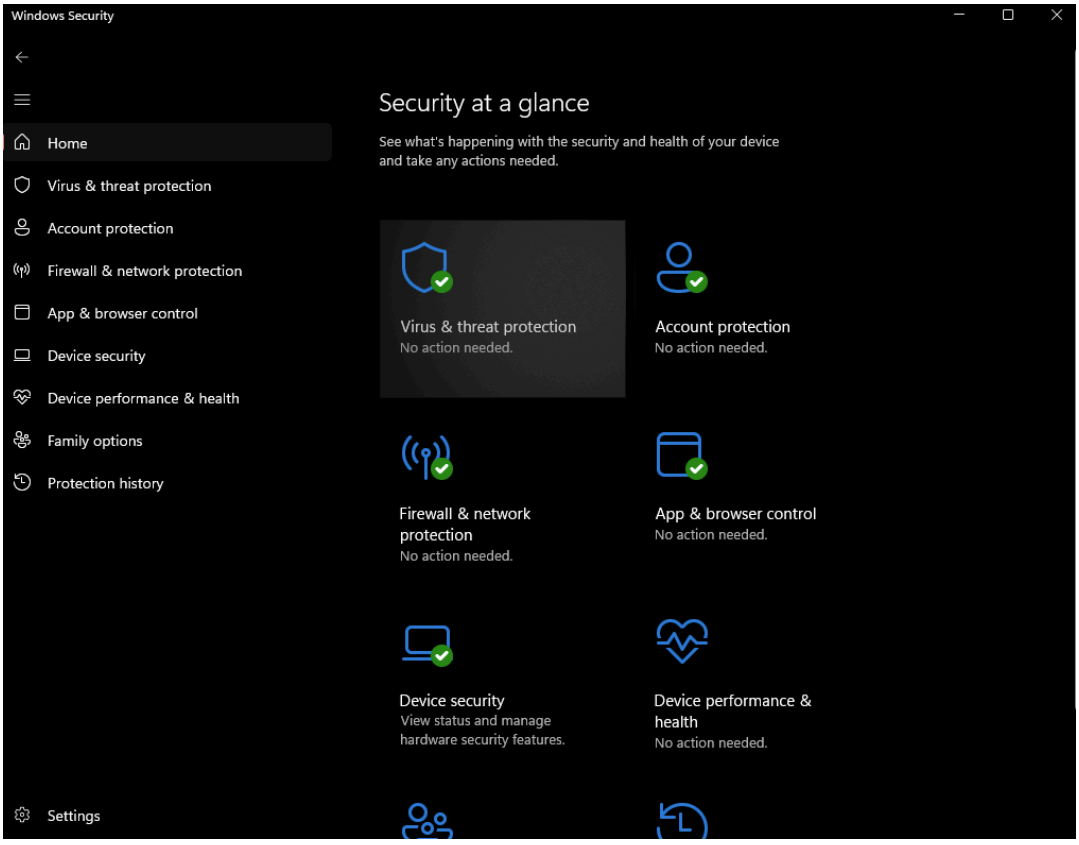
When using MetaDefender Storage Security deployed on Windows systems, it is important exclude the OPSWAT directory from **all** existing Anti-Malware software programs on the machine. Failure to exclude the OPSWAT directory can lead to performance degradation under high load, as the Anti-Malware software will also scan files that are streamed from the storage integration to MetaDefender Core, and may also slow down communication between certain components of the product.

Some common Anti-Malware software to perform these exclusions on are, including but not limited to: Windows Defender, McAfee, Norton, Bitdefender, Malwarebytes, SentinelOne.

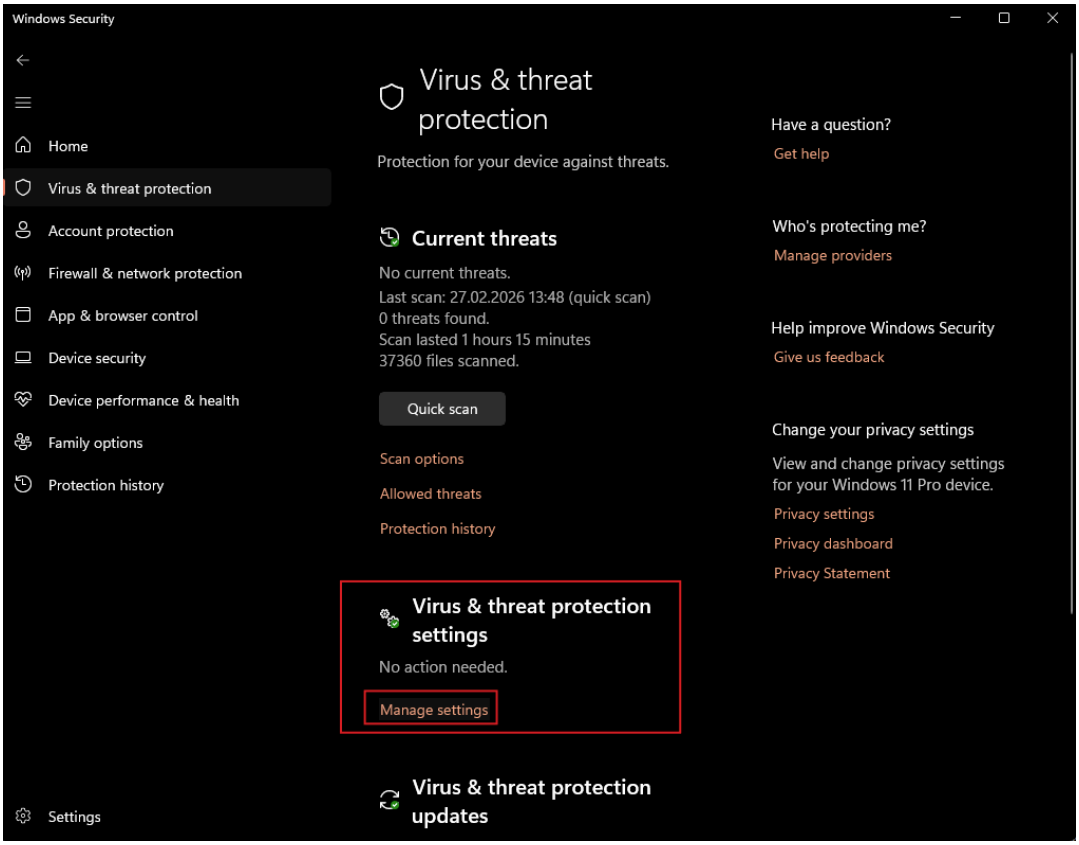
How to exclude on Windows Defender

Ensure you have **administrator privileges** before adding exclusions.

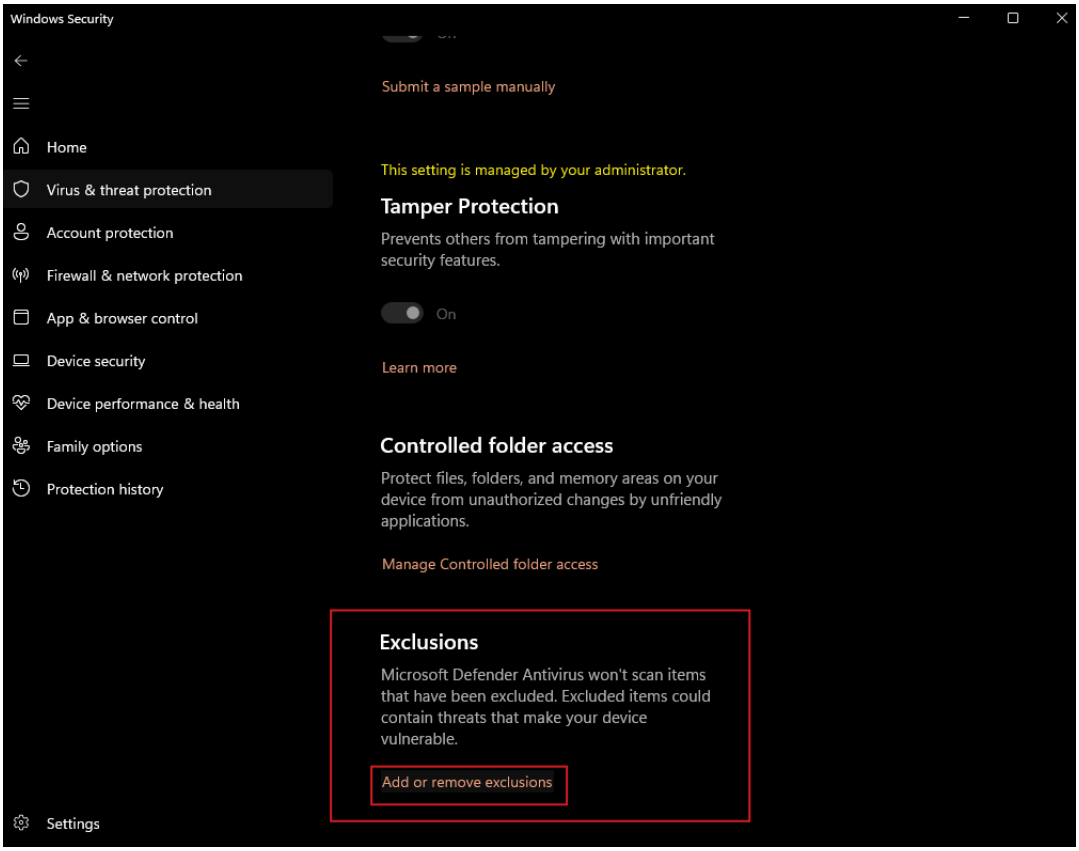
1. Navigate to “Windows Security” > “Virus & threat protection”



2. Under “Virus & threat protection settings” click “Manage settings”



3. Scroll down to “Exclusions” and click “Add or remove exclusions”



3. Add the “OPSWAT” folder. By default it is located in the installation directory: `C:\Program Files\OPSWAT`.
4. Add the “Temp” folder located at `C:\Users\<user>\AppData\Local\Temp`.

What are the Syslog Facilities and Levels in MetaDefender Storage Security?

> **Check Your Version:**
>
> This article applies to all supported versions of MetaDefender Storage Security.

A Syslog facility is a field within a Syslog message that identifies the source or category of the message, such as the kernel, mail system, or a specific application. This functions as a filter that actually instructs MetaDefender Storage Security to forward all the logs using the configured `Facility`.

These facilities, originally based on Unix processes and daemons, work with Syslog Levels (like Emergency, Critical, or Informational) to provide a more comprehensive way to manage and analyze logs in a system or network.

Numerical Code	Facility	Facility Description
0	kernel Messages	Messages generated by the operating system kernel
1	User Level Messages	Messages from user-level processes and applications
2	Mail System	Log entries related to mail servers and email processing
3	System Daemons	Messages from system daemons or background services
4	Security or Authorization Messages 1	Events related to user authentication and authorization, such as login or logout attempts
5	Internal Messages	Messages generated by the syslog daemon itself
6	Line Printer Subsystem	Messages concerning the printing system
7	Network News Subsystem	Log entries related to Usenet news services
8	UUCP Subsystem	Messages about Unix-to-Unix Copy (UUCP) communication
9	Clock Daemon 1	Messages from scheduled tasks or the cron daemon
10	Security or Authorization Messages 2	Private authentication-related events, often used for security-sensitive logs
11	FTP Daemon	Log entries for the FTP (File Transfer Protocol) service
12	NTP Subsystem	Messages from the Network Time Protocol daemon
13	Log Audit	General security-related events and authorization failures
14	Log Alert	Originating process or system component that sent a log message
15	Clock Daemon 2	-
16-23	Local User 0-7	User-defined facilities reserved for administrators or specific applications to

		use1414

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chat with our support engineer**.

How to protect the customer.env file in MetaDefender Storage Security?

> **Check Your Version:**
>
> This article applies to all MetaDefender Storage Security releases deployed on Windows and Linux systems.

The customer.env file stores the Postgres username and Postgres password in plaintext, these credentials are randomly generated at startup and are required for MetaDefender Storage Security to connect to the internal database that is not exposed outside of the machine.

Options for security:

- If you are connecting MDSS to an external database, it is recommended to setup **access control** for the external database at Client Authentication to ensure that only the server running MetaDefender Storage Security can access the database.
- The password must remain in plaintext, it cannot be hashed as it is sent in plaintext over the network, therefore ensure you have a **secure network**.
- Remove the `customer.env` file after MetaDefender Storage Security service starts, and place it back only when starting/stopping MetaDefender Storage Security service, or you may configure permissions for the file so that only authorized users have access to it, by default only **root user** has access to it.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

How Can I Improve Folder Discovery and Scanning Performance in MetaDefender Storage Security?

> **Check Your Product:**

> This article applies to all MetaDefender Storage Security releases deployed on Windows or Linux systems.

When using MetaDefender Storage Security (MDSS) to scan large or deeply nested folder structures, you may notice that the initial discovery process takes longer than expected. This article explains how MDSS performs folder discovery, why deep or complex directory structures can impact scanning speed, and how to optimize performance using event-based scanning.

The information provided applies to **all supported storage services** (e.g., Box, Amazon S3, Google Drive, local storage).

Discovery Service Workflow

MDSS uses a **Discovery Service** to locate and prepare files for malware scanning. The workflow proceeds as follows:

1. **Initial Discovery**
   - MDSS begins by enumerating all objects (files and folders) at the root level of the designated storage path.
2. **Analysis Phase**
   - Each discovered object is analyzed:
     - **Files** are immediately sent to MetaDefender Core for malware scanning.
     - **Folders** initiate a **recursive discovery** process, scanning their contents level by level.
3. **Sequential Traversal**
   - Discovery proceeds **hierarchically**, meaning that deeper folder structures take longer to complete because each subfolder is processed sequentially.

How to Improve Scanning Performance

If your environment contains a large number of folders or deep directory trees, the following methods can significantly reduce discovery and scanning time:

1. Implement Event-Based Scanning

Instead of relying solely on recursive discovery scans, use **Event-Based scanning** (also known as Real-Time Protection) to process files as they are created or modified.

Benefits:

- Files are scanned immediately, without full-folder traversal.
- Reduces overall discovery and processing time.

- Optimizes resource utilization and system performance.

2. Limit the Scope of Initial Discovery

If full discovery is required:

- Target only critical folders.
- Adjust scan depth (if applicable).
- Schedule discovery tasks during off-peak hours to minimize performance impact.

Additional Resources

- **Event-Based Handling Overview**
- **Event-Based Request Format**

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

How to create a PostgreSQL Database in AWS?

> **Check Your Version:**
>
> Affected product: MDSS 4.0.0 and newer

Overview:

MetaDefender Storage Security (MDSS) requires a PostgreSQL database to store configuration data, scan results, and operational metadata. This guide walks administrators through the complete process of provisioning a PostgreSQL instance on AWS (using Aurora or RDS), configuring the required database and permissions, and establishing the connection with MDSS.

Prerequisites

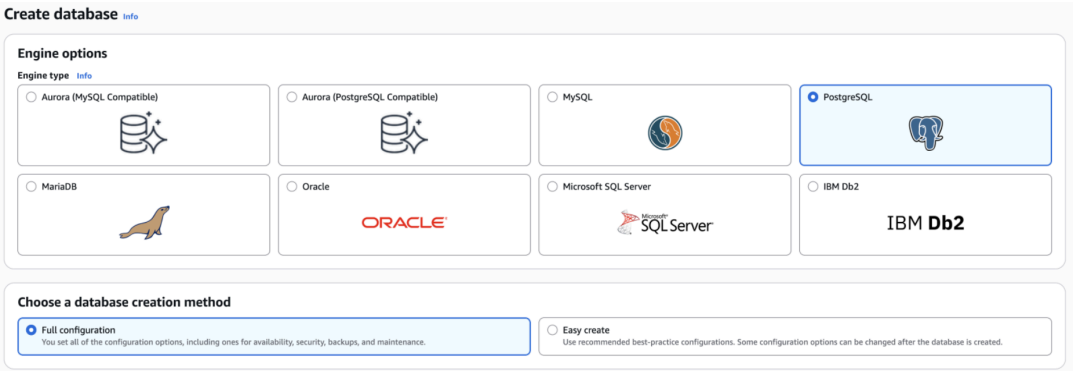
- An active AWS account with permissions to create RDS or Aurora PostgreSQL instances.
- Access to the target PostgreSQL server via pgAdmin or psql.
- MDSS 4.0.0 or newer installed, with access to its configuration files (customer.env).
- Network connectivity between the MDSS host and the AWS RDS/Aurora endpoint (security groups and firewall rules must allow traffic on port 5432).

Steps to configure:

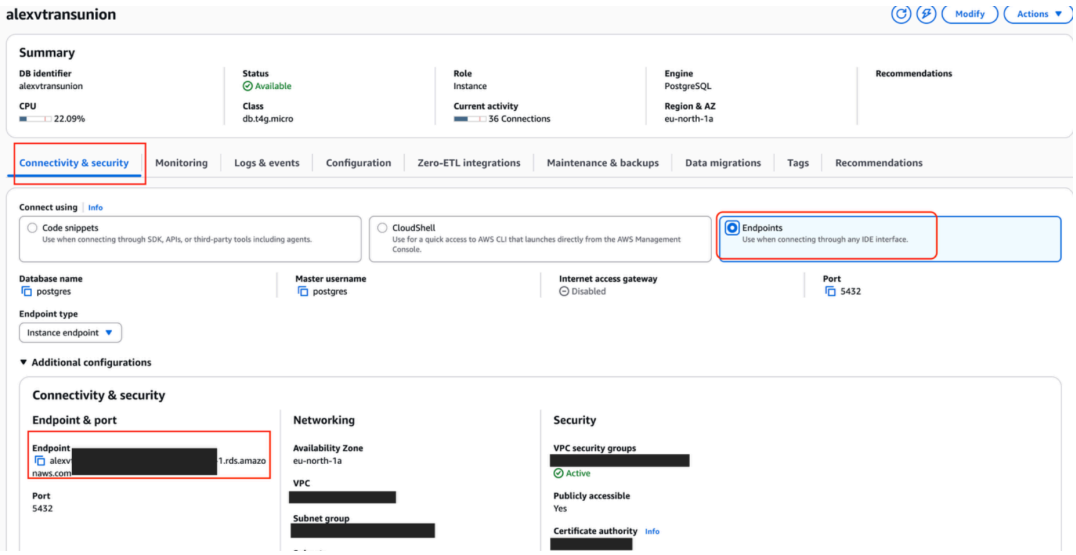
1. Create the PostgreSQL instance in AWS

In AWS, search for Aurora and RDS.

Select Create and choose full configuration.



After the instance is created, open the database details and copy the endpoint. This will be the PostgreSQL host. The default port is 5432. If the port has been changed, copy that value as well.

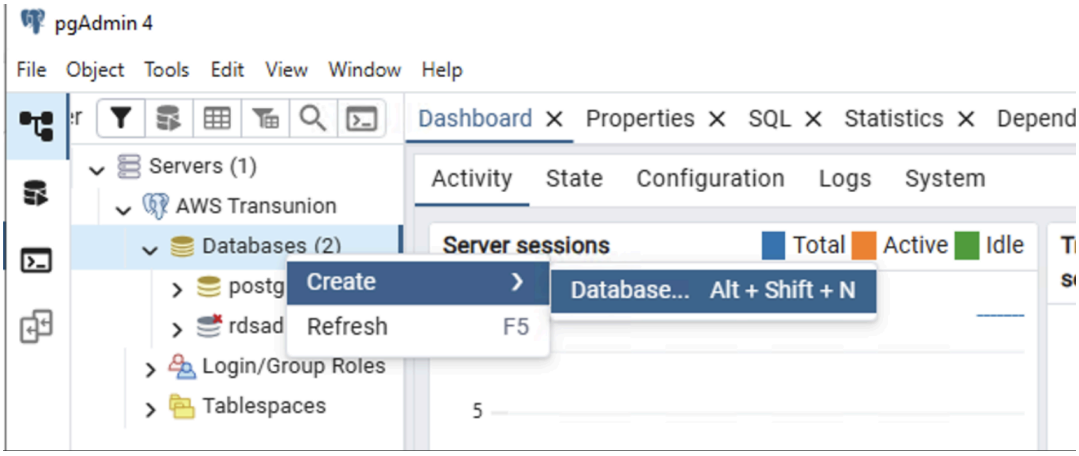


2. Create the MDSS database and assign permissions

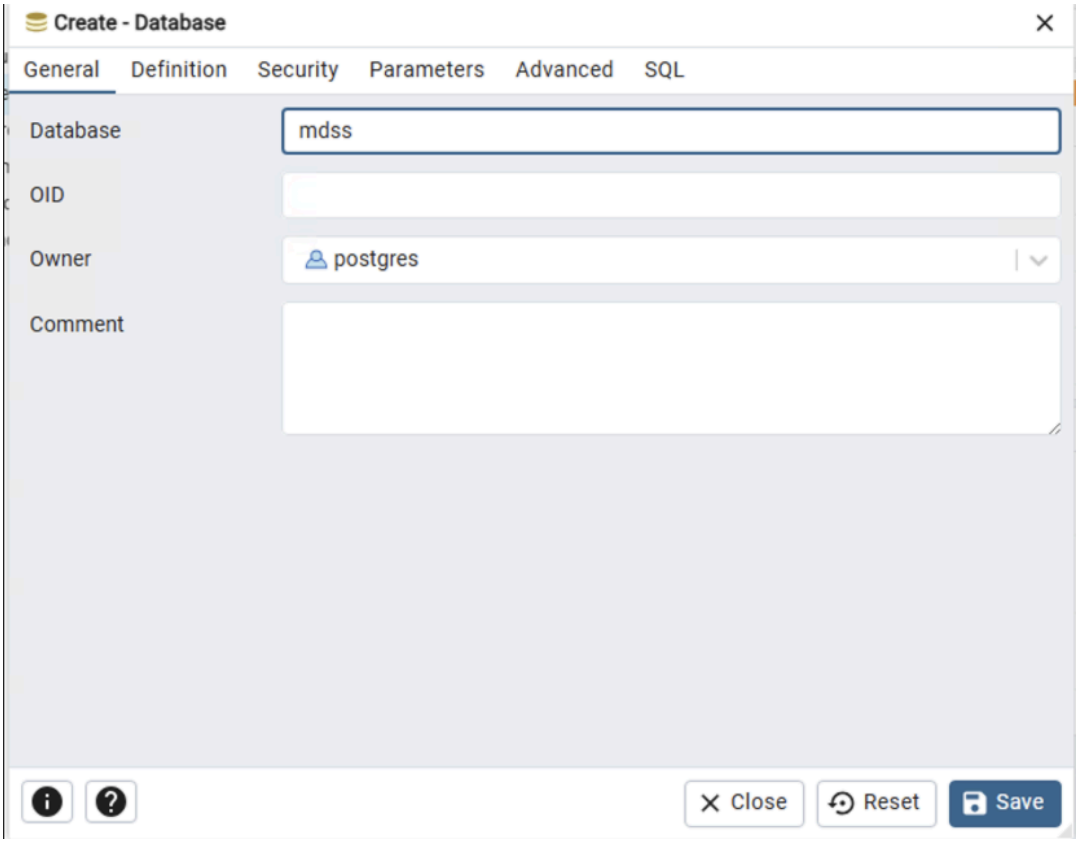
Option A: Using pgAdmin

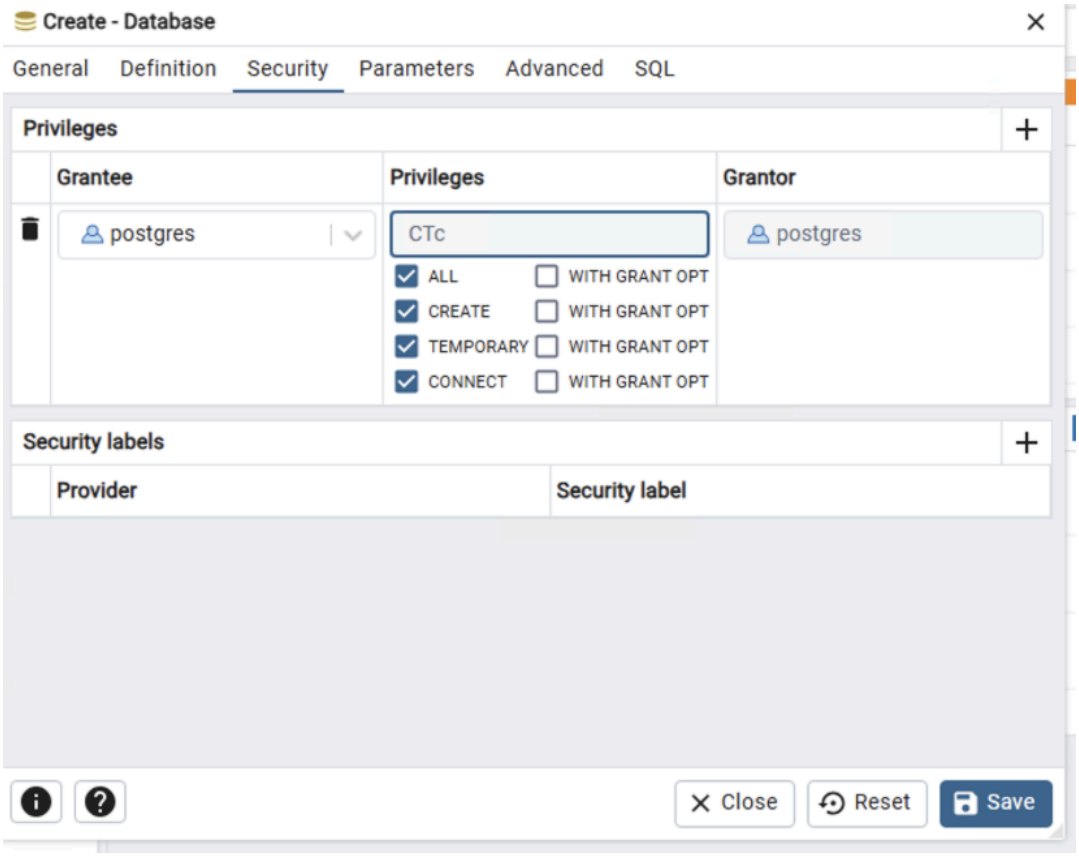
In pgAdmin, connect to the target PostgreSQL server.

Right-click Databases, then select Create > Database...



On the General tab, enter “mdss” as the database name. On the Security tab, assign the required privileges to the target user, then click Save.





Option B: Using psql

Use the following commands to connect, create the database, and grant the required permissions.

```
psql -h db.example.com -p 5432 -U myuser

 -- 1. Create database
CREATE DATABASE mdss;

 -- 2. Database connection
GRANT CONNECT ON DATABASE mdss TO myuser;

 -- 3. Schema permissions
GRANT CREATE, USAGE ON SCHEMA public TO myuser;

 -- 4. Full privileges on existing objects

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO myuser;
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO myuser;
GRANT ALL PRIVILEGES ON ALL FUNCTIONS IN SCHEMA public TO myuser;
GRANT ALL PRIVILEGES ON ALL PROCEDURES IN SCHEMA public TO myuser;

 -- 5. Default privileges for new objects
ALTER DEFAULT PRIVILEGES IN SCHEMA public
GRANT ALL PRIVILEGES ON TABLES TO myuser;

ALTER DEFAULT PRIVILEGES IN SCHEMA public
GRANT ALL PRIVILEGES ON SEQUENCES TO myuser;

ALTER DEFAULT PRIVILEGES IN SCHEMA public
GRANT ALL PRIVILEGES ON FUNCTIONS TO myuser;

 -- 6. For encryption extensions (if needed)
GRANT USAGE ON SCHEMA pg_catalog TO myuser;
```

3. Configure the new PostgreSQL server in MDSS

Add the connection details to customer.env.

```
POSTGRES_USER=user
POSTGRES_PASSWORD=password
POSTGRES_DB=mdss
POSTGRESQL_URL=Host=postgres;Port=5432;Username=user;Password=password;
Database=mdss
```

Save the file and start the MDSS service.

Verify functionality:

Once the MDSS service has started, use the following checks to confirm that the database connection is working correctly.

1. Check the database connection directly from the MDSS server

From the MDSS host or any machine with network access to the RDS/Aurora endpoint, run the following command to verify that the database is reachable and the credentials are valid:

```
psql -h <POSTGRES_HOST> -p 5432 -U <POSTGRES_USER> -d mdss -c
"\conninfo"
```

A successful response will display the connection details, confirming that the user, host, port, and database are all correct. If the connection fails, verify that the endpoint, credentials, and AWS security group rules are configured correctly.

2. Verify the MDSS web interface

Open a browser and navigate to the MDSS management interface. If the database connection is healthy, you will see the user configuration wizard. Any database connectivity issues will typically surface as a page showing errors connecting to the database.

> **Support:**
>
> If Further Assistance is required, please proceed to create a support case or chat with our support engineer.

How to set up Active Active configuration?

> **Check Your Version:**
>
> This article applies to all MetaDefender Storage Security version deployed in both Linux or Windows.

Summary: Guide for Configuring MDSS Active-Active Setup

Description: Below is a detailed guide to configure an Active-Active setup for MDSS, along with a recommended startup approach to ensure stability.

Step 1 — Deploy External Services

Set up the following services on dedicated servers (separate from MDSS nodes):

- **PostgreSQL** — Install and create the MDSS database and user
- **RabbitMQ** — Install and create a user with appropriate permissions
- **Redis** — Install and configure a password for remote access

Reference documentation: External Services Configuration - MetaDefender Storage Security

---

Step 2 — Configure MDSS to Use External Services

On each MDSS node, update the `/etc/mdss/customer.env` file:

```
# External RabbitMQ
RABBITMQ_URI=amqp://<user>:<password>@<rabbitmq_host>:5672
RABBITMQ_HOST=<rabbitmq_host>
RABBITMQ_PORT=5672
RABBITMQ_DEFAULT_USER=<user>
RABBITMQ_DEFAULT_PASS=<password>

# External Redis
CACHE_SERVICE_URI=<redis_host>:6379,user=default,password=
<password>,syncTimeout=10000
CACHE_SERVICE_URL=<redis_host>
CACHE_SERVICE_PORT=6379

# External PostgreSQL
POSTGRES_HOST=<postgres_host>
POSTGRES_PORT=5432
POSTGRES_DB=mdss
POSTGRES_USER=<user>
POSTGRES_PASSWORD=<password>
```

Step 3 — Disable Bundled Services

Since external services are used, disable the built-in services:

`DISABLED_SERVICES=postgres,rabbitmq,redis`

This ensures each MDSS node does not start local instances.

Step 4 — Start MDSS Nodes (Recommended Approach)

To ensure stable initialization:

1. Start only **one MDSS node** first
2. Verify all services are running properly
3. Perform a test scan (e.g., upload a sample file)
4. Once validated, start additional nodes using the same configuration

---

Step 5 — (Optional) Load Balancer

A load balancer is optional but recommended:

- Provides high availability for Web UI/API via a single endpoint
- Should include health checks (e.g., port 443)

Note: Not required for scanning, as workload distribution is handled via RabbitMQ.

Step 6 — (Optional) Automate Active-Passive Failover

If your requirement is failover rather than parallel processing, a reference script is available that monitors the active node and triggers a scan on the passive node once the active one stops responding:

OPSWAT/mdss-active-passive-failover

Configuration is done entirely through `PASSIVE_MODE_*` environment variables (required: active and passive URLs and ports, API key, storage ID, workflow ID and scan name; optional: priority, check interval, request timeout and failure threshold). The repository README documents the prerequisites, every variable and how to run the script as a service.

How It Works

- File events (e.g., RTP) are published to the shared RabbitMQ queue
- The first available MDSS node processes the request
- All nodes share the same PostgreSQL database, ensuring consistent state and no duplicate scanning

> **Support:**
>
> If Further Assistance is required, please proceed to **create a support case or chat with our support engineer.**

AWS Lambda with S3 Trigger for MDSS Real-Time Processing

> **Prerequisites**
>
> Before proceeding with the Lambda function creation, ensure the following requirements are met:

1. IAM User Permissions

- The IAM user you are using for Lambda creation must have the following permissions, if you are using an admin user then ignore this:
  - Create Lambda functions
  - Assign IAM roles
  - Configure S3 triggers
  - Access the target S3 buckets

2. Account-Level Verification

- Confirm there are no account-level naming restrictions that would prevent your chosen Lambda function name
- Verify your AWS account has sufficient permissions and service quotas to create new Lambda functions in the desired region
- Ensure you have access to the target AWS region where the Lambda function will be deployed

3. S3 Bucket Configuration

- Verify that S3 bucket policies are configured to allow access from the Lambda function's execution role
- Required S3 permissions may include:
  - s3:GetObject - to read objects from the bucket
  - s3:PutObject - to write objects to the bucket (if applicable)
  - s3:DeleteObject - to delete objects from the bucket (if applicable)

4. IAM Execution Role

- Ensure you have a pre-configured IAM role with:
- Lambda execution permissions
- S3 access permissions for your target bucket(s)
- CloudWatch Logs permissions for function monitoring

Create IAM execution role for Lambda

First create a policy that will be used by the LAMBDA execution role.

Go to IAM → Policies → Create Policy. Give it a name, for example: MDSSLambdaS3AccessPolicy and use the following JSON

```
{
  "Version": "2012-10-17",
  "Statement": [

    {
      "Sid": "S3BucketListAccess",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucketname"
      ]
    },
    {
      "Sid": "S3ObjectAccess",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucketname/*"
      ]
    },

    {
      "Sid": "CloudWatchLogsAccess",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    }
  ]
}
```

Now we need to create an IAM role that will be used by the Lambda function

Got to

- IAM → Roles → Create Role
- Trusted entity: AWS Service → Lambda
- Attach policy: Attach the policy you created above MDSSLambdaS3AccessPolicy
- Role name: Give it a name, for example MDSSLambdaExecutionRole

Create Lambda Function

Now navigate lambda in AWS management console

Click “Create function”

- Select “Author from scratch”
- Function name: Example: MetaDefenderStorageSecurityProcessor
- Runtime: Select Python (choose the latest compatible version)
- Under change default execution role : Execution role → Use another role → select the role you created for lambda

Configure Advanced Settings (Optional but Recommended)

1. Expand “Advanced configuration”
2. Enable “Tags”
3. Add relevant tags for resource organization:

- Key: Purpose | Value: MetaDefenderScan
- Key: Environment | Value: Production/Development
- Key: Owner | Value: [Your Team/Department]

Click create function

Configure S3 trigger

In the newly created lambda Lambda go to → Configuration → Triggers → Add Trigger

Under trigger configuration -> select a source, choose S3 from the drop down

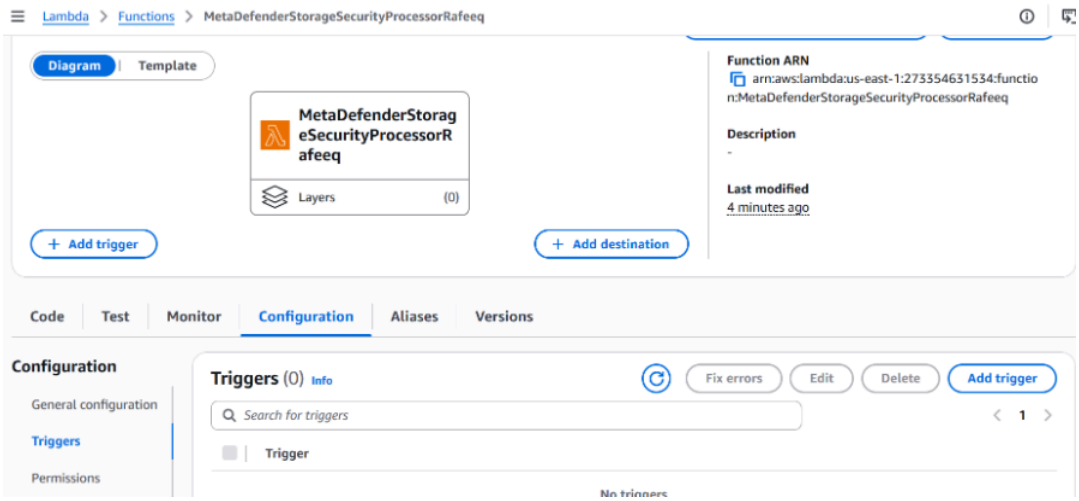
Select your bucket from the drop down

Under Event types: Choose the appropriate event type (Default is "All object create events" which will work for the Lambda). Alternative options: Object create, delete, or restore events based on your requirements.

Prefix (Optional): Specify a prefix to filter objects by path. Example: uploads/ to only trigger on objects in the uploads folder

Suffix (Optional): Specify a suffix to filter objects by file extension. Example: .pdf to only trigger on PDF files

Recursive invocation: Check this option to acknowledge potential recursive invocations

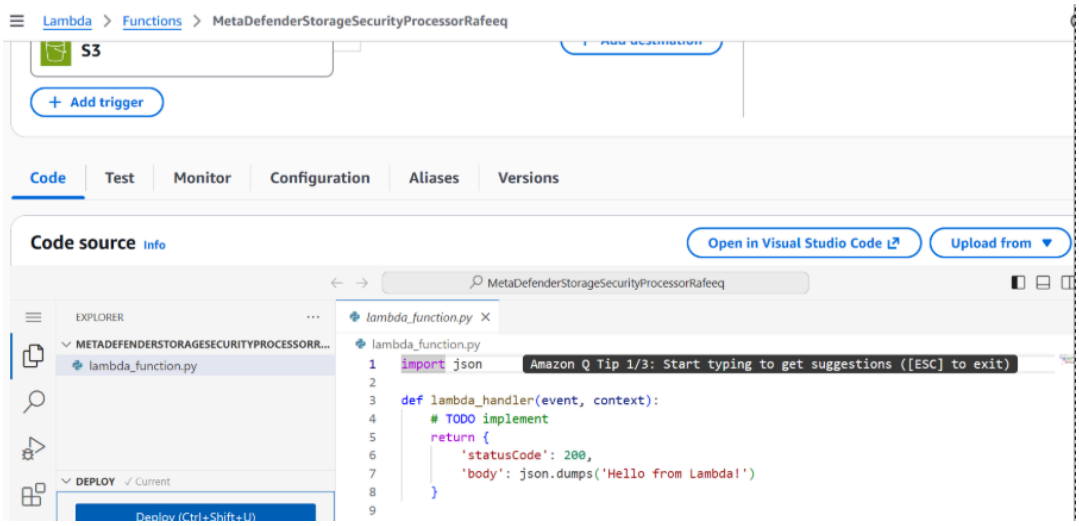


Click Add

Deploy Lambda Code

Now we need to update Lambda with the code that it should execute when it gets triggered

For that, go to code section of your lambda function



Delete the default code in there and copy paste the following code

```
import os
import json
import urllib.request
from urllib.parse import unquote
def lambda_handler(event, context):
    for eventRecord in event['Records']:
        eventRecord['s3']['object']['key']=
unquote(eventRecord['s3']['object']['key'].replace("+", " "))
        payload = {
            'metadata': json.dumps(eventRecord),
            'storageClientId': os.getenv('STORAGECLIENTID', "")
        }
        req = urllib.request.Request(
            url=os.getenv('APIENDPOINT', ""),
            data=json.dumps(payload).encode('utf-8'),
            headers={
                'ApiKey': os.getenv('APIKEY', ""),
                'Content-Type': 'application/json'
            },
            method='POST'
        )
        urllib.request.urlopen(req)
```

Click deploy to save your code

Configure environment variables for Lambda

Now we need to set environment variables for the lambda function so that we can configure MDSS endpoint with lambda

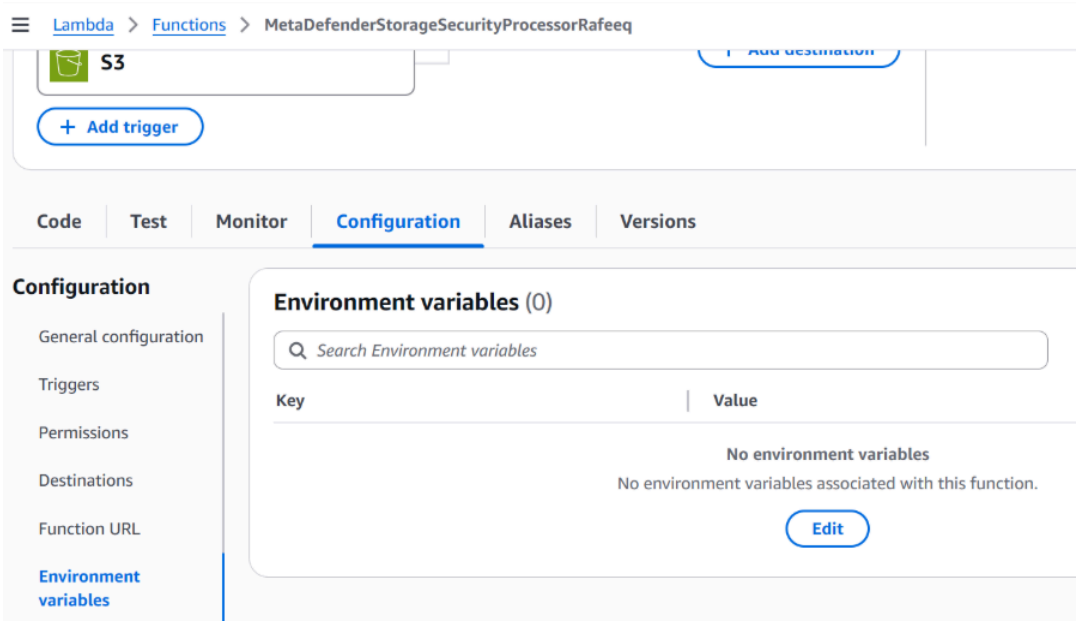
Go to environment variables under your lambda function

Add the following environment variables

APIENDPOINT – Your MDSS URL + /api/webhook/realtime Example: https://mdss-example.com/api/webhook/realtime

APIKEY – An api key generated from MDSS (Go to MDSS -> click on the dropdown next your profile -> click on profile information -> select configure api key -> click on generate api key -> click on copy and save)

STORAGECLIENTID -> Your storage client ID from MDSS. To obtain: Navigate to your desired storage configuration and copy the storageClientId (Click on storage units option in MDSS -> under listed storage units you will be able to see storage client ID and a copy button next to it)



Setup RTP scan on MDSS

Once lambda setup is done, we need to configure a new RTP scan on MDSS

Open MDSS web console → configure scan → select Real time processing → Type: Event based → select the storage unit → set priority → set optional scan name and partition → choose the workflow → Start RTP

Test

Upload a file to the s3 bucket and you should be able to see the file getting processed MDSS

How to configure SMTP server on MetaDefender Storage Security to send notification to Gmail mailbox?

> **Check Your Product:**
>
> This article applies to MetaDefender Storage Security releases version 3.3.0 or newer deployed on Windows or Linux systems.

To be able to send notification to your Gmail mailbox from MetaDefender Storage Security the following fields should be configured properly under MetaDefender Storage Security:

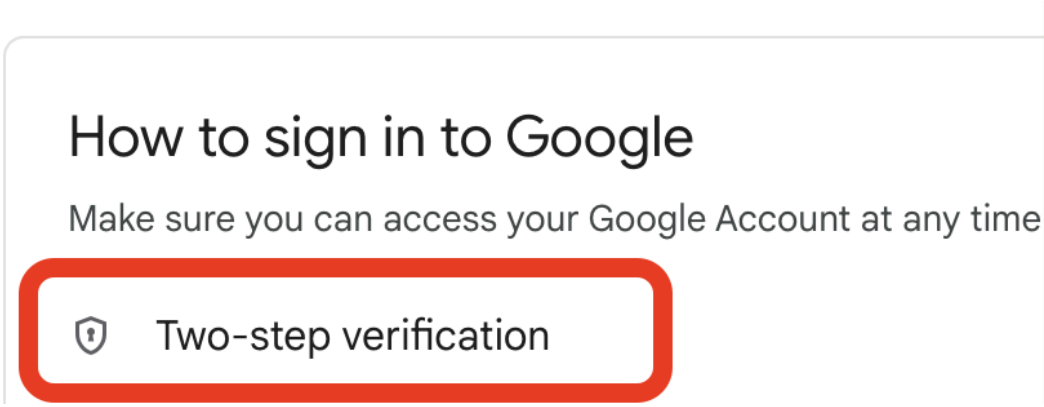
- HOST: smtp.gmail.com
- Port: 587
- Base URL: the address to use in the email for links; this address should point to the MetaDefender Storage Security web interface
- Sender email address (Optional): Ex.: MDSS_no-reply@test.com

- Sender name (Optional): Ex.: MDSS_No-Reply
- Username: username of the Gmail mailbox used for SMTP integration.
- Password: App password created for Gmail mailbox provided under “Email” field.
- Please choose a security option: Auto

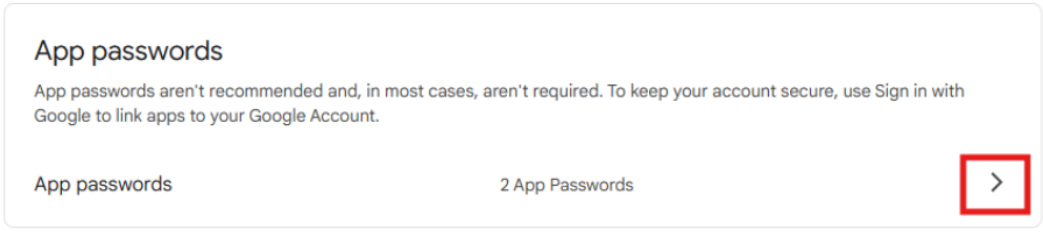
In order to achieve that, the following configuration should be done on Gmail mailbox account:

1. Create App password for Gmail account

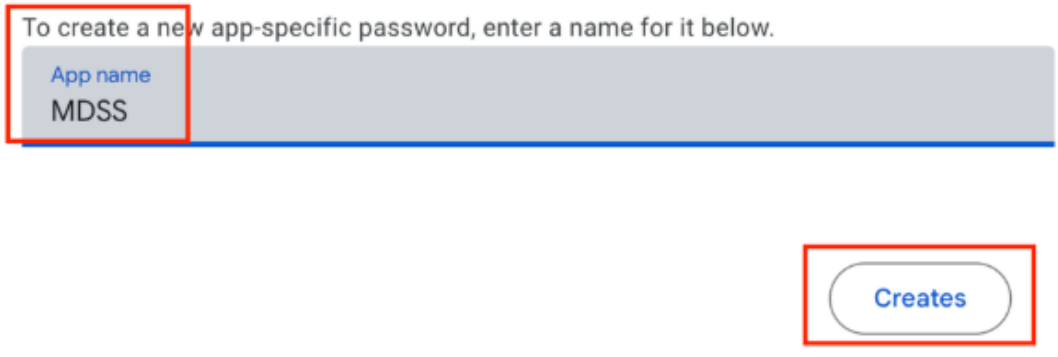
- Login to https://myaccount.google.com/security
- Under “Signing in to Google”:



- Click on Two-Step Verification
- Scroll down to section “App passwords” and click arrow in front of “App passwords”

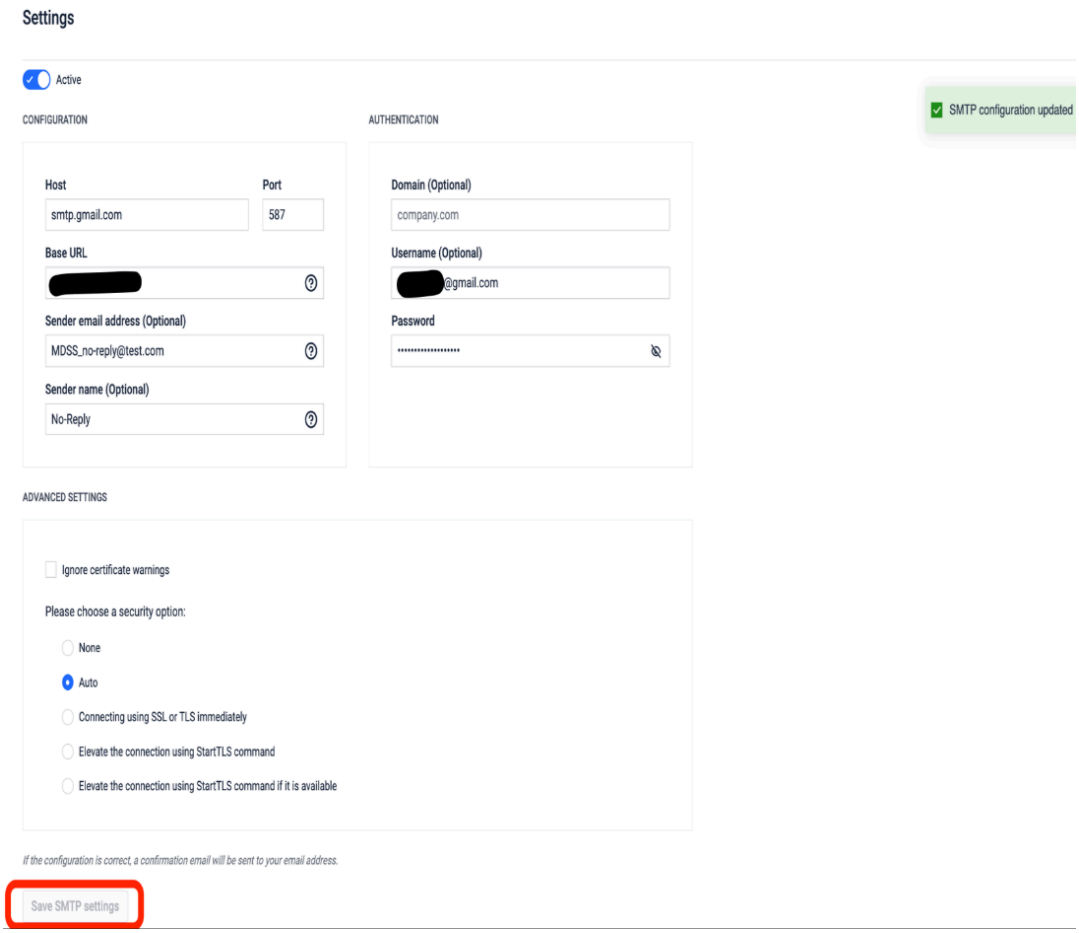


- In new page opened, write a name for your App password and click “Creates”



- Save the App password generated and click “Finished”. Further we will use the App password generated as password for mailbox used for MetaDefender Storage Security configuration as mentioned above and as in below example.

Example:



IMPORTANT: For field “Password” use App password created under Gmail account and not use the password which you use to login on Gmail mailbox.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

What are the Maximum and Minimum Data Retention Periods for MetaDefender Storage Security?

> **Check Your Version:**

> This article applies to all MetaDefender Storage Security releases deployed on Windows and Linux systems.

MetaDefender Storage Security allows you to define how many days scan reports are retained.

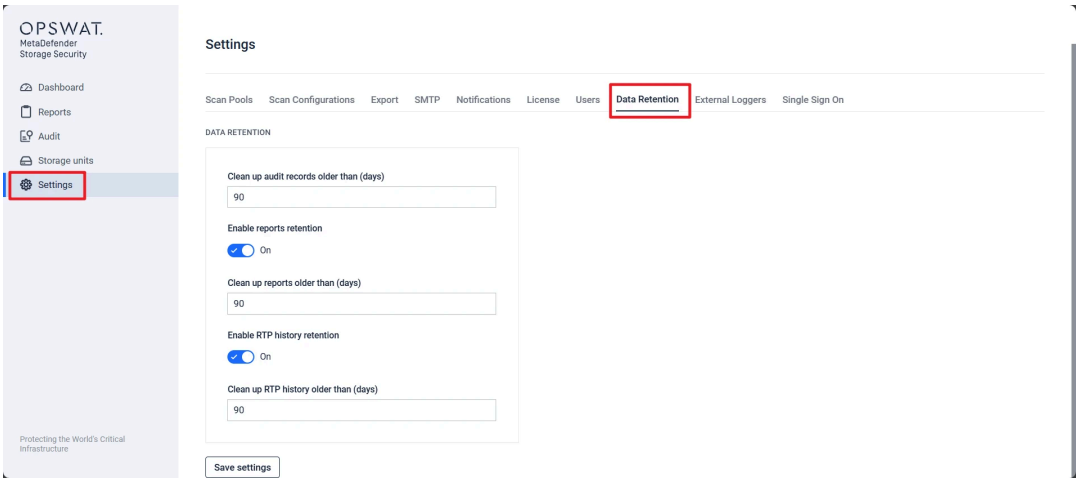
Data Retention Days Range

- **Maximum value:** 50,000 days
- **Minimum value:** 1 day

These values define the range you can enter when configuring the Data Retention Days setting in the MetaDefender Storage Security UI.

Steps to Configure

You can set the retention period by navigating to: Settings > Data Retention



For further details on data retention settings, refer to: Database Retention - MetaDefender Storage Security

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a support case or chatting with our support engineer.

How to change PostgreSQL DB port used by MDSS

Steps

1. Stop MDSS Services

Open Services (services.msc) and stop the following:

- MDSS service
- PostgreSQL service
- Mdss_rabbitmq

2. Edit Configuration
3. Navigate to:

C:\Program Files\OPSWAT\MetaDefender Storage Security\config

2. Open customer.env in a text editor.
3. Update the following parameters:

POSTGRESQL_PORT=<new_port> # e.g., 5435
POSTGRESQL_URL=jdbc:postgresql://localhost:<new_port>/mdss

4. Save and close the file.
5. Refresh Environment
6. Open PowerShell as Administrator.
7. Run:

cd “C:\Program Files\OPSWAT\MetaDefender Storage Security\scripts” .\mdss.ps1 -c refresh

This refreshes environment variables and restarts MDSS services.

4. Start Services

If not already started, manually start the services stopped in Step 1.

5. Verify

Confirm MDSS is listening on the new port:

netstat -ano | findstr :<new_port>

6. Troubleshooting

- Check MDSS logs:

C:\Program Files\OPSWAT\MetaDefender Storage Security\logs

- Ensure no other application is using the new port.

> **Support:**
>
> If Further Assistance is required, please proceed to log a support case or chatting with our support engineer.

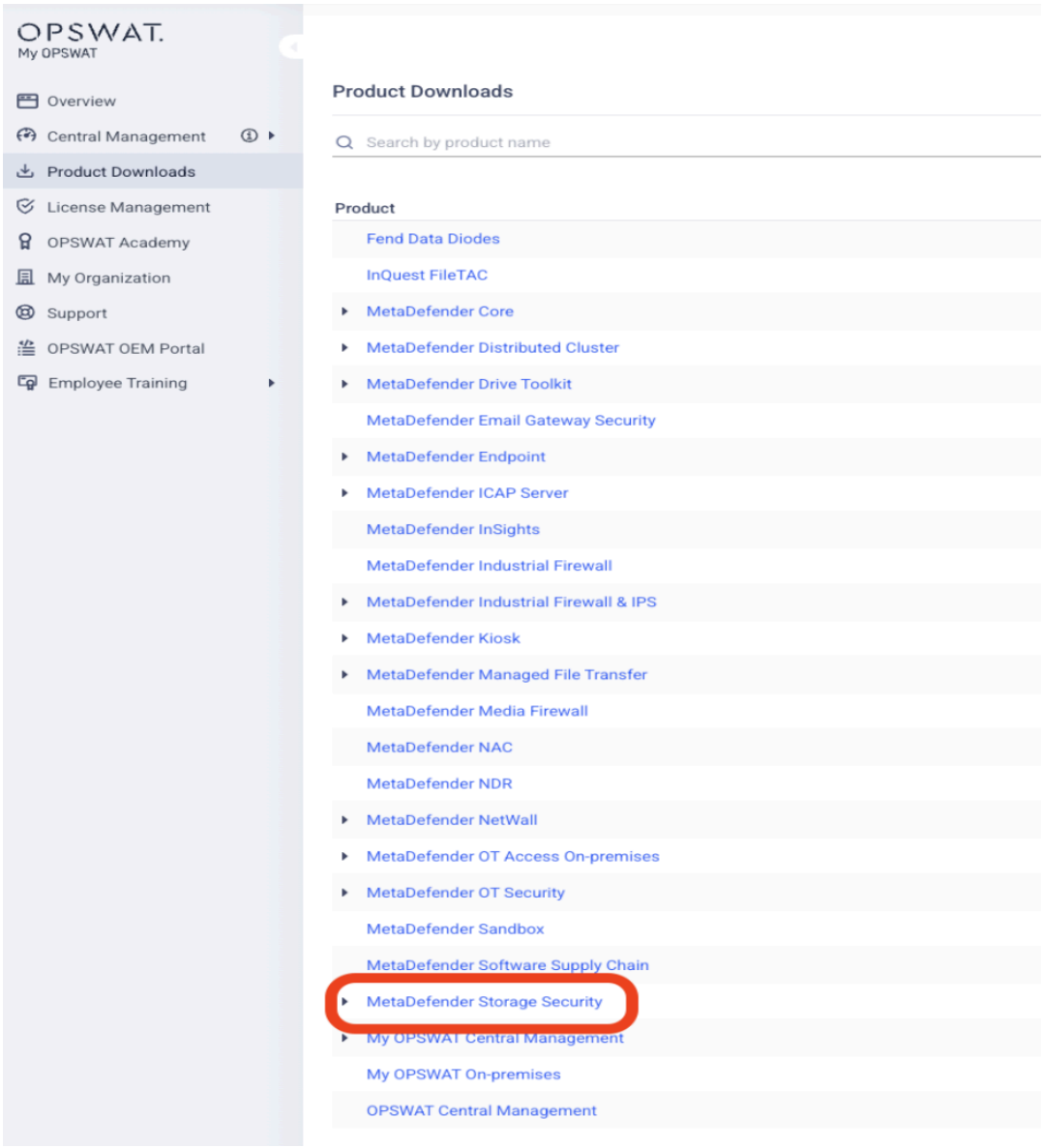
What is the latest MetaDefender Storage Security version?

> **Check Your Version:**
>
> This article applies to all MetaDefender Storage Security releases deployed on Windows and Linux systems.

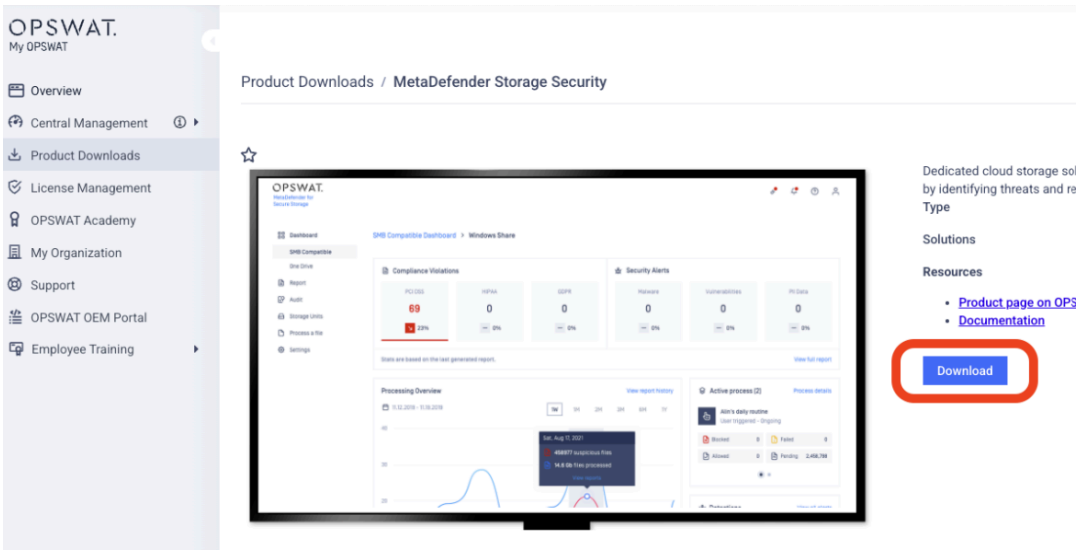
Browse MetaDefender Storage Security Release Notes for details on the latest version, version history, version comparisons and release notes.

Alternatively, go directly through the My OPSWAT for a list of available downloads.:

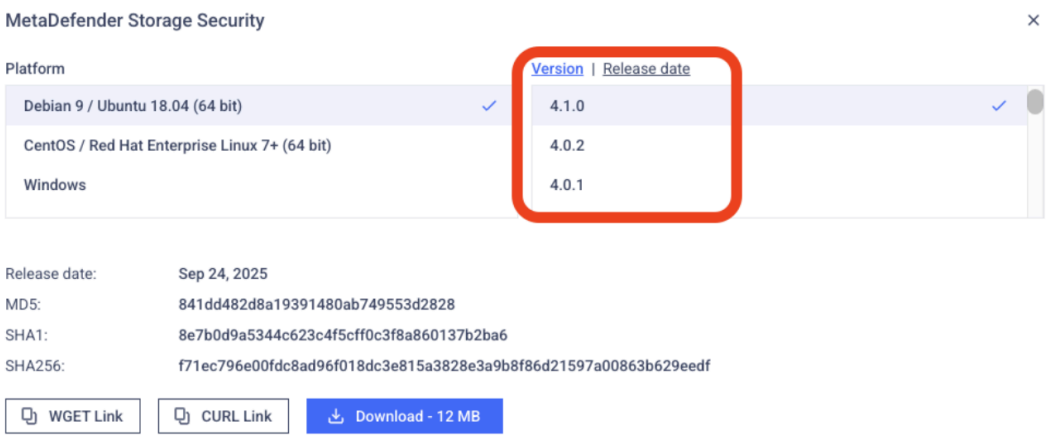
1. Login to My OPSWAT portal
2. Click on “MetaDefender Storage Security”



3. Click on “Download” button



4. In the new window opened, always the version from the top is the latest version



Pro Tip:

OPSWAT highly recommends that users upgrade to the latest version of MetaDefender Storage Security. Current versions incorporate state-of-the-art features and necessary bug fixes, and are freely available for all license holders to download.

Support:

Feel free to contact OPSWAT support for further guidance through your Upgrade process. Please follow these instructions on How To Create a Support Package, before logging a Support Ticket with the OPSWAT team.

Why is the Discovery Service becoming unresponsive due to RabbitMQ connection issues?

> **Check Your Version:**
>
> This article applies to MetaDefender Secure Storage version older than 4.1.0 deployed on Windows and Linux systems.

Issue:

The Discovery Service becomes unresponsive or stops working. Log entries show timeout and retry messages in the PublishHandler, along with RabbitMQ connection errors such as:

```
Already closed: The AMQP operation was interrupted
Close reason: code=406
```

Possible Causes:

- Missing or misconfigured target exchange in RabbitMQ.
- Transient network interruptions or broker restarts leading to dropped connections.
- RabbitMQ reaching connection or channel limits, causing timeouts.

Resolution:

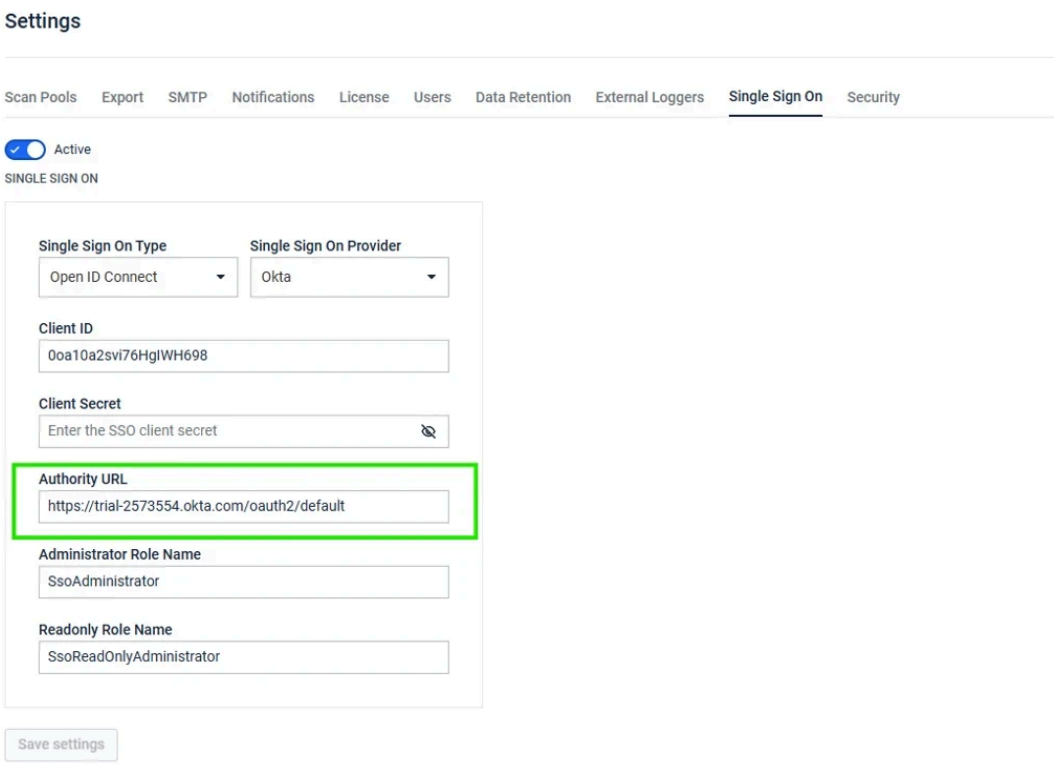
Upgrade the Helm chart to use MDSS version 4.1.0. This version includes fixes for Discovery Service stability and RabbitMQ connection handling.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a support case or chatting with our support engineer.

Changing the Authority URL in SSO will redirect to initial one

> **Check Your Version:**
>
> This article applies to Metadefender Storage Security V4.2.0

Issue: Once you update the **Authority URL** in the MetaDefender Storage Security SSO configuration page, save your changes, sign out, and then try logging in again via SSO. The system continues to redirect to the previous Authority URL.



Solution:

1. After changing the SSO configuration page with the new Authority URL and related data, save the settings.
2. **Restart MDSS:**

Windows : Restart the OPSWAT MetaDefender Security Storage service under Windows Services.

Linux: Restart the Metadefender Storage Security using the following command:

```
MDSS -c restart
```

Restarting the MDSS service clears the previously cached URL and forces MDSS to load the newly saved configuration.

Please note that this is fixed in MDSS V4.3.0.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

Scan Priorities & Throughput Tuning

> **Info**
>
> Available in **MetaDefender Storage Security 3.6.0** and later.

This guide explains how MetaDefender Storage Security (MDSS) controls **how fast** your storages are scanned, and how you can tune that speed. It covers three settings:

- **Scan Priority** (Low / Medium / High)
- **Prefetch Count** (`RABBITMQ_DISCOVERY_PREFETCH_COUNT`, `RABBITMQ_SCANNING_PREFETCH_COUNT`, `RABBITMQ_REMEDIATIONS_PREFETCH_COUNT`)
- **Consumer Multiplier** (`RABBITMQ_CONSUMER_MULTIPLIER`)

No prior knowledge of the internals is required — everything is explained with a simple analogy and worked examples.

How scanning works (in one minute)

When MDSS scans a storage, the work moves through an internal pipeline made up of three stages:

1. **Discovery** — MDSS walks through your storage and finds every file that needs to be scanned.
2. **Scanning** — each file found is then scanned for threats.
3. **Remediation** — when a file needs action (for example, a threat is found), MDSS applies the configured action to it, such as moving, deleting, or quarantining the file.

All three stages process files using parallel **workers**. The settings in this guide simply control **how many workers run** and **how much work each worker holds at a time**.

> **Info**
>
> Moving-team analogy — used throughout this guide. Think of scanning your files like a team clearing a pile of boxes:
>
> - Workers (Consumer Multiplier) = how many movers are working.
> - Prefetch Count = how many boxes each mover carries per trip.
> - Scan Priority = how many movers you put on a job.
>
> More movers, each carrying more per trip = the pile clears faster — as long as you have enough people (your server's CPU and memory) to keep up.

---

1. Scan Priority (Low / Medium / High)

Every scan you start is given a **priority**. The priority decides how much processing power that scan receives relative to other scans. Higher-priority scans are automatically assigned **more parallel workers**, so they finish faster.

Priority	Relative processing power	Best for
Low *(default)*	Baseline	Routine, scheduled, or background scans where finishing time isn't critical
Medium	3× the workers of Low	Standard on-demand scans a user is actively waiting on
High	6× the workers of Low	Urgent, time-sensitive scans that must complete as quickly as possible

Key things to know:

- **Low is the default.** If you don't choose a priority, the scan runs as Low.
- **Priority is set per scan.** Two scans on the same storage can run at different priorities at the same time.
- **Higher priority does not pause lower-priority scans.** All priorities run in parallel — High simply gets many more workers, so it drains its queue much faster and effectively finishes ahead.

---

2. Prefetch Count

What it is: the number of files a **single worker** pulls and holds at one time before reporting them as done.

In the moving-team analogy, this is how many boxes each mover carries per trip. Carrying more per trip means fewer empty-handed walks back — the mover keeps working instead of waiting for the next box.

Setting	Controls	Default	Maximum
`RABBITMQ_DISCOVERY_PREFETCH_COUNT`	Discovery stage (finding files)	10	400
`RABBITMQ_SCANNING_PREFETCH_COUNT`	Scanning stage (scanning files)	10	400
`RABBITMQ_REMEDIATIONS_PREFETCH_COUNT`	Remediation stage (acting on files)	10	400

Effect of the value:

- **Too low (e.g. 1):** each worker finishes one file, then waits to be handed the next. Small idle gaps add up and reduce throughput.
- **Balanced (e.g. 10–50):** the worker always has the next files ready, so it never sits idle. Recommended for most deployments.
- **Too high (e.g. 400):** a single worker can hoard a large batch of files while other workers sit idle, and memory usage rises. Rarely beneficial.

> **Info**
>
> Example: `RABBITMQSCANNINGPREFETCH_COUNT=20` Each scanning worker keeps up to 20 files lined up. As soon as it finishes one, the next is already waiting — no time wasted asking for more work.

> **Info**
>
> Any value above the maximum (400) is automatically capped.

---

3. Consumer Multiplier

What it is: how many **workers** the service runs in parallel. This is how many movers you put to work at once.

Setting	Default	Maximum
`RABBITMQ_CONSUMER_MULTIPLIER`	1	10

The important detail: the multiplier is a **base number** that is automatically scaled up for higher priorities. You set one value, and MDSS applies it proportionally:

Priority	Workers created
Low	multiplier × 1
Medium	multiplier × 3
High	multiplier × 6

> **Info**
>
> Example: `RABBITMQCONSUMERMULTIPLIER=2`
>
> - Low-priority scans → 2 workers
> - Medium-priority scans → 6 workers (2 × 3)

- High-priority scans → 12 workers (2 × 6)

> **Info**
>
> This is exactly why High-priority scans are faster — they automatically receive 6× the workers of Low.

When to increase it: when scans are running slower than you'd like, files are clearly piling up, **and** your server still has spare CPU and memory. Each extra worker scans another file in parallel.

When to leave it at the default (1): most deployments. Raising it above 3–5 is rarely necessary and increases load on the system. Any value above the maximum (10) is automatically capped.

---

4. Putting it all together

The total number of files a stage can process **at once** is:

```
workers × prefetch count
```

Remember that the number of **workers** depends on both the multiplier *and* the priority (Low ×1, Medium ×3, High ×6).

A step-by-step tuning walkthrough

The best way to understand these settings is to follow a real tuning session. Imagine a customer scanning a large file share. We start with the out-of-the-box defaults and adjust one thing at a time, checking the effect after each change.

Iteration 0 — Start with the defaults

```
RABBITMQ_CONSUMER_MULTIPLIER     = 1   (default)
RABBITMQ_SCANNING_PREFETCH_COUNT = 10  (default)
```

Files the Scanning stage can process at once, per priority:

Priority	Workers (multiplier × tier)	Prefetch each	Files in progress at once
Low	1 (1×1)	10	**10**
Medium	3 (1×3)	10	**30**
High	6 (1×6)	10	**60**

> **Info**
>
> Observation: A High-priority scan completes reasonably fast, but a large Low-priority scan feels slow. Checking the server, CPU and memory still have plenty of headroom — so there's room to push harder.

Iteration 1 — Increase the prefetch count

Each worker is finishing its files and then waiting for the next batch. We give every worker a bigger armful by raising the prefetch count from 10 to 20:

```
RABBITMQ_CONSUMER_MULTIPLIER     = 1
RABBITMQ_SCANNING_PREFETCH_COUNT = 20   ← changed
```

Priority	Workers	Prefetch each	Files in progress at once
Low	1	20	**20** *(was 10)*
Medium	3	20	**60** *(was 30)*
High	6	20	**120** *(was 60)*

> **Info**
>
> Observation: Throughput roughly doubles and workers no longer sit idle between files. But the Low-priority scan still isn't keeping up with demand, and the CPU is still not fully used. The bottleneck now is the number of workers, not how much each one holds.

Iteration 2 — Increase the consumer multiplier

We add more workers by raising the multiplier from 1 to 2. This doubles the workers at **every** priority:

```
RABBITMQ_CONSUMER_MULTIPLIER     = 2    ← changed
RABBITMQ_SCANNING_PREFETCH_COUNT = 20
```

Priority	Workers (multiplier × tier)	Prefetch each	Files in progress at once
Low	2 (2×1)	20	**40** *(was 20)*
Medium	6 (2×3)	20	**120** *(was 60)*
High	12 (2×6)	20	**240** *(was 120)*

> **Info**
>
> Observation: Scanning is now fast at every priority — a High scan can move up to 240 files through scanning at once, 6× a Low scan. The CPU is now well used but not maxed out, which is the healthy target.

Iteration 3 — Tune the other stages if needed

Scanning is no longer the bottleneck, but the customer notices that after scanning, **acting on flagged files (remediation) is lagging behind**. Each stage has its own prefetch count, so we raise the remediation stage too:

```
RABBITMQ_CONSUMER_MULTIPLIER          = 2
RABBITMQ_SCANNING_PREFETCH_COUNT      = 20
RABBITMQ_REMEDIATIONS_PREFETCH_COUNT  = 20   ← changed
RABBITMQ_DISCOVERY_PREFETCH_COUNT     = 20   ← optional, if finding
files is also slow
```

> **Info**
>
> Observation: The whole pipeline — finding, scanning, and acting on files — now flows smoothly. Note that `RABBITMQCONSUMERMULTIPLIER` is a single setting that applies to all stages, while each stage has its own prefetch count, so you can fine-tune any stage independently.

When to stop

Keep increasing only while the queues are backing up **and** the server still has spare CPU and memory. The moment CPU approaches 80%, stop — adding more workers to a maxed-out server makes things slower, not faster. At that point the answer is more hardware, not bigger numbers.

---

5. Quick reference

You want to...	Do this
Finish a specific scan faster	Start it as **High** priority (6× the workers)
Run a non-urgent background scan	Leave it as **Low** (the default)
Speed up scanning overall (CPU has headroom)	Increase `RABBITMQ_CONSUMER_MULTIPLIER`
Smooth out workers that idle between files	Increase the relevant `*_PREFETCH_COUNT`
Speed up the **finding** of files	Tune `RABBITMQ_DISCOVERY_PREFETCH_COUNT`
Speed up the **scanning** of files	Tune `RABBITMQ_SCANNING_PREFETCH_COUNT`
Speed up **acting on** flagged files	Tune `RABBITMQ_REMEDIATIONS_PREFETCH_COUNT`

Default and maximum values

Setting	Default	Maximum
`RABBITMQ_DISCOVERY_PREFETCH_COUNT`	10	400
`RABBITMQ_SCANNING_PREFETCH_COUNT`	10	400
`RABBITMQ_REMEDIATIONS_PREFETCH_COUNT`	10	400
`RABBITMQ_CONSUMER_MULTIPLIER`	1	10

Important notes

- **These settings increase parallelism, not raw machine speed.** They help only when workers are waiting for work while your CPU and memory still have room to spare. On a server that is already at full CPU, raising these values will **not** help — add more resources instead.
- **Changing the environment variables requires a service restart** to take effect. Set them in the service's environment configuration (`.env` / `customer.env`).
- **Values above the maximum are automatically capped** to the limits shown above.

Service Instance Scaling on Linux

> **Info**

> This article is applicable only to MetaDefender Storage Security deployments that are running on **Linux**.
>
> Available in **MetaDefender Storage Security 4.4.0** and later.

This guide explains how to run **extra instances** of MetaDefender Storage Security (MDSS) services so your deployment can handle more work in parallel. It's an optional setting for larger or busier environments.

If you've read the Scan Priorities & Throughput Tuning guide, here's the key difference:

- That guide tunes how hard a **single** service container works (more workers, bigger batches).
- This guide adds **more containers** of the same service, so the work is shared across them.

> **Info**
>
> Analogy: if a single service container is a moving team clearing boxes, then instance scaling adds more teams — each with their own truck — all working the same job at the same time.

How it works

By default, each scalable service runs as **one** container. You can add up to **two extra** containers per service using the `*_ADDITIONAL_INSTANCES` settings.

The value is simply **how many extra containers to add on top of the default one**:

Value	Total containers that run
`0`	1 (default — just the original)
1	2
2	3 (the maximum)

The maximum is **2 additional** (3 total). Any higher value is automatically reduced to 2.

When MDSS starts, it reads these values and launches the extra containers automatically. They join the same workload and start sharing it immediately — no other configuration is needed.

What you can scale

Three central services can be scaled. These are the core engines of the scanning pipeline:

Setting	Service	What it does
`DISCOVERY_SERVICE_ADDITIONAL_INSTANCES`	Discovery	Walks through your storage and finds files to scan
`SCANNING_SERVICE_ADDITIONAL_INSTANCES`	Scanning	Scans each file for threats
`REMEDIATIONS_SERVICE_ADDITIONAL_INSTANCES`	Remediation	Acts on flagged files (move, delete, quarantine, etc.)

> **Storage types**
>
> There are no per-storage-type settings. Discovery, scanning and remediation for every storage type run inside the Discovery, Scanning and Remediations services, so scaling those three services scales the scanning pipeline for every storage type at once. The Storages service, which handles storage setup and connection tests, always runs as a single instance.
>
> The NFS and SMB sidecar services always run as a single instance. `NFS_ADDITIONAL_INSTANCES` and `SMB_ADDITIONAL_INSTANCES` are present in `.env` but currently have no effect.

Examples

Example 1 — Default (no scaling)

```
SCANNING_SERVICE_ADDITIONAL_INSTANCES=0
```

One scanning container runs. This is the default and is right for most deployments.

Example 2 — Add one extra scanning container

```
SCANNING_SERVICE_ADDITIONAL_INSTANCES=1
```

Two scanning containers run and share the scanning workload — roughly doubling scanning capacity.

Example 3 — Scale the whole central pipeline

```
DISCOVERY_SERVICE_ADDITIONAL_INSTANCES=1
SCANNING_SERVICE_ADDITIONAL_INSTANCES=2
REMEDIATIONS_SERVICE_ADDITIONAL_INSTANCES=1
```

Discovery runs 2 containers, Scanning runs 3, and Remediation runs 2 — useful for a large, busy environment where every stage needs more throughput.

When should you scale instances?

Add instances when **a single container is already working at full capacity** and your server still has spare CPU and memory to run more containers.

A good order to follow:

1. **First, tune the single container** using the prefetch and consumer-multiplier settings (see *Scan Priorities & Throughput Tuning*). This is the simplest change and often enough.
2. **Then, add instances** if one container — even fully tuned — still can't keep up, and you have the hardware to run more.

If your server's CPU is already near 100%, adding more containers will **not** help — they'll compete for the same resources. In that case, the answer is more hardware (a bigger server, or additional servers), not more instances.

How to apply changes

1. Set the desired `*_ADDITIONAL_INSTANCES` values in `/etc/mdss/customer.env` (see Custom configuration)
2. Restart MDSS so the extra containers are created

> **Info**
>
> Changes only take effect after a restart, because the extra containers are created at startup.

Things to keep in mind

- **The default (0) is right for most deployments.** Only scale when you have a clear throughput need and spare hardware.

- **The maximum is 2 additional instances (3 total) per service.** Higher values are automatically capped.
- **Only the services listed in this guide can be scaled.** Other MDSS services are designed to run as a single instance and should not be duplicated.
- **More instances need more resources.** Each extra container uses additional CPU and memory. Make sure your server has the capacity before scaling.
- **Instance scaling and worker tuning work together.** You can both add instances *and* tune each container's workers/prefetch for maximum throughput.

How to move the location of the MetaDefender Storage Security services_logs folder?

> **Check Your Version:**
>
> This article applies to all versions of MetaDefender Storage Security Linux deployments.

Background

When dealing with disk space issues related to the logging created by MDSS in environments where MDSS default log partition might be highly restricted, MDSS supports the ability to use symlinks to point the logs folder to a different partition/location.

Steps

The following provides steps on how to add a symlink for the `/etc/mdss/services_logs` directory/folder:

1. **Change name of the current services_logs folder and it backup**

```
sudo mv services_logs/ services_logs_v1/
```

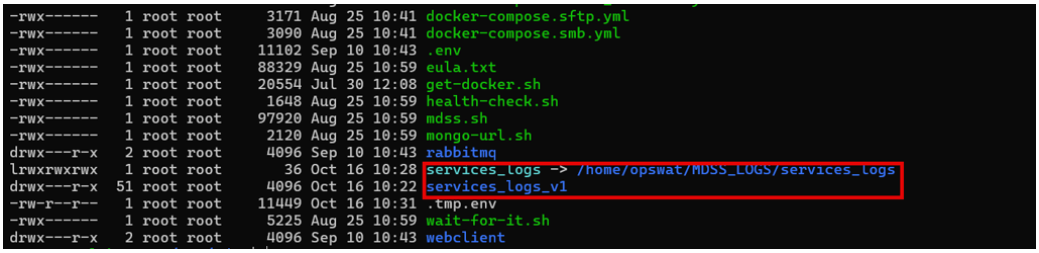
2. **Create a symlink:**

```
sudo ln -s [absolute_path_to_destination_folder]/services_logs
[MDSS_dicretory]
example: sudo ln -s /home/opswat/MDSS_LOGS/services_logs /etc/mdss/
```

3. Restart MDSS:

```
sudo mdss -c restart
```

4. Once the product restarts, you will be able to see the symlink created by executing the command: `ls -al`. Sample symnlink created shown below:



> Notes:
>
> After restart, a new file structure and logs will be created on the destination of the symlink.
>
> Further restarts of MDSS will not affect symlink functionality.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a support case or chat with our support engineer.

How to install RabbitMQ on Windows as an external service?

> Check your version:
>
> This article applies to MDSS 4.x (Windows) and later.

**Overview **

This guide covers the installation of RabbitMQ 4.1.8 as an external service for MDSS (MetaDefender Storage Security) on Windows. It includes Erlang installation, RabbitMQ configuration, known issues and their resolutions, and the final MDSS customer.env configuration.

**Prerequisites **

- Windows Server with administrative access
- PowerShell (run as administrator for all steps)
- Only one Erlang version installed at a time

- Installation paths must use ASCII characters only — no spaces in directory names
- Firewall access to ports: 5672, 15672, 4369, 25672

**Part 1 — Install Erlang **

RabbitMQ 4.1.8 requires Erlang 26.2 minimum. Erlang 27.x is recommended and fully supported.

**Step 1. **Download the latest Erlang 27.x Windows 64-bit installer from:

https://erlang.org/download/otp_versions_tree.html

**Step 2. **Right-click the installer and select Run as administrator.

**Step 3. **Follow the installation wizard. Install to a path with no spaces, e.g.:

C:\Erlang

**Step 4. **Verify the installation. Open PowerShell as administrator and run:

```
erl -version
```

**Important: **If erl -version fails in cmd.exe but works in PowerShell, this is a PATH caching issue in cmd. Use PowerShell for all remaining steps. A full system reboot will fix cmd permanently.

**Part 2 — Install RabbitMQ 4.1.8 **

**Step 5. **Download the RabbitMQ 4.1.8 Windows installer:

https://github.com/rabbitmq/rabbitmq-server/releases/download/v4.1.8/rabbitmq-server-4.1.8.exe

**Step 6. **Right-click the installer and select Run as administrator.

**Step 7. **Follow the wizard. RabbitMQ will install and register itself as a Windows Service.

**Step 8. **Verify the service is running:

```
Get-Service -Name RabbitMQ
```

**Part 3 — Configure RabbitMQ for MDSS **

**3.1 Create the MDSS User **

Open PowerShell as administrator and navigate to the sbin directory:

```
cd "C:\Program Files\RabbitMQ Server\rabbitmq_server-4.1.8\sbin"
```

Create a dedicated user (replace and with your credentials):

```
.\rabbitmqctl.bat add_user <username> <password>

.\rabbitmqctl.bat set_permissions -p / <username> ".*" ".*" ".*"

.\rabbitmqctl.bat set_user_tags <username> administrator
```

Example:

```
.\rabbitmqctl.bat add_user mdss 2eH50&i_\CLwx%CU:6=y_<1,EObKA6jN

.\rabbitmqctl.bat set_permissions -p / mdss ".*" ".*" ".*"

.\rabbitmqctl.bat set_user_tags mdss administrator
```

**3.2 Enable the Management Plugin **

```
.\rabbitmq-plugins.bat enable rabbitmq_management
```

**3.3 Configure Consumer Timeout **

MDSS requires consumer timeout to be disabled to support long-running processing consumers. This must be done using advanced.config — NOT rabbitmq.conf, which does not support the undefined value for this setting.

**Critical: **Do not add consumer_timeout to rabbitmq.conf. The values “undefined” and “false” are invalid in the sysctl config format and will cause RabbitMQ to fail on startup with: “undefined cannot be converted to a(n) integer”.

Create or edit the advanced.config file:

```
%APPDATA%\RabbitMQ\advanced.config
```

Powershell command:

```
notepad "$env:APPDATA\RabbitMQ\advanced.config"
```

Add the following content exactly, including the trailing dot:

```
[{rabbit, [{consumer_timeout, undefined}]}].
```

**3.4 Restart the Service **

Restart RabbitMQ to apply all configuration changes:

```
Stop-Service RabbitMQ

Start-Sleep -Seconds 3

Start-Service RabbitMQ
```

**Part 4 — Open Firewall Ports **

Run the following in an admin PowerShell to add the required firewall rules:

```
New-NetFirewallRule -DisplayName "RabbitMQ AMQP" -Direction Inbound -
Protocol TCP -LocalPort 5672 -Action Allow

New-NetFirewallRule -DisplayName "RabbitMQ Management" -Direction
Inbound -Protocol TCP -LocalPort 15672 -Action Allow

New-NetFirewallRule -DisplayName "RabbitMQ EPMD" -Direction Inbound -
Protocol TCP -LocalPort 4369 -Action Allow

New-NetFirewallRule -DisplayName "RabbitMQ Distribution" -Direction
Inbound -Protocol TCP -LocalPort 25672 -Action Allow
```

Port reference:

Port	**Purpose**
5672	AMQP — used by MDSS
15672	Management UI and HTTP API
4369	Erlang peer discovery (epmd)
25672	Inter-node and CLI communication

**Part 5 — Configure MDSS (customer.env) **

On the MDSS host, edit the customer.env file (typically at /etc/mdss/customer.env) and add the following, replacing with the IP or hostname of the RabbitMQ server and the credentials from Part 3:

```
RABBITMQ_URI=amqp://<username>:<password>@<rabbitmq-host>:5672

RABBITMQ_HOST=<rabbitmq-host>

RABBITMQ_PORT=5672

RABBITMQ_DEFAULT_USER=<username>

RABBITMQ_DEFAULT_PASS=<password>

Disable the embedded RabbitMQ container in MDSS:

DISABLED_SERVICES=rabbitmq
```

**Part 6 — Verification **

**Step 1. **Confirm RabbitMQ is listening on the expected ports:

```
netstat -ano | findstr :5672
netstat -ano | findstr :15672
```

**Step 2. **Check node status from the sbin directory:

```
.\rabbitmqctl.bat status
```

**Step 3. **Access the management UI in a browser:

http://localhost:15672

Log in with the credentials created in Part 3.

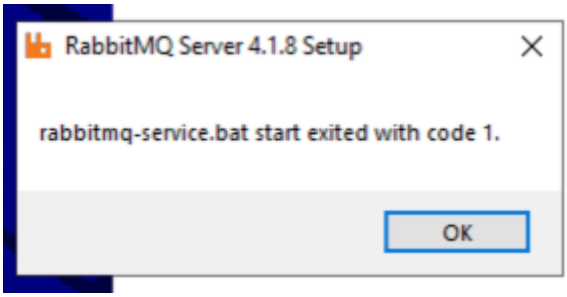
**Step 4. **Once MDSS is running, monitor the key queue:

```
.\rabbitmqctl.bat list_queues name messages messages_ready
messages_unacknowledged consumers | Select-String
"object_ready_for_scan_queue"
```

**Troubleshooting **

**RabbitMQ process fails to start — exit code 1 **

Service exited with code 1.



Check if RabbitMQ ports are not in use by another application or instance:

```
netstat -ano | findstr :5672
netstat -ano | findstr :15672
netstat -ano | findstr :25672
```

**RabbitMQ node fails to start — consumer_timeout error **

Symptom: Boot failure with message:

“undefined” cannot be converted to a[n] integer

Cause: consumer_timeout = undefined was added to rabbitmq.conf.

Fix: Remove consumer_timeout from rabbitmq.conf entirely. Set it in advanced.config instead:

```
%APPDATA%\RabbitMQ\advanced.config
[{rabbit, [{consumer_timeout, undefined}]}].
```

**rabbitmqctl.bat cannot connect to the node **

Symptom:

Error: unable to perform an operation on node rabbit@

TCP connection succeeded but Erlang distribution failed

Cause: The Erlang cookie used by the CLI tool does not match the one used by the Windows Service.

Fix: Copy the service cookie to the current user profile:

```
Copy-Item
"$env:SystemRoot\system32\config\systemprofile\.erlang.cookie"
"$env:HOMEDRIVE$env:HOMEPATH\.erlang.cookie" -Force
```

Then restart the service and retry.

**erl -version not found in cmd.exe **

Symptom: erl is not recognized as an internal or external command in cmd.exe, but works in PowerShell.

Cause: cmd.exe caches the PATH at launch and does not pick up new entries until a reboot.

Fix: Use PowerShell (as administrator) for all steps. A full system reboot will resolve cmd.exe permanently.

**Management UI not accessible at localhost:15672 **

Check that the management plugin is enabled:

```
.\rabbitmq-plugins.bat list
```

Look for [E*] rabbitmq_management. If not enabled:

```
.\rabbitmq-plugins.bat enable rabbitmq_management
```

Then restart the service. If the plugin is enabled but the port is still not listening, confirm the node itself started successfully (see consumer_timeout issue above).

> **Support:**
>
> **Further Assistance** is required, please proceed to create a support case or chat with our support engineer.

Endpoint Protection Exclusions for MetaDefender Storage Security running on Windows

> **Applies to:** MetaDefender Storage Security (MDSS) **4.6.x or later**, Windows Basic Deployment, and the OPSWAT ONTAP Connector. Linux and Kubernetes deployments are out of scope.

This article lists what to exclude from an endpoint protection product (antivirus, EDR, XDR, application control) on the Windows machines that run MDSS, and why. It is vendor neutral: each item names the protection category it applies to; map that category to the matching module in your product.

MDSS is itself a file security product. It reads files that may be malicious, deletes and moves files on storage in bulk, and runs a supervised tree of processes and PowerShell scripts. An endpoint agent is built to stop exactly these behaviours. Without exclusions it slows MDSS down, breaks scans and remediations, and in the worst case deletes customer files from the protected storage before MDSS acts on them.

Section 3 is the recommended rule set: a few generic rules that cover everything MDSS does. **Appendix A** is the full inventory behind those rules, for products without wildcard or parent-process features, or for policies that require every exclusion to be enumerated.

---

1. Symptoms of interference

Symptom	Usual cause
Restarting MDSS takes 5 to 15 minutes. `data\webclient\startup-log.txt` shows minutes spent in "Loading CA certificates", "Refreshing environment and nginx configuration" or "Applying HTTP/HTTPS configuration".	The agent inspects every PowerShell process the MDSS service starts (script control, .NET instrumentation, application control).
`powershell.exe -NoProfile -File .\mdss.ps1 -c load-ca` takes more than 10 seconds (healthy: 1 to 3).	Same as above. If the exclusions do not fix it, Windows is waiting on blocked revocation checks for the signed scripts; see Slow 'mdss.ps1' commands on Windows.
The `mdss` service or one of its child processes is terminated, or the console stays on "Initializing".	Behavioural, anti-ransomware or child-process protection blocks a process in the MDSS tree.
Alerts about a service starting `powershell.exe -ExecutionPolicy Bypass`, `sc.exe`, `icacls.exe`, `msiexec.exe`, `psql.exe` or `nginx.exe`, or about a new root certificate, firewall rule, scheduled task or changed machine environment variables.	The `mdss` service supervising its process tree, and its start-up scripts. Expected behaviour.

---

2. Principles

- **Exclude by full path, not by file name.** Every MDSS process lives under the installation folder or is a private copy of a runtime under it. Excluding the MDSS `dotnet.exe` by path does not exclude other .NET applications on the host.
- **Use folder and process exclusions together.** A folder exclusion protects the files MDSS *owns*. A process exclusion protects what MDSS *does*, and most of that happens

outside the installation folder: reading files on shares, writing temp copies, bulk deletes and moves, spawning PowerShell from `C:\Windows`.

- **Trust the process tree.** "A process and its descendants" for the `mdss` service is the single most effective rule in this document. If your product lacks it, exclude each executable in A.3.
- **Prefer process exclusions over disabling a module**, and use the OPSWAT, Inc. code signature where the product allows it (A.7).
- **Do not copy this list to the MetaDefender Core host.** MD Core has its own requirements.

---

3. Recommended exclusion rules

`<INSTALL_DIR>` is the installation folder, by default `C:\Program Files\OPSWAT\MetaDefender Storage Security`. The installer lets the customer choose another drive or folder; the real path is in the registry value `InstallDir` under `HKLM\SOFTWARE\OPSWAT\MetaDefender Storage Security`.

3.1 Folders

Exclude from on-access (real-time) scanning, scheduled scanning and behavioural file monitoring:

1. **<INSTALL_DIR>**, the whole tree (A.1).
2. **C:\ProgramData\OPSWAT**: upgrade state files and, under `mdss\smb-mounts`, one directory symbolic link per mounted SMB share pointing at `\\server\share`. **Critical:** anything the agent does under `smb-mounts` happens on the customer's storage. Scanning there scans the whole share through the MDSS server; quarantining there deletes the customer's file. The location is set by `SMBSERVICE_MOUNT_DIRECTORY`; exclude the configured value if it was changed.
3. **C:\ProgramData\OPSWAT-ONTAP-Connector**, plus the folder named by `LogPath` in its `config.ini` if one is set, when the ONTAP Connector runs on this host (section 4).
4. **The temp folder of the account the mdss service runs as**: `C:\Windows\Temp` (or `C:\Windows\SystemTemp` on current Windows Server builds) for `LocalSystem`, `C:\Users\<account>\AppData\Local\Temp` for a custom account. MDSS writes temporary copies of the files it uploads during remediation there. They can be the original, possibly malicious, file a workflow is moving to quarantine; if the agent removes them the remediation fails. If this is too broad for your policy, the process rules in 3.2 give the same result.

3.2 Processes

Exclude from on-access scanning, network file scanning, behavioural analysis, anti-ransomware, exploit and memory protection, and .NET or process instrumentation (CLR profilers, startup hooks, `APPDOMAIN_MANAGER_*` injection):

1. **<INSTALL_DIR>\services\opswat.mdss.windows.service\opswat.mdss.windows.service.exe and all its descendants.** This is the `mdss` Windows service. It starts and supervises every other MDSS process: the private .NET runtime (`services\dotnet\dotnet.exe`, 12 or more instances hosting the microservices), nginx, the Garnet cache, the SMB sidecar, and every `powershell.exe`, `sc.exe`, `icacls.exe`, `psql.exe`, `nginx.exe`, `node.exe` and `msiexec.exe` it launches at start-up.
2. **Any executable under <INSTALL_DIR>\services** (`<INSTALL_DIR>\services\**\*.exe`). Covers the two components that run as their own Windows services rather than under the supervisor (PostgreSQL as `mdss_postgresql`, RabbitMQ and Erlang as `mdss_rabbitmq`) and the tools administrators run by hand (`pg_dump.exe`, `opswat.mdss.reencryption.cli.exe`).
3. **C:\ProgramData\OPSWAT-ONTAP-Connector\OPSWAT-ONTAP-Connector.exe**, when installed on this host.

Rule 1 also covers the SMB sidecar reading customer files through `smb-mounts` and the remediation temp copies. Every service is a .NET 10 application that compiles code at run time, so memory scanning and instrumentation add latency to each process, and the supervisor restarts any child that exits (it polls every 10 seconds), so a blocked process becomes a restart loop. Executables, signers and ports: A.3 and A.4.

3.3 Scripts

Under script control, PowerShell protection and .NET Framework instrumentation, allow every `.ps1` file in `<INSTALL_DIR>\scripts\`, and the `powershell.exe` that runs it, when started by the `mdss` service (rule 3.2.1), by `msiexec.exe` during install, upgrade or uninstall, or by an administrator from an elevated prompt. They are Windows PowerShell 5.1 scripts signed by OPSWAT, Inc. and launched with `-ExecutionPolicy Bypass … -File`, so key the rule on publisher or path, not on the execution policy. The installer additionally runs inline PowerShell commands (`powershell.exe -Command`), VBScript custom actions inside `msiexec.exe` (the UAC relaunch, WMI queries of ports, processes and services) and `cmd.exe /c net stop … sc delete …`. A.5 lists what each script does.

3.4 Services, scheduled task, firewall rules and registry

Allow the installer and the processes in 3.2 to create, change and delete:

- Windows services `mdss`, `mdss_postgresql`, `mdss_rabbitmq`, legacy `mdss_mongodb`, and `opswat-ontap-connector`.
- The scheduled task `MDSS_InstallSharePointComponents` (SharePoint On-Premises module only).
- Windows Firewall inbound rules whose display name starts with `MetaDefender Storage Security:`.
- Registry keys under `HKLM\SOFTWARE\OPSWAT`, `HKLM\SOFTWARE\Ericsson\Erlang\ErlSrv`, `HKLM\SYSTEM\CurrentControlSet\Services\mdss*` and `…\Services\opswat-ontap-connector`, the machine environment key `HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment`, the machine Trusted Root and Personal certificate stores, and (ONTAP Connector only) `HKLM\SOFTWARE\WOW6432Node\Data ONTAP\Clustered Data ONTAP Antivirus Connector\v1.0\data-lifs`.

Details: A.2 and A.6.

3.5 Network

- **Inbound:** 80 and 443 (`WEB_PORT`, `WEB_PORT_SSL`): console, REST API, storage webhooks and ONTAP Connector traffic, all through nginx.
- **Loopback:** all MDSS components talk to each other on 127.0.0.1. Do not filter loopback traffic for the processes in 3.2 (ports in A.4).
- **Outbound:**
  - Scan engines: MetaDefender Core (HTTP or HTTPS, default port 8008); `api.metadefender.com:443` when MetaDefender Cloud is used.
  - OPSWAT: `activation.dl.opswat.com:443` for license activation and, only when `ENABLE_TELEMETRY=yes`, for telemetry (OTLP over HTTPS). The previous telemetry endpoint `mdss-telemetry.metadefender.com:4317` is used only when configured explicitly.
  - Protected storages: SMB 445; NFS 2049 and 111; SFTP 22; FTP 21 plus passive ports; 443 to the cloud storage endpoints (Amazon S3, Azure, Google Cloud, Alibaba, Oracle, Box, Microsoft Graph, SharePoint).
  - Customer infrastructure: domain controllers or LDAP servers (389, 636, 3268, 3269), the SMTP server, webhook targets, and an external PostgreSQL or RabbitMQ when configured.

3.6 Behaviours to recognise

These are normal for MDSS. Excluding the processes in 3.2 from the behavioural, anti-ransomware and child-process modules handles all of them; the list helps the security team recognise the alerts.

1. **Bulk file operations on network shares.** Remediation workflows delete, move, copy, rename and overwrite many files per minute, which triggers mass-delete, mass-rename, canary-file and “encryption-like write” heuristics.
2. **Reading, and temporarily copying, malicious files** before they are sent to MetaDefender Core or moved to quarantine.
3. **Symbolic links to UNC paths.** The SMB sidecar maps shares with `WNetAddConnection2` and creates directory symbolic links under `smb-mounts` with `CreateSymbolicLink`, on demand, removing them after 300 seconds idle. Agents that follow reparse points scan, and may quarantine on, the customer’s storage.
4. **A service spawning PowerShell and management tools.** At every start the `mdss` service runs `powershell.exe -ExecutionPolicy Bypass`, which spawns more PowerShell, `sc.exe`, `icacls.exe`, `nginx.exe`, `psql.exe`, `node.exe` and `msiexec.exe`, registers services and changes their logon account.
5. **Start-up and installer changes to the machine.** Every start imports CA certificates into the Trusted Root store, rewrites about 300 machine environment variables and broadcasts `WM_SETTINGCHANGE`. The installer relaunches itself elevated through UAC, queries WMI, adds firewall rules and a scheduled task, and writes state files to `C:\ProgramData\OPSWAT`. A support package runs `systeminfo`, `route print` and `ipconfig /all` and creates large archives under `<INSTALL_DIR>`.

3.7 Application control (allow-listing)

1. Publisher rule for **OPSWAT, Inc.**: all MDSS executables, libraries and PowerShell scripts, and the ONTAP Connector.
2. Publisher rule for **Microsoft Corporation**: the private .NET runtime, Garnet, PowerShell, `msiexec.exe`, `sc.exe`, `icacls.exe`.
3. Path rule for everything under **<INSTALL_DIR>**, including SQL and JavaScript files. Covers the third-party runtimes (PostgreSQL, Erlang, RabbitMQ, Python, Node.js, MongoDB) whatever their signer, and `nginx.exe`, which is not signed by OPSWAT in 4.6.x. The folder is writable only by administrators, which is what makes a path rule acceptable.

If policy requires publisher rules for every file, enumerate the signers with the command in A.7 and add hash rules for the unsigned files it reports. The `mdss` service performs its own integrity check: it refuses to start an MDSS binary that is not signed by OPSWAT, or a runtime executable (`dotnet.exe`, `GarnetServer.exe`, `python.exe`) that carries no signature. Do not let the agent modify, wrap or re-sign these files.

4. OPSWAT ONTAP Connector (NetApp ONTAP)

A separate Windows service, usually on its own host next to the NetApp Antivirus Connector (`ontapavc`). Installed to `C:\ProgramData\OPSWAT-ONTAP-Connector\` by MDSS or manually; the MDSS installer updates and removes it together with MDSS.

- **Service:** `opswat-ontap-connector`, executable `OPSWAT-ONTAP-Connector.exe`, signed by OPSWAT.
- **Files:** `config.ini` (rewritten through `config.ini.tmp` whenever the set of protected storages changes), `data.json`, `logs\YYYYMMDD.log` and `YYYYMMDD_HHMMSS.log` (or the configured `LogPath`). On uninstall, `config.ini` is backed up to `C:\ProgramData\OPSWAT\ontap-connector-config.ini.bak`.
- **Named pipes:** `\pipe\ntapvsrq` (scan requests from the NetApp AV Connector) and `\pipe\ntapvscp` (registration and verdicts), RPC over `ncacn_np` to 127.0.0.1.
- **Registry:** writes and deletes `data-lif::<host>` values under `HKLM\SOFTWARE\WOW6432Node\Data ONTAP\Clustered Data ONTAP Antivirus Connector\v1.0\data-lifs`, creating the key if it is missing. Since 4.6.0 this runs automatically every 15 seconds (`StorageSyncIntervalInSeconds`), not only at installation.
- **Network:** HTTP or HTTPS to the MDSS web port (80 or 443) for `/api/webhook/realtime`, `/api/file/*`, `/api/health` and `/api/account` (polled every 15 seconds). SMB 445 to the ONTAP shares.
- **Behaviour:** reads every file a user opens or modifies on the protected shares through `\\?\UNC\…` paths and posts it to MDSS. These files may be malicious. Optionally writes the `Zone.Identifier` alternate data stream (`ZoneIdentifierFiles`).

Exclude the folder, the process and its UNC file reads from on-access scanning. An agent that scans the network files this process opens scans, and may quarantine on, the ONTAP volume before MDSS returns a verdict, and it doubles the latency of every file open for every ONTAP user.

---

5. What to keep enabled

- Everything outside the paths and processes above. MDSS does not need the agent disabled.
- On-access scanning of files that administrators copy onto the server themselves, and network protection for destinations not listed in 3.5.
- Audit-only logging of the behaviours in 3.6, if the security team wants visibility. Only blocking and quarantine actions need to be excluded.

---

6. Verification

1. PowerShell start-up time, as administrator. Expect well under one second for the first command and one to three seconds for the second.

```
Measure-Command { powershell.exe -NoProfile -NonInteractive -Command
exit }
cd "<INSTALL_DIR>\scripts"
Measure-Command { powershell.exe -NoProfile -File .\mdss.ps1 -c
load-ca }
```

2. Restart MDSS, read the step timings and confirm every process and service is up. Every step except database migrations should finish within seconds; a healthy restart takes two to four minutes.

```
.\mdss.ps1 -c restart
Get-Content "<INSTALL_DIR>\data\webclient\startup-log.txt"
.\mdss.ps1 -c status
Get-Service mdss, mdss_postgresql, mdss_rabbitmq
```

3. Check the agent console for detections referencing the paths in 3.1, the processes in 3.2 or the behaviours in 3.6. Any remaining detection means an exclusion is missing, or is by file name where a full path is required.

4. Scan a storage that holds a known-safe test file and, if remediation is used, run a workflow that moves a file to quarantine. Confirm in the MDSS audit log that both complete, and on the storage that nothing other than MDSS removed a file.

Appendix A. Detailed inventory

Use this appendix when your product has no wildcard or parent-process trust feature, or when policy requires each exclusion to be enumerated.

A.1 Installation folder layout

Folder	Contents
`services\`	One folder per microservice (its `.dll` files and a `logs\` folder), the private .NET runtime (`dotnet`), Garnet (`garnet`), nginx (`opswat.mdcs.webclient`), PostgreSQL (`postgres`), RabbitMQ (`rabbitmq`), Erlang (`erlang`), the SMB sidecar (`opswat.mdss.smb.service`), the service host (`opswat.mdss.windows.service`), legacy Python (`python`) and MongoDB (`mongo`). The Scanning service keeps its storage providers in a `providers` subfolder. Hundreds of files load at every start; logs are written continuously.
`data\`	PostgreSQL data, the RabbitMQ message store, the web client files and start-up log (`data\webclient`), legacy MongoDB data. Rewritten constantly; scanning causes latency and lock conflicts.
`logs\`	PostgreSQL, RabbitMQ, migration and installation logs.
`config\`	`.env`, `customer.env`, nginx configuration and TLS certificates (`config\nginx\certificates`), RabbitMQ, PostgreSQL and cache configuration. Rewritten at every start.
`scripts\`	The PowerShell scripts in A.5.
`dependencies\`	Compressed runtimes extracted during install and upgrade.
`Utils\`	SQL migration scripts and installer prerequisites (SharePoint client components MSI, Visual C++ runtime).
`ca_certificates\`	Customer CA certificates imported into the Windows trust store at every start.
`tmp\`, `<INSTALL_DIR>\*.zip`	Working folder and output of the support package, log, settings and database exports; staging folder for ONTAP Connector updates.

Outside the installation folder: `C:\ProgramData\OPSWAT\` (`mdss_upgrade_customer.env` and `mdss_upgrade_service_account.txt` during upgrades, `ontap-connector-config.ini.bak`, `mdss\smb-mounts\`), `C:\ProgramData\OPSWAT-ONTAP-Connector\`, the service account's temp folder (3.1), and `%TEMP%\MetaDefenderStorageSecurity-Setup-<timestamp>.log` of the administrator who ran the installer.

A.2 Windows services, scheduled task and firewall rules

Item	Executable or definition
Service `mdss`	`<INSTALL_DIR>\services\opswat.mdss.windows.service\opswat`
Service `mdss_postgresql`	`<INSTALL_DIR>\services\postgres\bin\pg_ctl.exe`, runs `postgre`
Service `mdss_rabbitmq`	`<INSTALL_DIR>\services\erlang\erts-*\bin\erlsrv.exe`, runs `er`
Service `mdss_mongodb`	`<INSTALL_DIR>\services\mongo\mongod.exe`
Service `opswat-ontap-connector`	`C:\ProgramData\OPSWAT-ONTAP-Connector\OPSWAT-ONTAP-Connec`
Scheduled task `MDSS_InstallSharePointComponents`	`powershell.exe -ExecutionPolicy Bypass -WindowStyle Hidde` `<INSTALL_DIR>\scripts\install-sharepoint-components.ps1",`
Firewall rules `MetaDefender Storage Security: Allow inbound traffic <NAME> - <PORT>`	Inbound TCP

A.3 Executables

`services.exe` starts `opswat.mdss.windows.service.exe`, which starts `powershell.exe` for the start-up orchestration; that starts more `powershell.exe` instances, `nginx.exe -s reload`, `psql.exe`, `node.exe` and `msiexec.exe`. After orchestration the service starts and keeps alive every process below, restarting any that exits.

Executable (under <INSTALL_DIR>\services\ unless stated)	Signer	Role and ports
opswat.mdss.windows.service\opswat.mdss.windows.service.exe	OPSWAT	The mdss service. Single that loads its assemblies (some agents flag this as validates the signature o before starting it.
dotnet\dotnet.exe	Microsoft	Private .NET 10 runtime: d in A.4, plus one per scan (WINDOWS_SCANNING_ 1 to 30). Exclude by this e
opswat.mdss.smb.service\opswat.mdss.smb.service.exe	OPSWAT	SMB sidecar: authenticat the smb-mounts links, the shares. 127.0.0.1:6002
opswat.mdcs.webclient\nginx.exe	not OPSWAT	Console web server and master and one worker. F
garnet\GarnetServer.exe	Microsoft	Redis-compatible cache
postgres\bin\postgres.exe, pg_ctl.exe, psql.exe, pg_dump.exe, pg_restore.exe, pg_isready.exe, initdb.exe	PostgreSQL build vendor	Database and tools; one connection. 127.0.0.1:543
erlang\erts-*\bin\erlsrv.exe, erl.exe, epmd.exe, inet_gethost.exe; rabbitmq\sbin\rabbitmq-service.bat, rabbitmqctl.bat	Erlang / RabbitMQ vendors, may be unsigned	Message broker. 127.0.0.1 4369.
opswat.mdss.reencryption.cli\opswat.mdss.reencryption.cli.exe	OPSWAT	Re-encrypts stored cred administrators.
python\python.exe	Python Software Foundation	Legacy SMB sidecar (ser on 6002 to 6006 and 700 USE_OPSWAT_SMB_SE
mongo\mongod.exe, mongosh.exe; opswat.mdcs.mongomigrations\node.exe	MongoDB Inc., Node.js	Legacy database and mi systems upgraded from
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	Microsoft	Runs the scripts in A.5 a commands, with -Exec Bypass.
C:\Windows\System32\msiexec.exe, sc.exe, icacls.exe, cmd.exe, net.exe, systeminfo.exe, route.exe, ipconfig.exe	Microsoft	Installer and scripts: Sha components MSI, service

		cleanup, permissions on `customer.env`, host information for the support package.
`C:\ProgramData\OPSWAT-ONTAP-Connector\OPSWAT-ONTAP-Connector.exe`	OPSWAT	Section 4.

A.4 Microservices hosted by `dotnet.exe`, and loopback ports

If your product filters on the command line rather than the process path, allow `dotnet.exe` when its argument is one of these files (paths under `<INSTALL_DIR>\services\`; each service runs with its own folder as the working directory). Ports are bound to 127.0.0.1 unless the customer changed the listening IP.

Service	Command-line argument	Lo po
Licensing	`opswat.mdcs.license.service\opswat.mdcs.license.service.dll`	50 (gl
API Gateway	`opswat.mdcs.api.gateway\opswat.mdcs.api.gateway.dll`	80 80
Identity	`opswat.mdss.identity.service\opswat.mdss.identity.service.dll`	70
Logging	`opswat.mdcs.logging.service\opswat.mdcs.logging.service.dll`	–
Job Dispatcher	`opswat.mdcs.job.dispatcher\opswat.mdcs.job.dispatcher.dll`	–
Notification	`opswat.mdcs.notification.service\opswat.mdcs.notification.service.dll`	–
Scanning (1 to 30 instances)	`opswat.mdcs.scanning.service\opswat.mdcs.scanning.service.dll`	–
Discovery	`opswat.mdss.discovery.service\opswat.mdss.discovery.service.dll`	–
Remediations	`opswat.mdss.remediations.service\opswat.mdss.remediations.service.dll`	–
Storages	`opswat.mdss.storages.service\opswat.mdss.storages.service.dll`	–
Workflow Manager	`opswat.mdss.workflowmanager.service\opswat.mdss.workflowmanager.service.dll`	–
NFS sidecar (NFS module only)	`opswat.mdss.nfs.service\opswat.mdss.nfs.service.dll`	50

Other loopback ports: 6002 and 7002 SMB sidecar (legacy Python sidecar: 6002 to 6006 and 7002 to 7006); 5432 PostgreSQL; 5672, 15672, 25672 and 4369 RabbitMQ, its management interface, Erlang distribution and port mapper; 6379 Garnet cache; 27017 and 27778 legacy MongoDB and its migration tool.

All storage providers (Amazon S3, Azure, Google Cloud, Alibaba, Oracle, Box, Microsoft Graph, SharePoint, SFTP, FTP, SMB and NFS) are libraries loaded in-process by the Discovery, Storages, Remediations and Scanning services; there are no per-provider processes. The Scanning service verifies the Authenticode signature of its providers with Windows before loading them. An agent that blocks or delays library loading, or that strips or alters signatures, prevents these services from starting.

A.5 PowerShell scripts

All scripts live in `<INSTALL_DIR>\scripts\`, are signed by OPSWAT, Inc., and run on Windows PowerShell 5.1 (.NET Framework). Agents that hook the .NET Framework through the `APPDOMAIN_MANAGER_ASM` and `APPDOMAIN_MANAGER_TYPE` machine environment variables inspect every one of these processes and can add minutes to each.

Script	Started by	Purpose
`startup-orchestration.ps1`	The `mdss` service, at every start	Clears the RabbitMQ queue store, imports CA certificates, refreshes configuration and the web client `env-config.js`, applies HTTP or HTTPS, syncs the environment variables to Machine scope, reloads nginx, installs SharePoint components, runs database migrations.
`mdss.ps1`	The orchestration (`-c load-ca`, `-u enable_https`, `-u disable_https`) and administrators (`-c start/stop/restart/status`, `-u support_package`, `-u export_db`, others)	`load-ca` writes each certificate to a temp `.crt` file and imports it into `Cert:\LocalMachine\Root`; `support_package` runs `systeminfo`, `route print` and `ipconfig /all` and compresses logs; `export_db` runs `pg_dump.exe`.
`init.ps1`	The installer; the orchestration (`-setenv`)	Writes machine environment variables, broadcasts `WM_SETTINGCHANGE`, checks ports, creates firewall rules and the SharePoint scheduled task; `-uninstall` reverses it.
`dependencies.ps1`	The installer; the orchestration (`-refresh`)	Extracts and registers PostgreSQL, RabbitMQ, Erlang, nginx, Garnet, .NET, Python and MongoDB (`sc.exe`, `pg_ctl.exe`, `rabbitmq-service.bat`, `mongod.exe --install`); regenerates nginx and PostgreSQL configuration; updates or removes the ONTAP Connector.
`migrations_postgres.ps1`, `migrations.ps1`, `migrations_mongo_to_postgres.ps1`	The orchestration	Database migrations with `psql.exe` and, on upgraded systems, `node.exe`.
`install-sharepoint-components.ps1`	The orchestration; the scheduled task	Runs `msiexec.exe` to install the SharePoint client components MSI.

`mdss-service-logon.ps1`, `write-customer-env.ps1`, `InstallationStatus.ps1`, `monitor-installation.ps1`, `mdss-resp-client.ps1`	The installer; the other scripts	Restore the service logon account after an upgrade (`sc.exe config`, `icacls`), write `customer.env`, report installer progress, talk to the Garnet cache.

A.6 Registry keys

- `HKLM\SOFTWARE\OPSWAT\MetaDefender Storage Security`: `InstallDir` and installation state, written by the installer and read back on upgrade.
- `HKLM\SYSTEM\CurrentControlSet\Services\mdss`, `mdss_postgresql`, `mdss_rabbitmq`, `mdss_mongodb`, `opswat-ontap-connector`: service definitions, recovery options and logon account, written by the installer and the scripts.
- `HKLM\SOFTWARE\Ericsson\Erlang\ErlSrv\1.1\MDSS_RabbitMQ`: written by the RabbitMQ service registration.
- `HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Environment`: about 300 MDSS machine environment variables (ports, connection strings, feature flags, `DOTNET_ROOT`, `ERLANG_HOME`, `RABBITMQ_*`), rewritten at every service start.
- `HKLM\SOFTWARE\Microsoft\SystemCertificates\ROOT\Certificates`: customer CA certificates imported by `mdss.ps1 -c load-ca`. `enable_https` also uses the machine Personal (`My`) store temporarily.
- `HKLM\SOFTWARE\WOW6432Node\Data ONTAP\Clustered Data ONTAP Antivirus Connector\v1.0\data-lifs`: one `data-lif::<host>` value per protected NetApp SVM, maintained automatically by the ONTAP Connector.

A.7 Code signers

- **OPSWAT, Inc.:** `opswat.mdss.windows.service.exe`, `opswat.mdss.smb.service.exe`, `opswat.mdss.reencryption.cli.exe`, `OPSWAT-ONTAP-Connector.exe`, every `opswat.*.dll`, every `.ps1` in `<INSTALL_DIR>\scripts\`, and the installer itself.
- **Microsoft Corporation:** `dotnet.exe` and the .NET runtime libraries under `services\dotnet`, `GarnetServer.exe`, `powershell.exe`, `msiexec.exe`, `sc.exe`, `icacls.exe`.
- **Third-party runtime vendors** (PostgreSQL build vendor, Erlang/OTP, RabbitMQ, Python Software Foundation, Node.js, MongoDB Inc.): the corresponding runtimes under `services\`.
- **Not signed by OPSWAT:** `services\opswat.mdcs.webclient\nginx.exe`, and possibly some third-party runtime files. Use path or hash rules for these.

Enumerate the real signers on the installed system before writing publisher rules, because third-party runtime versions change between MDSS releases:

```
Get-ChildItem "<INSTALL_DIR>\services" -Recurse -Include *.exe,*.dll |
  Get-AuthenticodeSignature |
  Group-Object { if ($_.SignerCertificate) {
$_.SignerCertificate.Subject } else { "UNSIGNED" } } |
  Select-Object Count, Name | Sort-Object Count -Descending
```

Application control must also allow the PowerShell scripts in `<INSTALL_DIR>\scripts\` and the SQL and JavaScript files in `<INSTALL_DIR>\Utils\` and `<INSTALL_DIR>\services\opswat.mdcs.mongomigrations\`.

Appendix B. Changes since the 4.5.x article

If you applied the 4.5.x version of this list, update these items:

- **SFTP sidecar removed.** `opswat.mdss.sftp.service`, its loopback port 5003 and its firewall rule no longer exist. SFTP, FTP and all cloud providers run in-process inside the Discovery, Storages, Remediations and Scanning services.
- **Telemetry endpoint.** Telemetry goes to `activation.dl.opswat.com:443` by default; `mdss-telemetry.metadefender.com:4317` is used only when configured explicitly.
- **Installer behaviour.** The installer relaunches itself elevated through UAC, checks ports through WMI, runs inline PowerShell commands, and keeps upgrade state in `C:\ProgramData\OPSWAT` instead of `%APPDATA%\OPSWAT\MetaDefender Storage Security\backup\`. It also updates and uninstalls the ONTAP Connector.
- **ONTAP Connector synchronisation.** The connector polls `/api/account` every 15 seconds and rewrites `config.ini` and the `data-lif` registry values on its own. Its log folder is configurable [`LogPath`].
- **Signed scripts.** The PowerShell scripts carry an OPSWAT, Inc. Authenticode signature, so publisher-based script rules are possible. Blocked revocation checks for that signature are a separate cause of slow start-up [section 1].
- **Environment variables.** The machine environment holds about 300 MDSS variables, up from about 200.

Scanning NetApp Volumes with Tiered (Hot and Cold) data

Applies to: MetaDefender Storage Security protecting NetApp ONTAP volumes that tier inactive data: FabricPool on AFF and FAS systems, Amazon FSx for NetApp ONTAP, Azure NetApp Files with cool access, and Google Cloud NetApp Volumes with auto-tiering.

NetApp can move data that nobody has used for a while from fast SSD storage to cheaper object storage, and bring it back when it is used again. Before pointing MDSS at such a volume, customers usually ask three questions:

- Does scanning pull cold data back to the SSD tier?
- Does Real-Time Protection keep data hot, so that it never tiers?
- What does a scan cost on a volume that is mostly cold?

This article answers them, explains what happens behind the scenes, and walks through five examples.

---

1. The short answer

Question	Answer
Does MDSS move data between tiers?	No. Only ONTAP moves data. MDSS reads files through SMB or NFS like any other client.
Does discovery or polling change anything?	No. Discovery reads only metadata (names, sizes, dates), and metadata always stays on the SSD tier.
Does scanning a file pull it back to the SSD tier?	Not with NetApp's default settings. MDSS reads each file once from start to end, which ONTAP treats like an antivirus or backup read: cold data is served from the object tier and stays cold.
Does scanning make data cold?	No. Only time without reads makes data cold. Reading a file that is still on SSD restarts its cooling clock.
What does a scan of cold data cost?	Time and, on cloud platforms, object-store reads. The object tier is slower than SSD, and cloud platforms bill the reads.

One volume setting changes this picture: the **cloud retrieval policy**. If it is set to **On-read**, every file MDSS scans is written back to the SSD tier. See section 5.

---

2. How NetApp decides what is hot and what is cold

NetApp tracks temperature per 4 KB block, not per file.

1. When data is written, its blocks are **hot** and live on the SSD tier.
2. A background scan lowers the temperature of blocks that are not read: hot, then **warm**, then **cold**. A block becomes cold after the volume's **cooling period**: 31 days by default with the Auto policy, adjustable from 2 to 183 days.
3. Cold blocks are packed into 4 MB objects and moved to the object tier. The file keeps its name, size, dates and permissions. Users and applications see no difference, except that opening the file takes longer.
4. Reading a block that is still on SSD makes it hot again. What a read does to a cold block depends on how the file is read.

"Warm" is only a step on the way from hot to cold. There is no warm tier.

What happens	Effect on temperature
A file is written or changed	Its changed blocks are hot
A file on SSD is read	Its blocks are hot again; cooling starts over
A cold file is opened and browsed by an application (random reads)	Its blocks come back to SSD and are hot (Auto policy)
A cold file is read once from start to end (backup, antivirus, indexing, MDSS)	Its blocks stay cold and are served from the object tier
A folder is listed or file properties are read	Nothing. Metadata never tiers
Nothing happens for the cooling period	Blocks go hot, warm, cold, then move to the object tier

The same mechanism has a different name on each platform:

Platform	Fast tier	Cold tier	Feature name
ONTAP on AFF and FAS	Local (performance) tier	Cloud tier: StorageGRID, S3, Azure Blob, Google Cloud Storage	FabricPool
Amazon FSx for NetApp ONTAP	SSD storage	Capacity pool	Data tiering
Azure NetApp Files	Hot tier	Cool tier	Cool access
Google Cloud NetApp Volumes	Hot tier	Cold tier	Auto-tiering

Two volume settings matter for this article:

- **Tiering policy**: what may be tiered. *Auto* tiers cold user data and snapshots (the default in the FSx console). *Snapshot-only* tiers only snapshot blocks, so live files never go cold (the default from the FSx API and the ONTAP CLI). *All* tiers everything soon after it is written. *None* tiers nothing.
- **Cloud retrieval policy**: what a read of cold data does. *Default* follows the table above. *On-read* brings every block that is read back to SSD. *Never* brings nothing back. *Promote* brings the whole volume back.

---

3. What MDSS does on a tiered volume

MDSS does three things on a share. Only the last one reads file content.

Discovery. MDSS walks the folder tree and reads each file's name, size and timestamps. On NetApp this is metadata, served from SSD, and it changes nothing. Real-Time Protection in polling mode repeats this walk continuously, starting a new one as soon as the previous one finishes. A tiered file looks exactly like any other file during discovery; MDSS cannot tell hot files from cold ones.

Filtering. The workflow filters (file type, size, "Include files modified after") are applied to the discovered metadata. A file that is filtered out is never read.

Scanning. Each remaining file is downloaded once from start to end and sent to MetaDefender Core. This is the only step that touches data blocks, with the effects from section 2: blocks on SSD become hot again, cold blocks are read from the object tier and stay cold.

Which files reach the scanning step depends on the scan type:

Scan type	Files that are read
On-demand and scheduled scans	Every file in scope that passes the workflow filters
Real-Time Protection, polling mode	Only files created or changed after the real-time start date. The start date is the moment Real-Time Protection was enabled, unless "Pick files modified since" was set
On-access scanning through the NetApp ONTAP integration (Vscan)	The file a user is opening or closing. The user is reading it anyway

4. Examples

Example 1: a user saves a new file; Real-Time Protection is in polling mode. At 09:00 a user saves `Q3-report.xlsx` (2 MB) to the share. Its blocks are written to SSD and are hot. Within seconds MDSS lists the folder, sees a file with a new modification time, waits until the file stops changing, downloads it and sends it to MetaDefender Core. The file was already hot, so the read changes nothing. Every later polling cycle only lists the folder and never reads the file again. If nobody opens the file, ONTAP marks it cold after the cooling period and tiers it, exactly as if MDSS were not there.

Example 2: an application appends to a large file that is mostly cold. A 4 GB application log on the share has not been read for months and sits on the cold tier. Once a month the application appends about 50 MB. Those 50 MB are hot on SSD; the other 3.95 GB are still cold. MDSS sees the new modification time and downloads the whole file: 50 MB from SSD and 3.95 GB from the object tier. Afterwards the 50 MB are still hot and the 3.95 GB are still cold. The download took longer than it would from SSD, and nothing was written back.

Example 3: first on-demand scan of a mostly cold volume. A 20 TB volume uses the Auto tiering policy and the Default retrieval policy. 6 TB is hot and 14 TB is cold. MDSS lists the whole tree (metadata, from SSD), then reads every file: 6 TB from SSD and 14 TB from the object tier. After the scan the volume still holds 6 TB hot and 14 TB cold. The cooling clock of the 6 TB restarted, so files that were about to tier will tier a little later. Plan for the read speed of the object tier and, on FSx, Azure NetApp Files or Google Cloud, for the cost of reading 14 TB from it.

Example 4: the same scan with the On-read retrieval policy. Same volume, but the cloud retrieval policy is On-read. Now every cold block MDSS reads is written back to SSD, so the scan tries to bring 14 TB onto an SSD tier that holds 10 TB. On FSx, ONTAP stops bringing data back once the SSD tier is 90 percent full and stops all tiering at 98 percent. The scan still finishes, but the SSD tier is full and the volume is in a state nobody wanted. Set the retrieval policy back to Default before scanning a tiered volume.

Example 5: Real-Time Protection enabled with an old "Pick files modified since" date. Real-Time Protection is enabled on 1 October with "Pick files modified since" set to 1 January 2020. The first polling pass treats every file changed since 2020 as new and reads it, cold files included. This behaves like Example 3. Leave the field empty to protect only files that change from now on.

---

5. Recommendations

- **Keep the cloud retrieval policy at Default** (or Never) on volumes that MDSS scans. Do not use On-read or Promote on them.
- **Know how much is cold before the first full scan.** ONTAP's inactive data reporting, the FSx console and the Azure NetApp Files metrics show the size of the cold tier. Expect that part of the scan to run at object-tier speed and, on cloud platforms, to be billed as object-store reads.
- **Skip old files without reading them** with the workflow filter "Include files modified after". Filters work on metadata, so files outside the range are never downloaded. Files not modified within the cooling period are usually the cold ones.
- **Use Real-Time Protection for ongoing coverage.** It reads only files that change, and changed files are hot by definition.
- **Leave "Pick files modified since" empty** unless you need to catch up on a known period.
- **Keep SSD headroom.** AWS recommends keeping the FSx SSD tier under 80 percent. Normal user activity after a scan can still bring cold data back.
- **Run large first scans outside business hours.** Cold reads share the object tier's bandwidth with users who open cold files.

---

6. Frequently asked questions

Does MDSS move files from cold to warm, or from warm to cold? No. MDSS cannot move data between tiers and cannot see a file's temperature. Discovery and polling read metadata only. Scanning reads data, which makes SSD blocks hot and leaves cold blocks cold.

Will scanning undo the savings from tiering? No, with the Default or Never retrieval policy. Cold data stays cold. Only the On-read and Promote policies write data back.

Does the polling cycle keep my data hot? No. Listing folders and reading file properties does not touch data blocks.

Can MDSS skip cold files? Not by temperature, because SMB and NFS do not expose it. Use the "Include files modified after" filter instead; files not modified within the cooling period are usually the cold ones.

Is the NetApp ONTAP (Vscan) integration different? On-access scanning runs when a user opens or closes a file, so the data is being read anyway. On-demand scans through that integration read files the same way as an SMB share and follow this article.

What about the All tiering policy? New files move to the object tier soon after they are written, so even Real-Time Protection may read them from there. They are read from start to end and stay cold.

Further reading

- NetApp, FabricPool tiering policies
- NetApp, TR-4598 FabricPool best practices
- AWS, Volume storage capacity and data tiering for FSx for ONTAP
- Microsoft, Azure NetApp Files storage with cool access
- Google, Manage auto-tiering for Google Cloud NetApp Volumes

System Health APIs: which one to use

MDSS exposes three health endpoints on the API gateway. They answer different questions, so pick the one that matches what you want to know.

Endpoint	Question it answers	Credentials	Available
`GET /api/health`	Is the API gateway process up?	None from 4.6.0. Before 4.6.0, the `ApiKey` header of any user.	All versions
`GET /api/readyz`	Can MDSS serve requests right now?	None	From 4.7.0
`GET /api/healthz`	How is MDSS performing, and why?	`ApiKey` header of an administrator	All versions

Which one should I use?

- **A load balancer or monitor that needs up or down.** `GET /api/readyz` on MDSS 4.7.0 or later. On 4.6.x use `GET /api/health`; it stays `200` during a database, cache or broker outage, so it detects only a dead gateway. On 4.5.x and earlier `/api/health` needs an API key; a Read Only Administrator key is enough.
- **Kubernetes.** Liveness probe on `/api/health`, readiness probe on `/api/readyz`.
- **A script that waits for MDSS to start.** Poll `/api/health` until the process answers, then `/api/readyz` until MDSS is ready to log in and scan.
- **Slow scans, a pause or throttling.** `GET /api/healthz` with an administrator key, or the System tab in the UI. It explains the state and recommends an action.

What each endpoint sees

	`/api/health`	`/api/readyz`	`/api/healthz`
API gateway process	yes	yes	yes
Cache, database and message broker connections	no	yes	no (database capacity only)
Discovery, Scanning and Remediations reporting	no	yes	yes
Storage guardrail pause and throttling	no	yes, as `Degraded`	yes, with reasons and limits
MetaDefender Core instances	no	yes, `Degraded` when none is reachable	yes, per instance
Host resources, throughput, recommendations	no	no	yes

Never point a health check at `/`. It is a static page served by the web server and answers `200` even while the API is down.

Examples

```
# Is the gateway process up?
curl -i https://<mdss-host>/api/health

# Can MDSS serve? 200 or 503, no credentials (4.7.0 and later)
curl -i https://<mdss-host>/api/readyz

# Full operational status, administrator API key
curl -s -H "ApiKey: <administrator-api-key>" https://<mdss-
host>/api/healthz
```

Detailed pages

- Liveness API: GET /api/health
- Readiness API: GET /api/readyz
- System Health API: GET /api/healthz

Liveness API: GET /api/health

`GET /api/health` tells you whether the MDSS API gateway process is up and answering HTTP requests. It checks nothing else.

Available in all versions. No API key is needed from MDSS 4.6.0. On 4.5.x and earlier, send the `ApiKey` header of any user; a Read Only Administrator key is enough.

Request

Parameter	Required	Effect
`verbose=true`	no	Adds the body `{"apiStatus":"Ok"}`. Without it the `200` answer has an empty body.

The endpoint runs no query against the database, the cache or the message broker, and it does not ask the other MDSS services. It is safe to poll at any interval.

Response

HTTP	Body	Meaning
200	empty, or `{"apiStatus":"Ok"}` with `verbose=true`	The gateway process is up.
502	web server error page	The gateway is down or restarting. The web server in front of it could not reach it.
401		Only on 4.5.x and earlier, when no `ApiKey` header was sent.

What it does not tell you

`/api/health` keeps answering `200` while the database, the cache or the message broker is unreachable, and while Discovery, Scanning or Remediations are stopped. Users cannot log in or scan in those states, yet the process is alive.

- On MDSS 4.7.0 and later, use `GET /api/readyz` for that question. It needs no credentials either and answers `503` when MDSS cannot serve.
- On MDSS 4.6.x and earlier, `GET /api/healthz` with an administrator API key shows the state of each service.

Examples

Call it

```
curl -i https://<mdss-host>/api/health
curl -i "https://<mdss-host>/api/health?verbose=true"
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{"apiStatus":"Ok"}
```

On 4.5.x and earlier:

```
curl -i -H "ApiKey: <read-only-administrator-api-key>" https://<mdss-host>/api/health
```

Wait for the gateway to come up

```
until [ "$(curl -s -o /dev/null -w '%{http_code}' https://<mdss-
host>/api/health)" = "200" ]; do
  sleep 5
done
echo "API gateway is up"
```

Kubernetes liveness probe

```
livenessProbe:
  httpGet:
    path: /api/health
    port: https
  periodSeconds: 10
  timeoutSeconds: 5
  failureThreshold: 3
```

Why not `/`?

A web server that has lost its connection to the gateway still serves the login page at `/` with `200`. The same web server answers `502` on `/api/health`. Point health checks at `/api/health` or `/api/readyz`, never at `/`.

Notes

- Each API gateway answers for itself. Behind a load balancer, probe every instance.
- MDSS Cloud and other multi-tenant deployments: from 4.7.0 the endpoint answers without a tenant cookie.

Readiness API: GET /api/readyz

Applies to: MetaDefender Storage Security (MDSS) **4.7.0 and later**. Earlier versions do not have this endpoint and answer `401` or `404` on this path.

Credentials: none. The endpoint needs no API key, `Authorization` header or cookie. Credentials sent anyway are ignored.

`GET /api/readyz` reports whether this MDSS instance can serve requests. It is meant for load balancers, uptime monitors and Kubernetes readiness probes.

- Takes no parameters.
- The HTTP status code carries the verdict: `200` ready, `503` not ready. The body has the same shape in both cases.

- Every answer is current (`Cache-Control: no-store`) and is read from state the gateway already holds. A probe never queries the database, the cache or the broker itself, so any polling interval is safe.
- The body holds only statuses and short reason codes: no URLs, versions, sizes or keys.

```
{
  "status": "Healthy",
  "components": {
    "cache":         { "status": "Healthy" },
    "database":      { "status": "Healthy" },
    "messageBroker": { "status": "Healthy" },
    "discovery":     { "status": "Healthy" },
    "scanning":      { "status": "Healthy" },
    "remediation":   { "status": "Healthy" }
  }
}
```

Statuses

Status	HTTP	Meaning
`Healthy`	200	Everything reports fine.
`Degraded`	200	MDSS serves requests, but something needs attention: processing paused by the storage guardrail, throttling, a replica missing, or no scan instance reachable.
`Unhealthy`	503	MDSS cannot serve: the cache, database or message broker is unreachable, or a core service is not reporting.

The overall `status` is the worst component status. A component carries a `reason` only when it is not `Healthy`.

Reasons

Reason	Status	On	Meaning
`Unreachable`	Unhealthy	cache, database, messageBroker	The gateway's connection is down, or its check failed or timed out.
`Blocked`	Unhealthy	messageBroker	RabbitMQ has blocked the connection because of a memory or disk alarm on the broker. Nothing can be published.
`NotChecked`	Unhealthy	any	No check has completed yet (start-up), or the last result is older than 15 s.
`NotReporting`	Unhealthy	discovery, scanning, remediation	No replica of the service has sent a health report within its freshness window: the service is stopped, crashed or hung.
`Paused`	Degraded	discovery, scanning	Processing is paused by the storage guardrail: machine disk, database size or database disk at a critical level.
`NoInstanceReachable`	Degraded	scanning	Every MetaDefender Core instance returned errors on recent scan requests.
`Throttling`	Degraded	discovery, scanning, remediation	The service is throttling its intake because of resource or queue pressure.
`ReplicaMissing`	Degraded	discovery, scanning, remediation	One replica stopped reporting while another replica of the same service still reports.

When several reasons apply to one component, the first in this order is reported: `NotReporting`, `Paused`, `NoInstanceReachable`, `Throttling`, `ReplicaMissing`.

How the status is tracked

Component	What is watched	Refreshed	Goes down after	Comes back after
`cache`	The state of the gateway's own Redis connection	every 5 s	up to 5 s	up to 10 s: reconnect plus one cycle
`database`	The gateway's own queries. One `SELECT 1` with a 3 s timeout runs only when no query answered in the last 5 s or a connection failure was seen.	every 5 s	up to 10 s	up to 10 s
`messageBroker`	The state of the gateway's own RabbitMQ connection	every 5 s	up to 5 s	up to 5 s
`discovery`, `scanning`, `remediation`	Health reports every service broadcasts over RabbitMQ, every 10 s by default	on each report	`NotReporting` after 3 missed reports: about 30 s after a crash or hang, immediately after a clean stop	up to 10 s: the first report of the restarted service
`Paused`	The storage guardrail in the Job Dispatcher: checked every 30 s, sent in its health report every 60 s	on each report	up to 90 s after a capacity signal turns critical	up to 90 s after every signal is back to normal. Resume is automatic only while all signals can be measured.
`Throttling`	The autoscaler state of the service, plus the discovery queue thresholds	on each report	up to 10 s	when pressure is relieved or the service is idle. The autoscaler holds Throttle for at least 30 s.

Over time:

- **Start-up.** The gateway answers `503` until its first checks run and the first report of every service arrives, about 10 to 20 s after it starts. The components show `NotChecked` or `NotReporting`.
- **One instance at a time.** Each API gateway answers for itself. Behind a load balancer, probe every instance.
- **Forgotten replicas.** A replica silent for 180 s is removed. Until then it shows as `ReplicaMissing` if another replica of the same service still reports, or `NotReporting` if none does.
- **Logging.** The gateway logs a state change once: `Readiness check: the Database is Unhealthy (Unreachable)` and later `Readiness check: the Database is reachable again`. A `503` answered to a probe is logged at Information, not Error.

Examples

Call it

```
curl -i https://<mdss-host>/api/readyz
```

Only the HTTP code:

```
curl -s -o /dev/null -w "%{http_code}\n" https://<mdss-host>/api/readyz
```

Only the components that are not healthy:

```
curl -s https://<mdss-host>/api/readyz | jq '.components | to_entries[]
| select(.value.status != "Healthy")'
```

Database unreachable: `503 Service Unavailable`

```
{
  "status": "Unhealthy",
  "components": {
    "cache":         { "status": "Healthy" },
    "database":      { "status": "Unhealthy", "reason": "Unreachable"
},
    "messageBroker": { "status": "Healthy" },
    "discovery":     { "status": "Healthy" },
    "scanning":      { "status": "Healthy" },
    "remediation":   { "status": "Healthy" }
  }
}
```

`/api/health` still answers `200` in this state: the process is alive, but it cannot serve.

Paused by the storage guardrail: `200 OK`

```
{
  "status": "Degraded",
  "components": {
    "cache":          { "status": "Healthy" },
    "database":       { "status": "Healthy" },
    "messageBroker": { "status": "Healthy" },
    "discovery":      { "status": "Degraded", "reason": "Paused" },
    "scanning":       { "status": "Degraded", "reason": "Paused" },
    "remediation":    { "status": "Healthy" }
  }
}
```

The instance stays in the load balancer rotation. Free disk or database space, and the status returns to `Healthy` by itself.

Load balancer health check

Setting	Value
Path	`/api/readyz`
Success codes	`200`
Interval	10 to 30 s
Timeout	5 s
Unhealthy threshold	2
Healthy threshold	2

Do not point the health check at `/`: it is a static page served by the web server and answers `200` even while the API is down.

Kubernetes readiness probe

```
readinessProbe:
  httpGet:
    path: /api/readyz
    port: https
  periodSeconds: 10
  timeoutSeconds: 5
  failureThreshold: 3
```

What you see during an Amazon MQ (RabbitMQ) maintenance

1. A broker node restarts. Within 5 s `/api/readyz` answers `503`, `messageBroker: Unreachable`.
2. Every MDSS service restarts after losing the broker. While the API gateway restarts, the web server answers `502` on every endpoint, `/api/health` included, for a few seconds up to about half a minute.
3. The gateway is back: `503` with `NotChecked` or `NotReporting` for about 20 s while its checks and the service reports arrive.
4. Every service reports: `200`, `Healthy`.

If `503` with `NotReporting` persists for more than 2 to 3 minutes after the maintenance has ended, a service did not come back. Run `sudo mdss -c restart` on Linux, or restart the MDSS services on Windows.

Limitations

- A Redis that is frozen but keeps its connection open still shows `cache: Healthy`. A stopped or killed Redis is detected. Tracked in MDSS-15082.
- `NoInstanceReachable` reflects the errors the scanning service saw on recent scan requests. It is not a direct check of MetaDefender Core, and with no recent scans `scanning` stays `Healthy`.
- On MDSS Cloud and other multi-tenant deployments `/api/health` and `/api/readyz` answer without a tenant cookie and report deployment-level state only.

System Health API: GET /api/healthz

Applies to: MetaDefender Storage Security (MDSS) 4.6.x and earlier, `GET /api/healthz`.

The request needs an API key of an administrator, sent in the `ApiKey` header. A health check that needs no API key will be available starting with MDSS 4.7.0.

1. Two status fields per service

Each entry in `perServiceStats` has two independent fields:

Field	Question it answers	Values
`status`	Is the service processing normally?	`Healthy`, `Unhealthy`
`autoscalerStatus`	What is MDSS doing with the service's intake?	`Idle`, `Hold`, `Scaling Up`, `Scaling Down`, `Throttling`, `Paused`

`Healthy` with `Hold` is the normal state of a busy system: the service works at the level its host, its queues and the scan instances allow, and MDSS keeps its intake there. A service is `Unhealthy` only while it is `Throttling` or `Paused`.

2. Autoscaler states

Every 10 seconds each service checks its host CPU and memory, its queues and, for Scanning, every scan instance. Each check is **Relieved**, **Warning** or **Critical**. From these the Scanning service raises or lowers its intake, the number of files it takes from its queue at a time. Discovery and Remediations only monitor: they report `Idle` or `Hold` and never scale.

Value	Meaning	Enters when	Leaves when
`Idle`	Nothing to process.	The input queue is empty and nothing is in progress.	Work arrives.
`Scaling Up`	The intake is being increased.	Every check is Relieved with headroom to spare, confirmed 5 times in a row. `Hold` is shown while the confirmations are collected.	Right after the step. The next increase needs 5 new confirmations.
`Hold`	The intake stays where it is. The service runs at the level the system allows.	A check is in Warning but not yet for 2 consecutive runs, or every check is Relieved but close to its threshold, or confirmations for a scale-up are being collected, or throttling has just ended. Discovery and Remediations report `Hold` whenever they have work.	A check turns Critical (`Throttling`), a Warning persists (`Scaling Down`), headroom is confirmed (`Scaling Up`), or the work runs out (`Idle`).
`Scaling Down`	The intake is being reduced.	A check stayed in Warning for 2 consecutive runs: CPU above 90 %, memory above 80 %, the output queue above 70 % of its capacity, or a scan instance above its desired load.	Right after the step.
`Throttling`	The intake was halved at once to protect the system. The service is `Unhealthy`.	Any check is Critical: CPU at 95 % or more, memory at 90 % or more, the output queue at its critical size, or a scan instance returning too many errors. Discovery also throttles when one of its pending-file queues exceeds its limit.	After 30 seconds, if nothing is Critical any more; extended by 30 seconds while something still is. The service returns to `Hold`, never straight to `Scaling Up`.
`Paused`	File processing is stopped by the storage guardrail. All services are `Unhealthy`.	The MDSS disk, the database volume or the database size reaches 90 % of its capacity.	Every guardrail signal is back at 80 % or less. Checked every 30 seconds.

3. Overall status

The top-level `status` uses the same six values, chosen in this order:

1. `Paused`, if the guardrail has paused processing. `statusReason` is then `Disk` or `Database`.
2. `Throttling`, if Discovery is throttling.
3. Otherwise the most severe state of the Scanning replicas: `Throttling`, then `Scaling Up`, `Scaling Down`, `Hold`, `Idle`.
4. `Idle`, if no Scanning replica has reported in the last 3 minutes.

4. Scan instances

`scanningInstancesStats` reports each MetaDefender Core, Cloud or Cluster instance as the Scanning service sees it.

status	reason	Meaning
`Healthy`	`null`	The instance keeps up.
`Unhealthy`	`Overloaded`	The instance's queue is above the load MDSS wants to keep on it. MDSS sends it less.
`Unhealthy`	`429Responses`	More than 10 % of the last 100 responses were "429 Too Many Requests". MDSS scales down.
`Critical`	`BadResponses`	More than 10 % of the last 100 responses were errors. MDSS throttles scanning.

5. Limits

`activeLimit`, `perServiceStats[].currentLimit` and `perServiceStats[].reason` name the resource that stops a service from taking more work. `activeLimit` is `null` when nothing limits the system; otherwise it is the first of these reported by any service:

Value	Meaning
`Disk`	The MDSS disk is full enough to pause processing.
`Database`	The database volume or the database size is full enough to pause processing.
`Memory`	Host memory is the limit.
`Cpu`	Host CPU is the limit.
`Scanning Instances`	The scan instances are the limit.
`OutputThroughput`	Only in `reason`: the service throttles because its output queue cannot drain fast enough.

6. Recommendations

`currentRecommendation` is `null` when there is nothing to recommend. Otherwise `severity` is `Critical` while processing is paused, `Warning` while a limit is active and `Info` otherwise, and `key` is one of:

key	Shown when
REST_API_MSG_RECOMMENDATION_INCREASE_SPECS	The host is below the minimum specification: 16 vCPU and 32 GB, or 32 vCPU and 64 GB when MetaDefender Core runs on the same host.
REST_API_MSG_RECOMMENDATION_INCREASE_STORAGE	Processing is paused by the guardrail and the database is external.
REST_API_MSG_RECOMMENDATION_USE_EXTERNAL_DB	Processing is paused by the guardrail with a local database, or the local database is larger than 100 GB.
REST_API_MSG_RECOMMENDATION_INCREASE_CPU	Host CPU is the limit.
REST_API_MSG_RECOMMENDATION_INCREASE_MEMORY	Host memory is the limit.
REST_API_MSG_RECOMMENDATION_USE_EXTERNAL_MDCORE	The scan instances are the limit and MetaDefender Core runs on the MDSS host.
REST_API_MSG_RECOMMENDATION_INCREASE_MDCORE_PERFORMANCE	The scan instances are the limit and MetaDefender Core is external.
REST_API_MSG_RECOMMENDATION_USE_EXTERNAL_CACHE	Memory was the last limit and Redis runs on the MDSS host.

A recommendation stays visible after its limit clears, until another limit replaces it.

Which File Operations Trigger a Real-Time Processing Scan?

Applies to: MetaDefender Storage Security (MDSS) 4.6.x and later. Real-Time Processing on all storage categories, with either monitor type: Polling or Event-Based.

Real-Time Processing (RTP) scans files that are new or have changed. This article answers which everyday file operations count as a change: opening, downloading, editing, renaming, moving, or changing only a file's metadata. It gives the answer for each storage category and monitor type, then explains the rules behind it.

---

1. Storage categories

The answers depend on how a storage keeps and changes files, so they are given per category, as on the *Integrations* page. Storages in the same category behave the same. Where one of them differs, a note says so.

Category	Storages	Event-Based available for
Cloud Object Storages	Amazon S3, Wasabi, Cloudian, Oracle Cloud (S3 Compatible and Native), MinIO, Cubbit, Azure Blob, Google Cloud, Alibaba Cloud, Scality RING	All except Cubbit and Oracle Cloud (S3 Compatible)
File Storages	Box, OneDrive, SharePoint Online, Microsoft Teams, SharePoint On-Premise, Azure Files, OPSWAT MFT	Box and SharePoint Online
Network Attached Storages (NAS)	NFS, NFS with Data Wrapping, SMB, NetApp ONTAP, Amazon EFS, Dell Isilon, SFTP, FTP	SMB storages through an OPSWAT connector; any NAS storage through your own tool (see below)

Storages without Event-Based use Polling only.

Event-Based on NAS. MDSS does not watch NAS storages for changes: a notifier has to tell MDSS which file changed. OPSWAT provides notifiers for SMB only:

- **NetApp ONTAP**, through Vscan: ONTAP sends scan requests to the OPSWAT ONTAP Connector.
- **Other SMB-compatible storages** (Windows file servers, Samba and NAS shares), through the **OPSWAT SMB Connector**, a Windows service shown as OPSWAT SMB File Event Monitor, available from MDSS 4.7.0.

For any other NAS storage, such as NFS, SFTP or FTP, Event-Based works when your own tool notifies MDSS through the webhook described in *Event-Based request format*. Your own tool can also be used for SMB.

---

2. The short answer

- **Opening, viewing or downloading a file does not trigger a scan.** Reading a file changes nothing that MDSS looks at. If the application saves the file while it is open, for example with AutoSave, that save is an edit. The one exception is NetApp ONTAP, where ONTAP can ask for a scan when a file is opened (note 13).
- **Editing and saving a file triggers a scan of the new version.** On object storages, editing means uploading a new version over the object.
- **Renaming, moving and metadata-only changes depend on the category and the monitor type**, as shown below.

Polling

Operation	Cloud Object Storages	File Storages	NAS
Open or view a file (read only)	No	No	No (1)
Download a file	No	No	No
Edit and save a file	Yes	Yes	Yes
Rename a file	Yes (2)	Conditional (3)	Conditional (4)
Move a file to another folder	Yes (2)	Conditional (3)	Conditional (4)
Change only metadata or attributes	Conditional (5)	Conditional (6)	No (7)

1. Some applications create temporary files next to the file they open. For example, Microsoft Office creates a hidden owner file whose name starts with ~$ when a document on a network share is opened for editing. That file is new, so RTP can scan it. The document itself is not scanned again.
2. Object storages have no real rename or move: tools copy the object to its new name and delete the original. The copy is a new object with a new last-modified date, so it is scanned. A storage that renames in place, such as Azure Blob with a hierarchical namespace, can keep the object's dates; note 3 then applies.
3. Yes if the file was created or last modified on or after the job's start date (see section 3), or if the operation gives the file a new modified date. Otherwise No.
4. Yes only if the file was created or last modified on or after the job's start date. File servers keep a file's dates when it is renamed or moved within the same volume, including a move into the monitored folder from elsewhere on that volume. A move to another volume is a copy followed by a delete; on Windows file servers the copy gets a new created date, so it is scanned.

5. Yes when the change rewrites the object or gives it a new last-modified date. Editing user metadata does this on S3-compatible storages and Alibaba Cloud, which copy the object onto itself, and on Azure Blob and Google Cloud. No for changes that keep the date and the ETag, such as Amazon S3 object tags.
6. Yes if the change gives the file a new modified date. Otherwise No. On SharePoint On-Premise and Azure Files, a change that gives the file a new ETag is also scanned when the file is dated on or after the job's start date.
7. File servers do not change a file's dates or size when its attributes, permissions or owner change. Properties stored inside a file, such as Office document properties edited in File Explorer, are part of its content: changing them is an edit.

Event-Based

Operation	Cloud Object Storages	File Storages: SharePoint Online	File Storages: Box	NAS: OPSWAT SMB Connector	NAS: NetApp ONTAP or your own tool
Open or view a file (read only)	No	No	No	No	Depends on the notifier (13)
Download a file	No	No	No (11)	No	Depends on the notifier (13)
Edit and save a file	Yes	Yes	Yes (11)	Yes (12)	Depends on the notifier (13)
Rename a file	Yes (8)	Yes (10)	No (11)	Yes (12)	Depends on the notifier (13)
Move a file to another folder	Yes (8)	Yes (10)	Conditional (11)	Yes (12)	Depends on the notifier (13)
Change only metadata or attributes	Conditional (9)	Yes (10)	No (11)	No (12)	Depends on the notifier (13)

8. The documented cloud setups forward object-creation events (see section 4). A rename or move copies the object to its new name, which is an object creation, so it is scanned. An in-place rename (note 2) is not an object creation and is not scanned.
9. Yes when the change copies the object onto itself, which is an object creation: editing user metadata on S3-compatible storages and Alibaba Cloud. No on Azure Blob and Google Cloud, where editing metadata is not an object creation, and No for object tags.

10. For files inside the storage unit's scope: its document library, or its folder if one is set. A file moved out of the scope is not scanned. SharePoint reports every change it records on a file, including edits to library columns. In MDSS 4.6.x this includes some updates SharePoint makes on its own shortly after an upload, such as image tags or Office document properties, so a new file can be scanned a second time.
11. Box notifies MDSS through the webhook you create during setup, with the trigger **File Uploaded** (see *Event-based handling for Box*). Box sends File Uploaded when a file is uploaded, when a new version is saved, and when a file is moved into the folder that has the webhook. It does not send it for previews, downloads, renames or metadata changes.
12. The OPSWAT SMB Connector reports every file that is created, written or renamed in a watched share, once the file has settled, and ignores temporary names such as `~$report.docx` and `*.tmp`. A move within a share is a rename and a move into a share creates the file, so both are scanned. Attribute and permission changes on a file are not reported (connector 1.0.4 and later). Renaming a folder in place, or changing its permissions, makes the connector read the whole folder again, so its files are scanned again.
13. With NetApp ONTAP, ONTAP sends the OPSWAT ONTAP Connector a scan request when a file is accessed or modified, as its Vscan on-access policy decides, so opening or reading a file can also trigger a scan. With your own tool, MDSS scans every file the tool reports, so the answers depend on what it sends.

---

3. How Polling decides that a file needs a scan

Polling lists the files of the storage unit in cycles; a new cycle starts shortly after the previous one ends. A listing reads names, sizes and dates, never file content. A file is scanned when it passes two checks.

Check 1: the file is recent enough. The later of the file's created date and modified date must be on or after the job's start date. The start date is the moment the real-time job was last started, or the date chosen in **Pick files modified since**. The `DISCOVERY_SERVICE_<PROTOCOL>_RTP_HANDLING` setting of earlier versions is no longer used. A file changed in the last 10 seconds waits for a later cycle, so that MDSS reads complete files.

Check 2: MDSS has not scanned this version yet. MDSS remembers each version it scans by the file's path, its modified date and a version value from the storage:

Category	Dates MDSS reads	Version value
Cloud Object Storages	Last modified, and created where the storage reports it	ETag (Google Cloud: MD5 or CRC32C checksum)
File Storages	Created and Modified	Item ID (OneDrive, SharePoint Online, Microsoft Teams), SHA-1 of the content (Box), ETag (SharePoint On-Premise, Azure Files), file ID (MFT)
NAS	Created (where available) and Modified	File size and modified date (FTP: the server's checksum, when it provides one)

This explains the Polling table:

- **Open, view, download:** nothing changes, so the version is already known.
- **Edit and save:** the file gets a new modified date, so this is a new version.
- **Rename and move:** the path changes, so this is a version MDSS has not scanned, and it is scanned if it passes check 1. On object storages the renamed object is a new copy with a new date, so it always passes. Elsewhere, a file dated before the start date passes only if the operation gives it a new date.
- **Metadata or attributes only:** the path stays the same, so this is a new version only if the modified date or the version value changes, and check 1 still applies.

Two consequences to plan for:

- **Renaming or moving a folder** changes the path of every file in it. On object storages every object under the folder is copied, so all of them are scanned. On File Storages and NAS, every file in the folder that passes check 1 is scanned again.
- **The start date can move forward.** When **RTP history retention** (Data Retention) removes old results of a running job, MDSS moves the job's start date to the retention cut-off, so that the removed files are not scanned again. Files dated before the new start date then no longer pass check 1.

---

4. How Event-Based decides that a file needs a scan

In Event-Based mode the storage, or a system acting for it, tells MDSS which file changed, and MDSS scans that file. MDSS does not compare dates or versions and does not look at the event type, and **Pick files modified since** does not apply. Every notification for a file in the storage unit's scope can lead to a scan, including a scan of a version MDSS has already scanned.

What triggers a scan therefore depends on what is notified:

- **Cloud Object Storages:** the event setup you create in the cloud decides (see Event-Based Real-Time Cloud Functions Examples). The documented setups forward object-creation events: uploads, overwrites and copies.
- **SharePoint Online:** MDSS creates and renews a Microsoft Graph subscription for each document library in the unit's scope. SharePoint reports every file added or changed, including renames, moves and column edits. MDSS ignores deleted items and folders.
- **Box:** the triggers of the Box webhook decide (note 11).
- **NAS:** the notifier decides: the OPSWAT SMB Connector (note 12), NetApp ONTAP through Vscan, or your own tool (note 13).

Because every notification can lead to a scan, Event-Based mode can scan the same file several times. AutoSave and co-authoring, for example, can save several versions of a document within minutes, and each one can be scanned. Polling scans the version it finds in each cycle, once the file has been unchanged for 10 seconds.

Polling and Event-Based compared

	Polling	Event-Based
What starts a scan	A new version found by the next listing cycle	A notification
Files dated before the start date	Not scanned, even after a rename or move, unless the operation gives them a new date (object storage copies always get one)	Scanned when notified; the start date does not apply
Same version scanned again	No	Possible, once per notification
Available for	All storages	See section 1

5. Recommendations

- **To scan renamed and moved files whatever their age** on File Storages and NAS, create the Polling job with **Pick files modified since** set to **Use the oldest available timeframe**. Every file then passes check 1: the first cycles scan the existing files in scope, and after that every rename or move is scanned. Expect a large first backlog on big storages.
- **To rescan files that have not changed**, for example after an engine update, run an on-demand or scheduled scan. Real-Time Processing only scans new versions.
- **Cover the gaps of Event-Based.** Notifications that arrive while an Event-Based job is stopped are refused, so changes made in that time may never be scanned. Run an on-

demand scan for that period, and consider a scheduled scan at a long interval, for example weekly, as a safety net.

- **Forward only the events that the setup guides use.** MDSS scans the file named in every notification, whatever the event type. On object storages, forward only object-creation events: delete events end as “file not found” errors, and other events add scans. On Box, keep File Uploaded as the only trigger: File Previewed or File Downloaded would scan a file every time it is opened or downloaded.
- **Run one OPSWAT SMB Connector per share.** Two connectors watching the same share make MDSS scan every file twice.
- **Workflow filters still apply.** A changed file that the workflow’s filters exclude is not scanned.
- **To find out why Polling skipped a file**, search the Discovery service log (included in the support package) for `predates the real-time start date`. MDSS logs this line for each file dated before the job’s start date.

Errors and Troubleshooting

Why does MetaDefender Storage Security get stuck in a remediation loop, causing inaccessibility?

> **Check Your Version:**
>
> This article applies to all MetaDefender Storage Security releases deployed on Windows and Linux systems.

Scenario:

In certain environments, MetaDefender Storage Security (MDSS) repeatedly retries remediation actions on specific files, even when the remediation fails or when files are already processed as *allowed* (depending on customer configuration).

As a result, multiple files may become stuck in the remediation process, eventually causing the MDSS web interface to become unresponsive until the MDSS service is restarted.

Root Cause

The remediation loop occurs when MDSS retries remediation actions multiple times without an effective retry limit per real-time process or global configuration.

By default, MDSS performs up to **3 remediation retries** with **30-second intervals**. When too many real-time processes (RTPs) are running on a single MDSS instance, **resource contention** can further increase the likelihood of remediation loops or degraded responsiveness.

Resolution

- **Distribute workload** Deploy MDSS in an **active-active** model by distributing RTP workloads across multiple MDSS instances. For example: one instance handling 3 RTPs and another handling 4 RTPs instead of running all RTPs on a single instance.
- **Review remediation retry configuration** Adjust the remediation process retry parameters according to the official documentation: https://www.opswat.com/docs/mdss/configuration/remediation-process-retries
- **Investigate failed remediations** Check the scan report details of failed files to identify the underlying causes, such as:
  - Permission or ownership issues
  - File locking by other applications

- Interference from antivirus or endpoint protection

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

How to fix 'Name does not resolve' error in RHEL / Rocky Linux 8 with Podman

Overview

By default, **Rocky Linux / RHEL 8.x** uses the **CNI DNS plugin** as its container networking DNS backend (most common with Podman + CNI). This can cause services to fail to communicate with the database or external network.

Rocky Linux / RHEL 9.x ships with **aardvark-dns** and **netavark**, Podman's modern network stack, which resolves these issues.

> **Info**
>
> **Long-term recommendation: Upgrade to Rocky Linux / RHEL 9.** The steps below are a workaround for systems that cannot yet be upgraded.

Temporary solution: Change DNS library

> **Info**
>
> ⚠️ Note: This procedure requires an active internet connection and must be performed with `sudo` (root) privileges.

Step 1: Stop MDSS

```
sudo mdss -c stop
```

Step 2 — Install `netavark` and `aardvark-dns`

```
sudo dnf install -y netavark aardvark-dns
```

Step 3 — Configure Podman to Use the new network backend

Edit `/etc/containers/containers.conf` and add the following section:

```
[network]
network_backend = "netavark"
dns_enabled = true
```

> **Info**
>
> If the file does not exist, create it at `/etc/containers/containers.conf`.

Step 4 — Clean Up Old Podman Configuration

Remove the old CNI network configuration and restart Podman:

```
sudo podman container prune -f
sudo podman network prune -f
sudo rm -rf /etc/cni/net.d/*
sudo rm -rf /var/lib/cni/networks/*
sudo systemctl restart podman
```

Step 5 - Validate the new DNS library is used

Run these commands:

```
echo "Network Backend: $(sudo podman info --format
'{{.Host.NetworkBackend}}')"
echo "DNS Packages: $(rpm -qa | grep -E 'dnsname|netavark|aardvark')"
```

You should see something like this:

```
Network Backend: netavark
DNS Packages: aardvark-dns-1.16.0-1.el9.x86_64
```

Step 6 — Start MDSS

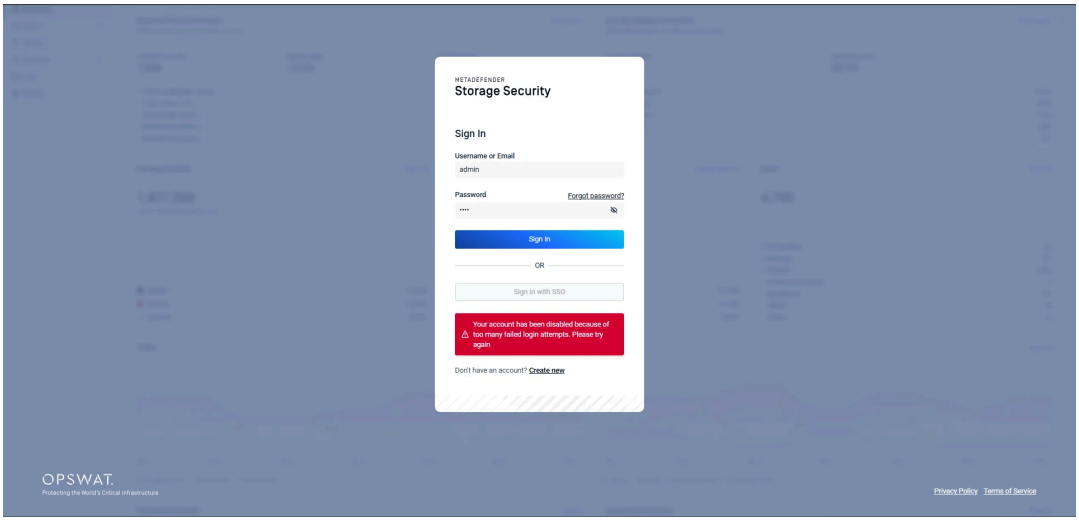
```
sudo mdss -c start
```

It should now work as expected.

Why is my MetaDefender Storage Security account disabled due to too many failed login attempts?

> **Check Your Version:**
>
> This article applies to MetaDefender Storage Security v4.0.x releases deployed on Windows and Linux systems.

Users may encounter the error message “Your account has been disabled because of too many failed login attempts. Please try again” when attempting to log into the MetaDefender Storage Security dashboard.



Starting with version v4.0.x, MetaDefender Storage Security implemented an enhanced security mechanism to **block users who repeatedly enter incorrect passwords**, even with the **correct username or email**.

This mechanism temporarily blocks the user from logging in. To regain access, the user must **wait for a specific interval** and then **enter the correct password**.

To prevent security breaches, the system applies a progressive delay mechanism after each failed login attempt. The delay increases exponentially, doubling after every failed attempt and starting from the very first failure. This approach helps slow down repeated login attempts while still allowing legitimate users to retry

The delay progression works as follows:

After **waiting for the specified interval**, you **can attempt to log in again** using the **correct password**.

Failed attempt	Delay
1st	0s → **1s**
2nd	1s → **2s**
3rd	2s → **4s**
4th	4s → **8s**
...	doubles each time
after ~12 failures	capped at **60 minutes**1st1st

> **Support:**
>
> If Further Assistance is required, please proceed to create a support case or chat with our support engineer.

Troubleshooting errors when configuring HTTPS communication for webserver

> **Check Your Version:**
>
> This article applies to all MetaDefender Storage Security releases deployed on Windows and Linux.

Issue:

The MetaDefender Storage Security (MDSS) web GUI is inaccessible via HTTPS despite enabling HTTPS and uploading SSL certificates. Symptoms include:

- The HTTPS enablement script reports SSL certificate validation failure.
- The web browser shows an endless loading loop when trying to connect via HTTPS.

Root Cause:

- Depending on the error message when running `mdss.ps1 -u enable_https`:
  - "**WARNING: Invalid certificate structure!**" indicates the SSL certificate is not in the correct PEM format required by MDSS, causing SSL certificate validation to fail (e.g. if the certificate is in DER format, it will cause this error).
  - "**WARNING: Invalid key structure!**" indicates the private key might have been extracted from a .pfx file with a passphrase, which MDSS does not currently support.

- The internal CA certificate also needs to be installed in the Windows "Trusted Root Certification Authorities" store to be trusted by MDSS (or `/etc/mdss/ca_certificates/` on Linux).

Resolution:

1. Verify that the SSL certificate and private key are in **PEM format**, indicated by headers such as "-----BEGIN CERTIFICATE-----" and "-----BEGIN PRIVATE KEY-----".
2. If extracting the certificate and private key from the .pfx file, use OpenSSL commands to extract without a passphrase:
   - Extract certificate: `openssl pkcs12 -in certificate.pfx -clcerts -nokeys -out ssl.crt`
   - Extract private key **without passphrase**: `openssl pkcs12 -in certificate.pfx -nocerts -out ssl.key -nodes`
3. Validate the certificate and key formats using OpenSSL:
   - Verify certificate: `openssl x509 -in ssl.crt -text -noout`
   - Verify private key: `openssl rsa -in ssl.key -check` (should output "RSA key ok")
4. Confirm that the MD5 hashes of the certificate and private key match:
   - `openssl x509 -noout -modulus -in ssl.crt | openssl md5`
   - `openssl rsa -noout -modulus -in ssl.key | openssl md5`
5. Ensure the internal CA root certificate is installed in the Windows "Trusted Root Certification Authorities" store so MDSS can trust the certificate chain (or `/etc/mdss/ca_certificates/` on Linux).
   - Refer to: https://www.opswat.com/docs/mdss/configuration/custom-ca-certificates
6. Disable HTTPS on MDSS before re-enabling it manually:
   - Run: `mdss.ps1 -u disable_https` (Windows)
   - Copy the validated `ssl.crt` and `ssl.key` files to `C:\Program Files\OPSWAT\MetaDefender Storage Security\config\nginx\certificates` on Windows (or `/etc/mdss/webclient/ on Linux`)
   - Run: `mdss.ps1 -u enable_https` (Windows)
7. Access the MDSS web console via HTTPS URL: `https://`
8. If issues persist, generate and upload a support package for further investigation.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a support case or chat with our support engineer.

Why does a report fail to load with error "API took too long to respond"?

> **Check Your Version:**
>
> This article applies to all MDSS versions

> When opening a report in MDSS, the UI displays the following error:

`API took too long to respond. Please try again later`

The database queries are taking more than 60 seconds to return, triggering the default web server (nginx) timeout.

Increase the **nginx proxy timeout** and **PostgreSQL query timeout** values.

Both changes are required.

For Windows

1. Stop MDSS services
2. Increase Nginx Timeout Create a backup of the nginx configuration file at `C:\Program Files\OPSWAT\MetaDefender Storage Security\webclient\default.conf`, then edit the **default.conf** file Add or modify the following values in the appropriate `location` or `http` block:

```
proxy_connect_timeout 300s;
proxy_send_timeout 300s;
proxy_read_timeout 300s;
```

3. Increase PostgreSQL Timeout Edit the MDSS environment configuration file at `C:\Program Files\OPSWAT\MetaDefender Storage Security\config\customer.env` Modify the `POSTGRESQL_URL` parameter to include increased timeout values. Example:

```
POSTGRESQL_URL=Host=postgres;Port=5432;Username=mdss;Password=
<password>;Database=MDSS;Timeout=300;CommandTimeout=300
```

4. Start MDSS services

For Linux

1. Stop MDSS service

```
sudo mdss -c stop
```

2. Increase Nginx Timeout Run the bellow command

```
sudo awk '/proxy_intercept_errors off;/{print; print "
proxy_connect_timeout 300s;"; print "        proxy_send_timeout
300s;"; print "        proxy_read_timeout    300s;"; next}1'
/etc/mdss/webclient/default.conf | sudo tee
/etc/mdss/webclient/default.conf.tmp && sudo mv
/etc/mdss/webclient/default.conf.tmp /etc/mdss/webclient/default.conf
```

This will create a backup of the config file `/etc/mdss/webclient/default.conf`and increase the nginx timeout to 300ms

3. Increase the timeout for the PostgreSQL database by modifying the `POSTGRESQL_URL` in the `/etc/mdss/customer.env` file:
   - Set `Timeout` to 300.
   - Set `CommandTimeout` to 300.

Here's an example of how the `POSTGRESQL_URL` should look:

```
POSTGRESQL_URL=Host=postgres;Port=5432;Username=mdss;Password=a3ZPzF1VI
N3RzCu5YWrl;Database=MDSS;Timeout=300;CommandTimeout=300
```

4. Start MDSS service

```
sudo mdss -c start
```

> **Note**
>
> - Increasing the timeout allows **long-running report queries** to complete successfully.
> - If the issue occurs frequently, it may indicate:
>   - very large report datasets
>   - insufficient database resources
>   - missing database indexes
>
> In those cases, further **database performance tuning** may be required.

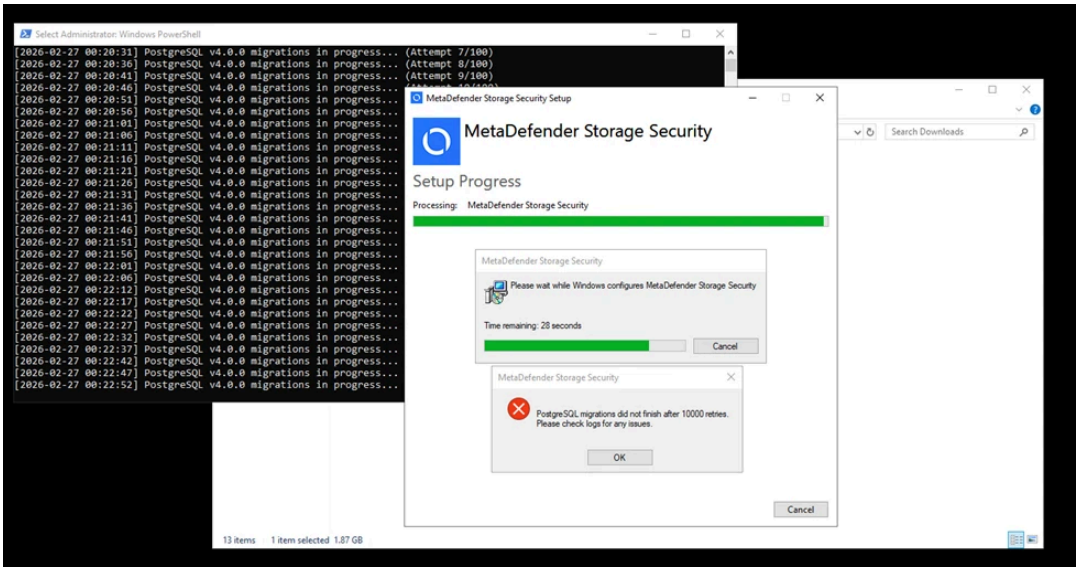
> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

Why am I receiving the error "PostgreSQL migrations did not finish" when installing MetaDefender Storage Security?

> **Check Your Version:**
>
> This article applies to MetaDefender Storage Security v4.0.x releases deployed on Windows systems.

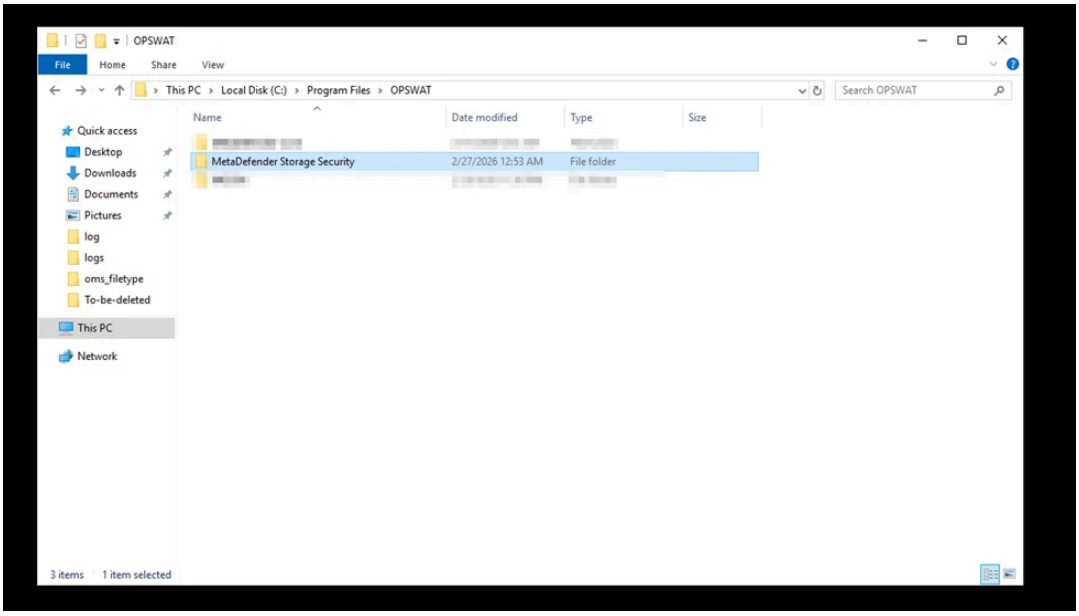
Issue:

- When installing MetaDefender Storage Security on Windows server, the error "PostgreSQL migrations did not finish after 10000 retries. Please check logs for any issues.“ is received, even on a fresh installation.



Main Cause:

- This occurs because MetaDefender Storage Security was previously installed on the machine but not completely removed after an uninstall or problem. This leftover data causes the installer to recognize an existing installation and attempt an upgrade instead of a fresh install.



Resolution:

Please follow the steps below to safely remove and reinstall MetaDefender Storage Security (MDSS):

1. Stop MDSS Services

Stop all related MDSS services from **Services (services.msc)** or by using the following commands:

```
net stop mdss
net stop mdss_postgresql
net stop mdss_rabbitmq
```

---

2. Remove MDSS Windows Services

Run the following commands as Administrator to remove existing services:

```
sc.exe delete "mdss_postgresql"
sc.exe delete "mdss_rabbitmq"
sc.exe delete "mdss"
```

---

3. Remove Installation Directory

Delete the default installation folder:

```
C:\Program Files\OPSWAT\MetaDefender Storage Security
```

Ensure all files are removed.

---

4. Handle Locked Files (if any)

If some files cannot be deleted:

- Ensure all MDSS services are stopped
- Close any open log files
- Restart the system if needed

Advanced:

- Use **Process Explorer** to identify which process is locking the files

---

5. Remove Registry Entry

Clean up the registry by running:

```
reg delete "HKEY_LOCAL_MACHINE\SOFTWARE\OPSWAT\MetaDefender Storage Security" /f
```

---

6. Reboot the System

Restart the machine to ensure all remaining components are fully cleared.

7. Reinstall MDSS

Run the installer again after completing all steps above.

> **Support:**
>
> If Further Assistance is required, please proceed to create a support case or chat with our support engineer.

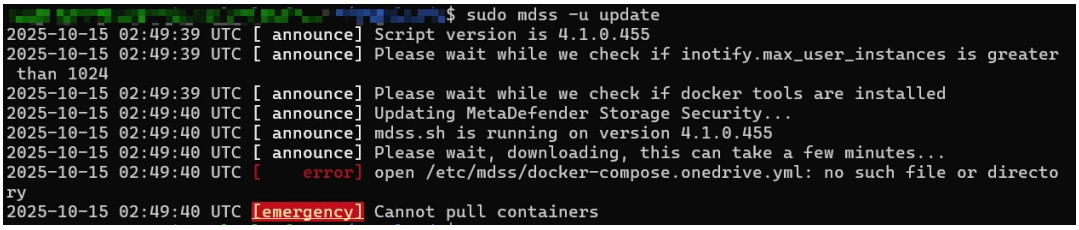
Why am I getting a "stat /etc/mdss/docker-compose.onedrive.yml: no such file or directory" error when upgrading MetaDefender Storage Security from 3.6.1 to 4.1.0?

> **Check Your Version:**

> This article applies to the upgrade from MetaDefender Storage Security **3.6.1** to **4.1.0** on Linux systems. Starting with 4.6.0, the upgrade no longer fails on an unknown module name: a value in `ENABLED_MODULES` that has no matching compose file is skipped silently. Replacing `onedrive` with `graph` is still recommended, because `onedrive` no longer enables anything.

Issue

During an upgrade from **MetaDefender Storage Security** version **3.6.1** to **4.1.0**, you may encounter the following error:



`[error] stat /etc/mdss/docker-compose.onedrive.yml: no such file or directory [emergency] Cannot pull containers`

This error occurs when the `ENABLED_MODULES` variable in the configuration file `/etc/mdss/customer.env` includes the value `onedrive`, which is no longer valid in version 4.1.0.

Cause

In version 4.1.0, the `onedrive` module was renamed to `graph`, and the `/etc/mdss/docker-compose.onedrive.yml` file was replaced by `/etc/mdss/docker-compose.graph.yml`.

The upgrade process expects the updated module name, and referencing `onedrive` in customer.env causes it to look for a non-existent configuration file [`docker-compose.onedrive.yml`], resulting in the error.

Resolution

To fix the issue:

1. Open the file `/etc/mdss/customer.env`.
2. Locate the line that starts with `ENABLED_MODULES=`.

```
ENABLED_MODULES=smb,box,onedrive,sftp,nfs,mft,ftp
```

3. If `onedrive` is listed, replace it with `graph`.

```
ENABLED_MODULES=smb,box,graph,sftp,nfs,mft,ftp
```

Example:

- **Before:** ENABLED_MODULES=smb,box,**onedrive**,sftp,nfs,mft,ftp
- **After:** ENABLED_MODULES=smb,box,**graph**,sftp,nfs,mft,ftp

4. Save the file and re-run the upgrade command: `sudo mdss -u update`

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a support case or chatting with our support engineer.

Why is RabbitMQ consuming excessive memory in MetaDefender Secure Storage (MDSS) Kubernetes deployment?

> **Check Your Version:**
>
> This article applies to **MetaDefender Storage Security (MDSS)** deployment on **Kubernetes** environment.

Overview

This article addresses a specific memory exhaustion issue affecting RabbitMQ components within MetaDefender Secure Storage (MDSS) Kubernetes deployments. It details the symptoms of the memory leak and provides a configuration change to the Helm Chart to permanently resolve the instability.

Issue

RabbitMQ v4.1.0 instances might experience continuous memory growth (memory leaks). This is specifically caused by the improper cleanup of “deleted consumer metrics” within the `rabbitmq:4.1.0-management-alpine` container image.

Symptoms include:

- Steadily climbing RAM usage without release.
- High resource consumption attributed to the Management Plugin.
- Potential Out of Memory (OOM) crashes or pod restarts.

Solution

To resolve this, you will need to switch the container image to the standard Alpine version, which excludes the Management Plugin.

1. Modify the Configuration

Open the `values.yaml` file used for your MDSS deployment.

2. Update the Image Tag

Locate the RabbitMQ image section and remove the `-management` suffix.

3. Change From:

```
image: rabbitmq:4.1.0-management-alpine
```

Change To:

```
image: rabbitmq:4.1.0-alpine
```

4. Apply the Changes

Execute the Helm upgrade command to apply the new configuration:

```
helm upgrade <release-name> ./<chart-path> -f values.yaml -n <namespace>
```

5. Monitor

Verify the pod is running the new image and monitor memory usage over the next 24 hours to confirm stability.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

Is the Scheduled Scan in MetaDefender Storage Security Based on OS Time or an Internal Timer?

> **Check Your Version:**
>
> This article is applied to MetaDefender Storage Security deployments.

Overview

This article explains how the **scheduled scan timing** works in **MetaDefender Storage Security**, specifically whether it relies on the operating system's time or an internal timer mechanism within the product.

Question

There may be uncertainty about how **scheduled scans** are triggered in MetaDefender Storage Security — specifically whether the scheduling is determined by the **operating system's local time** or by an **internal timer** within the product.

Clarification

- The **scheduled scan** in MetaDefender Storage Security runs according to the **operating system's local time**.
- For example, if a scheduled scan is configured to run **every 1 hour**, and the system time is currently **4:03**, the next scan will initiate at **5:03 (OS time)**.

Additional Notes

- Ensure that the **system clock** and **timezone settings** on the host operating system are accurate, as these directly affect the timing of scheduled scans.
- Any drift or misconfiguration in system time may result in delayed or early scan execution.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

Why does MDSS installation might fail with "net/http: TLS handshake timeout" when pulling Docker containers?

> **Check Your Version:**
>
> This article applies to all MDSS releases deployed as a Docker Image.

Overview

During first start or installation, **MetaDefender Storage Security (MDSS)** pulls multiple Docker images from Docker Hub's CDN (`production.cloudflare.docker.com`). In some environments, this process fails with errors like:

```
failed to copy: httpReadSeeker: failed open: failed to do request:
Get "https://production.cloudflare.docker.com/registry-
v2/docker/registry/v2/blobs/sha256/...":
net/http: TLS handshake timeout

[emergency] Cannot pull containers
```

Key observations from the affected systems:

- DNS, ping and `curl` tests to Docker Hub and Cloudflare all worked (TLS handshakes succeeded).
- Pulling **one MDSS image at a time** with `docker pull opswat/<image>:<tag>` worked.
- The error occurred **only when MDSS tried to pull many images in parallel**.

Conclusion: the network itself was healthy, but **too many concurrent image downloads** caused some TLS handshakes to time out (combination of Docker's defaults, CDN behavior, and/or network limits). This is consistent with Docker's documented behavior where reducing `max-concurrent-downloads` stabilizes image pulls under load.

Resolution

1. Reduce Docker's concurrent downloads

Goal: Force Docker to pull layers sequentially (or with lower parallelism), reducing pressure on the network and avoiding TLS handshake timeouts.

1. Create or edit /etc/docker/daemon.json on the MDSS host:

```
{
"max-concurrent-downloads": 1
}
```

You can later tune this to `2` or `3`, but using `1` is the safest starting point to verify the fix. Docker's `max-concurrent-downloads` is an officially documented setting controlling concurrent layer downloads.

2. Restart Docker:

```
sudo systemctl restart docker
```

3. Re-run the MDSS start/activation:

```
./mdss.sh -c start
```

2. Optional: Re-try with a modest concurrency value

If installation works with `1` but you want faster future upgrades:

1. Adjust `daemon.json`, for example:

```
{
  "max-concurrent-downloads": 2
}
```

2. Restart Docker and test an image pull or a service update.

If timeouts reappear, reduce back to 1.

3. Verify network devices (optional but recommended)

Even with reduced concurrency, it is good practice to verify with the network team that:

- No firewall, IDS/IPS, or TLS-inspection device is limiting or throttling HTTPS connections to:
  - `registry-1.docker.io`
  - `production.cloudflare.docker.com`
- There are no strict connection or rate limits for outbound HTTPS from that host.

Many Docker handshake timeout reports involve some interaction between Docker's parallel pulls and upstream devices, so confirming this can help avoid future issues.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chat with one of our support engineers**.

How to troubleshoot slow historical data retrieval from the AVEVA PI Server?

> **Check Your Version:** This article applies to MetaDefender NetWall used with the OPSWAT PI Connector.

Issue

Historical data transfer from the Blue side to the Red side is slower than expected. Symptoms include delayed backfill of historical values, slow snapshot updates, and general performance degradation that becomes more noticeable under higher load or during large backfill operations.

Possible Causes

This issue may occur in either of the following situations:

- The AVEVA PI Data Archive on the Red side is taking longer than expected to process historical-data requests. The Red Flanker measures this time but does not control it.
- The PI Buffer Subsystem is not enabled on the Red Flanker, so writes to the destination PI Server are not buffered.

Retrieval time is measured from the moment the Red Flanker issues the AVEVA PI AF SDK call `PIPoint.RecordedValues()` until the call returns. This call retrieves historical data from the PI Data Archive. If the call itself is slow, the delay is occurring inside the PI Data Archive, not in the OPSWAT components.

Troubleshooting Steps

1. Enable the PI Buffer Subsystem on the Red Flanker

Confirm that the PI Buffer Subsystem is enabled on the Red Flanker node. Buffering allows writes to the destination PI Server to be queued rather than blocking on each write, and is required for the reference performance figures below to apply.

2. Collect the PiHistoryStore statistics

The `PiHistoryStore` statistics are generated once per minute and recorded in both syslog and the Windows Event Log on the Red Flanker. Collect several consecutive samples covering a period when the slowness is observed.

The following sample is from an OPSWAT reference environment and is shown here as a **healthy baseline**, not as an example of the problem:

> 09:26:28.67 PI_STATS 3460-pi HistoryStoresTotal 18082
>
> 09:26:28.67 PI_STATS 3460-pi HistoryStoresPerSecond 0.0
>
> 09:26:28.67 PI_STATS 3460-pi Last History Store Time 7/28/2026 9:22:30 AM

> 09:26:28.67 PI_STATS 3460-pi Collective Main Connection Status: Primary
>
> 09:26:28.67 PI_STATS 3460-pi PiHistoryStore(09:25:53) 3460-pi Blocks:12116 Fetches:12115 elapsed 13.01secs, Red Fetch longest:11ms, avg:1.07ms, total:12.96secs, BlueSort 1ms
>
> 09:26:28.67 PI_STATS 3460-pi PiHistoryStore(09:25:53) longest red fetch length:84 avg:28.37 # of values fetched:343672 Total Blue Values:343672

In this sample:

- 12,116 history blocks containing 343,672 values were transferred from the Blue side to the Red side.
- The Red Flanker performed 12,115 historical-data reads and retrieved 343,672 values.
- The longest individual read took 11 milliseconds; the average read took 1.07 milliseconds.
- The reads consumed 12.96 seconds of read time within a 13.01 second sampling interval.
- Each read returned an average of 28.37 values, and the largest returned 84 values.

3. Compare your figures against the reference baseline

Metric	Reference environment	Indicates slow PI Data Archive response
Average read time (`avg`)	~1 ms	Tens of milliseconds — for example, ~42 ms
Longest read time (`longest`)	~11 ms	Consistently above ~50 ms
Average values per read (`avg length`)	~28 values	Higher averages increase per-read cost

An average read time in the tens of milliseconds against a reference of roughly 1 millisecond means the PI Data Archive is taking substantially longer to satisfy each request. Larger average read lengths also raise the cost of each individual read, so consider both figures together.

4. Tuning Historical Backfill for best results

If the comparison in step #3 looks good, please refer to the following documentation for guidance on tuning the configuration to optimize performance:Optimizing AVEVA Pi Historical Backfill - MetaDefender Optical Diode

5. Escalate to AVEVA with supporting evidence (If Needed)

If the `PIPoint.RecordedValues()` call consistently takes significantly longer than the reference results, open a support case with AVEVA and provide:

- PI Server logs covering the same period
- Red Flanker event logs covering the same period

AVEVA Support can determine why historical-data retrieval is taking longer than expected.

Reference Environment

The reference figures above were produced in an OPSWAT environment with the following characteristics:

- The PI System contains approximately 264,000 tags.
- Two physical computers connected through a 1 Gbps network.
- One computer runs the OPSWAT Red Flanker and acts as the PI API node.
- The other computer runs the PI Server.
- The PI Buffer Subsystem is enabled.

Recommendation

Capture `PiHistoryStore` statistics routinely rather than only during an incident, so a healthy baseline for your own environment is available for comparison.

> **Support:** If **Further Assistance** is required, please proceed to log a support case or chatting with our support engineer.

How to resolve “Remediation Reached the Configured Timeout” for SMB Storage?

> **Check Your Version:** This article applies to all MetaDefender Storage Security

Issue

When running a **Discovery workflow with Remediation** against SMB storage, some files may fail with the following error:

> opswat.mdcs.job.dispatcher.ObjectProcessingStuckException: Remediation reached the configured timeout —> System.TimeoutException: 1 or more hours passed since MetaDefender Storage Security started remediations for the file — End of inner exception stack trace —

Possible issues

By default, MetaDefender Storage Security waits up to **1 hour** for a remediation task to complete. If the remediation process takes longer than the configured timeout, the file is marked as failed.

This can happen when:

- The server does not meet the recommended system requirements.
- The SMB storage configuration is not optimized for MetaDefender Storage Security.
- The storage contains a large number of files or large file sizes.
- SMB remediation takes longer than NFS remediation due to protocol or environment differences.

******

Resolutions**

1. Verify the Server Meets System Requirements

Confirm that the server hosting MetaDefender Storage Security meets the required system specifications.

Refer to the official system requirements:

General System Requirements - MetaDefender Storage Security

2. Use the Recommended SMB Storage Approach

For SMB-compatible storage, OPSWAT highly recommends using the recommended integration approach documented below:

Recommended approach when adding SMB units on Linux deployments - MetaDefender Storage Security

This approach can help improve reliability and reduce remediation-related timeout issues when working with SMB shares.

3. Increase the Remediation Timeout

If remediation tasks require more than the default timeout of **1 hour**, increase the configured remediation timeout (to 3 hours for example).

On the MetaDefender Storage Security server, edit the `customer.env` file and add the following value:

```
REMEDIATION_STUCK_TIMEOUT_HOURS=3
```

If this value already exists, update it to:

```
REMEDIATION_STUCK_TIMEOUT_HOURS=3
```

For instructions on modifying `customer.env`, refer to:

Custom configuration - MetaDefender Storage Security

> **Support:** If **Further Assistance** is required, please collect a new MetaDefender Storage Security support package and proceed to log a support case or chatting with our support engineer.

Slow 'mdss.ps1' commands on Windows

Applies to: MetaDefender Storage Security 4.5.2 and later, installed on Windows Server.

Symptoms

- `mdss.ps1 -c restart` takes about 10 minutes instead of 1 to 2 minutes.
- `mdss.ps1 -c load-ca` and other `mdss.ps1` commands take 1 to 2 minutes.
- Every PowerShell script in `<install folder>\scripts` waits about 50 seconds before it starts. No error is logged and the scripts complete normally.

Cause

Starting with MDSS 4.5.2, the PowerShell scripts shipped with MDSS are digitally signed. When a Software Restriction Policy (SRP) is applied to the server through Group Policy, even with the default security level "Unrestricted", Windows verifies the signature of every script before PowerShell runs it. Part of that verification is online:

- a revocation check of the signing certificate against DigiCert (OCSP and CRL), and

- a refresh of Microsoft's trusted-root and disallowed-certificate lists from `ctldl.windowsupdate.com`.

If the server cannot reach these hosts and the firewall silently drops the connections, every attempt runs to its timeout: about 20 seconds for the revocation check and 2 x 15 seconds for the Microsoft lists, about 50 seconds per script. MDSS runs many scripts per command, so the waits add up to minutes.

Confirm

Run in an elevated PowerShell on the MDSS server. Adjust the path to your install folder.

```
Measure-Command {
[System.Management.Automation.Security.SystemPolicy]::GetLockdownPolicy
("C:\Program Files\OPSWAT\MetaDefender Storage
Security\scripts\init.ps1", $null) }
```

Expected: under 1 second. If `TotalSeconds` is in the tens, this article applies.

Optional checks:

```
reg query
"HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers"
Test-NetConnection ocsp.digicert.com -Port 80
Test-NetConnection ctldl.windowsupdate.com -Port 80
```

If the registry key exists, an SRP is applied to the server. `TcpTestSucceeded : False` after a long wait means the firewall drops the connection.

Resolution (recommended)

Allow outbound HTTP (TCP port 80, not 443) from the MDSS server to:

- `ocsp.digicert.com`
- `crl3.digicert.com`
- `crl4.digicert.com`
- `cacerts.digicert.com`
- `ctldl.windowsupdate.com`

The checks then complete in milliseconds and Windows caches the results.

Workaround for servers without Internet access

Both changes only stop Windows from waiting for connections that fail anyway. Validated on Windows Server 2025 with MDSS 4.5.2.

1. Stop the Microsoft certificate-list downloads. In an elevated command prompt:

```
reg add
"HKLM\SOFTWARE\Policies\Microsoft\SystemCertificates\AuthRoot" /v
DisableRootAutoUpdate /t REG_DWORD /d 1 /f
reg add
"HKLM\SOFTWARE\Policies\Microsoft\SystemCertificates\AuthRoot" /v
EnableDisallowedCertAutoUpdate /t REG_DWORD /d 0 /f
```

2. Make the DigiCert lookups fail immediately. Add these lines to `C:\Windows\System32\drivers\etc\hosts`, then run `ipconfig /flushdns`:

```
0.0.0.0 ocsp.digicert.com
0.0.0.0 crl3.digicert.com
0.0.0.0 crl4.digicert.com
0.0.0.0 cacerts.digicert.com
```

   This step has no effect if the server uses a WinHTTP proxy. Check with `netsh winhttp show proxy`.

3. Open a new PowerShell window and run the command from **Confirm** again. Expected: well under 1 second. No reboot is needed. MDSS picks up the change at the next `mdss.ps1 -c restart`.

Note: Microsoft recommends keeping the disallowed-certificate list update enabled. On a server without Internet access that update cannot run anyway. If you later allow the access, revert the workaround so the updates resume.

Revert the workaround

```
reg delete
"HKLM\SOFTWARE\Policies\Microsoft\SystemCertificates\AuthRoot" /v
DisableRootAutoUpdate /f
reg delete
"HKLM\SOFTWARE\Policies\Microsoft\SystemCertificates\AuthRoot" /v
EnableDisallowedCertAutoUpdate /f
```

Remove the four lines from the hosts file and run `ipconfig /flushdns`.

Reference

- Microsoft: Configure trusted roots and disallowed certificates

Is MetaDefender Storage Security (MDSS) Affected by PHP2Shell (CVE-2026-109618 and CVE-2026-109619)?

> ## Check Your Version
>
> Based on the product-level security assessment, no PHP component was identified in the reviewed MetaDefender Storage Security application or its bundled container dependencies.

> ## Security Assessment: Not Affected
>
> MetaDefender Storage Security (MDSS) is not affected by CVE-2026-109618 or CVE-2026-109619 within the assessed product scope.

Overview

PHP2Shell refers to two security vulnerabilities in the PHP interpreter: **CVE-2026-109618** and **CVE-2026-109619**. Under certain conditions, an affected PHP runtime may allow remote code execution.

Product Assessment

The assessment reviewed the application's container images and software dependencies and did not identify PHP.

Is Any Action Required?

No product-specific patch, hotfix or configuration change is required for the assessed product to address these two PHP vulnerabilities. PHP installed separately as part of another application or service must be assessed independently.

Note: This article addresses only the two CVEs above. It does not assess other vulnerabilities or independently installed third-party software.

Additional Resources

- PHP Security Release Information — October 21, 2026

Support

If you have questions about this security assessment or a reported vulnerability finding, please contact OPSWAT Support.

This article was last updated on 2026-10-22.

Troubleshooting Guide

Diagnostic

Logs location

Unix

Type of logs	Location
Internal services logs	/etc/mdss/services_logs/{service-name}/
Installation logs	/var/log/mdss/
Script logs	/etc/mdss/logs

Windows

Type of logs	Location
Script logs	C:\Program Files\OPSWAT\MetaDefender Storage Security\logs\
Internal services logs	C:\Program Files\OPSWAT\MetaDefender Storage Security\services{service-name}\logs\
Windows Service logs	C:\Windows\System32\opswat.mdss.windows.service\
Installation logs	C:\Users\AppData\Local\Temp\ C:\Program Files\OPSWAT\MetaDefender for Secure Storage\logs\

How to read the logs

The log files are plain text files that can be opened with any text editor.

For MetaDefender Storage Security, the default logging level is set to **Information**. To access additional log levels, you need to enable Extra Logging in the Configuration. Extra logging is recommended only for troubleshooting purposes and should be enabled only when necessary.

For more details, refer to this article: How to Enable Extra Logging in the Configuration?

File Locations

- **Linux**: Log files are located at `/etc/mdss/services_logs`
- **Windows**: The default path is `C:\Program Files\OPSWAT\MetaDefender Storage Security\services\`

Log Format

The messages stored in the service logs follow this format:

```
[TIMESTAMP] [LEVEL] service.name: Message
```

Example:

```
[2024-06-12 10:17:55Z] [Information] nfs.service: HTTP POST
/ObjectOperations/Download responded 200 in 37.0824 ms
```

Log Levels

- **Debug : Provides detailed information useful for troubleshooting.**
- **Information**: The default logging level, capturing general operational messages.
- **Warning**: Highlights potentially harmful situations.
- **Error**: Logs issues that affect functionality or cause interruptions.
- **Fatal**: Reserved for very critical issues that require immediate attention.

> **Support:**
>
> If you have any difficulty reading MetaDefender Storage Security logs, please create a support case or chat with our support engineer.

Analysis Guide

This guide helps customer support representatives efficiently locate error messages in product logs when troubleshooting customer issues.

Storage-type specific code (Amazon S3, Azure, Google Cloud, Box, SharePoint, SFTP, FTP, and so on) runs inside the central Storages, Discovery and Remediations services, and inside the Scanning service when it downloads files, so their errors appear in those services' logs. Only NFS and SMB have a separate sidecar service with its own log folder.

Behavior	Log folder under `services_logs`
Failed to add a storage unit	`mdss_storagesservice` `mdintegrations_nfsservice` / `mdintegrations_smbservice` (NFS and SMB storages only)
Failed to add MD Core or Cloud instance	`mdcs_scanningservice` (connection, API key and saving the instance) `mdss_workflowmanager` (workflow rule checks)
Failed to start a scan No files are discovered	`mdss_workflowmanager` (scan start) `mdss_discoveryservice` `mdcs_jobdispatcher` (scan scheduling and job state) `mdintegrations_nfsservice` / `mdintegrations_smbservice` (NFS and SMB storages only)
Failed to scan a file No files are scanned	`mdcs_scanningservice`
Failed to remediate a file Remediations not applied Sanitization failed Sanitization not applied	`mdss_remediationsservice` `mdintegrations_nfsservice` / `mdintegrations_smbservice` (NFS and SMB storages only)
Login or API key problems	`mdss_identityservice` `mdcs_apigateway`
License not applied or expired	`mdcs_licensingservice`
Notifications (email, webhook) not sent	`mdcs_notificationservice`
Syslog or other external log output not received	`mdcs_loggingservice`

How to generate a support package

Linux deployment

Run the following command:

```
sudo mdss -u support_package
```

In case of a successful generation, you should see the following message:

```
Support package was exported to
/etc/mdss/mdss_3.6.1.10601_support_package_2025-06-03_16-16-44.tar
```

Windows deployment

> **PowerShell Compatibility Notice**
>
> Our product has been thoroughly tested and optimized for compatibility with **PowerShell 5.x**, which is the default Windows PowerShell version included with Windows Server 2019, 2022 and 2025.
>
> When generating a support package for troubleshooting purposes, please ensure that you use **Windows PowerShell 5.x** and not PowerShell 7 (PowerShell Core). This will make sure that all diagnostic information is correctly captured and formatted for our analysis tools.

Before you begin, please run the following command in an elevated **PowerShell** to allow the utility script (**mdss.ps1**) to be executed

```
PS> Set-ExecutionPolicy RemoteSigned
```

In order to generate a support package please run the following commands in the scripts folder within the installation folder (by default *C:\Program Files\OPSWAT\MetaDefender Storage Security\scripts*)

```
PS> .\mdss.ps1 -u support_package
```

In case of a successful generation, you should see the following message:

```
MetaDefender Storage Security support package was exported to
C:\Program Files\OPSWAT\MetaDefender Storage
Security\support_package_{date}.zip
```

For Kubernetes Deployment

In case of issues that require further investigation by OPSWAT support, a package can be generated using the `sp.sh` script what contains pod logs, configurations and details regarding the Kubernetes deployment. In order to generate the support package, make the `sp.sh` script executable and run the following command:

./sp.sh

Notes:

- The namespace name where the application is deployed needs to be specified as an argument, otherwise the default namespace is used.

- The script uses `kubectl` to extract the required information, make sure `kubectl` is installed and configured with the correct context.
- The script generates the support package as a zip file in the same directory where it is executed.

Maximum limit

The package can grow very large on systems with extensive logs. In order to prevent this, you can use the -m / --max-size argument as such:

Linux

```
sudo mdss -u support_package -m 500M
sudo mdss -u support_package -m 1G
sudo mdss -u support_package -m 1.5G
```

Windows

```
.\mdss.ps1 -u support_package
.\mdss.ps1 -u support_package -MaxSize 500M
.\mdss.ps1 -u support_package -MaxSize 1G
.\mdss.ps1 -u support_package -m 1.5G
```

How the size budget is allocated internally

When -m is provided, the total budget is split into two parts:

1. Settings/host info reservation — 50MB reserved for non-log data (settings, host info, env files). Configurable via env var **RESERVED_FOR_SETTINGS**.
2. Logs budget = max_size - 50MB — everything remaining goes to logs.

The logs budget itself is then divided into:

- Container logs — each container is capped at the last 25,000 lines (configurable via **CONTAINER_LOG_LINES**)
- Script logs — always included in full (usually small)
- Service logs — whatever budget remains after container + script logs

How service logs are collected (the two-phase approach)

Phase 1 — Fair representation: Takes the N most recent log files from each service (default: 1 file, configurable via **MIN_FILES_PER_SERVICE**). This guarantees every service has at least some representation in the package, even if the budget is tight. **Phase 2** —

Fill remaining budget: Any budget left after Phase 1 is filled with the most recently modified log files across all services, globally sorted by timestamp — most recent first.Files are never arbitrarily truncated — they are only truncated if they exceed the remaining budget, and when truncated, the last N bytes are kept (most recent content) via tail -c.

Environment variable overrides

Variable	Default	Description
RESERVED_FOR_SETTINGS	50MB	Space reserved for settings/host info
MIN_FILES_ PER_SERVICE	1	Minimum files per service in Phase 1
CONTAINER_LOG_LINES	25000	Max container log lines when -m is active
SERVICES_LOG_PATH	./services_logs	Path to service log files

> **Default limit**
>
> There is a maximum size set by default, which, if not modified it limits the support package to **2GB** on Windows and **5GB** on Linux.

How to Enable Extra Logging in the Configuration?

Logging supports different log levels to have the desired granularity. Default logging is set to **Information**.

To change log level adjust **MinimumLevel.Default** as follows:

- Debug: provides detailed information useful for troubleshooting.
- Information: the default logging level, capturing general operational messages.
- Warning: only highlights potentially harmful situations.
- Error: logs only issues that affect functionality or cause interruptions.
- Fatal: reserved for very critical issues that require immediate attention.

A typical `appsettings.json` configuration file appears as follows depends on the desired Serilog Sink:

- for **File Serilog Sink**

> **Info**
>
> Default maximum size of a log file is 500Mb which can be adjust by changing `fileSizeLimitBytes`

> Rolling interval is configured to be daily by the `rollingInterval`or on file size limit `rollOnFileSizeLimit: true`
>
> Default limit of retained files is set to 10 using the `retainedFileCountLimit` property

```
{
  "Serilog": {
    "Using": [],
    "MinimumLevel": {
      "Default": "Information",
      "Override": {
        "Microsoft": "Warning",
        "System": "Warning",
        "Microsoft.Hosting.Lifetime": "Warning"
      }
    },
    "Enrich": [ "FromLogContext", "WithMachineName", "WithProcessId",
"WithProcessName", "WithThreadId" ],
    "WriteTo": [
      {
        "Name": "File",
        "Args": {
          "buffered": false,
          "fileSizeLimitBytes": 500000000,
          "outputTemplate": "[{Timestamp:u}] [{Level}] {Namespace}:
{Message:lj}{NewLine}{Exception}",
          "path": "logs/log.txt",
          "retainedFileCountLimit": "10",
          "rollingInterval": "Day",
          "rollOnFileSizeLimit": true
        }
      }
    ]
  }
}
```

- for **HTTP Serilog Sink**

```
{
  "Serilog": {
    "MinimumLevel": {
      "Default": "Information",
      "Override": {
        "Microsoft": "Warning",
        "System": "Warning",
        "Microsoft.Hosting.Lifetime": "Information"
      }
    },
    "Enrich": [
      "FromLogContext",
      "WithMachineName",
      "WithProcessId",
      "WithProcessName",
      "WithThreadId"
    ],
    "WriteTo": [
      {
        "Name": "Http",
        "Args": {
          "requestUri": "<REQUEST_URI>",
          "queueLimitBytes": null
        }
      }
    ]
  }
}
```

- for **Syslog Serilog Sink**

```
{
  "Serilog": {
    "Using": [
      "Serilog.Sinks.Syslog"
    ],
    "MinimumLevel": {
      "Default": "Information",
      "Override": {
        "Microsoft": "Warning",
        "System": "Warning",
        "Microsoft.Hosting.Lifetime": "Information"
      }
    },
    "Enrich": [
      "FromLogContext",
      "WithMachineName",
      "WithProcessId",
      "WithProcessName",
      "WithThreadId"
    ],
    "WriteTo": [
      {
        "Name": "TcpSyslog",
        "Args": {
          "host": "<HOST>",
          "port": 514,
          "appName": "MetaDefenderStorageSecurity",
          "format": "RFC5424",
          "facility": "<SYSLOG_FACILITY_NAME>",
          "outputTemplate": "[{Timestamp:u}] [{Level}] {SourceContext}:
{Message:lj}{NewLine}{Exception}"
        }
      }
    ]
  }
}
```

Linux-based systems

Use the **appsettings.json** file from the `configurations` directory

Note: The default path on Linux is `/etc/mdss/configurations`.

> On Linux, logging settings are applied immediately after any changes, so a system restart is not required.

Windows-based systems

On Windows systems, each service has its own logging configuration file. You can locate the `appsettings.json` file by navigating to the service's directory and then accessing the specific `<service>` folder.

Note: The default path on Windows is `C:\Program Files\OPSWAT\MetaDefender Storage Security\services\`.

> On Windows, logging settings take effect only after the service has been restarted.

Enable debug logs for script commands

In order to troubleshoot please run :

```
sudo LOG_LEVEL=7 [mdss-command]
```

```
PS> .\mdss.ps1 [mdss-command] -log_level=7
```

where **mdss-command** is the command that has an issue. For example, if MetaDefender Storage Security cannot be started please run the following command:

```
sudo LOG_LEVEL=7 mdss -c start
```

```
PS> .\mdss.ps1 -c start -log_level 7
```

Resolving error code 1722 when OPSWAT ONTAP Connector tries to connect to ONTAP AV Connector

Overview

This document describes a troubleshooting and resolution procedure for a scenario where the **OPSWAT ONTAP Connector** receives **RPC_S_SERVER_UNAVAILABLE**** (error 1722, 0x6BA)** when connecting to a named pipe on a hardened Windows Server, even though the pipe is confirmed to exist on the server.

The issue was observed against a server running NetApp ONTAP AV Connector, where the OPSWAT ONTAP Connector connects to the pipe `\\127.0.0.1\pipe\ntapvscp`. The same pattern applies to any same-host RPC-over-named-pipes integration that uses an SMB-

style pipe path (`\\127.0.0.1\pipe\...`, `\\localhost\pipe\...`, or `\\<hostname>\pipe\...`).

Symptoms

The OPSWAT ONTAP Connector logs an entry similar to the following when the registration call fails:

```
[2026-05-06 11:23:49] [Error] [1722] VS_Registration2 failed
```

Additional indicators:

- The Win32 error code `1722` (`RPC_S_SERVER_UNAVAILABLE`) appears in the OPSWAT ONTAP Connector log.
- The named pipe is confirmed to exist on the server (visible under `\\.\pipe\`).
- The connection target is the loopback interface (`127.0.0.1`), so no network device or external firewall is involved.
- The host has been hardened according to a security baseline (CIS, STIG, or vendor-specific hardening).

Root Cause

Named-pipe access through an SMB-style path (for example `\\127.0.0.1\pipe\ntapvscp`) routes through the SMB redirector and requires:

1. A listener on TCP 445.
2. A valid inbound firewall rule allowing the connection.
3. SMB Server bound to a network interface (which is what creates the listener).

On a hardened host, all three of these layers are commonly disabled to reduce the attack surface. The combination produces the observed failure chain:

```
App opens \\127.0.0.1\pipe\ntapvscp
    -> Routed through the SMB redirector
        -> SMB attempts to connect to 127.0.0.1:445
            -> No listener on 445
                -> SMB returns ERROR_NETNAME_DELETED (64) or
ERROR_UNEXP_NET_ERR (59)
                    -> RPC layer surfaces RPC_S_SERVER_UNAVAILABLE
(1722)
```

Note that the AV Connector itself, the named pipe, and the local RPC subsystem are healthy. Only the SMB transport layer used to reach the pipe is broken.

Diagnostic Procedure

Run the following commands in an elevated PowerShell session on the affected host.

1. Confirm the named pipe exists

```
Get-ChildItem \\.\pipe\ | Where-Object Name -like "*ntapvscp*"
```

2. Confirm RPC over the local NPFS path works (bypasses SMB)

```
rpcping -s . -t ncacn_np -e \pipe\ntapvscp -v 3
```

A successful result (`Completed 1 calls in N ms`) proves the pipe and the local RPC stack are healthy.

3. Reproduce the failure over the SMB path

```
rpcping -s 127.0.0.1 -t ncacn_np -e \pipe\ntapvscp -v 3
```

A failure with `Exception 1722` and an inner status of `0x40` (`ERROR_NETNAME_DELETED`) or `0x3B` (`ERROR_UNEXP_NET_ERR`) confirms the SMB path is broken.

4. Test TCP 445 reachability on loopback

```
Test-NetConnection 127.0.0.1 -Port 445
```

`TcpTestSucceeded : False` indicates nothing is listening on TCP 445.

5. Identify which layer is broken

```
# Is the Server service running?
Get-Service LanmanServer, LanmanWorkstation

# Is anything listening on 445?
Get-NetTCPConnection -LocalPort 445 -State Listen
netstat -ano | findstr :445

# Is SMB Server bound to any network interface?
Get-SmbServerNetworkInterface

# What does the firewall rule for SMB-In look like?
Get-NetFirewallRule -DisplayName "File and Printer Sharing (SMB-In)" |
  Format-Table DisplayName, Enabled, Profile

# Is "File and Printer Sharing for Microsoft Networks" bound to the
active adapter?
Get-NetAdapterBinding -ComponentID ms_server

# What is the active network profile?
Get-NetConnectionProfile
```

A typical hardened-host signature is:

- `LanmanServer` is running.
- `Get-NetTCPConnection -LocalPort 445 -State Listen` returns nothing.
- `Get-SmbServerNetworkInterface` returns nothing.
- The `File and Printer Sharing (SMB-In)` firewall rule is `Enabled: False`.
- `Get-NetAdapterBinding -ComponentID ms_server` shows `Enabled: False` on the active adapter.

Resolution

The fix restores SMB inbound on the host while keeping the customer's hardening intact for everything except loopback.

Run the following in an elevated PowerShell session:

```
# 1. Enable the inbound firewall rule for SMB
Enable-NetFirewallRule -DisplayName "File and Printer Sharing (SMB-In)"

# 2. Scope the rule to loopback only (do not expose 445 on the network)
Set-NetFirewallRule -DisplayName "File and Printer Sharing (SMB-In)" -
RemoteAddress 127.0.0.1

# 3. Re-bind "File and Printer Sharing for Microsoft Networks" to the
active adapter
Enable-NetAdapterBinding -Name "Ethernet" -ComponentID ms_server

# 4. Restart the Server service so it picks up the new binding
Restart-Service LanmanServer -Force
```

If the active adapter is named something other than `Ethernet`, substitute the correct name from `Get-NetAdapterBinding -ComponentID ms_server`.

Verification

After applying the fix, run the following to confirm each layer is healthy:

```
# Adapter binding restored
Get-NetAdapterBinding -ComponentID ms_server
# Expected: active adapter shows Enabled : True

# SMB Server bound to interface
Get-SmbServerNetworkInterface
# Expected: at least one row returned

# TCP listener up on 445
Get-NetTCPConnection -LocalPort 445 -State Listen
# Expected: LocalAddress :: , LocalPort 445, State Listen,
OwningProcess 4

# Loopback reachability
Test-NetConnection 127.0.0.1 -Port 445
# Expected: TcpTestSucceeded : True

# SMB session establishment works
net use \\127.0.0.1\IPC$
# Expected: "The command completed successfully"

# RPC over the SMB path now works
rpcping -s 127.0.0.1 -t ncacn_np -e \pipe\ntapvscp -v 3
# Expected: Completed 1 calls in N ms
```

Once `rpcping` succeeds against the SMB path, reproduce the original failure with the OPSWAT ONTAP Connector. The connector should now open the pipe and complete the RPC

call without `RPC server is unavailable`, and `VS_Registration2` should complete successfully.

Why This Happens on Hardened Hosts

Common hardening baselines disable inbound SMB to reduce lateral-movement risk. Typical changes include:

- Disabling the `File and Printer Sharing (SMB-In)` firewall rule for all profiles.
- Unchecking “File and Printer Sharing for Microsoft Networks” on every network adapter, which removes the binding the SMB Server uses to create its TCP 445 listener.
- Stopping or disabling the `LanmanServer` service.

Any one of these disables inbound SMB. The hardened host in this case had two of the three applied (the firewall rule and the adapter binding). The Server service was still running, but with no adapter binding it had nothing to listen on, so port 445 was effectively closed even on loopback.

Recommendations

For the customer

1. Document the exception in the host’s hardening baseline. A subsequent hardening sweep applied through GPO, Intune, or a hardening tool will revert these settings unless the exception is encoded in the baseline.
2. Keep the firewall rule scoped to `127.0.0.1` only. This restores SMB exactly enough for same-host named-pipe IPC and does not expose port 445 to the network.
3. If an EDR product is present, verify it does not re-block SMB at the session layer. Some EDRs flag SMB sessions as lateral movement and reset them even when the OS allows the traffic.

Reference: Error Codes Encountered

Code	Hex	Name	Meaning
1722	0x6BA	RPC_S_SERVER_UNAVAILABLE	Top-level RPC error surfaced to the client
64	0x40	ERROR_NETNAME_DELETED	The specified network name is no longer available (no SMB listener)
59	0x3B	ERROR_UNEXP_NET_ERR	An unexpected network error occurred (SMB session/negotiation failure)
1706	0x6AA	RPC_S_NO_ENDPOINT_FOUND	rpcping endpoint string formatted incorrectly (no real failure)

Reference: rpcping Endpoint Syntax

When `-s` already specifies the server, the `-e` argument must be the bare endpoint name without server prefix:

Form	Result
`-s 127.0.0.1 -e \pipe tapvscp`	Correct (SMB path)
`-s . -e \pipe tapvscp`	Correct (local NPFS path)
`-s 127.0.0.1 -e \\.\pipe tapvscp`	Returns 1706
`-s 127.0.0.1 -e \\127.0.0.1\pipe tapvscp`	Returns 1706

INSTALLATION

Unable to remove docker network

If you encounter some message regarding your network:

network mdss_stack id xxx has active endpoints

while you tried to remove or update MDSS, please try to restart docker:

```
systemctl restart docker
```

Running behind a proxy

Application-level proxy (customer.env)

MDSS services read proxy settings from `/etc/mdss/customer.env` (see Custom configuration).

`NO_PROXY` must list every hostname the MDSS containers reach directly on the Docker network, so that service-to-service traffic is not sent through the proxy. The list below covers both deployment modes: the bundled platform container (`md-platform`, the default) and the individual-services deployment (`COMMON_SERVICES_ENABLED=no`). Entries for services that are not running in your deployment are harmless.

```
HTTP_PROXY="http://proxy_ip:proxy_port"
HTTPS_PROXY="https://proxy_ip:proxy_port"

NO_PROXY=localhost,127.0.0.1,apigateway,mdcs_apigateway,core,discoverys
ervice,identityservice,jobdispatcher,licensingservice,loggingservice,md
-
platform,mongodb,mongomigrations,nfsservice,node,notificationservice,pg
migrations,postgres,rabbitmq,redis,remediationsservice,scanningservice,
smbservice,storagesservice,webclient,workflowmanagerservice
```

```
HTTP_PROXY="http://username:password@proxy_ip:proxy_port"
HTTPS_PROXY="http://username:password@proxy_ip:proxy_port"

NO_PROXY=localhost,127.0.0.1,apigateway,mdcs_apigateway,core,discoverys
ervice,identityservice,jobdispatcher,licensingservice,loggingservice,md
-
platform,mongodb,mongomigrations,nfsservice,node,notificationservice,pg
migrations,postgres,rabbitmq,redis,remediationsservice,scanningservice,
smbservice,storagesservice,webclient,workflowmanagerservice
```

What the entries are:

Entry	Deployment	Purpose
`md-platform`	Bundled (default)	The single container that hosts Identity, Licensing, Job Dispatcher, Logging, Scanning, Notification, Discovery, Remediations, Storages and Workflow Manager
`discoveryservice`, `identityservice`, `jobdispatcher`, `licensingservice`, `loggingservice`, `notificationservice`, `remediationsservice`, `scanningservice`, `storagesservice`, `workflowmanagerservice`	Individual services (`COMMON_SERVICES_ENABLED=no`)	The same services, one container each
`apigateway`, `mdcs_apigateway`, `webclient`, `pgmigrations`, `mongomigrations`	Both	API Gateway, web client and the database migration jobs
`postgres`, `rabbitmq`, `redis`, `mongodb`	Both	Infrastructure. Replace with your own hostnames when using external PostgreSQL, RabbitMQ or Redis, if they are reachable without a proxy
`nfsservice`, `smbservice`	Both, when the NFS or SMB module is enabled	The NFS and SMB protocol sidecars. All other storage types run in-process and need no extra entry

`core`, `node`	Both, when MetaDefender Core is deployed alongside MDSS	The bundled MetaDefender Core instance

> **Warning**
>
> The configured scan instance address should also be added to the `NO_PROXY` list if connecting to it does not require a proxy.

Docker daemon proxy (pulling MDSS images)

If a proxy is required to access to docker hub in order to pull MDSS images, then this guide should be followed: https://docs.docker.com/network/proxy/

Create the drop-in config file or edit it if it already exists:

```
sudo mkdir -p /etc/systemd/system/docker.service.d
sudo nano /etc/systemd/system/docker.service.d/http-proxy.conf
```

Set the proxy settings:

```
[Service]
Environment="HTTP_PROXY=http://proxy_ip:proxy_port"
Environment="HTTPS_PROXY=http://proxy_ip:proxy_port"
Environment="NO_PROXY=localhost,127.0.0.1"
```

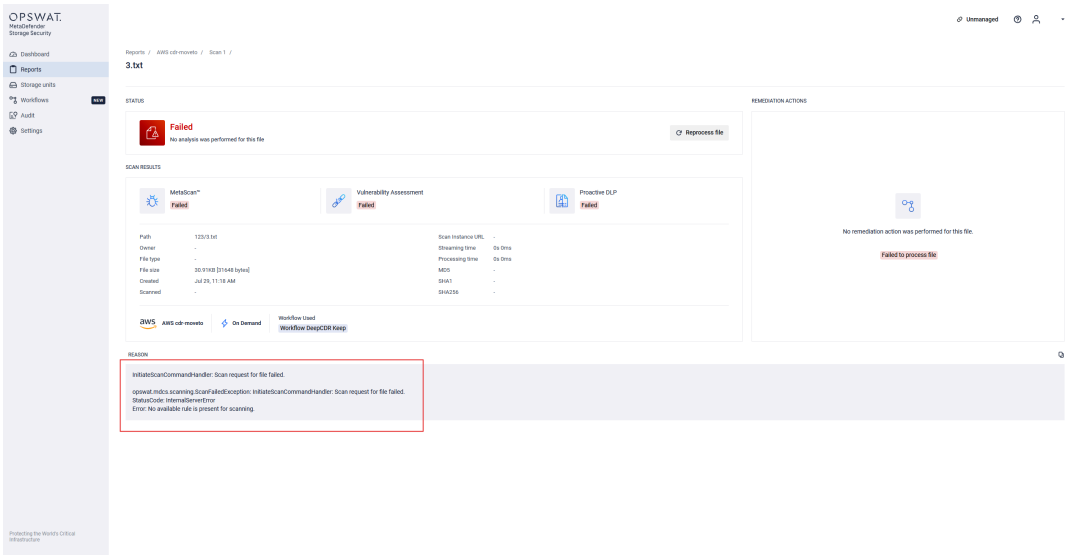
Apply the changes:

```
sudo systemctl daemon-reload
sudo systemctl restart docker
```

Why I am getting error "No available rule is present for scanning"

When triggering a scan, you may encounter a scan failure with the following error message:

> **Error**
>
> opswat.mdcs.scanning.ScanFailedException: InitiateScanCommandHandler: Scan request for object failed. StatusCode: InternalServerError Error: No available rule is present for scanning.



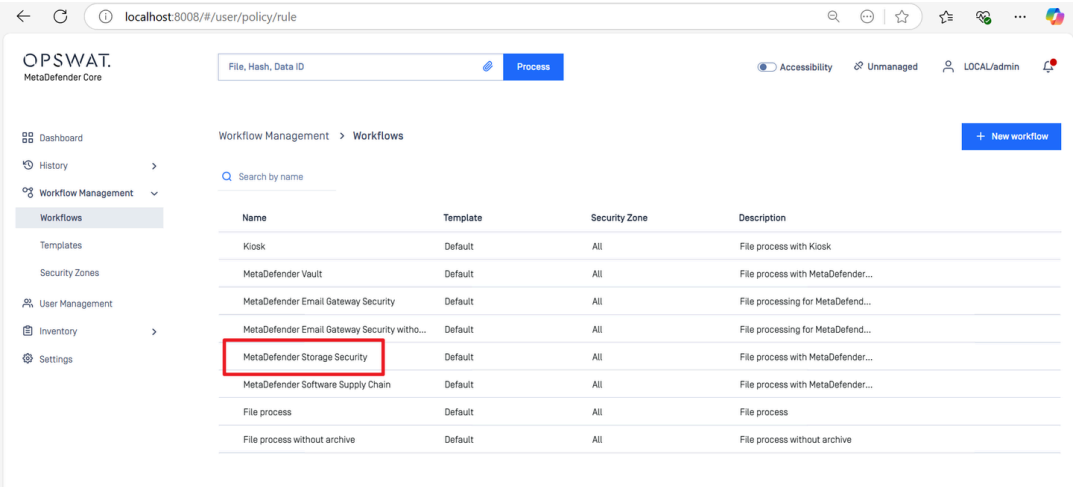
Cause

This error typically occurs when the workflow name configured in MetaDefender for Secure Storage (MDSS) does not match the workflow name set in MetaDefender Core.

MetaDefender Storage Security

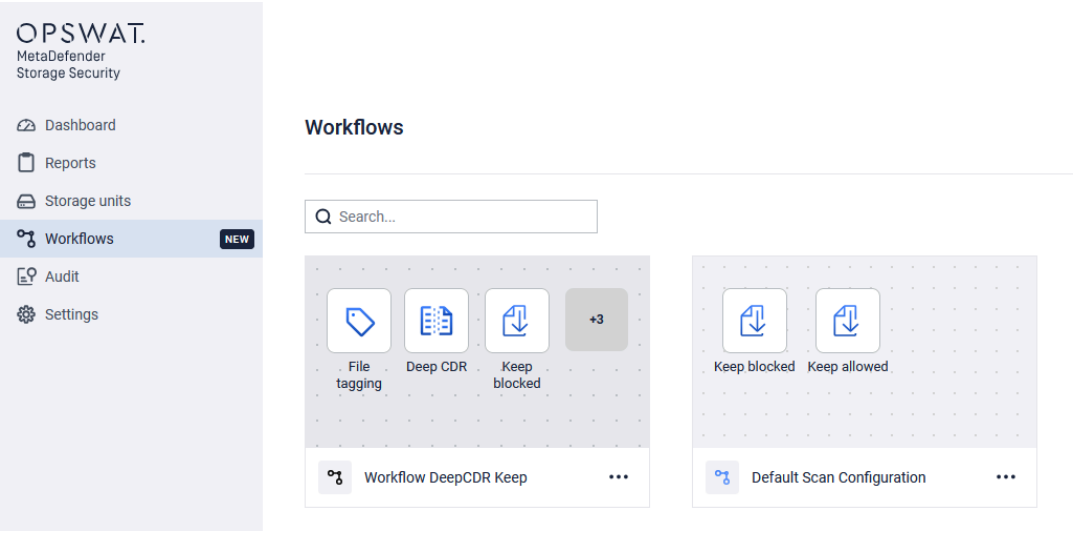


MetaDefender Core



Steps to Resolve:

1. **Log into MetaDefender Core**
   - Navigate to Workflows.
   - Identify the workflow associated with MDSS.
   - The workflow name should be either: “MetaDefender Storage Security” or “MetaDefender for Secure Storage”
2. **Verify and Update the Workflow in MDSS**
   - If the workflow name in Core is “MetaDefender Storage Security”, log into MDSS Configuration.
   - Go to **Workflows** and click on your workflow



- Edit the scan configuration and update the rule name to “MetaDefender Storage Security”.



- Ensure the workflow name exactly matches the one in Core.

3. **Apply Changes and Restart the Service**

- Click “Save” to save the configuration.
- Restart the scanning service if necessary and retry the scan to verify if the issue is resolved.

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a **support case or chatting with our support engineer**.

Why am I getting the 'Configured user limit on inotify instances has been reached' error in MDSS on Linux/Kubernetes?

Linux uses **inotify** to monitor file system events, but it enforces limits on how many files and directories a single user can watch. The default limits (often **128 instances** and **8,192 watches**) are too low for Kubernetes workloads. You will need to raise this value higher to run MDSS.

> **Please Note:**
>
> If using a **cloud-managed cluster** (e.g., GKE, EKS, AKS), nodes are preconfigured, and changes may not be necessary or possible. Ensure this setting is adjusted on **all Kubernetes nodes** if needed.

Sample Error Log:

```
[2025-03-27 08:02:01Z] [Fatal] : Unable to start Discovery service
|
 System.IO.IOException: The configured user limit (128) on the number
of inotify instances has been reached, or the per-process limit on the
number of open file descriptors has been reached.        |
   at System.IO.FileSystemWatcher.StartRaisingEvents()
|
    at
Microsoft.Extensions.FileProviders.Physical.PhysicalFilesWatcher.TryEna
bleFileSystemWatcher()     |
    at
Microsoft.Extensions.FileProviders.Physical.PhysicalFilesWatcher.Create
FileChangeToken(String filter)
|
    at
Microsoft.Extensions.Primitives.ChangeToken.ChangeTokenRegistration`1..
ctor(Func`1 changeTokenProducer, Action`1 changeTokenConsumer, TState
state)                                    |
    at Microsoft.Extensions.Primitives.ChangeToken.OnChange(Func`1
changeTokenProducer, Action changeTokenConsumer)
|
    at
Microsoft.Extensions.Configuration.FileConfigurationProvider..ctor(File
ConfigurationSource source)
|
    at
Microsoft.Extensions.Configuration.Json.JsonConfigurationSource.Build(I
ConfigurationBuilder builder)
|
    at Microsoft.Extensions.Configuration.ConfigurationBuilder.Build()
|
    at
Microsoft.Extensions.Hosting.HostBuilder.InitializeAppConfiguration()
|
    at Microsoft.Extensions.Hosting.HostBuilder.Build()
|
    at opswat.mdss.discovery.service.Program.Main(String[] args) in
/app/Backend/src/Discovery/opswat.mdss.discovery.service/Program.cs:lin
e 27
```

You can check your current inotify instances of your nodes by running these commands:

```
sysctl fs.inotify.max_user_instances
sysctl fs.inotify.max_user_watches
```

```
root@cxk8s-server-01:~# sysctl fs.inotify.max_user_instances
fs.inotify.max_user_instances = 128
root@cxk8s-server-01:~# sysctl fs.inotify.max_user_watches
fs.inotify.max_user_watches = 8192
```

Solution:

1. To immediately raise the limits without restarting, run:

```
sudo sysctl -w fs.inotify.max_user_instances=8192
sudo sysctl -w fs.inotify.max_user_watches=1048576
```

This takes effect immediately but will reset after a reboot. You can modify the values to be higher based on your needs.

2. To ensure these limits remain after a restart, add them to the system configuration file:

```
echo "fs.inotify.max_user_instances=8192" | sudo tee -a
/etc/sysctl.conf
echo "fs.inotify.max_user_watches=1048576" | sudo tee -a
/etc/sysctl.conf
```

3. Once updated, apply the new settings by running:

```
sudo sysctl -p
```

> **Support:**
>
> If **Further Assistance** is required, please proceed to log a support case or chat with our support engineer.

How to handle empty objects in Discovery?

To skip the discovery of empty objects, please add the following property in your `customer.env` file:

```
DISCOVERY_SERVICE_IGNORE_EMPTY_OBJECTS=1
```

By default, this value is set to 0, which means we will **not** ignore empty objects.

After the property is added in your `customer.env` file, please restart MetaDefender Storage Security for changes to take effect.

Permission denied while uninstalling on Linux

On some Linux distributions that have **SELinux (Security-Enhanced Linux)** enabled by default you may encounter a “Permission denied” message when uninstalling MDSS. The uninstall still completes successfully and only some docker containers might still be left running if `mdss -u clean` was not executed before the uninstall. This is due to SELinux

policies and can be avoided by disabling SELinux temporarily before uninstalling by running:

```
sudo setenforce 0
```

INTEGRATIONS

NFS storage unit is not detecting any files

Root Cause

The `nolock` flag was not included in the NFS mount command previously. While it is generally not required, certain environments, especially those using older NFS servers or specific NFS configurations, depend on this flag to ensure proper file access during MDSS scans. In such cases, omitting the `nolock` flag can lead to scanning failures.

Some older NFS servers lock files when opened by another application, which prevents MDSS from scanning them. To prevent this issue, the `nolock` flag is required in the mount command.

Solution

[Linux]

1. Execute the following command to mount the NFS storage: `sudo mount -o nolock {nfs_server_ip}:/srv/company-shared-files /mnt/mdss/nfs/company-files`

This command mounts the shared files from the NFS Server to the company-files directory on the NFS Client.

2. Ensure Persistent Mounting To maintain the mount across reboots, add the following entry to the `/etc/fstab` file: `{nfs_server_ip}:/srv/company-shared-files /mnt/mdss/nfs/company-files nfs rw,hard,intr,nolock 0 0`

Without this step, the mount will be lost when the NFS Client machine is restarted.

For more details, refer to this article: [Linux] How to add an NFS storage unit

[Windows]

For instructions on setting up NFS storage in Windows, refer to the following article: [Windows] How to add an NFS storage unit

Following the instruction mentioned in the article and in the terminal, run the following command:

```
mount -o "nolock,sec=sys" {nfs-server-ip}:{nfs-share-location} {mounted-drive}
```

For example, the NFS share will be mounted in the Y drive:

```
mount -o "nolock,sec=sys" 192.168.207.139:/srv/company-public-files/ Y:
```

> **Support:**
>
> If **Further Assistance** is required, please proceed to create a support case or chat with our support engineer.

Why my files are not moving on SharePoint On-Prem with MDSS, despite UI confirmation?

Issue

When moving files within SharePoint On-Prem, the UI indicates that the file has been successfully moved. However, upon verification, the file is not transferred to the target destination.

Root Cause

The issue occurs due to missing handling for maintaining folder structures during the file movement process.

Solution

Upgrade MetaDefender Storage Security to version **3.5.3 or later**, where proper handling for folder structure preservation in SharePoint On-Prem has been implemented.

> **Support:**
>
> If you encounter any issues during the upgrade process or require further assistance, please contact OPSWAT support via My OPSWAT at: Home - My OPSWAT
>
> Our support team will be able to provide you with the necessary guidance and resolution for any upgrade or compatibility concerns.

Scan stuck in the "processing" state on the SMB Service

> **Check Your Version:**
>
> This article applies only to releases before 3.5.2, all of which used the Python SMB service. The issue is fixed in 3.5.2 and later, including the legacy SMB service shipped with 4.6.0.

Symptom

The log appears to be stuck with the following entries:

```
Starting processing server. Listening on 127.0.0.1:6002
Starting health server. Listening on 127.0.0.1:7002
Using configured transport receive timeout of: 600
Starting worker
```

Under normal conditions, every SMB process keeps logging, each in its own log folder named after its port (6002 to 6006), instead of remaining stuck at this stage.

Possible Cause

SMB Service process stuck due to insufficient CPU resources The `psutil.open_files()` function hangs or freezes for several minutes, issue occurs specifically on Windows.

Solution

The issue was fixed starting with Version 3.5.2

> **Support:**
>
> If Further Assistance is required, please proceed to create a support case or chat with our support engineer.

Release Notes

Release notes

MetaDefender™ Storage Security 4.6.2

Release date: November 16, 2026

End of support: May 16, 2028

Improvements

- **PostgreSQL 17.12.** MDSS now ships PostgreSQL 17.12 on Linux and Windows, which fixes the vulnerabilities listed in the PostgreSQL November 12, 2026 security release. Upgrade to 4.6.2 to get it; Linux installations with internet access can also run `mdss -u update`.

MetaDefender™ Storage Security 4.6.1

Release date: October 14, 2026

End of support: April 14, 2028

> **Breaking change alert**
>
> From MDSS 4.6.0, the Lambda or cloud function must send the full file path as reported by the cloud provider. MDSS trims the configured folder location itself.
>
> **Impact:** Customers using event-based Real-Time Protection through a Lambda or cloud function on a cloud storage unit **added with a folder location**.
>
> **Additional details:** This keeps all cloud integrations consistent, and the function needs no code changes whether the unit has a folder location or not.
>
> **Mitigation Q:** Is your storage unit added with a folder location? **Yes:** When upgrading to 4.6.0, remove the code that strips the folder location from your cloud function. **No:** nothing to do.

Enhancements

- **Workflow file filters in the web UI.** The file type, size and last modified filters introduced in 4.6.0 can now be set on the workflow diagram, in a File filters node between File discovery and Scan pool. You can pick whole file type categories or single extensions, add extensions outside the catalog under Other extensions, set a size range and a last modified window, and see the filters a workflow uses in the Workflows

list, in the New job and Job details drawers, and as optional columns in Reports and Inventory.

- **Larger file type catalog and custom extensions.** The file type catalog lists 378 extensions, up from 111, taken from the MetaDefender Core supported file type lists for Deep CDR and Proactive DLP, and any other extension can be added by hand in the UI or through the API, with the default limit per workflow raised from 250 to 500. The filter compares the extension in the file name and never inspects the content, so a file without an extension always passes. A size or date range whose maximum is not above its minimum is refused on save and on settings import instead of being saved and skipping every file, and a workflow saved earlier with such a range is refused on its next update until the range is corrected.

Improvements

- **Arabic and Hebrew file names in reports.** File and folder names in Arabic, Persian or Hebrew display correctly in the Reports file list, in file details and in the PDF report. Each name reads in its own direction, the path runs from the root folder to the file, and the extension follows the file name. The PDF report, which drew Arabic letters reversed and unjoined, renders them as written, and names that mix Arabic or Hebrew with Latin words lay out the same way in the PDF as in the UI.

Bug Fixes

- **Rescans no longer stuck in Cancelled.** Rescanning a file from a completed or cancelled scan was accepted, but the file went straight back to Cancelled and never ran, and no further rescan could recover it. Rescans now run and end with a verdict, and cancelling a scan or a single file still stops the work queued for it.
- **NFS and SMB scans retry transient listing errors.** In 4.6.0, a folder listing that hit a short-lived error, such as a dropped connection or the NFS or SMB service not yet ready, was skipped without a retry, and a scan started while that service was unreachable ended within seconds with no files. Listings are retried again, and a folder that still fails after the retries is reported as a warning while the scan continues. The warning shows the underlying status, so a timeout can be told from a refused connection, and the SKIP_FOLDER_ON_MAX_TRANSIENT_ERROR_RETRIES setting is no longer read.

MetaDefender™ Storage Security 4.6.0

Release date: October 5, 2026

End of support: April 5, 2028

Important change alert

From MDSS 4.6.0, **DISCOVERY_SERVICE_{PROTOCOL}_RTP_HANDLING** (SMB, SFTP, NFS, FTP) is no longer used. Polling Real-Time Protection takes its start date from the scan's "Pick files modified since" setting instead.

Impact: Customers running polling Real-Time Protection on SMB, SFTP, NFS or FTP storage units with DISCOVERY_SERVICE_{PROTOCOL}_RTP_HANDLING=1 in customer.env.

Additional details: The 4.6.0 upgrade moves the "Pick files modified since" date of every running polling RTP scan on a protocol whose setting is 1 to 1 January 2000, so those scans keep covering existing files. For new scans, please follow this article: Real Time Processing configuration for NAS

Mitigation Q: Is DISCOVERY_SERVICE_{PROTOCOL}_RTP_HANDLING=1 set in your customer.env? **Yes:** Keep it there while upgrading to 4.6.0, then remove it. For RTP scans created after the upgrade, choose "Use the oldest available timeframe" in the scan wizard. **No:** nothing to do.

New Features

- **Workflow file filters.** A workflow can now skip files by type, size and last modified date before they reach a scan pool, saving scan capacity for the files you care about. In this release the filters are set through the REST API. Skipped files are listed in the report and the PDF export with the filter that excluded them, and each gets an audit entry.
- **Scan pool routing rules.** A scan pool can now mix MetaDefender Core and MetaDefender Cloud instances. In Settings, Scan Pools, Configure Routing you choose a balancing algorithm (least recently used or lowest outstanding), add ordered rules that send files to specific instances by type or size, and decide what happens to files no rule matches. Configuration export and import include the routing settings.
- **Automatic storage unit sync for the NetApp ONTAP Connector.** The OPSWAT ONTAP Connector reads the storage units under event-based real-time protection from MDSS and keeps its own configuration current, adding units when protection starts and removing them when it stops, with no manual edits and no service restart. Share path and server address changes made in MDSS are picked up as well.
- **Usage dashboard.** A new Usage tab on the Dashboard shows, for one storage unit, a timeline of past, running and upcoming scans and a heat map of files processed per hour over the selected date range, up to 90 days.

Enhancements

- **New SMB service on Linux.** Linux deployments use a new SMB service that mounts shares through the host's CIFS kernel module, so the host needs that module available. It supports custom SMB ports, reports wrong credentials and unreachable servers when you add the account, and reports a missing share path when you add the storage unit. Where the host cannot grant the service its mount privileges, you can switch back to the previous SMB service.
- **Target load per MetaDefender Core instance.** Each MetaDefender Core scan instance has a Desired load setting from 20 to 80 percent, 40 by default, editable in the UI and included in configuration export. Scanning is no longer capped by the message queue consumer limit, and when an instance set above 40 percent starts refusing work, the System tab recommends lowering it.
- **API key required for MetaDefender Core instances.** Adding or editing a MetaDefender Core scan instance now requires an API key. After an upgrade, workflows that use a scan pool with a keyless instance are flagged, cannot start new jobs, and keep a banner on screen until those instances have a key.
- **Clearer scan results in reports.** Report file tables now show the MDSS verdict, Allowed or Blocked, next to the result returned by the scan instance, so you can see why each file was allowed or blocked. The Remediations column shows the action that ran on each file.
- **Workflow details in Inventory.** Storage unit lists in Inventory can show the details of the workflow each unit's latest job ran with. Display options on Reports, Inventory and Accounts are grouped in sections you can select or clear at once, and the Reports page now opens with six columns by default instead of all of them.
- **External services on a dedicated host.** On Linux, `mdss -c start -e db,cache,queue` starts PostgreSQL, Redis and RabbitMQ on a separate host with generated credentials and prints the settings to paste into customer.env on the product host. Any subset of the three services can be selected.

Improvements

- **Faster, lighter storage integrations.** Most storage integrations now run inside the main services instead of in their own containers, so discovery, downloads and remediation skip a message broker round trip, and there are fewer containers to run and patch. NFS and SMB keep a small dedicated container.
- **Clearer System tab.** Informational recommendations no longer turn a healthy system amber, severity bands follow the reported severity, status lines are written in third person, and the tab marks its data as stale when health checks stop arriving. Product naming on the tab is corrected to MetaDefender Storage Security.

- **Syslog format notice.** The external logger settings state that only the logs sent to the external logger use the RFC 5424 format, while MDSS service and container logs keep their own format.
- **Platform component upgrades.** RabbitMQ is upgraded to 4.3.5 and Redis to 8.6.6, and on Windows, Garnet is upgraded to 1.1.10.

Bug Fixes

- **Upgrade from 4.2.0 no longer fails on credentials.** The account credential migration aborted the upgrade with a base64 decoding error when credentials were already in the current format, leaving the deployment unable to finish upgrading. The migration now completes, converts only credentials still in the old format, and can be repeated without error.
- **Windows upgrade no longer hangs for standard users.** Double-clicking the installer as a standard user on a machine with MDSS installed left the wizard on the preparing screen indefinitely. The installer now asks for administrator credentials right after launch and continues with the existing settings, including the configured ports.
- **Move and Copy remediation details kept after upgrade.** Upgrading from a version before 4.2.0 no longer resets the Move and Copy remediation details shown in job reports to Keep. Files were always moved and copied as configured.
- **NFS scans with inaccessible folders.** An on-demand scan of an NFS share containing a folder the service could not read completed with zero files. Inaccessible entries are now skipped and shown as a discovery warning in the report, and all accessible files are scanned.
- **Dropped uploads retried.** A file whose upload to a scan instance was interrupted no longer fails at once; the upload is retried, and a clear error is reported if the retry fails too.
- **Reprocessed files stay processed.** A failed file that was reprocessed successfully could be marked Failed again by the hourly stuck-file check, consuming its retry budget until reprocessing gave up. The check now measures from the retry, so a successful retry is left alone while a file that stalls is still caught.
- **Zero-byte files are remediated regardless of extension.** With Deep CDR enabled, an empty file whose extension was not .txt was recorded as a failed sanitization and its Move or Delete remediation was skipped, although MetaDefender Core had reported success. Empty files of every extension now follow the same path.
- **Storage errors no longer throttle scan instances.** Files that failed to read from storage no longer count as scan instance errors, and an instance that rejects its API key is taken out of the pool until the key works again.
- **Database disk on a separate filesystem is monitored.** When the Docker data root sat on its own volume, the database could fill it to 100 percent without processing being paused. MDSS now measures free space on the filesystem that holds the database,

pauses at the same thresholds as for the machine disk, and the System tab names the database disk as the cause. Where the database role cannot run that check, a start-up warning points to the GUARDRAILS_PG_DISK_SIZE_BYTES setting.

- **Memory badge on the System tab.** A machine with enough RAM was shown as below the minimum memory spec whenever the CPU count was below spec, because one flag covered both cards. CPU and memory are judged separately, and the memory value is labelled GiB.
- **Storage and account edits locked during scans.** The web UI no longer lets you change a storage unit's source or its account's credentials while a scan is running on it.
- **Scan cleanup and scheduled jobs survive restarts.** Deleting an on-demand scan during a service restart no longer leaves its file records behind, and one failed scheduled job no longer stops the others in its batch.
- **KMS setting saved on Amazon S3 accounts.** Enabling KMS client-side encryption on an Amazon S3 account was not shown as enabled after saving. The setting and its secret name are now saved and shown, and the Secrets Manager and KMS service URLs are optional.
- **Deep CDR status for files deflected by OPSWAT Alin AI.** A file that OPSWAT Alin AI Deflection cleared before sanitization showed Deep CDR as "Unsupported file type" and went through the CDR unsupported remediation. It now shows sanitization as skipped and follows the normal Allowed or Blocked remediation.
- **Configuration import without storage units.** Importing a configuration with no storage units during first-time setup now completes the setup instead of leaving the wizard stuck.
- **Message broker disk alarm.** During a RabbitMQ disk or memory alarm, queue-depth checks leaked channels until the connection could open no more, which blinded the autoscaler and flooded the logs until a restart. Each service now logs the blocked connection once, keeps its channel count stable, and resumes probing when the alarm clears.
- **Garnet CPU usage on Windows.** The Garnet cache service on Windows could get stuck at 100 percent CPU after running for some time. Its thread pool is now capped, and upgrades apply the new setting.

Changelog

MetaDefender Storage Security v4.5.2

Release Date: August 19, 2026

Version supported until February 19, 2028

Maintenance Release
- Deep CDR carries dedicated workflow branches for file types it does not sanitize, so you can keep, move, copy or delete them and clear the source folder in one pass. - Workflows validate before a scan runs, and Deep CDR requires an API key on every scan instance in the primary and failover pools. - Reports name the exact outcome per file, including unsupported file types, the sanitization detail returned by MD Core, and the remediation you configured. - Scanning applies a retry budget while MD Core is busy and holds its scaling steady while an instance is unavailable, so throughput stays predictable under load. - Stopping a real-time scan completes in seconds, the report refreshes on its own, and rescan sits in the report actions. - Schedules coordinate so a location runs one scan at a time, consecutive FTP scans run back to back, and real-time scans follow newly created child folders. - The portable package and the offline Docker toolkit deploy on a clean host, and upgrades carry databases holding non-Latin file names. - MDSS runs on the cache you already have, including AWS ElastiCache, and reads CPU and memory from pod limits on Kubernetes. - The NetApp ONTAP Connector ships at 1.0.44 with the full share path, so several SVMs connect to one MDSS. - OPSWAT SAML sign-on, email notifications and telemetry collectors follow your configuration, and inventory tables let you pick the columns you need. **Maintenance Release (v4.5.1)** Release Date: July 30, 2026 Version supported until January 30, 2028 - Real-time protection jobs each apply their own workflow, so multiple policies can run against a single storage unit. - Disk usage in the warning band keeps scanning running, scan health reflects real load, and autoscaling ships enabled on all deployment types with per-instance sizing recommendations. - The NetApp ONTAP Connector delivers files over HTTPS, and you can configure the response it returns to NetApp when a file scan times out. - Real-time protection keeps its status and discovery cycle detail, and partition-based FTP scans resume without reconfiguration. - Stopped jobs report accurately, their files can be rescanned in place, FTP scans run to completion, and event-based scans handle partitioned storage units. - Remediation actions deliver to a second destination, tags follow files moved into folder-based units, and Central Management copy workflows run as configured. - PDF exports honor the filters you applied accurately per storage unit. - The dashboard loads faster against a 10 million file scan with a searchable unit selector, and job schedules support valid month days and 52-week cycles. - Single sign-on from the virtual appliance keeps its identity provider, notification targets are validated, and S3 accounts show the correct region and endpoint fields
Highlights (v4.5.0) Release Date: July 22, 2026 Version supported until January 22, 2028 MetaDefender Storage Security 4.5.0 introduces a redesigned user interface, centralized management through OPSWAT Central Management (OCM) v10, a new autoscaling engine, significant security enhancements, and multiple platform and storage integration improvements.
New Features (v4.5.0) Redesigned User Interface The application has been rebuilt using the Blue Line design system, delivering a modern user experience with: - Updated Dashboard, Reports, Inventory, Jobs, Workflows, Settings, and Users pages - Improved reporting with dedicated Threats, Sanitization, and Data Loss Prevention views - Guided onboarding for first-time deployments - Dark theme support - Backward-compatible legacy URL redirects **OPSWAT Central Management (OCM) v10 Integration** MetaDefender Storage Security now supports centralized management through OCM v10, including: - Instance enrollment and license management - Remote configuration - Deployment ID synchronization across managed environments **Autoscaling Engine** A new autoscaling engine dynamically adjusts system resources based on workload. Key capabilities include: - CPU, memory, and file throughput monitoring - Scaling recommendations - `/healthz` endpoint for health monitoring - Dashboard visibility into scaling status Supported deployment models include MetaDefender Core, Cloud and Distributed Cluster. **New SMB Service for Windows deployments** The

Windows SMB Service is a new, fully in-house SMB client implementation for Windows deployments. Built with no third-party dependencies, it delivers improved reliability and resolves the connection issues found in earlier versions.
Security Improvements (v4.5.0) This release includes multiple security enhancements, including: - Fixes for OData injection vulnerabilities - License forgery protection - Penetration testing remediations, including JWT secret removal, improved SSO security, and sensitive data protection - Non-root container enforcement for Kubernetes deployments - Improved Windows file permission security
Platform Improvements (v4.5.0) Platform updates include: - Updated infrastructure components (RabbitMQ, Garnet, Alpine Linux) - Improved job dispatcher with automatic recovery for orphaned scan jobs - Configurable scan result retention policies - Database growth monitoring and protection - Decoupled scanning service architecture for cross-product reuse - Prevention of overlapping scans within storage groups - Optimized scanning and results storage for archives
Storage and Integration Improvements (v4.5.0) Storage connectors have been enhanced with improved reliability and functionality, including: - Better error reporting for NFS and SFTP - Improved SharePoint Cloud real-time protection - Microsoft Graph connection throttling for OneDrive and SharePoint - NetApp ONTAP enhancements - Autodiscovery support for all S3-compatible storage providers - Updated Diode integration documentation
Telemetry (v4.5.0) Telemetry has been renamed **Usage Report** and now includes: - Kubernetes containerized collector - Improved usage reporting accuracy
Resolved Issues (v4.5.0) This release resolves several issues, including: - Scheduled scans starting at incorrect times due to timezone offsets - Recovery of cancelled or stuck scan jobs after service restart - Dashboard threat counts matching report data - Improved stability when scanning large file volumes - RabbitMQ startup synchronization issues - Improved error handling across storage integrations
Known limitation: Remediation counters on reports for scans performed before version 4.0.0 (v4.5.0) After upgrading from a version earlier than 4.0.0, reports for scans that completed on the older version may display inconsistent remediation counters: - The **Keep** counter may show 0 even when files were kept. - The **Remediated** counter excludes kept files, so it may show 0 when keeping files was the only configured remediation action. All other values on these reports remain accurate, including the allowed and blocked file totals and the **Move**, **Copy**, **Delete**, and **Tagging** counters. This limitation applies only to scans completed before the upgrade. Scans performed after the upgrade display all counters correctly.

MetaDefender Storage Security v4.4.2

Release Date: May 14, 2026

Version supported until November 14, 2027

Maintenance Release
- **SFTP scan reliability.** Fixed an issue where SFTP scans could fail with repeated connection errors against servers that limit concurrent sessions. - **Windows upgrade from v3.6.1 and v4.4.0.** Fixed a crash during the MongoDB migration step that blocked Windows upgrades to the latest version. - **Offline toolkit package.** Fixed the offline installer so air-gapped deployments include all required components and the platform service starts as expected. **Maintenance Release (v4.4.1)** - **NGINX security update.** Updated NGINX to 1.31.0, which patches CVE-2026-42945. Previous MDSS versions are not affected by this CVE; the update ships as a precaution. - **RTP validation for circular remediations.** Real-time scans now reject workflows whose move or copy remediation targets the same storage unit being scanned, preventing the retry loop seen previously. - **Additional S3 region.** Added the New Zealand (ap-southeast-6) region to the AWS S3 and S3-compatible storage selectors. - **Reduced error log noise.** Cleaned up spurious error log entries from background job cleanups and certain file-download failure paths.
Highlights - **New Storage Integration: Scality RING -** Added native integration for Scality RING S3-compatible object storage, with full coverage for multiscanning, CDR, DLP, file tagging, and remediation actions. - **Azure Blob storage autodiscovery.** Azure Blob accounts can now automatically discover and onboard new containers as they are created. Container selection can be fully automatic or scoped to a manual subset, and the feature can be toggled per account. - **Access Control (ABAC Phase I).** New Settings → Access Control tab showing default roles and their permissions. - **Real-Time Processing start date.** Added an optional "Pick files modified since" date picker to the RTP scan modal, so older files (e.g., OneDrive uploads that retain their original `LastModified`) can be included in the scan. - **Scan size visibility.** Individual scan reports and active scans now show `Total Files Size` and `Processed Files Size`, and `Total Files Count` is back in the report view. - **Users included in config export/import.** Configuration export can now optionally include MDSS users (local and SSO), with matching users reconciled on import.
Improvements - **Scan reliability and gRPC performance.** Reused gRPC channels across NFS/SFTP discovery, object operations, and telemetry to prevent port exhaustion; added watchdog timeouts and retries to prevent stuck scans and stuck remediations; fixed timing edge cases that could mark instant scans complete prematurely; and added parallel consumers on non-priority RPC queues for better authentication throughput. - **Logging and diagnostics.** Added a separate error/warning log file with its own rotation, persisted PostgreSQL/Redis/RabbitMQ logs across container restarts, identified the specific MD Core instance in API key failure logs, and clarified the support package export error around `LOGS_PER_SERVICE`. - **Cancel on-demand scans from Reports.** On-demand scans can now be cancelled directly from the Reports tab, matching the existing RTP cancel behavior. - **Persistent service scaling.** Extra service instances can now be configured in `customer.env` and survive upgrades. - **Discovery throttling for Discovery-Only scans.** Added configurable queue-depth-aware throttling so Discovery-Only workflows back off when downstream queues are saturated and resume once they drain. - **Scan reliability guardrails.** Scans no longer get stuck on slow storage or remediation, and instant scans now wait for discovery to actually finish before reporting completion. - **Workflow display in scan details and reports.** Active scan details now show only the currently running workflow, while reports keep the full workflow history with a clearer "current vs. previously used" layout. - **Adaptive Analysis status accuracy.** Adaptive Analysis and Sandbox now show "Skipped" (not "Failed") when MD Core skipped the file, and clean verdicts display a green badge. - **Adding storage no longer auto-starts a full RTP scan.** New storage units are created without an automatic RTP scan, so the Reports view stays clean until you explicitly start one. - **Delete**

RTP scans from Reports. RTP scans (including partition scans) can now be deleted from the Reports page, not just cleaned, with support for bulk and mixed-type deletion. - **Timestamps and cleaner messages on RTP errors.** Expanded error details now show when each error occurred and present a friendlier message without raw stack traces. - **Upgrade reliability.** Fixed Windows upgrades silently overwriting `customer.env`, and restored RTP partition scans that stopped working after upgrading to 4.3.0. - **Stability at scale.** Reduced Garnet memory usage on Windows and fixed a service crash triggered by a large number of scans in the database. - **Integration fixes.** Resolved a SharePoint Online issue where slow Graph subscription creation prevented RTP from starting, and blocked the same GCP bucket from being added twice through different credential types (ADC vs. keys). - **Dependency upgrades.** Upgraded RabbitMQ to 4.2.4, Redis to 8.6.0, and Garnet to 1.0.99.

Info - Upgrades from v4.3.0 or earlier

RTP behavior during upgrade.*This applies when upgrading from any version older than v4.3.1. Customers already on v4.3.1 or later are not affected.*

Active RTP scans are paused during the upgrade and resume automatically once it completes. Files added or modified during the upgrade window can be picked up by running an RTP report cleanup afterward with RTP Handling with Comprehensive Detection enabled. On NAS storage units this cleanup reprocesses all files and may take some time depending on volume.

MetaDefender Storage Security v4.3.2

Release Date: April 03, 2026

Version supported until: October 03, 2027

Highlights

Workflow Cards: Storage Unit Destination

- Workflow cards now show which storage units are set as destinations for remediation actions, so you can quickly see where your data ends up.
- **How to enable:** Navigate to Workflows and select Extended View from the top right.

Improvements

- Certificates are now picked up automatically after a machine restart, without requiring a product restart separately.
- Fixed a race condition during Podman restarts that previously caused orphaned containers running in the background.

MetaDefender Storage Security v4.3.1

Release Date: March 06, 2026

Version supported until: September 06, 2027

Improvements

- Resolved issue that malformed scanning paths for active RTP scans and Scheduled Scans, that target partitions, on FTP, NFS, SFTP, SMB, and SharePoint On-Premise storage units
- Resolved issue that forced RTP scans on NAS storage units to reprocess files processed before upgrade

MetaDefender Storage Security v4.3.0

Release Date: March 05, 2026

Version supported until: September 05, 2027

This release introduces Microsoft Teams support for enhanced collaboration security, auto-discovery for SharePoint Online to streamline setup, a new reprocess failed files feature for easier remediation, stability improvements, and concurrent scanning scenarios.

Highlights

- **Multitenancy (MDSS Cloud)**
  - Support for multiple tenants within a single MDSS Cloud deployment
- **OpenTelemetry (Initial Release)**
  - Collect and export telemetry data for monitoring and observability
- **Microsoft Teams Scanning**
  - Scan and sanitize file attachments shared in Microsoft 365 Teams channels and chats
- **SharePoint Online Auto-Discovery**
  - Automatically discover and connect SharePoint Online sites—no manual configuration required
- **Reprocess Failed Files**
  - Create new scans targeting only previously failed files, with the flexibility to apply any workflow
- **Flexible Scan Scheduling**
  - Schedule scans down to the minute for precise timing control
- **Real-Time Protection Limits (Cloud)**
  - Configure scan limits for MDSS Cloud deployments to manage resource usage

Improvements

- **Scan Priority UX**
  - Added tooltips and reordered priority options for clearer configuration
- **Storage Unit Display**
  - Storage unit name now visible in scan configuration for easier identification
- **Sanitization Workflow Control**
  - Original files deleted only after successful sanitization. New option to configure delete behavior separately for allowed and blocked files—enabling workflows like Deep CDR for blocked files only while leaving clean files untouched
- **Connection Reliability**
  - Automatic retry on temporary connection failures
- **Performance**
  - Loading indicator on download button when generating large scan reports
- **Report Export UX**
  - Faster remediation processing for large-scale deployments
- **SMTP Configuration**
  - Input validation for host field with helpful error messages for invalid entries
- **SharePoint Storage Units**
  - Cloud and On-Prem storage units now use Site URL and Document Library URL fields. Existing configurations remain compatible, with a post-upgrade prompt to update
- **Bug Fixes & Documentation**
  - Fixed authentication failures when running multiple SMB scans simultaneously
  - Resolved partition scan errors across all users
  - Reports now correctly deleted only after retention period expires and only for completed scans
  - Notification imports now save valid settings even when some notification types fail validation
  - Updated documentation to reflect recent changes that include UI, windows installs, API keys, and scan reprocess actions.

MetaDefender Storage Security v4.2.0

Release Date: December 15, 2025

Version supported until: May 25, 2026

MetaDefender Storage Security 4.2.0 delivers significant enhancements to cloud integration, identity scanning capabilities, and workflow automation, alongside critical stability improvements and bug fixes.

Highlights

- **GCP Autodiscovery**
  - Automated bucket discovery for Google Cloud Platform
- **Oracle Cloud Enhancements**
  - Improved support for Oracle Native while keeping its S3-compatible storage configurations
- **Cross Domain**
  - Compliant data transfer across classification domains for high-security environments of workloads with isolated toxic handling, RabbitMQ-orchestrated processing, and webhook-based discovery
- **Identity Detection**
  - Identity detection across backup filesystems with PDF reporting and remediation integration. Supports on-demand and scheduled scans with intelligent incremental scanning. Real-time processing (RTP) not supported.
- **PDF Report Enhancements**
  - Improved file distribution visualization, enhanced tooltips, and identity scan results inclusion
- **PostgreSQL Optimization**
  - Resolved stability and performance issues for large-scale deployments

Improvements

- Discovery workflow execution and event-based scanning capabilities
- Advanced file versioning support and remediation workflow performance
- Optimized FTP connection pooling and file discovery mechanisms
- Enhanced MDSS Cloud rate limiting and tenant administrator access controls
- Improved SharePoint OnPrem workflow persistence
- Robust storage account validation and credential management
- Advanced password security in account management interfaces
- Refined form validation for account creation and RabbitMQ configuration
- Improved UI responsiveness and rendering performance across viewing contexts
- Streamlined onboarding experience and intelligent default behaviors
- Enhanced file distribution visualization and reporting accuracy
- Comprehensive RabbitMQ validation, audit logging, and notification delivery

- Refined SMTP lifecycle management
- Improved webhook configuration interface and user guidance
- Optimized database operation efficiency during export operations

MetaDefender Storage Security v4.1.0

Release Date: September 23, 2025

Version supported until March 23, 2027

Highlights

- **Sandbox Analysis Integration**
  - Full sandbox analysis capabilities with MITRE technique mapping and threat intelligence dashboard integrated with MD Cloud and MD Core deployments for advanced threat detection.
- **Storage Management Redesign**
  - New dual view modes (Groups and Accounts) for organizing storage units with comprehensive account-level operations including editing, deletion, and bulk scanning capabilities.
- **HTTPS Configuration**
  - Enable HTTPS directly through the Security settings interface with real-time certificate validation and clear error messages, eliminating lengthy timeout periods.
- **Enhanced Real-Time Protection**
  - Configure multiple RTP scans for different partitions within the same storage unit for all scanning configurations per partition.
- **SharePoint Historical Versions**
  - Scan all historical versions of SharePoint files, not just the latest version, ensuring comprehensive security coverage across file version history.
- **Multiple Workflow Destinations**
  - Configure up to 5 copy/move destinations in workflow remediations

Improvements

- **User Interface**
  - Form consistency across SMB/SFTP and FTP configurations with interactive guided tour for new users
  - Enhanced inline notifications with auto-scroll functionality and completely rebuilt audit system with improved filtering

- **Integration & Connectivity**
  - FTP storage protocol support
  - Enhanced SFTP ports beyond port 22
  - Individual file webhooks for real-time integration and secure syslog transmission with TLS support
- **Security & Platform**
  - Streamlined licensing without legacy SKUs
  - Workflow processing without multiscanning requirements
- **Reporting & Analytics**
  - Enhanced PDF reports with new file statistics and detailed remediation information
  - Proper handling of files with successful scans but failed remediations
- **Bug Fixes & Documentation**
  - Resolved audit expand functionality, filtering issues, and file path handling improvements
  - Comprehensive API documentation updates and myOPSWAT setup instructions

This release focuses on advanced threat detection capabilities and storage management flexibility while enhancing security and user experience across the platform.

MetaDefender Storage Security v4.0.1 [Hotfix]

Release Date: August 25, 2025

Version supported until February 25, 2027

- SMTP Re-activation Issue
  - Fixed a critical functionality bug where SMTP service could not be re-activated after being deactivated, rendering email notifications permanently unusable until system intervention.
- Database Migration Report Recovery
  - Resolved issues where scan reports became inaccessible after upgrading from previous versions (3.5.1/3.6.1), particularly affecting canceled scans and PDF export functionality during database migration processes.
- **Multi-tenancy Filter Disabled for MDSS Cloud**
  - Temporarily disabled PostgreSQL tenant ID filtering for cloud deployments as the multitenancy feature is not yet complete, preventing authentication and permission issues for Tenant Admin users until proper tenant configuration is implemented.

MetaDefender Storage Security v4.0.0

Release Date: August 12, 2025

Version supported until February 12, 2027

Highlights

- Migrated from MongoDB to **PostgreSQL 17** to align with MetaDefender Core and ICAP Server. This simplifies deployments and improves performance for complex queries and large datasets.
- The new **Workflows** feature creates an automated security processing pipeline:
  - 5-stage security processing: Scan Configuration → Advanced Threats → Vulnerabilities → Sanitization → File Tagging
  - Automatically sorts files into Original, Allowed, Blocked, and Sanitized categories
  - Executes automated actions (Copy, Move, Delete) based on scan results
  - Process thousands of files without manual intervention
- **Security Updates**
  - New account creation and login flows
  - Enhanced encryption key handling and secure key rotation
  - Security patches for known vulnerabilities

Improvements

- **Performance**
  - Upgraded RabbitMQ to version 4.1
  - Enhanced SharePoint Cloud real-time processing
  - Custom date range functionality in Dashboard
- **Integration**
  - Upgraded Microsoft Graph SDK to version 5.79.0
  - Added automatic file tagging for Google Cloud Platform
  - Extended AWS regional support
- **Platform**
  - Updated PostgreSQL documentation
  - Improved Azure AD configuration guide
  - Better error handling

This release focuses on database modernization and workflow automation while maintaining platform stability.

MetaDefender Storage Security v3.6.1

Release Date: May 20, 2025

Version supported until November 20, 2026

Highlights

- Streamlined Autodiscovery of S3 Buckets so users can now easily add S3 storage by providing AWS credentials. MetaDefender Storage Security can now automatically discover available buckets, allowing you to enable continuous discovery or manually select buckets for integration via a simplified interface.
- Advanced Remediation Options
  - Configure 'Empty Folder Cleanup' to preserve specific subfolders during parent folder deletion, preventing critical data loss.
- Expanded SharePoint Support
  - Integrate with SharePoint Online Government Cloud (GCC) using a simple configuration checkbox for correct authentication.
  - Discover, scan, and manage content within SharePoint On-Premises subsites, providing comprehensive coverage for hierarchical structures.
- Granular File Rescan so users can rescan individual files directly from scan reports without needing to rescan the entire storage unit or move files, significantly improving efficiency for addressing specific issues.

Improvements

- My OPSWAT Central Management Integration: Access My OPSWAT Central Management settings (SMTP, Notifications, License, Data Retention, Syslog) more easily within MetaDefender Storage Security and utilize integrated read-only user roles, including permissions for managing user invitations.
- Improved Error Handling & Feedback: Enhanced API error handling provides more specific, actionable messages (distinguishing connection, permission, config issues) in a standardized format, improving troubleshooting and UI feedback. Added a clearer error message when attempting to add storage without existing scan configurations.
- UI & UX Refinements
  - The MetaDefender Storage Security Cloud UI now hides unavailable integration options for better clarity.
  - Dashboard charts have an improved "no data" state visualization.
  - User Management features backend-powered search with input validation and minor bug fixes.
  - File remediation details are displayed more comprehensively in the UI.

  - The “Manage Storage Unit” interface now allows separate editing of storage settings and underlying account credentials.

- API & Backend Improvements
  - Real-Time Processing (RTP) scan reports now have a default, non-null name for consistent API identification.
  - API key security is enhanced: copying is restricted to the moment of generation, and GET requests for keys are disallowed.
  - The ‘Scan Folder’ API now prevents conflicts with ‘Delete Empty Folders’ remediation by returning a `400 Bad Request`.
- Deployment & Scalability
  - The MongoDB database name used by MetaDefender Storage Security is now configurable, supporting external MongoDB instances with predefined names.
  - Improved reliability for scanning large SharePoint On-Premises deployments by managing request rates and using retry mechanisms.
  - Enhanced internal component version identification for cross-platform consistency.
- More Reliable Windows Deployments by reducing failures and simplified setup through preemptive port validation and improved installation configuration.
- Stronger Security & Data Accuracy by ensuring secured MetaDefender Core connectivity (HTTPS) and corrected S3 file tagging for dependable security verdicts.
- Faster Windows Performance & Simpler Stack by improving speed and streamlined dependencies by migrating Windows caching to Garnet.

MetaDefender Storage Security v3.6.0

Release Date: March 31, 2025

Version supported until September 30, 2026

Highlights

- MDSS has undergone a major overhaul, featuring a redesigned UI and achieving feature parity with previous versions.
- Users now control scanning priority (performance or thoroughness) for storage unit scans, allowing for tailored resource allocation.
- A real-time processing overview provides immediate monitoring of storage scan activity and a detailed On-Demand Scan Report.
- Users can now cancel the scanning of individual files, optimizing resource management and preventing unnecessary processing.

- Introduced dedicated permissions for Tenant Admins in MetaDefender Storage Security Cloud, enabling granular tenant management without full admin access.
- Users can now specify a OneDrive group, user, and folder location for scanning. This allows for granular security policies and targeted scanning within OneDrive groups. Remediation actions are updated to accommodate group and user specifications.
- The configuration options that are irrelevant (My OPSWAT Central Management integration) are hidden in MetaDefender Storage Security Cloud deployments, simplifying the UI.
- A "Destination Storage for Remediations" section is added to the dashboard, centralizing remediation action visibility.
- Implemented a configurable debouncing mechanism for RTP scans of productivity files (.docx, .xlsx, .pptx) to prevent exceeding MetaDefender Cloud API limits during high-volume live document editing. This applies to the SharePoint Online Integration and provides configurable debounce settings (interval) and information on MDCloud API request limits.
- Storage Units are now editable in My OPSWAT Central Management. My OPSWAT Central Management configuration is simplified with a dedicated modal and active/inactive/loading states, and user management now features sorting and ordering capabilities.
- Remediation data is included in scan details, providing a comprehensive remediation overview.
- The /file API endpoint is updated to retrieve file lists with filtering options. File remediation history is improved by including remediation information in API responses.
- An optional "Delete empty folders after remediation" setting is added for SMB/NFS/SFTP shares, with audit logging of deleted folders.
- Password security is enhanced to Level 1 compliance, the old password requirement is enforced, existing users migrate on change, and strong authentication is maintained.
- Expanded syslog messages now include Data ID, threat details, and comprehensive scan results for improved SIEM integration.

Improvements

- The shutdown process for all services is improved for a clean and controlled shutdown.
- Authorization issues with Box Webhooks are resolved with a dedicated endpoint.
- A non-dismissible warning/error message is displayed if the database version is less than MongoDB 6.
- The `privateprocessing` header is added to MetaDefender Cloud scan requests by default, ensuring immediate file deletion from MetaDefender Cloud servers after scanning.

MetaDefender Storage Security v3.5.3

Release Date: 23 January 2025

Version supported until 23 July 2026

This release of MetaDefender Storage Security focuses on enhanced functionality, improved performance, and bug fixes to provide a more robust and efficient security solution for your storage environments.

Highlights

- Manage storage units (read-only) remotely via OPSWAT Cloud Management (OCM) with vPack integration.
- Preserve AWS checksum metadata during file transfers between storage locations, including S3-compatible integrations.
- Added support for Hitachi S3, including object deletion without version ID.

Improvements

- Significantly reduced scan times for small files on S3.
- Resolved issues impacting S3 scanning, audit log accuracy, and OCM integration.
- Refined the storage unit management interface in OCM for better usability.

MetaDefender Storage Security v3.5.2

Release Date: 18 December 2024

Version supported until 18 June 2026

Highlights

- MDSS can now connect with ADFS servers that utilize larger OpenID tokens (over 4KB).
- Windows deployments can now leverage SharePoint OnPrem for storage and remediation actions.
- Move existing storage units between groups with the updated Storage Units API.
- Integrate MDSS with Kafka to streamline external logging, including audit events.

Improvements

- The offline package now includes Docker binaries and a dedicated package for seamless offline MongoDB upgrades.
- JSON formatting for console logs allows easier integration with popular log management tools like Corallogix.

- MDSS Cloud now efficiently retrieves sanitized files from MetaDefender Cloud scan pools.
- Database Retention settings documentation has been updated to provide comprehensive information on data segregation, isolation, retention policies, and file handling in S3 and EC2 environments.
- Windows Upgrade Enhancements:
  - MDSS automatically clears RabbitMQ queues upon restart after upgrading from version 3.4.2.
  - The upgrade process now seamlessly handles MongoDB database upgrades from version 6 to 7.

MetaDefender Storage Security v3.5.1

Release Date: 19 November 2024

Version supported until 19 May 2026

Highlights

- We've boosted security for cloud deployments with support for AWS PrivateLink, allowing users to connect to MetaDefender Storage Security securely within their Amazon Virtual Private Cloud (VPC)
- Significantly improved performance for remediation tasks, enabling quicker responses to security threats.
- Single Sign-On (SSO) via SAML 2.0 - Business network administrators can now easily integrate with their existing identity providers using SAML 2.0 protocol. This allows for streamlined user authentication and authorization, enhancing security and simplifying access management.

Improvements

- We've addressed stability issues affecting SMB on Windows environments
- Users can now use custom URLs for S3 integration, enabling secure access to S3 buckets via AWS PrivateLink.
- We've improved the OPSWAT Cloud Management (OCM) synchronization job to reduce unnecessary configuration downloads and ensure the latest version of the vPack package is always available.
- To improve performance and scalability, we've added data retention settings for real-time processing for large datasets.
- We've added PIN functionality to OPSWAT Cloud Management, ensuring only authorized users can access and modify your configurations.

MetaDefender Storage Security v3.5.0

Release Date: 23 October 2024

Version supported until 23 April 2026

Highlights

New Integration: Native NetApp Storage

- New Integration: Native NetApp Storage using the ONTAP connector.
- New Integration: OPSWAT MetaDefender Managed File Transfer (MFT)
- New Integration: Cubbit Storage
- Support for PingFederate as an SSO (single sign-on) provider via OpenID Connect
- Scan management now includes balanced priority queues for real-time processing and instant scans

Improvements

- Improved API documentation for some API endpoints.
- Updated the Serilog package to version 4.0.1 to enhance integration capabilities and security.
- Documentation is now available describing the usage of MetaDefender Core and MetaDefender Cloud APIs, including required roles and permissions for correct access.
- Upgraded multiple node packages to enhance security and reliability.

MetaDefender Storage Security API

MetaDefender Storage Security API v1

> Scroll down for example requests and responses.

This section of the user guide describes how you can programmatically interact with the MetaDefender Storage Security REST API. Below are some common tasks that can be done using the available REST APIs:

- Authenticate to obtain a JSON Web Token(JWT)
- Start or stop a process(scan)
- Add / remove storage units

About this REST API

The exposed endpoint is located by default at **http(s)://md-storage-server/api/** (for example, the authentication endpoint is available at **http(s)://md-storage-server/api/user/authenticate**). All requests are handled by the NGINX web server before being proxied to the backend API Gateway service.

All endpoints perform authentication and authorization checks. For these checks to succeed, a valid token should be presented in the Authorization header in the form of Bearer.

Please note that all issued tokens have a timestamp and signature associated in order to prevent long-term usage without re - authentication.The lifespan of the token is currently set to 60 minutes, meaning you will have to request a new token before it expires in order to avoid error responses.

Useful links

- https://swagger.io/tools/swagger-ui/

Base URLs:

- http://<MDSS_HOST>

Authentication

- HTTP Authentication, scheme: bearer As mentioned earlier, all endpoints perform authentication and authorization checks. In order for these checks to succeed, a valid token should be presented in the Authorization header in the form of Bearer .

Suggested application logic

If you plan to integrate MetaDefender Storage Security in your custom application or workflow, please consider the following scenarios for successfully making REST API requests:

Scenario	Possible use cases
Short-lived integration	You are building or enhancing an application that requires sporadic or on-demand access to MetaDefender Storage Security REST APIs.
	The application is not expected to make more than a few REST API calls per hour.
	The application does not need to preserve a session.
Long-lived integration	You are building or enhancing an application that requires continuous, uninterrupted, or hard to predict access to MetaDefender Storage Security REST APIs.
	Requests are being triggered based on external factors and your application should maintain connectivity with MetaDefender Storage Security REST API.
	Session preserving is necessary and authentication should happen without user-interaction.
	Your application will make a significant number of REST API requests and you need increased performance
API Key integration	You are building or enhancing an application that requires continuous, uninterrupted, or hard to predict access to MetaDefender Storage Security REST APIs.
	Requests are being triggered based on external factors and your application should maintain connectivity with MetaDefender Storage Security REST API.
	Authentication should happen without user-interaction.
	Your application will make a significant number of REST API requests and you need increased performance.

Short-lived integration

- Obtain a signed accessToken by calling /api/user/authenticate API
- Use this token to call your desired REST API by providing it in the Authorization header
- Expire the token by calling /api/user/logout
- Repeat steps 1-3 the next time your application needs to call a REST API

Long-lived integration

- Obtain a signed token by calling /api/user/authenticate API
- Securely save the received accessToken and the refreshToken
- Use the accessToken to call your desired REST API by providing it in the Authorization header
- Add an exception handler in case you receive a 401 Unauthorized response because the JWT has expired call /api/user/refreshToken to obtain a new accessToken by providing the saved refreshToken.

  > a) the accessToken expires after an hour of creation; the expiry time is represented in UTC format by the accessTokenExpiryTime value.

> b) the refreshToken expires after an hour of creation; the expiry time is represented in UTC format by the refreshTokenExpiryTime value. c) if the refreshToken has expired as well, obtain a signed token by calling /api/user/authenticate API

- Use the newly issued accessToken to call your desired REST API by providing it in the Authorization header

API Key integration

- In the MetaDefender Storage Security interface, Navigate to Settings -> Users
- Find your user entry in the user list, click on the three dots on the right side of the entry, and then click on Generate an API Key
- Generate the key, copy it, and store it somewhere safe
- Use the API Key to call your desired REST API by providing it as a header with the Key: “ApiKey” and Value:

General considerations

The access token expiration date cannot be extended. By default, the access token is valid for an hour after calling /api/user/authenticate API to obtain it. The refresh token is also valid for an hour but can be extended by calling /api/user/refreshToken and it is also automatically extended with an hour with each non-GET request.

A refresh token is used to request a new access token when the current one expires without requiring re-authentication using a username and password.

The refresh token is used to forcibly expire any previously issued JWT when the refresh token expires or is removed by calling /api/user/logout.

A 3rd party application that needs persistent connectivity with MetaDefender Storage Security should implement a timeout mechanism to ensure that the refresh token is renewed before it expires by calling /api/user/refreshToken whenever the JWT (access token) is expired but before the refresh token expires as well.

Abac

Attribute-Based Access Control management endpoints

Get Roles

`GET /api/abac/roles`

Retrieve all ABAC roles. Set includePermissions to true to include the full permission list for each role.

Parameters

Name	In	Type	Required	Description
includePermissions	query	boolean	false	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "roles": [
    {
      "id": "string",
      "name": "string",
      "permissions": [
        {
          "id": "string",
          "action": "string",
          "resource": "string"
        }
      ]
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetRolesQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Create Role

`POST /api/abac/roles`

Create a new ABAC role. Provide an optional templateRoleId to inherit all permissions from an existing role; any IDs in permissionIds are merged on top of the template.

> Body parameter

```
{
  "name": "string",
  "permissionIds": [
    "string"
  ],
  "templateRoleId": "string"
}
```


Parameters

Name	In	Type	Required	Description
body	body	CreateRoleParameters	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "role": {
    "id": "string",
    "name": "string",
    "permissions": [
      {
        "id": "string",
        "action": "string",
        "resource": "string"
      }
    ]
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	CreateRoleCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get Role by ID

`GET /api/abac/roles/{roleId}`

Retrieve a single role and its associated permissions by role ID.

Parameters

Name	In	Type	Required	Description
roleId	path	string(guid)	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "role": {
    "id": "string",
    "name": "string",
    "permissions": [
      {
        "id": "string",
        "action": "string",
        "resource": "string"
      }
    ]
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetRoleByIdQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update Role

`PUT /api/abac/roles/{roleId}`

Replace an existing role's name and full permission set in a single operation. The supplied permissionIds become the role's complete permission list (previously assigned permissions not in the list are removed). Predefined system roles cannot be updated.

> Body parameter

```
{
  "name": "string",
  "permissionIds": [
    "string"
  ]
}
```

Parameters

Name	In	Type	Required	Description
roleId	path	string(guid)	true	none
body	body	UpdateRoleParameters	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateRoleCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Delete Role

`DELETE /api/abac/roles/{roleId}`

Permanently delete a role. Predefined system roles cannot be deleted.

Parameters

Name	In	Type	Required	Description
roleId	path	string(guid)	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	DeleteRoleCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get Permissions

`GET /api/abac/permissions`

Retrieve all available ABAC permissions (action + resource pairs).

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "permissions": [
    {
      "id": "string",
      "action": "string",
      "resource": "string"
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetPermissionsQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Create Permission

`POST /api/abac/permissions`

Create a new ABAC permission defined by an action and a resource.

> Body parameter

```
{
  "action": "string",
  "resource": "string"
}
```


Parameters

Name	In	Type	Required	Description
body	body	CreatePermissionParameters	true	none

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "permission": {
    "id": "string",
    "action": "string",
    "resource": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	CreatePermissionCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Delete Permission

`DELETE /api/abac/permissions/{permissionId}`

Permanently delete a permission by ID. Predefined base permissions cannot be deleted.

Parameters

Name	In	Type	Required	Description
permissionId	path	string(guid)	true	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	DeletePermissionCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Assign Permission to Role

`POST /api/abac/roles/{roleId}/permissions`

Assign a permission to the specified role.

Body parameter

```
{
  "permissionId": "string"
}
```

Parameters

Name	In	Type	Required	Description
roleId	path	string(guid)	true	none
body	body	RolePermissionParameters	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	AssignPermissionToRoleCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Remove Permission from Role

`DELETE /api/abac/roles/{roleId}/permissions/{permissionId}`

Remove a permission from the specified role. Permissions that were assigned to a predefined role during initialization cannot be removed.

Parameters

Name	In	Type	Required	Description
roleId	path	string(guid)	true	none
permissionId	path	string(guid)	true	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	RemovePermissionFromRoleCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Assign Role to User

`POST /api/abac/users/{userId}/roles`

Assign a role to a user, optionally with an expiry date after which the assignment is no longer valid.

Body parameter

```
{
  "roleId": "string",
  "expiresAt": "2019-08-24T14:15:22Z"
}
```


Parameters

Name	In	Type	Required	Description
userId	path	string	true	none
body	body	AssignRoleToUserParameters	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "roleBindingId": "string"
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	AssignRoleToUserCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Replace User Roles

`PUT /api/abac/users/{userId}/roles`

Atomically replace all role assignments for a user with the provided set of role IDs. All existing role bindings are removed and new ones are created in a single transaction.

> Body parameter

```
{
  "roleIds": [
    "string"
  ]
}
```

Parameters

Name	In	Type	Required	Description
userId	path	string	true	none
body	body	ReplaceUserRolesParameters	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	ReplaceUserRolesCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Remove Role from User

`DELETE /api/abac/users/{userId}/roles/{roleBindingId}`

Remove a role binding from a user using the role binding ID.

Parameters

Name	In	Type	Required	Description
userId	path	string	true	none
roleBindingId	path	string(guid)	true	none

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	RemoveRoleFromUserCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Assign Permission to User

`POST /api/abac/users/{userId}/permissions`

Assign a permission directly to a user, optionally with an expiry date.

> Body parameter

```
{
  "permissionId": "string",
  "expiresAt": "2019-08-24T14:15:22Z"
}
```


Parameters

Name	In	Type	Required	Description
userId	path	string	true	none
body	body	AssignPermissionToUserParameters	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "userPermissionId": "string"
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	AssignPermissionToUserCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Remove Permission from User

`DELETE /api/abac/users/{userId}/permissions/{userPermissionId}`

Remove a directly assigned permission from a user using the user permission ID.

Parameters

Name	In	Type	Required	Description
userId	path	string	true	none
userPermissionId	path	string(guid)	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	RemovePermissionFromUserCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Account

Manage your accounts

Add an account

`POST /api/account`

Add a new storage account to the system

Body parameter

```
CredentialsFile: string
```

Parameters

Name	In	Type	Required	Description
Name	query	string	false	display name for the account
Description	query	string	false	description of the account purpose or scope
Credentials	query	string	false	access credentials for the storage service
ProtocolType	query	any	false	type of protocol (0 = SMB, 1 = S3, etc.)
VendorType	query	any	false	vendor type of storage (0 = AmazonS3, 1 = OneDrive, etc.)
AutoDiscovery	query	boolean	false	if true, enables automatic storage detection
GroupId	query	string	false	id of the group account storages should be added to by default
StorageIds	query	array[string]	false	id's of specific storage units under this account
body	body	object	false	none
» CredentialsFile	body	string(binary)¦null	false	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "account": {
    "id": "string",
    "name": "string",
    "description": "string",
    "credentials": "string",
    "protocolType": 0,
    "vendorType": 0,
    "autoDiscovery": true,
    "autodiscoveryDestinationGroupId": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	AddAccountCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch all accounts

`GET /api/account`

Retrieve a list of all configured accounts

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "accounts": [
    {
      "id": "string",
      "name": "string",
      "description": "string",
      "storages": [
        {
          "id": "string",
          "groupId": "string",
          "accountId": "string",
          "name": "string",
          "storageClientId": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "scansInProgress": true,
          "isRtpInProgress": true,
          "isScheduledScanInProgress": true,
          "storageFunctionality": "string",
          "$type": "string"
        }
      ],
      "credentials": "string",
      "autoDiscovery": true,
      "groupId": "string",
      "protocolType": 0,
      "vendorType": 0
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetAccounts

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update an account

`PATCH /api/account`

Update an existing account's configuration

> Body parameter

```
CredentialsFile: string
```

Parameters

Name	In	Type	Required	Description
AccountId	query	string	false	id of the account to update
Name	query	string	false	display name for the account
Description	query	string	false	description of the account purpose or scope
Credentials	query	string	false	access credentials for the storage service
ProtocolType	query	any	false	type of protocol (0 = SMB, 1 = S3, etc.)
VendorType	query	any	false	vendor type of storage (0 = AmazonS3, 1 = OneDrive, etc.)
AutoDiscovery	query	boolean	false	if true, enables automatic storage detection
GroupId	query	string	false	id of the group account storages should be added to by default
body	body	object	false	none
» CredentialsFile	body	string(binary)¦null	false	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "accountId": "string"
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateAccountCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Delete an account

`DELETE /api/account/{accountId}`

Permanently delete an account and its storage references

Parameters

Name	In	Type	Required	Description
accountId	path	string	true	id of the account to delete

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "accountId": "string"
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	DeleteAccountCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch account by ID

`GET /api/account/{accountId}`

Retrieve a single account's details by ID

Parameters

Name	In	Type	Required	Description
accountId	path	string	true	id of the account to get

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "account": {
    "id": "string",
    "name": "string",
    "description": "string",
    "storages": [
      {
        "id": "string",
        "groupId": "string",
        "accountId": "string",
        "name": "string",
        "storageClientId": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "scansInProgress": true,
        "isRtpInProgress": true,
        "isScheduledScanInProgress": true,
        "storageFunctionality": "string",
        "$type": "string"
      }
    ],
    "credentials": "string",
    "autoDiscovery": true,
    "groupId": "string",
    "protocolType": 0,
    "vendorType": 0
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetAccountById

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch available storage units for an account

`GET /api/account/{accountId}/storages`

List all storage units linked to a particular account

Parameters

Name	In	Type	Required	Description
accountId	path	string	true	ID of the account

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "autodiscoverySource": "string",
      "autodiscoverySourceUrl": "string"
    }
  ],
  "protocolType": 0,
  "vendorType": 0
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	EnumerateAccountStoragesQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch the number of accounts

`GET /api/account/count`

Get the total number of accounts in the system

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "accountCount": {
    "count": 0
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetAccountsCount

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch account sources

`GET /api/account/{accountId}/sources`

Fetch account sources

Parameters

Name	In	Type	Required	Description
filter	query	string	false	none
accountId	path	string	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    "string"
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	EnumerateAccountSourcesQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Audit

List audit information

Fetch audit logs

`GET /api/audit/{startIndex}/{count}`

Parameters

Name	In	Type	Required	Description
startIndex	path	string	true	none
count	path	string	true	none
logType	query	string	false	none
categoryType	query	string	false	none
logLevel	query	string	false	none
searchText	query	string	false	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "logLevel": 0,
      "details": "string",
      "logType": 0,
      "userId": "string",
      "category": 0,
      "timeStamp": "2019-08-24T14:15:22Z",
      "userName": "string"
    }
  ],
  "filterCount": 0,
  "totalCount": 0
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	EnumerateLogsQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Configuration

Import or export configuration file

Export configuration file

`GET /api/configuration/export`

Export the current configuration settings to an archive. The file will be encrypted using the provided password.

Parameters

Name	In	Type	Required	Description
exportLicense	query	boolean	false	none
exportSmtp	query	boolean	false	none
exportNotifications	query	boolean	false	none
exportRetention	query	boolean	false	none
exportUsers	query	boolean	false	none
password	query	string	false	none

Example responses

200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Import configuration file

`POST /api/configuration/import`

> Body parameter

```
ConfigurationFile: string
password: string
```

Parameters

Name	In	Type	Required	Description
body	body	object	false	none
» ConfigurationFile	body	string(binary)¦null	false	none
» password	body	string¦null	false	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "areSmtpSettingsImported": true,
  "isLicenseImported": true,
  "isRetentionImported": true,
  "areNotificationsImported": true,
  "hasNotificationsPartialFailure": true,
  "activationDto": {
    "result": 0,
    "responseKey": "string",
    "responseMessage": "string",
    "responseMessageParams": {
      "property1": "string",
      "property2": "string"
    }
  },
  "newLocalUserId": "string",
  "areUsersImported": true
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	ImportSettingsCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get enabled modules

`GET /api/configuration/enabledModules`

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "enabledModules": [
    0
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetEnabledModules

This operation does not require authentication

ExternalLogger

Manage external loggers

Retrieve external loggers

`GET /api/externallogger`

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "isEnabled": true,
      "loggerType": 0,
      "connectionSettings": {
        "$type": "string"
      }
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	EnumerateExternalLoggersQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update external logger state

`PUT /api/externallogger/status`

> Body parameter


```
{
  "id": "string",
  "isEnabled": true
}
```


Parameters

Name	In	Type	Required	Description
body	body	UpdateExternalLoggerStatusParameters	true	none

Example responses

200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Add a new Syslog server configuration

`POST /api/externallogger/udpSyslog`

Body parameter

```
{
  "isEnabled": true,
  "serverAddress": "string",
  "port": 0,
  "loggerType": 0,
  "facility": 0,
  "format": 0,
  "$type": "string"
}
```

Parameters

Name	In	Type	Required	Description
body	body	AddUdpSyslogExternalLoggerParameters	true	none

Example responses

200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update a Syslog server configuration

`PUT /api/externallogger/udpSyslog`

> Body parameter

```
{
  "isEnabled": true,
  "serverAddress": "string",
  "port": 0,
  "loggerType": 0,
  "facility": 0,
  "format": 0,
  "$type": "string",
  "id": "string"
}
```

Parameters

Name	In	Type	Required	Description
body	body	UpdateUdpSyslogExternalLoggerParameters	true	none

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Add a new Kafka server configuration

`POST /api/externallogger/kafka`

> Body parameter

```
{
  "isEnabled": true,
  "serverAddressWithPortList": "string",
  "loggerType": 0,
  "topic": "string",
  "configuration": "string",
  "$type": "string"
}
```

Parameters

Name	In	Type	Required	Description
body	body	AddKafkaExternalLoggerParameters	true	none

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update a Kafka server configuration

`PUT /api/externallogger/kafka`

> Body parameter

```
{
  "isEnabled": true,
  "serverAddressWithPortList": "string",
  "loggerType": 0,
  "topic": "string",
  "configuration": "string",
  "$type": "string",
  "id": "string"
}
```


Parameters

Name	In	Type	Required	Description
body	body	UpdateKafkaExternalLoggerParameters	true	none

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Delete external logger

`DELETE /api/externallogger/{id}`

Parameters

Name	In	Type	Required	Description
id	path	string	true	none

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

File

Retrieve processed files information

Rescan a file on demand

`PUT /api/file`

This request is used to update a scanned file with passwords, in case it is an encrypted archive andit could not be scanned because the passwords to decrypt it were not provided. It can also be used withoutproviding any passwords to simply rescan a specific file from a finished scan.

> Body parameter

```
{
  "fileId": "string",
  "fileIds": [
    "string"
  ],
  "scanId": "string",
  "storageProtocolType": 0,
  "passwords": [
    "string"
  ],
  "workflowId": "string"
}
```

Parameters

Name	In	Type	Required	Description
body	body	OnDemandScanParameters	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	OnDemandScanCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Enumerate processed files

`GET /api/file`

Parameters

Name	In	Type	Required	Description
StartIndex	query	integer(int32)	false	starting position for pagination (0-based index)
Count	query	integer(int32)	false	none
StartDate	query	string(date-time)	false	filter files by scan date
EndDate	query	string(date-time)	false	filter files by scan date
Search	query	string	false	text filter
ScanType	query	array[integer]	false	scan type such as (on-demand, scheduled, real time)
StorageId	query	array[string]	false	filter based on a list of possible storage id
ScanId	query	array[string]	false	filter based on a list of possible scan id
ScanWorkflowSnapshotId	query	array[string]	false	filter based on a list of possible scan workflow snapshot id
ProcessingState	query	array[integer]	false	none
Result	query	array[integer]	false	filter by scan result (0 = Clean, 1 = Blocked, 2 = Processing)
ScanAllResultI	query	array[integer]	false	filter by MD Core scan result (0 = NoThreatDetected, 1 = Infected, 2 = Suspicious, etc.)
ScanResultICount	query	array[integer]	false	the number of engine results that are not “No threat detected”
SandboxVerdict	query	array[integer]	false	filter by Sandbox verdict (0 = NoThreat, 1 = Malicious, 2 = Suspicious)
VulnerabilityCount	query	array[integer]	false	the number of vulnerabilities detected
HasVulnerabilities	query	boolean	false	filter by presence of vulnerabilities
DlpDetection	query	boolean	false	DLP detection
HasBlockedFileVersions	query	boolean	false	filter by presence of blocked file versions

Remediations	query	array[integer]	false	filter by remediation performed on the file
SkippedScan	query	boolean	false	filter by whether the scan was skipped
SortBy	query	array[integer]	false	none
SortOrder	query	array[integer]	false	none
Expand	query	array[string]	false	Possible values for expand: workflow_snapshot

Enumerated Values

Parameter	Value
ScanType	0
ScanType	1
ScanType	2
ProcessingState	0
ProcessingState	1
ProcessingState	2
ProcessingState	3
ProcessingState	4
Result	0
Result	1
Result	2
ScanAllResultI	0
ScanAllResultI	1
ScanAllResultI	2
ScanAllResultI	3
ScanAllResultI	7
ScanAllResultI	8
ScanAllResultI	9
ScanAllResultI	10
ScanAllResultI	12
ScanAllResultI	13
ScanAllResultI	14
ScanAllResultI	15
ScanAllResultI	16
ScanAllResultI	17
ScanAllResultI	18
ScanAllResultI	19

ScanAllResultI	20
ScanAllResultI	21
ScanAllResultI	22
ScanAllResultI	23
ScanAllResultI	24
ScanAllResultI	25
ScanAllResultI	26
ScanAllResultI	27
ScanAllResultI	28
ScanAllResultI	29
ScanAllResultI	30
ScanAllResultI	31
ScanAllResultI	32
ScanAllResultI	33
ScanAllResultI	34
ScanAllResultI	36
ScanAllResultI	38
ScanAllResultI	39
ScanAllResultI	40
ScanAllResultI	41
ScanAllResultI	42
ScanAllResultI	43
ScanAllResultI	255
ScanAllResultI	1014
ScanAllResultI	9999
SandboxVerdict	0
SandboxVerdict	1
SandboxVerdict	2
SandboxVerdict	3

SandboxVerdict	4
SandboxVerdict	5
SandboxVerdict	999
Remediations	0
Remediations	5
Remediations	6
Remediations	7
Remediations	8
Remediations	9
Remediations	10
Remediations	11
Remediations	12
Remediations	13
Remediations	14
Remediations	15
Remediations	16
Remediations	17
Remediations	18
Remediations	19
Remediations	20
Remediations	21
Remediations	22
Remediations	23
Remediations	24
Remediations	25
Remediations	100
Remediations	101
Remediations	102
SortBy	0

SortBy	1
SortBy	2
SortBy	3
SortBy	4
SortBy	5
SortBy	6
SortBy	7
SortBy	8
SortOrder	0
SortOrder	1

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "result": 0,
      "responseKey": "string",
      "responseMessage": "string",
      "responseMessageParams": {
        "property1": "string",
        "property2": "string"
      },
      "id": "string",
      "scanId": "string",
      "name": "string",
      "path": "string",
      "hash": "string",
      "fileOwner": {
        "userId": "string",
        "fullName": "string",
        "userName": "string",
        "emailAddress": "string"
      },
      "size": 0,
      "fileTypeCategory": 0,
      "extension": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageName": "string",
      "storageLocation": "string",
      "scanName": "string",
      "progress": {
        "upload": 0,
        "scan": 0
      },
      "scanResult": {
        "extractedFiles": {
          "dataId": "string",
          "scanResultI": 0,
          "detectedBy": 0,
          "filesExtractedCount": 0
        },
        "scanResults": {
```

```
{
              "engineName": "string",
              "threatFound": "string",
              "scanResultI": 0,
              "defTime": "string",
              "scanTime": 0.1,
              "engId": "string",
              "waitTime": 0.1
            }
          ],
          "dataId": "string",
          "scanAllResultI": 0,
          "scanAllResultA": "string",
          "startTime": "string",
          "totalTime": 0.1,
          "totalAvs": 0,
          "totalDetections": 0,
          "progressPercentage": 0
        },
        "fileInfo": {
          "fileSize": 0,
          "uploadTimestamp": "string",
          "md5": "string",
          "sha1": "string",
          "sha256": "string",
          "fileTypeCategory": "string",
          "fileTypeDescription": "string",
          "fileTypeExtension": "string",
          "displayName": "string",
          "fileTypeId": "string"
        },
        "fileScanInfo": {
          "finalVerdict": {
            "blocked": true,
            "verdict": "string",
            "confidence": 0.1,
            "threatLevel": 0.1
          },
          "overviewReport": {
            "analysisState": "string",
            "signalGroups": [
              {
                "description": "string",
                "verdict": "string",
                "signals": [
                  {
                    "originType": "string",
                    "signalReadable": "string",
                    "strength": 0.1
                  }
                ]
              }
            ]
```

```
"errorCategory": "string",
          "errorCode": 0,
          "errorDetails": "string"
        },
        "processInfo": {
          "postProcessing": {
            "actionsRan": "string",
            "actionsFailed": "string",
            "convertedTo": "string",
            "copyMoveDestination": "string",
            "convertedDestination": "string",
            "sanitizationDetails": {
              "failureCategory": "string",
              "details": "string"
            }
          },
          "progressPercentage": 0,
          "userAgent": "string",
          "profile": "string",
          "result": "string",
          "blockedReason": "string",
          "fileTypeSkippedScan": true,
          "processingTime": 0.1,
          "queueTime": 0.1,
          "resultI": 0
        },
        "dataId": "string",
        "dlpInfo": {
          "hits": null,
          "verdict": 0
        },
        "sbomInfo": {
          "finalVerdict": {
            "blocked": true,
            "licensingPackageCounts": {
              "allowed": 0,
              "blocked": 0,
              "unknown": 0,
              "unsupported": 0
            },
            "severity": 0,
            "totalPackageCount": 0,
            "verdict": "string",
            "vulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            }
          },
          "overviewReport": {
            "isPartialReport": true,
```

```
"osName": "string",
            "osVersion": "string"
          },
          "scanMode": "string",
          "steps": [
            {
              "command": "string",
              "packages": [
                {
                  "authors": "string",
                  "releaseDate": "2019-08-24T14:15:22Z",
                  "ecosystem": "string",
                  "group": "string",
                  "archive": true,
                  "executable": true,
                  "uid": "string",
                  "licenses": {
                    "allowed": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            null
                          ],
                          "conditions": [
                            null
                          ],
                          "limitations": [
                            null
                          ]
                        }
                      }
                    ],
                    "blocked": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            null
                          ],
                          "conditions": [
                            null
                          ],
                          "limitations": [
                            null
                          ]
                        }
                      }
                    ]
```

```
"packageName": "string",
                    "packageVersion": "string",
                    "hashes": [
                      {
                        "sha256": "string",
                        "fileName": "string"
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    },
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "arch": "string",
                    "osName": "string",
                    "osVersion": "string",
                    "dependencies": [
                      {
                        "ecosystem": "string",
                        "group": "string",
                        "packageName": "string",
                        "packageVersion": "string",
                        "vulnerabilities": [
                          {
                            "id": "string",
                            "severity": 0,
                            "source": "string",
                            "fixedVersions": "string",
                            "cwes": [
                              null
                            ]
                          }
                        ],
                        "vulnerabilityCounts": {
                          "critical": 0,
                          "high": 0,
                          "medium": 0,
                          "low": 0,
                          "unknown": 0
                        }
```

```
]
                  }
                ],
                "licenseStatus": "string",
                "licensingPackageCounts": {
                  "allowed": 0,
                  "blocked": 0,
                  "unknown": 0,
                  "unsupported": 0
                },
                "step": 0,
                "vulnerabilityCount": 0,
                "sha256": "string",
                "severity": 0,
                "vulnerabilityCounts": {
                  "critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
                }
              }
            ],
            "packages": [
              {
                "authors": "string",
                "releaseDate": "2019-08-24T14:15:22Z",
                "ecosystem": "string",
                "group": "string",
                "archive": true,
                "executable": true,
                "uid": "string",
                "licenses": {
                  "allowed": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ],
                  "blocked": [
                    {
```

```
"url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            "string"
                          ],
                          "conditions": [
                            "string"
                          ],
                          "limitations": [
                            "string"
                          ]
                        }
                      }
                    ]
                  },
                  "packageName": "string",
                  "packageVersion": "string",
                  "target": "string",
                  "hashes": [
                    {
                      "sha256": "string",
                      "fileName": "string"
                    }
                  ],
                  "vulnerabilities": [
                    {
                      "id": "string",
                      "severity": 0,
                      "source": "string",
                      "fixedVersions": "string",
                      "cwes": [
                        "string"
                      ]
                    }
                  ],
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  },
                  "dependencies": [
                    {
                      "ecosystem": "string",
                      "group": "string",
                      "packageName": "string",
                      "packageVersion": "string",
                      "vulnerabilities": [
                        {
                          "id": "string",
                          "severity": 0,
```

```
"fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    }
                  }
                ]
              }
            ],
            "dependenciesPackageCount": 0,
            "dependenciesVulnerabilityTotal": 0,
            "dependenciesVulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            }
          },
          "resultTemplateHash": "string"
        },
        "vulnerabilityInfo": {
          "vulnerabilityInfoResult": {
            "code": 0,
            "hash": "string",
            "method": 0,
            "timeStamp": "string",
            "timing": 0,
            "detectedProduct": {
              "hasVulnerability": true,
              "isCurrent": true,
              "product": {
                "id": 0,
                "name": "string"
              },
              "remediationLink": "string",
              "severity": "string",
              "sigName": "string",
              "signature": 0,
              "vendor": {
                "id": 0,
                "name": "string"
              },
              "version": "string",
              "versionData": {
```

```
"feedId": 0,
                "version": "string"
              },
              "vulnerabilities": [
                {
                  "description": "string",
                  "details": {
                    "cpe": "string",
                    "cve": "string",
                    "cvss": {
                      "accessComplexity": "string",
                      "accessVector": "string",
                      "authentication": "string",
                      "availabilityImpact": "string",
                      "confidentialityImpact": "string",
                      "generatedOnEpoch": "string",
                      "integrityImpact": "string",
                      "score": "string",
                      "source": "string"
                    },
                    "cvss3": {
                      "attackComplexity": "string",
                      "attackVector": "string",
                      "availabilityImpact": "string",
                      "baseScore": "string",
                      "baseSeverity": "string",
                      "confidentialityImpact": "string",
                      "exploitabilityScore": "string",
                      "impactScore": "string",
                      "integrityImpact": "string",
                      "privilegesRequired": "string",
                      "scope": "string",
                      "userInteraction": "string",
                      "vectorString": "string"
                    },
                    "cwe": "string",
                    "lastModifiedEpoch": "string",
                    "publishedEpoch": "string",
                    "references": [
                      "string"
                    ]
                  },
                  "severity": "string",
                  "severityIndex": 0,
                  "staticId": 0
                }
              ]
            }
          }
        },
        "sessionCookie": "string",
        "fileTypeInfo": {
          "fileInfoExtended": {
```

```
"groupId": "string",
            "typeId": "string"
          }
        },
        "scanInstanceUrl": "string",
        "scanWorkflowSnapshotId": "string",
        "coreEngines": [
          {
            "engineId": "string",
            "engineName": "string",
            "engineVersion": "string",
            "dbVersion": "string",
            "definitionTime": "string"
          }
        ]
      },
      "processingState": 0,
      "exception": {
        "exceptionDetails": "string",
        "exceptionMessage": "string"
      },
      "created": "2019-08-24T14:15:22Z",
      "lastModified": "2019-08-24T14:15:22Z",
      "discoveryStarted": "2019-08-24T14:15:22Z",
      "fileProcessingDetails": {
        "streamingStartedTime": "2019-08-24T14:15:22Z",
        "streamingCompletedTime": "2019-08-24T14:15:22Z",
        "streamingTimeSpanMs": 0,
        "scanCompletedTime": "2019-08-24T14:15:22Z",
        "processingTimeSpanMs": 0
      },
      "coreEngines": [
        {
          "engineId": "string",
          "engineName": "string",
          "engineVersion": "string",
          "dbVersion": "string",
          "definitionTime": "string"
        }
      ],
      "remediation": {
        "isTagged": true,
        "isDeleted": true,
        "isMoved": true,
        "isCopied": true,
        "deepCdr": {
          "sanitizationDetails": {
            "failureCategory": "string",
            "details": "string"
          },
          "cdrObjectState": 0,
          "fileWasReplaced": true
        },
```

```
{
            "objectActionType": 0,
            "exception": {
              "exceptionDetails": "string",
              "exceptionMessage": "string"
            }
          }
        ],
        "objectRemediationActions": [
          {
            "objectActionType": 0,
            "objectActionState": 0,
            "cdrObjectState": 0,
            "exceptions": [
              "string"
            ],
            "id": "string",
            "fileId": "string",
            "fileScanCompletedTime": "2019-08-24T14:15:22Z",
            "remediationId": "string",
            "tenantId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z"
          }
        ]
      },
      "metaDefenderCoreUrl": "string",
      "discoveryStartedDaySec": 0,
      "objectStorageMetadata": "string",
      "scanWorkflowSnapshotId": "string",
      "scanWorkflowSnapshot": {
        "id": "string",
        "scanId": "string",
        "workflow": {
          "id": "string",
          "scanPoolId": "string",
          "scanProcessFlow": 0,
          "isIdentityScanningEnabled": true,
          "scanIntervalDays": 0,
          "rules": [
            "string"
          ],
          "type": 0,
          "workflowSupportedTechnologies": {
            "dlp": {
              "status": 0
            },
            "deepCdr": {
              "status": 0
            },
            "sandBox": {
              "status": 0
            },
            "metascan": {
```

```
}
          },
          "name": "string",
          "userAgent": "string",
          "scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanPoolId": "string",
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "filters": {
            "id": "string",
            "minimumSize": 0,
            "maximumSize": 0,
            "minimumDateTime": "2019-08-24T14:15:22Z",
            "maximumDateTime": "2019-08-24T14:15:22Z"
          },
          "fileDefinition": {
            "blockingStrategy": 0,
            "configurableBlockingStrategy": {
              "shouldTreatFilesWithSensitiveDataAsBlocked": true,
              "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
            }
          },
          "remediations": [
            {
```

```
"workflowId": "string",
              "isEnabled": true,
              "configuration": "string",
              "type": 0,
              "handlingType": 0
            }
          ],
          "tenantId": "string"
        },
        "startTimestamp": 0,
        "stopTimestamp": 0,
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
```

```
{
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "storage": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        },
        "workflowStatistics": {
          "taggingStatistics": {
            "total": 0
          },
          "keepAllowedStatistics": {
            "total": 0
          },
          "keepBlockedStatistics": {
            "total": 0
          },
          "keepSanitizedStatistics": {
            "total": 0
          },
          "movedAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
```

```
"storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedSanitizedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
```

```
"vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
```

```
"statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "deleteAllowedStatistics": {
            "total": 0
          },
          "deleteBlockedStatistics": {
            "total": 0
          },
          "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
        },
        "account": {
          "id": "string",
          "name": "string",
          "description": "string",
          "credentials": "string",
          "protocolType": 0,
          "vendorType": 0,
          "autoDiscovery": true,
          "autodiscoveryDestinationGroupId": "string"
        },
        "realTimeStartDate": "2019-08-24T14:15:22Z"
      },
      "versions": [
        {
          "result": 0,
          "responseKey": "string",
          "responseMessage": "string",
          "responseMessageParams": {
            "property1": "string",
            "property2": "string"
          },
          "id": "string",
          "scanId": "string",
          "name": "string",
          "path": "string",
          "hash": "string",
          "fileOwner": {
            "userId": "string",
            "fullName": "string",
            "userName": "string",
            "emailAddress": "string"
          },
          "size": 0,
          "fileTypeCategory": 0,
          "extension": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
```

```
"vendorType": 0
          },
          "storageName": "string",
          "storageLocation": "string",
          "scanName": "string",
          "progress": {
            "upload": 0,
            "scan": 0
          },
          "scanResult": {
            "extractedFiles": {
              "dataId": "string",
              "scanResultI": 0,
              "detectedBy": 0,
              "filesExtractedCount": 0
            },
            "scanResults": {
              "scanDetails": [
                {
                  "engineName": "string",
                  "threatFound": "string",
                  "scanResultI": 0,
                  "defTime": "string",
                  "scanTime": 0.1,
                  "engId": "string",
                  "waitTime": 0.1
                }
              ],
              "dataId": "string",
              "scanAllResultI": 0,
              "scanAllResultA": "string",
              "startTime": "string",
              "totalTime": 0.1,
              "totalAvs": 0,
              "totalDetections": 0,
              "progressPercentage": 0
            },
            "fileInfo": {
              "fileSize": 0,
              "uploadTimestamp": "string",
              "md5": "string",
              "sha1": "string",
              "sha256": "string",
              "fileTypeCategory": "string",
              "fileTypeDescription": "string",
              "fileTypeExtension": "string",
              "displayName": "string",
              "fileTypeId": "string"
            },
            "fileScanInfo": {
              "finalVerdict": {
                "blocked": true,
                "verdict": "string",
```

```
"threatLevel": 0.1
            },
            "overviewReport": {
              "analysisState": "string",
              "signalGroups": [
                {
                  "description": "string",
                  "verdict": "string",
                  "signals": [
                    {
                      "originType": "string",
                      "signalReadable": "string",
                      "strength": 0.1
                    }
                  ]
                }
              ]
            },
            "errorCategory": "string",
            "errorCode": 0,
            "errorDetails": "string"
          },
          "processInfo": {
            "postProcessing": {
              "actionsRan": "string",
              "actionsFailed": "string",
              "convertedTo": "string",
              "copyMoveDestination": "string",
              "convertedDestination": "string",
              "sanitizationDetails": {
                "failureCategory": "string",
                "details": "string"
              }
            },
            "progressPercentage": 0,
            "userAgent": "string",
            "profile": "string",
            "result": "string",
            "blockedReason": "string",
            "fileTypeSkippedScan": true,
            "processingTime": 0.1,
            "queueTime": 0.1,
            "resultI": 0
          },
          "dataId": "string",
          "dlpInfo": {
            "hits": null,
            "verdict": 0
          },
          "sbomInfo": {
            "finalVerdict": {
              "blocked": true,
              "licensingPackageCounts": {
```

```
"blocked": 0,
                  "unknown": 0,
                  "unsupported": 0
                },
                "severity": 0,
                "totalPackageCount": 0,
                "verdict": "string",
                "vulnerabilityCounts": {
                  "critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
                }
              },
              "overviewReport": {
                "isPartialReport": true,
                "osInfo": {
                  "osName": "string",
                  "osVersion": "string"
                },
                "scanMode": "string",
                "steps": [
                  {
                    "command": "string",
                    "packages": [
                      {
                        "authors": "string",
                        "releaseDate": "2019-08-24T14:15:22Z",
                        "ecosystem": "string",
                        "group": "string",
                        "archive": true,
                        "executable": true,
                        "uid": "string",
                        "licenses": {
                          "allowed": [
                            {}
                          ],
                          "blocked": [
                            {}
                          ]
                        },
                        "packageName": "string",
                        "packageVersion": "string",
                        "hashes": [
                          {
                            "sha256": "string",
                            "fileName": "string"
                          }
                        ],
                        "vulnerabilityCounts": {
                          "critical": 0,
                          "high": 0,
```

```
"low": 0,
                        "unknown": 0
                      },
                      "vulnerabilities": [
                        {
                          "id": "string",
                          "severity": 0,
                          "source": "string",
                          "fixedVersions": "string",
                          "cwes": [
                            null
                          ]
                        }
                      ],
                      "arch": "string",
                      "osName": "string",
                      "osVersion": "string",
                      "dependencies": [
                        {
                          "ecosystem": "string",
                          "group": "string",
                          "packageName": "string",
                          "packageVersion": "string",
                          "vulnerabilities": [
                            null
                          ],
                          "vulnerabilityCounts": {}
                        }
                      ]
                    }
                  ],
                  "licenseStatus": "string",
                  "licensingPackageCounts": {
                    "allowed": 0,
                    "blocked": 0,
                    "unknown": 0,
                    "unsupported": 0
                  },
                  "step": 0,
                  "vulnerabilityCount": 0,
                  "sha256": "string",
                  "severity": 0,
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  }
                }
              ],
              "packages": [
                {
```

```
"releaseDate": "2019-08-24T14:15:22Z",
                    "ecosystem": "string",
                    "group": "string",
                    "archive": true,
                    "executable": true,
                    "uid": "string",
                    "licenses": {
                      "allowed": [
                        {
                          "id": "string",
                          "url": "string",
                          "category": "string",
                          "usageRestriction": {
                            "permissions": [
                              null
                            ],
                            "conditions": [
                              null
                            ],
                            "limitations": [
                              null
                            ]
                          }
                        }
                      ],
                      "blocked": [
                        {
                          "id": "string",
                          "url": "string",
                          "category": "string",
                          "usageRestriction": {
                            "permissions": [
                              null
                            ],
                            "conditions": [
                              null
                            ],
                            "limitations": [
                              null
                            ]
                          }
                        }
                      ]
                    },
                    "packageName": "string",
                    "packageVersion": "string",
                    "target": "string",
                    "hashes": [
                      {
                        "sha256": "string",
                        "fileName": "string"
                      }
                    ],
```

```
{
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    },
                    "dependencies": [
                      {
                        "ecosystem": "string",
                        "group": "string",
                        "packageName": "string",
                        "packageVersion": "string",
                        "vulnerabilities": [
                          {
                            "id": "string",
                            "severity": 0,
                            "source": "string",
                            "fixedVersions": "string",
                            "cwes": [
                              null
                            ]
                          }
                        ],
                        "vulnerabilityCounts": {
                          "critical": 0,
                          "high": 0,
                          "medium": 0,
                          "low": 0,
                          "unknown": 0
                        }
                      }
                    ]
                  }
                ],
                "dependenciesPackageCount": 0,
                "dependenciesVulnerabilityTotal": 0,
                "dependenciesVulnerabilityCounts": {
                  "critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
```

```
},
            "resultTemplateHash": "string"
          },
          "vulnerabilityInfo": {
            "vulnerabilityInfoResult": {
              "code": 0,
              "hash": "string",
              "method": 0,
              "timeStamp": "string",
              "timing": 0,
              "detectedProduct": {
                "hasVulnerability": true,
                "isCurrent": true,
                "product": {
                  "id": 0,
                  "name": "string"
                },
                "remediationLink": "string",
                "severity": "string",
                "sigName": "string",
                "signature": 0,
                "vendor": {
                  "id": 0,
                  "name": "string"
                },
                "version": "string",
                "versionData": {
                  "countBehind": 0,
                  "feedId": 0,
                  "version": "string"
                },
                "vulnerabilities": [
                  {
                    "description": "string",
                    "details": {
                      "cpe": "string",
                      "cve": "string",
                      "cvss": {
                        "accessComplexity": "string",
                        "accessVector": "string",
                        "authentication": "string",
                        "availabilityImpact": "string",
                        "confidentialityImpact": "string",
                        "generatedOnEpoch": "string",
                        "integrityImpact": "string",
                        "score": "string",
                        "source": "string"
                      },
                      "cvss3": {
                        "attackComplexity": "string",
                        "attackVector": "string",
                        "availabilityImpact": "string",
                        "baseScore": "string",
```

```
"confidentialityImpact": "string",
                          "exploitabilityScore": "string",
                          "impactScore": "string",
                          "integrityImpact": "string",
                          "privilegesRequired": "string",
                          "scope": "string",
                          "userInteraction": "string",
                          "vectorString": "string"
                        },
                        "cwe": "string",
                        "lastModifiedEpoch": "string",
                        "publishedEpoch": "string",
                        "references": [
                          "string"
                        ]
                      },
                      "severity": "string",
                      "severityIndex": 0,
                      "staticId": 0
                    }
                  ]
                }
              }
            },
            "sessionCookie": "string",
            "fileTypeInfo": {
              "fileInfoExtended": {
                "extensions": "string",
                "groupId": "string",
                "typeId": "string"
              }
            },
            "scanInstanceUrl": "string",
            "scanWorkflowSnapshotId": "string",
            "coreEngines": [
              {
                "engineId": "string",
                "engineName": "string",
                "engineVersion": "string",
                "dbVersion": "string",
                "definitionTime": "string"
              }
            ]
          },
          "processingState": 0,
          "exception": {
            "exceptionDetails": "string",
            "exceptionMessage": "string"
          },
          "created": "2019-08-24T14:15:22Z",
          "lastModified": "2019-08-24T14:15:22Z",
          "discoveryStarted": "2019-08-24T14:15:22Z",
          "fileProcessingDetails": {
```

```
"streamingCompletedTime": "2019-08-24T14:15:22Z",
            "streamingTimeSpanMs": 0,
            "scanCompletedTime": "2019-08-24T14:15:22Z",
            "processingTimeSpanMs": 0
          },
          "coreEngines": [
            {
              "engineId": "string",
              "engineName": "string",
              "engineVersion": "string",
              "dbVersion": "string",
              "definitionTime": "string"
            }
          ],
          "remediation": {
            "isTagged": true,
            "isDeleted": true,
            "isMoved": true,
            "isCopied": true,
            "deepCdr": {
              "sanitizationDetails": {
                "failureCategory": "string",
                "details": "string"
              },
              "cdrObjectState": 0,
              "fileWasReplaced": true
            },
            "postActionFailedInformation": [
              {
                "objectActionType": 0,
                "exception": {
                  "exceptionDetails": "string",
                  "exceptionMessage": "string"
                }
              }
            ],
            "objectRemediationActions": [
              {
                "objectActionType": 0,
                "objectActionState": 0,
                "cdrObjectState": 0,
                "exceptions": [
                  "string"
                ],
                "id": "string",
                "fileId": "string",
                "fileScanCompletedTime": "2019-08-24T14:15:22Z",
                "remediationId": "string",
                "tenantId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z"
              }
            ]
          },
```

```
"discoveryStartedDaySec": 0,
          "objectStorageMetadata": "string",
          "scanWorkflowSnapshotId": "string",
          "scanWorkflowSnapshot": {
            "id": "string",
            "scanId": "string",
            "workflow": {
              "id": "string",
              "scanPoolId": "string",
              "scanProcessFlow": 0,
              "isIdentityScanningEnabled": true,
              "scanIntervalDays": 0,
              "rules": [
                "string"
              ],
              "type": 0,
              "workflowSupportedTechnologies": {
                "dlp": {
                  "status": 0
                },
                "deepCdr": {
                  "status": 0
                },
                "sandBox": {
                  "status": 0
                },
                "metascan": {
                  "status": 0
                }
              },
              "name": "string",
              "userAgent": "string",
              "scanPool": {
                "id": "string",
                "name": "string",
                "scanPoolType": 0,
                "isDefault": true,
                "scanInstances": [
                  {
                    "id": "string",
                    "scanPoolId": "string",
                    "url": "string",
                    "apiKey": "string",
                    "timeout": "string",
                    "scanInstanceType": 0
                  }
                ]
              },
              "failOverScanPoolId": "string",
              "failOverScanPool": {
                "id": "string",
                "name": "string",
                "scanPoolType": 0,
```

```
"scanInstances": [
                  {
                    "id": "string",
                    "scanPoolId": "string",
                    "url": "string",
                    "apiKey": "string",
                    "timeout": "string",
                    "scanInstanceType": 0
                  }
                ]
              },
              "filters": {
                "id": "string",
                "minimumSize": 0,
                "maximumSize": 0,
                "minimumDateTime": "2019-08-24T14:15:22Z",
                "maximumDateTime": "2019-08-24T14:15:22Z"
              },
              "fileDefinition": {
                "blockingStrategy": 0,
                "configurableBlockingStrategy": {
                  "shouldTreatFilesWithSensitiveDataAsBlocked": true,
                  "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
                }
              },
              "remediations": [
                {
                  "id": "string",
                  "workflowId": "string",
                  "isEnabled": true,
                  "configuration": "string",
                  "type": 0,
                  "handlingType": 0
                }
              ],
              "tenantId": "string"
            },
            "startTimestamp": 0,
            "stopTimestamp": 0,
            "scanPool": {
              "id": "string",
              "name": "string",
              "scanPoolType": 0,
              "isDefault": true,
              "scanInstances": [
                {
                  "id": "string",
                  "scanPoolId": "string",
                  "url": "string",
                  "apiKey": "string",
                  "timeout": "string",
                  "scanInstanceType": 0
                }
```

```
},
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ],
            "failOverScanPool": {
              "id": "string",
              "name": "string",
              "scanPoolType": 0,
              "isDefault": true,
              "scanInstances": [
                {
                  "id": "string",
                  "scanPoolId": "string",
                  "url": "string",
                  "apiKey": "string",
                  "timeout": "string",
                  "scanInstanceType": 0
                }
              ]
            },
            "failOverScanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ],
            "storage": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
```

```
},
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            },
            "workflowStatistics": {
              "taggingStatistics": {
                "total": 0
              },
              "keepAllowedStatistics": {
                "total": 0
              },
              "keepBlockedStatistics": {
                "total": 0
              },
              "keepSanitizedStatistics": {
                "total": 0
              },
              "movedAllowedStatistics": {
                "total": 0,
                "destinationStorageInformation": {
                  "id": "string",
                  "name": "string",
                  "groupId": "string",
                  "clientId": "string",
                  "accountId": "string",
                  "source": "string",
                  "storageFunctionality": "string",
                  "storageType": {
                    "name": "string",
                    "categoryType": 0,
                    "protocolType": 0,
                    "vendorType": 0
                  },
                  "storageStatus": {
                    "message": "string",
                    "statusType": 0
                  },
                  "workflowId": "string",
                  "lastUpdated": "2019-08-24T14:15:22Z",
                  "tenantId": "string"
                }
              },
              "movedSanitizedStatistics": {
                "total": 0,
                "destinationStorageInformation": {
                  "id": "string",
                  "name": "string",
                  "groupId": "string",
                  "clientId": "string",
                  "accountId": "string",
                  "source": "string",
                  "storageFunctionality": "string",
```

```
"name": "string",
                    "categoryType": 0,
                    "protocolType": 0,
                    "vendorType": 0
                  },
                  "storageStatus": {
                    "message": "string",
                    "statusType": 0
                  },
                  "workflowId": "string",
                  "lastUpdated": "2019-08-24T14:15:22Z",
                  "tenantId": "string"
                }
              },
              "movedBlockedStatistics": {
                "total": 0,
                "destinationStorageInformation": {
                  "id": "string",
                  "name": "string",
                  "groupId": "string",
                  "clientId": "string",
                  "accountId": "string",
                  "source": "string",
                  "storageFunctionality": "string",
                  "storageType": {
                    "name": "string",
                    "categoryType": 0,
                    "protocolType": 0,
                    "vendorType": 0
                  },
                  "storageStatus": {
                    "message": "string",
                    "statusType": 0
                  },
                  "workflowId": "string",
                  "lastUpdated": "2019-08-24T14:15:22Z",
                  "tenantId": "string"
                }
              },
              "copyAllowedStatistics": {
                "total": 0,
                "destinationStorageInformation": {
                  "id": "string",
                  "name": "string",
                  "groupId": "string",
                  "clientId": "string",
                  "accountId": "string",
                  "source": "string",
                  "storageFunctionality": "string",
                  "storageType": {
                    "name": "string",
                    "categoryType": 0,
                    "protocolType": 0,
```

```
},
                  "storageStatus": {
                    "message": "string",
                    "statusType": 0
                  },
                  "workflowId": "string",
                  "lastUpdated": "2019-08-24T14:15:22Z",
                  "tenantId": "string"
                }
              },
              "copyBlockedStatistics": {
                "total": 0,
                "destinationStorageInformation": {
                  "id": "string",
                  "name": "string",
                  "groupId": "string",
                  "clientId": "string",
                  "accountId": "string",
                  "source": "string",
                  "storageFunctionality": "string",
                  "storageType": {
                    "name": "string",
                    "categoryType": 0,
                    "protocolType": 0,
                    "vendorType": 0
                  },
                  "storageStatus": {
                    "message": "string",
                    "statusType": 0
                  },
                  "workflowId": "string",
                  "lastUpdated": "2019-08-24T14:15:22Z",
                  "tenantId": "string"
                }
              },
              "deleteAllowedStatistics": {
                "total": 0
              },
              "deleteBlockedStatistics": {
                "total": 0
              },
              "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
            },
            "account": {
              "id": "string",
              "name": "string",
              "description": "string",
              "credentials": "string",
              "protocolType": 0,
              "vendorType": 0,
              "autoDiscovery": true,
              "autodiscoveryDestinationGroupId": "string"
            },
```

```
},
          "versions": [],
          "hasBlockedVersions": true,
          "versionsCount": 0,
          "objectType": 0,
          "skippedScan": true,
          "$type": "string",
          "url": "string",
          "fileId": "string"
        }
      ],
      "hasBlockedVersions": true,
      "versionsCount": 0,
      "objectType": 0,
      "skippedScan": true,
      "$type": "string"
    }
  ],
  "filterCount": 0,
  "totalCount": 0
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	EnumerateFiles

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Rescan multiple files on demand

`PUT /api/file/rescan`

This request is used to rescan and update existing files from a scan based on a filter.

> Body parameter

```
{
  "scanId": "string",
  "workflowId": "string",
  "passwords": [
    "string"
  ],
  "filterParameters": {
    "expand": [
      "string"
    ],
    "startDate": "2019-08-24T14:15:22Z",
    "endDate": "2019-08-24T14:15:22Z",
    "search": "string",
    "scanType": [
      0
    ],
    "storageId": [
      "string"
    ],
    "scanId": [
      "string"
    ],
    "scanWorkflowSnapshotId": [
      "string"
    ],
    "processingState": [
      0
    ],
    "result": [
      0
    ],
    "scanAllResultI": [
      0
    ],
    "scanResultICount": [
      0
    ],
    "sandboxVerdict": [
      0
    ],
    "vulnerabilityCount": [
      0
    ],
    "hasVulnerabilities": true,
    "dlpDetection": true,
    "hasBlockedFileVersions": true,
    "remediations": [
      0
    ],
    "skippedScan": true,
    "sortBy": [
      0
```

```
"sortOrder": [
      0
    ]
  },
  "storageProtocolType": 0
}
```

Parameters

Name	In	Type	Required	Description
body	body	OnDemandFilteredScanParameters	true	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	OnDemandScanCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch processing results for a file

`GET /api/file/{fileId}`

File processing is done asynchronously and each analysis request is tracked by a file ID. Because processing a file is a potentially time-consuming operation, scheduling a file for processing and retrieving the results needs to be done using two separate API calls.

This request needs to be made multiple times until the analysis is complete. Analysis completion can be tracked using the processingState and the progress values from the response...

Parameters

Name	In	Type	Required	Description
fileId	path	string	true	none
expand	query	array[string]	false	Possible values for expand: workflow_snapshot

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "id": "string",
  "scanId": "string",
  "name": "string",
  "path": "string",
  "hash": "string",
  "fileOwner": {
    "userId": "string",
    "fullName": "string",
    "userName": "string",
    "emailAddress": "string"
  },
  "size": 0,
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageName": "string",
  "storageLocation": "string",
  "scanName": "string",
  "progress": {
    "upload": 0,
    "scan": 0
  },
  "scanResult": {
    "extractedFiles": {
      "dataId": "string",
      "scanResultI": 0,
      "detectedBy": 0,
      "filesExtractedCount": 0
    },
    "scanResults": {
      "scanDetails": [
        {
          "engineName": "string",
          "threatFound": "string",
          "scanResultI": 0,
          "defTime": "string",
          "scanTime": 0.1,
          "engId": "string",
          "waitTime": 0.1
        }
      ],
```

```
"scanAllResultI": 0,
      "scanAllResultA": "string",
      "startTime": "string",
      "totalTime": 0.1,
      "totalAvs": 0,
      "totalDetections": 0,
      "progressPercentage": 0
    },
    "fileInfo": {
      "fileSize": 0,
      "uploadTimestamp": "string",
      "md5": "string",
      "sha1": "string",
      "sha256": "string",
      "fileTypeCategory": "string",
      "fileTypeDescription": "string",
      "fileTypeExtension": "string",
      "displayName": "string",
      "fileTypeId": "string"
    },
    "fileScanInfo": {
      "finalVerdict": {
        "blocked": true,
        "verdict": "string",
        "confidence": 0.1,
        "threatLevel": 0.1
      },
      "overviewReport": {
        "analysisState": "string",
        "signalGroups": [
          {
            "description": "string",
            "verdict": "string",
            "signals": [
              {
                "originType": "string",
                "signalReadable": "string",
                "strength": 0.1
              }
            ]
          }
        ]
      },
      "errorCategory": "string",
      "errorCode": 0,
      "errorDetails": "string"
    },
    "processInfo": {
      "postProcessing": {
        "actionsRan": "string",
        "actionsFailed": "string",
        "convertedTo": "string",
        "copyMoveDestination": "string",
```

```
"sanitizationDetails": {
          "failureCategory": "string",
          "details": "string"
        }
      },
      "progressPercentage": 0,
      "userAgent": "string",
      "profile": "string",
      "result": "string",
      "blockedReason": "string",
      "fileTypeSkippedScan": true,
      "processingTime": 0.1,
      "queueTime": 0.1,
      "resultI": 0
    },
    "dataId": "string",
    "dlpInfo": {
      "hits": null,
      "verdict": 0
    },
    "sbomInfo": {
      "finalVerdict": {
        "blocked": true,
        "licensingPackageCounts": {
          "allowed": 0,
          "blocked": 0,
          "unknown": 0,
          "unsupported": 0
        },
        "severity": 0,
        "totalPackageCount": 0,
        "verdict": "string",
        "vulnerabilityCounts": {
          "critical": 0,
          "high": 0,
          "medium": 0,
          "low": 0,
          "unknown": 0
        }
      },
      "overviewReport": {
        "isPartialReport": true,
        "osInfo": {
          "osName": "string",
          "osVersion": "string"
        },
        "scanMode": "string",
        "steps": [
          {
            "command": "string",
            "packages": [
              {
                "authors": "string",
```

```
"ecosystem": "string",
                "group": "string",
                "archive": true,
                "executable": true,
                "uid": "string",
                "licenses": {
                  "allowed": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ],
                  "blocked": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ]
                },
                "packageName": "string",
                "packageVersion": "string",
                "hashes": [
                  {
                    "sha256": "string",
                    "fileName": "string"
                  }
                ],
                "vulnerabilityCounts": {
                  "critical": 0,
```

```
"medium": 0,
                  "low": 0,
                  "unknown": 0
                },
                "vulnerabilities": [
                  {
                    "id": "string",
                    "severity": 0,
                    "source": "string",
                    "fixedVersions": "string",
                    "cwes": [
                      "string"
                    ]
                  }
                ],
                "arch": "string",
                "osName": "string",
                "osVersion": "string",
                "dependencies": [
                  {
                    "ecosystem": "string",
                    "group": "string",
                    "packageName": "string",
                    "packageVersion": "string",
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    }
                  }
                ]
              }
            ],
            "licenseStatus": "string",
            "licensingPackageCounts": {
              "allowed": 0,
              "blocked": 0,
              "unknown": 0,
              "unsupported": 0
            },
```

```
"vulnerabilityCount": 0,
            "sha256": "string",
            "severity": 0,
            "vulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            }
          }
        ],
        "packages": [
          {
            "authors": "string",
            "releaseDate": "2019-08-24T14:15:22Z",
            "ecosystem": "string",
            "group": "string",
            "archive": true,
            "executable": true,
            "uid": "string",
            "licenses": {
              "allowed": [
                {
                  "id": "string",
                  "url": "string",
                  "category": "string",
                  "usageRestriction": {
                    "permissions": [
                      "string"
                    ],
                    "conditions": [
                      "string"
                    ],
                    "limitations": [
                      "string"
                    ]
                  }
                }
              ],
              "blocked": [
                {
                  "id": "string",
                  "url": "string",
                  "category": "string",
                  "usageRestriction": {
                    "permissions": [
                      "string"
                    ],
                    "conditions": [
                      "string"
                    ],
                    "limitations": [
```

```
]
                    }
                  }
                ]
              },
              "packageName": "string",
              "packageVersion": "string",
              "target": "string",
              "hashes": [
                {
                  "sha256": "string",
                  "fileName": "string"
                }
              ],
              "vulnerabilities": [
                {
                  "id": "string",
                  "severity": 0,
                  "source": "string",
                  "fixedVersions": "string",
                  "cwes": [
                    "string"
                  ]
                }
              ],
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              },
              "dependencies": [
                {
                  "ecosystem": "string",
                  "group": "string",
                  "packageName": "string",
                  "packageVersion": "string",
                  "vulnerabilities": [
                    {
                      "id": "string",
                      "severity": 0,
                      "source": "string",
                      "fixedVersions": "string",
                      "cwes": [
                        "string"
                      ]
                    }
                  ],
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
```

```
"unknown": 0
                }
              }
            ]
          }
        ],
        "dependenciesPackageCount": 0,
        "dependenciesVulnerabilityTotal": 0,
        "dependenciesVulnerabilityCounts": {
          "critical": 0,
          "high": 0,
          "medium": 0,
          "low": 0,
          "unknown": 0
        }
      },
      "resultTemplateHash": "string"
    },
    "vulnerabilityInfo": {
      "vulnerabilityInfoResult": {
        "code": 0,
        "hash": "string",
        "method": 0,
        "timeStamp": "string",
        "timing": 0,
        "detectedProduct": {
          "hasVulnerability": true,
          "isCurrent": true,
          "product": {
            "id": 0,
            "name": "string"
          },
          "remediationLink": "string",
          "severity": "string",
          "sigName": "string",
          "signature": 0,
          "vendor": {
            "id": 0,
            "name": "string"
          },
          "version": "string",
          "versionData": {
            "countBehind": 0,
            "feedId": 0,
            "version": "string"
          },
          "vulnerabilities": [
            {
              "description": "string",
              "details": {
                "cpe": "string",
                "cve": "string",
                "cvss": {
```

```
"accessVector": "string",
                  "authentication": "string",
                  "availabilityImpact": "string",
                  "confidentialityImpact": "string",
                  "generatedOnEpoch": "string",
                  "integrityImpact": "string",
                  "score": "string",
                  "source": "string"
                },
                "cvss3": {
                  "attackComplexity": "string",
                  "attackVector": "string",
                  "availabilityImpact": "string",
                  "baseScore": "string",
                  "baseSeverity": "string",
                  "confidentialityImpact": "string",
                  "exploitabilityScore": "string",
                  "impactScore": "string",
                  "integrityImpact": "string",
                  "privilegesRequired": "string",
                  "scope": "string",
                  "userInteraction": "string",
                  "vectorString": "string"
                },
                "cwe": "string",
                "lastModifiedEpoch": "string",
                "publishedEpoch": "string",
                "references": [
                  "string"
                ]
              },
              "severity": "string",
              "severityIndex": 0,
              "staticId": 0
            }
          ]
        }
      }
    },
    "sessionCookie": "string",
    "fileTypeInfo": {
      "fileInfoExtended": {
        "extensions": "string",
        "groupId": "string",
        "typeId": "string"
      }
    },
    "scanInstanceUrl": "string",
    "scanWorkflowSnapshotId": "string",
    "coreEngines": [
      {
        "engineId": "string",
        "engineName": "string",
```

```
"dbVersion": "string",
        "definitionTime": "string"
      }
    ]
  },
  "processingState": 0,
  "exception": {
    "exceptionDetails": "string",
    "exceptionMessage": "string"
  },
  "created": "2019-08-24T14:15:22Z",
  "lastModified": "2019-08-24T14:15:22Z",
  "discoveryStarted": "2019-08-24T14:15:22Z",
  "fileProcessingDetails": {
    "streamingStartedTime": "2019-08-24T14:15:22Z",
    "streamingCompletedTime": "2019-08-24T14:15:22Z",
    "streamingTimeSpanMs": 0,
    "scanCompletedTime": "2019-08-24T14:15:22Z",
    "processingTimeSpanMs": 0
  },
  "coreEngines": [
    {
      "engineId": "string",
      "engineName": "string",
      "engineVersion": "string",
      "dbVersion": "string",
      "definitionTime": "string"
    }
  ],
  "remediation": {
    "isTagged": true,
    "isDeleted": true,
    "isMoved": true,
    "isCopied": true,
    "deepCdr": {
      "sanitizationDetails": {
        "failureCategory": "string",
        "details": "string"
      },
      "cdrObjectState": 0,
      "fileWasReplaced": true
    },
    "postActionFailedInformation": [
      {
        "objectActionType": 0,
        "exception": {
          "exceptionDetails": "string",
          "exceptionMessage": "string"
        }
      }
    ],
    "objectRemediationActions": [
      {
```

```
"objectActionState": 0,
        "cdrObjectState": 0,
        "exceptions": [
          "string"
        ],
        "id": "string",
        "fileId": "string",
        "fileScanCompletedTime": "2019-08-24T14:15:22Z",
        "remediationId": "string",
        "tenantId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z"
      }
    ]
  },
  "metaDefenderCoreUrl": "string",
  "discoveryStartedDaySec": 0,
  "fileDto": {
    "result": 0,
    "responseKey": "string",
    "responseMessage": "string",
    "responseMessageParams": {
      "property1": "string",
      "property2": "string"
    },
    "id": "string",
    "scanId": "string",
    "name": "string",
    "path": "string",
    "hash": "string",
    "fileOwner": {
      "userId": "string",
      "fullName": "string",
      "userName": "string",
      "emailAddress": "string"
    },
    "size": 0,
    "fileTypeCategory": 0,
    "extension": "string",
    "storageType": {
      "name": "string",
      "categoryType": 0,
      "protocolType": 0,
      "vendorType": 0
    },
    "storageName": "string",
    "storageLocation": "string",
    "scanName": "string",
    "progress": {
      "upload": 0,
      "scan": 0
    },
    "scanResult": {
      "extractedFiles": {
```

```
"scanResultI": 0,
        "detectedBy": 0,
        "filesExtractedCount": 0
      },
      "scanResults": {
        "scanDetails": [
          {
            "engineName": "string",
            "threatFound": "string",
            "scanResultI": 0,
            "defTime": "string",
            "scanTime": 0.1,
            "engId": "string",
            "waitTime": 0.1
          }
        ],
        "dataId": "string",
        "scanAllResultI": 0,
        "scanAllResultA": "string",
        "startTime": "string",
        "totalTime": 0.1,
        "totalAvs": 0,
        "totalDetections": 0,
        "progressPercentage": 0
      },
      "fileInfo": {
        "fileSize": 0,
        "uploadTimestamp": "string",
        "md5": "string",
        "sha1": "string",
        "sha256": "string",
        "fileTypeCategory": "string",
        "fileTypeDescription": "string",
        "fileTypeExtension": "string",
        "displayName": "string",
        "fileTypeId": "string"
      },
      "fileScanInfo": {
        "finalVerdict": {
          "blocked": true,
          "verdict": "string",
          "confidence": 0.1,
          "threatLevel": 0.1
        },
        "overviewReport": {
          "analysisState": "string",
          "signalGroups": [
            {
              "description": "string",
              "verdict": "string",
              "signals": [
                {
                  "originType": "string",
```

```
"strength": 0.1
                }
              ]
            }
          ]
        },
        "errorCategory": "string",
        "errorCode": 0,
        "errorDetails": "string"
      },
      "processInfo": {
        "postProcessing": {
          "actionsRan": "string",
          "actionsFailed": "string",
          "convertedTo": "string",
          "copyMoveDestination": "string",
          "convertedDestination": "string",
          "sanitizationDetails": {
            "failureCategory": "string",
            "details": "string"
          }
        },
        "progressPercentage": 0,
        "userAgent": "string",
        "profile": "string",
        "result": "string",
        "blockedReason": "string",
        "fileTypeSkippedScan": true,
        "processingTime": 0.1,
        "queueTime": 0.1,
        "resultI": 0
      },
      "dataId": "string",
      "dlpInfo": {
        "hits": null,
        "verdict": 0
      },
      "sbomInfo": {
        "finalVerdict": {
          "blocked": true,
          "licensingPackageCounts": {
            "allowed": 0,
            "blocked": 0,
            "unknown": 0,
            "unsupported": 0
          },
          "severity": 0,
          "totalPackageCount": 0,
          "verdict": "string",
          "vulnerabilityCounts": {
            "critical": 0,
            "high": 0,
            "medium": 0,
```

```
"unknown": 0
          }
        },
        "overviewReport": {
          "isPartialReport": true,
          "osInfo": {
            "osName": "string",
            "osVersion": "string"
          },
          "scanMode": "string",
          "steps": [
            {
              "command": "string",
              "packages": [
                {
                  "authors": "string",
                  "releaseDate": "2019-08-24T14:15:22Z",
                  "ecosystem": "string",
                  "group": "string",
                  "archive": true,
                  "executable": true,
                  "uid": "string",
                  "licenses": {
                    "allowed": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            "string"
                          ],
                          "conditions": [
                            "string"
                          ],
                          "limitations": [
                            "string"
                          ]
                        }
                      }
                    ],
                    "blocked": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            "string"
                          ],
                          "conditions": [
                            "string"
                          ],
```

```
"string"
                            ]
                          }
                        }
                      ]
                    },
                    "packageName": "string",
                    "packageVersion": "string",
                    "hashes": [
                      {
                        "sha256": "string",
                        "fileName": "string"
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    },
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "arch": "string",
                    "osName": "string",
                    "osVersion": "string",
                    "dependencies": [
                      {
                        "ecosystem": "string",
                        "group": "string",
                        "packageName": "string",
                        "packageVersion": "string",
                        "vulnerabilities": [
                          {
                            "id": "string",
                            "severity": 0,
                            "source": "string",
                            "fixedVersions": "string",
                            "cwes": [
                              "string"
                            ]
                          }
                        ],
                        "vulnerabilityCounts": {
```

```
"high": 0,
                        "medium": 0,
                        "low": 0,
                        "unknown": 0
                      }
                    }
                  ]
                }
              ],
              "licenseStatus": "string",
              "licensingPackageCounts": {
                "allowed": 0,
                "blocked": 0,
                "unknown": 0,
                "unsupported": 0
              },
              "step": 0,
              "vulnerabilityCount": 0,
              "sha256": "string",
              "severity": 0,
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              }
            }
          ],
          "packages": [
            {
              "authors": "string",
              "releaseDate": "2019-08-24T14:15:22Z",
              "ecosystem": "string",
              "group": "string",
              "archive": true,
              "executable": true,
              "uid": "string",
              "licenses": {
                "allowed": [
                  {
                    "id": "string",
                    "url": "string",
                    "category": "string",
                    "usageRestriction": {
                      "permissions": [
                        "string"
                      ],
                      "conditions": [
                        "string"
                      ],
                      "limitations": [
                        "string"
```

```
}
                    }
                  ],
                  "blocked": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ]
                },
                "packageName": "string",
                "packageVersion": "string",
                "target": "string",
                "hashes": [
                  {
                    "sha256": "string",
                    "fileName": "string"
                  }
                ],
                "vulnerabilities": [
                  {
                    "id": "string",
                    "severity": 0,
                    "source": "string",
                    "fixedVersions": "string",
                    "cwes": [
                      "string"
                    ]
                  }
                ],
                "vulnerabilityCounts": {
                  "critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
                },
                "dependencies": [
                  {
                    "ecosystem": "string",
                    "group": "string",
```

```
"packageVersion": "string",
                  "vulnerabilities": [
                    {
                      "id": "string",
                      "severity": 0,
                      "source": "string",
                      "fixedVersions": "string",
                      "cwes": [
                        "string"
                      ]
                    }
                  ],
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  }
                }
              ]
            }
          ],
          "dependenciesPackageCount": 0,
          "dependenciesVulnerabilityTotal": 0,
          "dependenciesVulnerabilityCounts": {
            "critical": 0,
            "high": 0,
            "medium": 0,
            "low": 0,
            "unknown": 0
          }
        },
        "resultTemplateHash": "string"
      },
      "vulnerabilityInfo": {
        "vulnerabilityInfoResult": {
          "code": 0,
          "hash": "string",
          "method": 0,
          "timeStamp": "string",
          "timing": 0,
          "detectedProduct": {
            "hasVulnerability": true,
            "isCurrent": true,
            "product": {
              "id": 0,
              "name": "string"
            },
            "remediationLink": "string",
            "severity": "string",
            "sigName": "string",
            "signature": 0,
```

```
"id": 0,
              "name": "string"
            },
            "version": "string",
            "versionData": {
              "countBehind": 0,
              "feedId": 0,
              "version": "string"
            },
            "vulnerabilities": [
              {
                "description": "string",
                "details": {
                  "cpe": "string",
                  "cve": "string",
                  "cvss": {
                    "accessComplexity": "string",
                    "accessVector": "string",
                    "authentication": "string",
                    "availabilityImpact": "string",
                    "confidentialityImpact": "string",
                    "generatedOnEpoch": "string",
                    "integrityImpact": "string",
                    "score": "string",
                    "source": "string"
                  },
                  "cvss3": {
                    "attackComplexity": "string",
                    "attackVector": "string",
                    "availabilityImpact": "string",
                    "baseScore": "string",
                    "baseSeverity": "string",
                    "confidentialityImpact": "string",
                    "exploitabilityScore": "string",
                    "impactScore": "string",
                    "integrityImpact": "string",
                    "privilegesRequired": "string",
                    "scope": "string",
                    "userInteraction": "string",
                    "vectorString": "string"
                  },
                  "cwe": "string",
                  "lastModifiedEpoch": "string",
                  "publishedEpoch": "string",
                  "references": [
                    "string"
                  ]
                },
                "severity": "string",
                "severityIndex": 0,
                "staticId": 0
              }
            ]
```

```
}
      },
      "sessionCookie": "string",
      "fileTypeInfo": {
        "fileInfoExtended": {
          "extensions": "string",
          "groupId": "string",
          "typeId": "string"
        }
      },
      "scanInstanceUrl": "string",
      "scanWorkflowSnapshotId": "string",
      "coreEngines": [
        {
          "engineId": "string",
          "engineName": "string",
          "engineVersion": "string",
          "dbVersion": "string",
          "definitionTime": "string"
        }
      ]
    },
    "processingState": 0,
    "exception": {
      "exceptionDetails": "string",
      "exceptionMessage": "string"
    },
    "created": "2019-08-24T14:15:22Z",
    "lastModified": "2019-08-24T14:15:22Z",
    "discoveryStarted": "2019-08-24T14:15:22Z",
    "fileProcessingDetails": {
      "streamingStartedTime": "2019-08-24T14:15:22Z",
      "streamingCompletedTime": "2019-08-24T14:15:22Z",
      "streamingTimeSpanMs": 0,
      "scanCompletedTime": "2019-08-24T14:15:22Z",
      "processingTimeSpanMs": 0
    },
    "coreEngines": [
      {
        "engineId": "string",
        "engineName": "string",
        "engineVersion": "string",
        "dbVersion": "string",
        "definitionTime": "string"
      }
    ],
    "remediation": {
      "isTagged": true,
      "isDeleted": true,
      "isMoved": true,
      "isCopied": true,
      "deepCdr": {
        "sanitizationDetails": {
```

```
"details": "string"
        },
        "cdrObjectState": 0,
        "fileWasReplaced": true
      },
      "postActionFailedInformation": [
        {
          "objectActionType": 0,
          "exception": {
            "exceptionDetails": "string",
            "exceptionMessage": "string"
          }
        }
      ],
      "objectRemediationActions": [
        {
          "objectActionType": 0,
          "objectActionState": 0,
          "cdrObjectState": 0,
          "exceptions": [
            "string"
          ],
          "id": "string",
          "fileId": "string",
          "fileScanCompletedTime": "2019-08-24T14:15:22Z",
          "remediationId": "string",
          "tenantId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z"
        }
      ]
    },
    "metaDefenderCoreUrl": "string",
    "discoveryStartedDaySec": 0,
    "objectStorageMetadata": "string",
    "scanWorkflowSnapshotId": "string",
    "scanWorkflowSnapshot": {
      "id": "string",
      "scanId": "string",
      "workflow": {
        "id": "string",
        "scanPoolId": "string",
        "scanProcessFlow": 0,
        "isIdentityScanningEnabled": true,
        "scanIntervalDays": 0,
        "rules": [
          "string"
        ],
        "type": 0,
        "workflowSupportedTechnologies": {
          "dlp": {
            "status": 0
          },
          "deepCdr": {
```

```
},
          "sandBox": {
            "status": 0
          },
          "metascan": {
            "status": 0
          }
        },
        "name": "string",
        "userAgent": "string",
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanPoolId": "string",
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "filters": {
          "id": "string",
          "minimumSize": 0,
          "maximumSize": 0,
          "minimumDateTime": "2019-08-24T14:15:22Z",
          "maximumDateTime": "2019-08-24T14:15:22Z"
        },
        "fileDefinition": {
          "blockingStrategy": 0,
          "configurableBlockingStrategy": {
```

```
"shouldTreatFilesWithVulnerabilitiesAsBlocked": true
          }
        },
        "remediations": [
          {
            "id": "string",
            "workflowId": "string",
            "isEnabled": true,
            "configuration": "string",
            "type": 0,
            "handlingType": 0
          }
        ],
        "tenantId": "string"
      },
      "startTimestamp": 0,
      "stopTimestamp": 0,
      "scanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "failOverScanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
```

```
"timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "failOverScanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "storage": {
        "id": "string",
        "name": "string",
        "groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      },
      "workflowStatistics": {
        "taggingStatistics": {
          "total": 0
        },
        "keepAllowedStatistics": {
          "total": 0
        },
        "keepBlockedStatistics": {
          "total": 0
        },
        "keepSanitizedStatistics": {
          "total": 0
        },
        "movedAllowedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
```

```
"name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "movedSanitizedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "movedBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
```

```
"source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "copyAllowedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "copyBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
```

```
"categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "deleteAllowedStatistics": {
          "total": 0
        },
        "deleteBlockedStatistics": {
          "total": 0
        },
        "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
      },
      "account": {
        "id": "string",
        "name": "string",
        "description": "string",
        "credentials": "string",
        "protocolType": 0,
        "vendorType": 0,
        "autoDiscovery": true,
        "autodiscoveryDestinationGroupId": "string"
      },
      "realTimeStartDate": "2019-08-24T14:15:22Z"
    },
    "versions": [
      {
        "result": 0,
        "responseKey": "string",
        "responseMessage": "string",
        "responseMessageParams": {
          "property1": "string",
          "property2": "string"
        },
        "id": "string",
        "scanId": "string",
        "name": "string",
        "path": "string",
        "hash": "string",
        "fileOwner": {
          "userId": "string",
          "fullName": "string",
          "userName": "string",
          "emailAddress": "string"
        },
```

```
"fileTypeCategory": 0,
        "extension": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageName": "string",
        "storageLocation": "string",
        "scanName": "string",
        "progress": {
          "upload": 0,
          "scan": 0
        },
        "scanResult": {
          "extractedFiles": {
            "dataId": "string",
            "scanResultI": 0,
            "detectedBy": 0,
            "filesExtractedCount": 0
          },
          "scanResults": {
            "scanDetails": [
              {
                "engineName": "string",
                "threatFound": "string",
                "scanResultI": 0,
                "defTime": "string",
                "scanTime": 0.1,
                "engId": "string",
                "waitTime": 0.1
              }
            ],
            "dataId": "string",
            "scanAllResultI": 0,
            "scanAllResultA": "string",
            "startTime": "string",
            "totalTime": 0.1,
            "totalAvs": 0,
            "totalDetections": 0,
            "progressPercentage": 0
          },
          "fileInfo": {
            "fileSize": 0,
            "uploadTimestamp": "string",
            "md5": "string",
            "sha1": "string",
            "sha256": "string",
            "fileTypeCategory": "string",
            "fileTypeDescription": "string",
            "fileTypeExtension": "string",
            "displayName": "string",
```

```
},
          "fileScanInfo": {
            "finalVerdict": {
              "blocked": true,
              "verdict": "string",
              "confidence": 0.1,
              "threatLevel": 0.1
            },
            "overviewReport": {
              "analysisState": "string",
              "signalGroups": [
                {
                  "description": "string",
                  "verdict": "string",
                  "signals": [
                    {
                      "originType": "string",
                      "signalReadable": "string",
                      "strength": 0.1
                    }
                  ]
                }
              ]
            },
            "errorCategory": "string",
            "errorCode": 0,
            "errorDetails": "string"
          },
          "processInfo": {
            "postProcessing": {
              "actionsRan": "string",
              "actionsFailed": "string",
              "convertedTo": "string",
              "copyMoveDestination": "string",
              "convertedDestination": "string",
              "sanitizationDetails": {
                "failureCategory": "string",
                "details": "string"
              }
            },
            "progressPercentage": 0,
            "userAgent": "string",
            "profile": "string",
            "result": "string",
            "blockedReason": "string",
            "fileTypeSkippedScan": true,
            "processingTime": 0.1,
            "queueTime": 0.1,
            "resultI": 0
          },
          "dataId": "string",
          "dlpInfo": {
            "hits": null,
```

```
},
          "sbomInfo": {
            "finalVerdict": {
              "blocked": true,
              "licensingPackageCounts": {
                "allowed": 0,
                "blocked": 0,
                "unknown": 0,
                "unsupported": 0
              },
              "severity": 0,
              "totalPackageCount": 0,
              "verdict": "string",
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              }
            },
            "overviewReport": {
              "isPartialReport": true,
              "osInfo": {
                "osName": "string",
                "osVersion": "string"
              },
              "scanMode": "string",
              "steps": [
                {
                  "command": "string",
                  "packages": [
                    {
                      "authors": "string",
                      "releaseDate": "2019-08-24T14:15:22Z",
                      "ecosystem": "string",
                      "group": "string",
                      "archive": true,
                      "executable": true,
                      "uid": "string",
                      "licenses": {
                        "allowed": [
                          {
                            "id": "string",
                            "url": "string",
                            "category": "string",
                            "usageRestriction": {
                              "permissions": [],
                              "conditions": [],
                              "limitations": []
                            }
                          }
                        ],
```

```
{
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [],
                          "conditions": [],
                          "limitations": []
                        }
                      }
                    ]
                  },
                  "packageName": "string",
                  "packageVersion": "string",
                  "hashes": [
                    {
                      "sha256": "string",
                      "fileName": "string"
                    }
                  ],
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  },
                  "vulnerabilities": [
                    {
                      "id": "string",
                      "severity": 0,
                      "source": "string",
                      "fixedVersions": "string",
                      "cwes": [
                        "string"
                      ]
                    }
                  ],
                  "arch": "string",
                  "osName": "string",
                  "osVersion": "string",
                  "dependencies": [
                    {
                      "ecosystem": "string",
                      "group": "string",
                      "packageName": "string",
                      "packageVersion": "string",
                      "vulnerabilities": [
                        {
                          "cwes": []
                        }
                      ],
                      "vulnerabilityCounts": {
```

```
"high": 0,
                            "medium": 0,
                            "low": 0,
                            "unknown": 0
                          }
                        }
                      ]
                    }
                  ],
                  "licenseStatus": "string",
                  "licensingPackageCounts": {
                    "allowed": 0,
                    "blocked": 0,
                    "unknown": 0,
                    "unsupported": 0
                  },
                  "step": 0,
                  "vulnerabilityCount": 0,
                  "sha256": "string",
                  "severity": 0,
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  }
                }
              ],
              "packages": [
                {
                  "authors": "string",
                  "releaseDate": "2019-08-24T14:15:22Z",
                  "ecosystem": "string",
                  "group": "string",
                  "archive": true,
                  "executable": true,
                  "uid": "string",
                  "licenses": {
                    "allowed": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            "string"
                          ],
                          "conditions": [
                            "string"
                          ],
                          "limitations": [
                            "string"
```

```
}
                    }
                  ],
                  "blocked": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ]
                },
                "packageName": "string",
                "packageVersion": "string",
                "target": "string",
                "hashes": [
                  {
                    "sha256": "string",
                    "fileName": "string"
                  }
                ],
                "vulnerabilities": [
                  {
                    "id": "string",
                    "severity": 0,
                    "source": "string",
                    "fixedVersions": "string",
                    "cwes": [
                      "string"
                    ]
                  }
                ],
                "vulnerabilityCounts": {
                  "critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
                },
                "dependencies": [
                  {
                    "ecosystem": "string",
                    "group": "string",
```

```
"packageVersion": "string",
                      "vulnerabilities": [
                        {
                          "id": "string",
                          "severity": 0,
                          "source": "string",
                          "fixedVersions": "string",
                          "cwes": [
                            "string"
                          ]
                        }
                      ],
                      "vulnerabilityCounts": {
                        "critical": 0,
                        "high": 0,
                        "medium": 0,
                        "low": 0,
                        "unknown": 0
                      }
                    }
                  ]
                }
              ],
              "dependenciesPackageCount": 0,
              "dependenciesVulnerabilityTotal": 0,
              "dependenciesVulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              }
            },
            "resultTemplateHash": "string"
          },
          "vulnerabilityInfo": {
            "vulnerabilityInfoResult": {
              "code": 0,
              "hash": "string",
              "method": 0,
              "timeStamp": "string",
              "timing": 0,
              "detectedProduct": {
                "hasVulnerability": true,
                "isCurrent": true,
                "product": {
                  "id": 0,
                  "name": "string"
                },
                "remediationLink": "string",
                "severity": "string",
                "sigName": "string",
                "signature": 0,
```

```
"id": 0,
                  "name": "string"
                },
                "version": "string",
                "versionData": {
                  "countBehind": 0,
                  "feedId": 0,
                  "version": "string"
                },
                "vulnerabilities": [
                  {
                    "description": "string",
                    "details": {
                      "cpe": "string",
                      "cve": "string",
                      "cvss": {
                        "accessComplexity": "string",
                        "accessVector": "string",
                        "authentication": "string",
                        "availabilityImpact": "string",
                        "confidentialityImpact": "string",
                        "generatedOnEpoch": "string",
                        "integrityImpact": "string",
                        "score": "string",
                        "source": "string"
                      },
                      "cvss3": {
                        "attackComplexity": "string",
                        "attackVector": "string",
                        "availabilityImpact": "string",
                        "baseScore": "string",
                        "baseSeverity": "string",
                        "confidentialityImpact": "string",
                        "exploitabilityScore": "string",
                        "impactScore": "string",
                        "integrityImpact": "string",
                        "privilegesRequired": "string",
                        "scope": "string",
                        "userInteraction": "string",
                        "vectorString": "string"
                      },
                      "cwe": "string",
                      "lastModifiedEpoch": "string",
                      "publishedEpoch": "string",
                      "references": [
                        "string"
                      ]
                    },
                    "severity": "string",
                    "severityIndex": 0,
                    "staticId": 0
                  }
                ]
```

```
}
          },
          "sessionCookie": "string",
          "fileTypeInfo": {
            "fileInfoExtended": {
              "extensions": "string",
              "groupId": "string",
              "typeId": "string"
            }
          },
          "scanInstanceUrl": "string",
          "scanWorkflowSnapshotId": "string",
          "coreEngines": [
            {
              "engineId": "string",
              "engineName": "string",
              "engineVersion": "string",
              "dbVersion": "string",
              "definitionTime": "string"
            }
          ]
        },
        "processingState": 0,
        "exception": {
          "exceptionDetails": "string",
          "exceptionMessage": "string"
        },
        "created": "2019-08-24T14:15:22Z",
        "lastModified": "2019-08-24T14:15:22Z",
        "discoveryStarted": "2019-08-24T14:15:22Z",
        "fileProcessingDetails": {
          "streamingStartedTime": "2019-08-24T14:15:22Z",
          "streamingCompletedTime": "2019-08-24T14:15:22Z",
          "streamingTimeSpanMs": 0,
          "scanCompletedTime": "2019-08-24T14:15:22Z",
          "processingTimeSpanMs": 0
        },
        "coreEngines": [
          {
            "engineId": "string",
            "engineName": "string",
            "engineVersion": "string",
            "dbVersion": "string",
            "definitionTime": "string"
          }
        ],
        "remediation": {
          "isTagged": true,
          "isDeleted": true,
          "isMoved": true,
          "isCopied": true,
          "deepCdr": {
            "sanitizationDetails": {
```

```
"details": "string"
            },
            "cdrObjectState": 0,
            "fileWasReplaced": true
          },
          "postActionFailedInformation": [
            {
              "objectActionType": 0,
              "exception": {
                "exceptionDetails": "string",
                "exceptionMessage": "string"
              }
            }
          ],
          "objectRemediationActions": [
            {
              "objectActionType": 0,
              "objectActionState": 0,
              "cdrObjectState": 0,
              "exceptions": [
                "string"
              ],
              "id": "string",
              "fileId": "string",
              "fileScanCompletedTime": "2019-08-24T14:15:22Z",
              "remediationId": "string",
              "tenantId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z"
            }
          ]
        },
        "metaDefenderCoreUrl": "string",
        "discoveryStartedDaySec": 0,
        "objectStorageMetadata": "string",
        "scanWorkflowSnapshotId": "string",
        "scanWorkflowSnapshot": {
          "id": "string",
          "scanId": "string",
          "workflow": {
            "id": "string",
            "scanPoolId": "string",
            "scanProcessFlow": 0,
            "isIdentityScanningEnabled": true,
            "scanIntervalDays": 0,
            "rules": [
              "string"
            ],
            "type": 0,
            "workflowSupportedTechnologies": {
              "dlp": {
                "status": 0
              },
              "deepCdr": {
```

```
},
            "sandBox": {
              "status": 0
            },
            "metascan": {
              "status": 0
            }
          },
          "name": "string",
          "userAgent": "string",
          "scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanPoolId": "string",
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "filters": {
            "id": "string",
            "minimumSize": 0,
            "maximumSize": 0,
            "minimumDateTime": "2019-08-24T14:15:22Z",
            "maximumDateTime": "2019-08-24T14:15:22Z"
          },
          "fileDefinition": {
            "blockingStrategy": 0,
            "configurableBlockingStrategy": {
```

```
"shouldTreatFilesWithVulnerabilitiesAsBlocked": true
            }
          },
          "remediations": [
            {
              "id": "string",
              "workflowId": "string",
              "isEnabled": true,
              "configuration": "string",
              "type": 0,
              "handlingType": 0
            }
          ],
          "tenantId": "string"
        },
        "startTimestamp": 0,
        "stopTimestamp": 0,
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
```

```
"timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ],
          "storage": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          },
          "workflowStatistics": {
            "taggingStatistics": {
              "total": 0
            },
            "keepAllowedStatistics": {
              "total": 0
            },
            "keepBlockedStatistics": {
              "total": 0
            },
            "keepSanitizedStatistics": {
              "total": 0
            },
            "movedAllowedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
```

```
"name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "movedSanitizedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "movedBlockedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
```

```
"source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "copyAllowedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "copyBlockedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
```

```
"categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "deleteAllowedStatistics": {
              "total": 0
            },
            "deleteBlockedStatistics": {
              "total": 0
            },
            "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
          },
          "account": {
            "id": "string",
            "name": "string",
            "description": "string",
            "credentials": "string",
            "protocolType": 0,
            "vendorType": 0,
            "autoDiscovery": true,
            "autodiscoveryDestinationGroupId": "string"
          },
          "realTimeStartDate": "2019-08-24T14:15:22Z"
        },
        "versions": [],
        "hasBlockedVersions": true,
        "versionsCount": 0,
        "objectType": 0,
        "skippedScan": true,
        "$type": "string",
        "url": "string",
        "fileId": "string"
      }
    ],
    "hasBlockedVersions": true,
    "versionsCount": 0,
    "objectType": 0,
    "skippedScan": true,
    "$type": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	GetFileResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Retrieve archive scan results

`GET /api/file/archive`

Retrieve the archive scan results using either the scanResultId or the parentId

Parameters

Name	In	Type	Required	Description
scanResultId	query	string	false	none
parentId	query	string	false	none
start	query	integer(int32)	false	none
count	query	integer(int32)	false	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "archiveScanResults": [
    {
      "extractedFiles": {
        "dataId": "string",
        "scanResultI": 0,
        "detectedBy": 0,
        "filesExtractedCount": 0
      },
      "scanResults": {
        "scanDetails": [
          {
            "engineName": "string",
            "threatFound": "string",
            "scanResultI": 0,
            "defTime": "string",
            "scanTime": 0.1,
            "engId": "string",
            "waitTime": 0.1
          }
        ],
        "dataId": "string",
        "scanAllResultI": 0,
        "scanAllResultA": "string",
        "startTime": "string",
        "totalTime": 0.1,
        "totalAvs": 0,
        "totalDetections": 0,
        "progressPercentage": 0
      },
      "fileInfo": {
        "fileSize": 0,
        "uploadTimestamp": "string",
        "md5": "string",
        "sha1": "string",
        "sha256": "string",
        "fileTypeCategory": "string",
        "fileTypeDescription": "string",
        "fileTypeExtension": "string",
        "displayName": "string",
        "fileTypeId": "string"
      },
      "fileScanInfo": {
        "finalVerdict": {
          "blocked": true,
```

```
"confidence": 0.1,
          "threatLevel": 0.1
        },
        "overviewReport": {
          "analysisState": "string",
          "signalGroups": [
            {
              "description": "string",
              "verdict": "string",
              "signals": [
                {
                  "originType": "string",
                  "signalReadable": "string",
                  "strength": 0.1
                }
              ]
            }
          ]
        },
        "errorCategory": "string",
        "errorCode": 0,
        "errorDetails": "string"
      },
      "processInfo": {
        "postProcessing": {
          "actionsRan": "string",
          "actionsFailed": "string",
          "convertedTo": "string",
          "copyMoveDestination": "string",
          "convertedDestination": "string",
          "sanitizationDetails": {
            "failureCategory": "string",
            "details": "string"
          }
        },
        "progressPercentage": 0,
        "userAgent": "string",
        "profile": "string",
        "result": "string",
        "blockedReason": "string",
        "fileTypeSkippedScan": true,
        "processingTime": 0.1,
        "queueTime": 0.1,
        "resultI": 0
      },
      "dataId": "string",
      "dlpInfo": {
        "hits": null,
        "verdict": 0
      },
      "sbomInfo": {
        "finalVerdict": {
          "blocked": true,
```

```
"allowed": 0,
            "blocked": 0,
            "unknown": 0,
            "unsupported": 0
          },
          "severity": 0,
          "totalPackageCount": 0,
          "verdict": "string",
          "vulnerabilityCounts": {
            "critical": 0,
            "high": 0,
            "medium": 0,
            "low": 0,
            "unknown": 0
          }
        },
        "overviewReport": {
          "isPartialReport": true,
          "osInfo": {
            "osName": "string",
            "osVersion": "string"
          },
          "scanMode": "string",
          "steps": [
            {
              "command": "string",
              "packages": [
                {
                  "authors": "string",
                  "releaseDate": "2019-08-24T14:15:22Z",
                  "ecosystem": "string",
                  "group": "string",
                  "archive": true,
                  "executable": true,
                  "uid": "string",
                  "licenses": {
                    "allowed": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            "string"
                          ],
                          "conditions": [
                            "string"
                          ],
                          "limitations": [
                            "string"
                          ]
                        }
                      }
```

```
"blocked": [
                        {
                          "id": "string",
                          "url": "string",
                          "category": "string",
                          "usageRestriction": {
                            "permissions": [
                              "string"
                            ],
                            "conditions": [
                              "string"
                            ],
                            "limitations": [
                              "string"
                            ]
                          }
                        }
                      ]
                    },
                    "packageName": "string",
                    "packageVersion": "string",
                    "hashes": [
                      {
                        "sha256": "string",
                        "fileName": "string"
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    },
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "arch": "string",
                    "osName": "string",
                    "osVersion": "string",
                    "dependencies": [
                      {
                        "ecosystem": "string",
                        "group": "string",
                        "packageName": "string",
```

```
"vulnerabilities": [
                        {
                          "id": "string",
                          "severity": 0,
                          "source": "string",
                          "fixedVersions": "string",
                          "cwes": [
                            "string"
                          ]
                        }
                      ],
                      "vulnerabilityCounts": {
                        "critical": 0,
                        "high": 0,
                        "medium": 0,
                        "low": 0,
                        "unknown": 0
                      }
                    }
                  ]
                }
              ],
              "licenseStatus": "string",
              "licensingPackageCounts": {
                "allowed": 0,
                "blocked": 0,
                "unknown": 0,
                "unsupported": 0
              },
              "step": 0,
              "vulnerabilityCount": 0,
              "sha256": "string",
              "severity": 0,
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              }
            }
          ],
          "packages": [
            {
              "authors": "string",
              "releaseDate": "2019-08-24T14:15:22Z",
              "ecosystem": "string",
              "group": "string",
              "archive": true,
              "executable": true,
              "uid": "string",
              "licenses": {
                "allowed": [
```

```
"id": "string",
                    "url": "string",
                    "category": "string",
                    "usageRestriction": {
                      "permissions": [
                        "string"
                      ],
                      "conditions": [
                        "string"
                      ],
                      "limitations": [
                        "string"
                      ]
                    }
                  }
                ],
                "blocked": [
                  {
                    "id": "string",
                    "url": "string",
                    "category": "string",
                    "usageRestriction": {
                      "permissions": [
                        "string"
                      ],
                      "conditions": [
                        "string"
                      ],
                      "limitations": [
                        "string"
                      ]
                    }
                  }
                ]
              },
              "packageName": "string",
              "packageVersion": "string",
              "target": "string",
              "hashes": [
                {
                  "sha256": "string",
                  "fileName": "string"
                }
              ],
              "vulnerabilities": [
                {
                  "id": "string",
                  "severity": 0,
                  "source": "string",
                  "fixedVersions": "string",
                  "cwes": [
                    "string"
                  ]
```

```
],
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              },
              "dependencies": [
                {
                  "ecosystem": "string",
                  "group": "string",
                  "packageName": "string",
                  "packageVersion": "string",
                  "vulnerabilities": [
                    {
                      "id": "string",
                      "severity": 0,
                      "source": "string",
                      "fixedVersions": "string",
                      "cwes": [
                        "string"
                      ]
                    }
                  ],
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  }
                }
              ]
            }
          ],
          "dependenciesPackageCount": 0,
          "dependenciesVulnerabilityTotal": 0,
          "dependenciesVulnerabilityCounts": {
            "critical": 0,
            "high": 0,
            "medium": 0,
            "low": 0,
            "unknown": 0
          }
        },
        "resultTemplateHash": "string"
      },
      "vulnerabilityInfo": {
        "vulnerabilityInfoResult": {
          "code": 0,
          "hash": "string",
          "method": 0,
```

```
"timing": 0,
          "detectedProduct": {
            "hasVulnerability": true,
            "isCurrent": true,
            "product": {
              "id": 0,
              "name": "string"
            },
            "remediationLink": "string",
            "severity": "string",
            "sigName": "string",
            "signature": 0,
            "vendor": {
              "id": 0,
              "name": "string"
            },
            "version": "string",
            "versionData": {
              "countBehind": 0,
              "feedId": 0,
              "version": "string"
            },
            "vulnerabilities": [
              {
                "description": "string",
                "details": {
                  "cpe": "string",
                  "cve": "string",
                  "cvss": {
                    "accessComplexity": "string",
                    "accessVector": "string",
                    "authentication": "string",
                    "availabilityImpact": "string",
                    "confidentialityImpact": "string",
                    "generatedOnEpoch": "string",
                    "integrityImpact": "string",
                    "score": "string",
                    "source": "string"
                  },
                  "cvss3": {
                    "attackComplexity": "string",
                    "attackVector": "string",
                    "availabilityImpact": "string",
                    "baseScore": "string",
                    "baseSeverity": "string",
                    "confidentialityImpact": "string",
                    "exploitabilityScore": "string",
                    "impactScore": "string",
                    "integrityImpact": "string",
                    "privilegesRequired": "string",
                    "scope": "string",
                    "userInteraction": "string",
                    "vectorString": "string"
```

```
"cwe": "string",
                  "lastModifiedEpoch": "string",
                  "publishedEpoch": "string",
                  "references": [
                    "string"
                  ]
                },
                "severity": "string",
                "severityIndex": 0,
                "staticId": 0
              }
            ]
          }
        }
      },
      "sessionCookie": "string",
      "fileTypeInfo": {
        "fileInfoExtended": {
          "extensions": "string",
          "groupId": "string",
          "typeId": "string"
        }
      },
      "scanInstanceUrl": "string",
      "scanWorkflowSnapshotId": "string",
      "coreEngines": [
        {
          "engineId": "string",
          "engineName": "string",
          "engineVersion": "string",
          "dbVersion": "string",
          "definitionTime": "string"
        }
      ]
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetArchiveScanResultResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Cancel a file in an ongoing scan

`POST /api/file/cancel`

Body parameter

```
{
  "fileIds": [
    "string"
  ]
}
```

Parameters

Name	In	Type	Required	Description
body	body	CancelFileParameters	true	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "id": "string",
  "scanId": "string",
  "name": "string",
  "path": "string",
  "hash": "string",
  "fileOwner": {
    "userId": "string",
    "fullName": "string",
    "userName": "string",
    "emailAddress": "string"
  },
  "size": 0,
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageName": "string",
  "storageLocation": "string",
  "scanName": "string",
  "progress": {
    "upload": 0,
    "scan": 0
  },
  "scanResult": {
    "extractedFiles": {
      "dataId": "string",
      "scanResultI": 0,
      "detectedBy": 0,
      "filesExtractedCount": 0
    },
    "scanResults": {
      "scanDetails": [
        {
          "engineName": "string",
          "threatFound": "string",
          "scanResultI": 0,
          "defTime": "string",
          "scanTime": 0.1,
          "engId": "string",
          "waitTime": 0.1
        }
      ],
```

```
"scanAllResultI": 0,
      "scanAllResultA": "string",
      "startTime": "string",
      "totalTime": 0.1,
      "totalAvs": 0,
      "totalDetections": 0,
      "progressPercentage": 0
    },
    "fileInfo": {
      "fileSize": 0,
      "uploadTimestamp": "string",
      "md5": "string",
      "sha1": "string",
      "sha256": "string",
      "fileTypeCategory": "string",
      "fileTypeDescription": "string",
      "fileTypeExtension": "string",
      "displayName": "string",
      "fileTypeId": "string"
    },
    "fileScanInfo": {
      "finalVerdict": {
        "blocked": true,
        "verdict": "string",
        "confidence": 0.1,
        "threatLevel": 0.1
      },
      "overviewReport": {
        "analysisState": "string",
        "signalGroups": [
          {
            "description": "string",
            "verdict": "string",
            "signals": [
              {
                "originType": "string",
                "signalReadable": "string",
                "strength": 0.1
              }
            ]
          }
        ]
      },
      "errorCategory": "string",
      "errorCode": 0,
      "errorDetails": "string"
    },
    "processInfo": {
      "postProcessing": {
        "actionsRan": "string",
        "actionsFailed": "string",
        "convertedTo": "string",
        "copyMoveDestination": "string",
```

```
"sanitizationDetails": {
          "failureCategory": "string",
          "details": "string"
        }
      },
      "progressPercentage": 0,
      "userAgent": "string",
      "profile": "string",
      "result": "string",
      "blockedReason": "string",
      "fileTypeSkippedScan": true,
      "processingTime": 0.1,
      "queueTime": 0.1,
      "resultI": 0
    },
    "dataId": "string",
    "dlpInfo": {
      "hits": null,
      "verdict": 0
    },
    "sbomInfo": {
      "finalVerdict": {
        "blocked": true,
        "licensingPackageCounts": {
          "allowed": 0,
          "blocked": 0,
          "unknown": 0,
          "unsupported": 0
        },
        "severity": 0,
        "totalPackageCount": 0,
        "verdict": "string",
        "vulnerabilityCounts": {
          "critical": 0,
          "high": 0,
          "medium": 0,
          "low": 0,
          "unknown": 0
        }
      },
      "overviewReport": {
        "isPartialReport": true,
        "osInfo": {
          "osName": "string",
          "osVersion": "string"
        },
        "scanMode": "string",
        "steps": [
          {
            "command": "string",
            "packages": [
              {
                "authors": "string",
```

```
"ecosystem": "string",
                "group": "string",
                "archive": true,
                "executable": true,
                "uid": "string",
                "licenses": {
                  "allowed": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ],
                  "blocked": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ]
                },
                "packageName": "string",
                "packageVersion": "string",
                "hashes": [
                  {
                    "sha256": "string",
                    "fileName": "string"
                  }
                ],
                "vulnerabilityCounts": {
                  "critical": 0,
```

```
"medium": 0,
                  "low": 0,
                  "unknown": 0
                },
                "vulnerabilities": [
                  {
                    "id": "string",
                    "severity": 0,
                    "source": "string",
                    "fixedVersions": "string",
                    "cwes": [
                      "string"
                    ]
                  }
                ],
                "arch": "string",
                "osName": "string",
                "osVersion": "string",
                "dependencies": [
                  {
                    "ecosystem": "string",
                    "group": "string",
                    "packageName": "string",
                    "packageVersion": "string",
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    }
                  }
                ]
              }
            ],
            "licenseStatus": "string",
            "licensingPackageCounts": {
              "allowed": 0,
              "blocked": 0,
              "unknown": 0,
              "unsupported": 0
            },
```

```
"vulnerabilityCount": 0,
            "sha256": "string",
            "severity": 0,
            "vulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            }
          }
        ],
        "packages": [
          {
            "authors": "string",
            "releaseDate": "2019-08-24T14:15:22Z",
            "ecosystem": "string",
            "group": "string",
            "archive": true,
            "executable": true,
            "uid": "string",
            "licenses": {
              "allowed": [
                {
                  "id": "string",
                  "url": "string",
                  "category": "string",
                  "usageRestriction": {
                    "permissions": [
                      "string"
                    ],
                    "conditions": [
                      "string"
                    ],
                    "limitations": [
                      "string"
                    ]
                  }
                }
              ],
              "blocked": [
                {
                  "id": "string",
                  "url": "string",
                  "category": "string",
                  "usageRestriction": {
                    "permissions": [
                      "string"
                    ],
                    "conditions": [
                      "string"
                    ],
                    "limitations": [
```

```
]
                    }
                  }
                ]
              },
              "packageName": "string",
              "packageVersion": "string",
              "target": "string",
              "hashes": [
                {
                  "sha256": "string",
                  "fileName": "string"
                }
              ],
              "vulnerabilities": [
                {
                  "id": "string",
                  "severity": 0,
                  "source": "string",
                  "fixedVersions": "string",
                  "cwes": [
                    "string"
                  ]
                }
              ],
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              },
              "dependencies": [
                {
                  "ecosystem": "string",
                  "group": "string",
                  "packageName": "string",
                  "packageVersion": "string",
                  "vulnerabilities": [
                    {
                      "id": "string",
                      "severity": 0,
                      "source": "string",
                      "fixedVersions": "string",
                      "cwes": [
                        "string"
                      ]
                    }
                  ],
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
```

```
"unknown": 0
                }
              }
            ]
          }
        ],
        "dependenciesPackageCount": 0,
        "dependenciesVulnerabilityTotal": 0,
        "dependenciesVulnerabilityCounts": {
          "critical": 0,
          "high": 0,
          "medium": 0,
          "low": 0,
          "unknown": 0
        }
      },
      "resultTemplateHash": "string"
    },
    "vulnerabilityInfo": {
      "vulnerabilityInfoResult": {
        "code": 0,
        "hash": "string",
        "method": 0,
        "timeStamp": "string",
        "timing": 0,
        "detectedProduct": {
          "hasVulnerability": true,
          "isCurrent": true,
          "product": {
            "id": 0,
            "name": "string"
          },
          "remediationLink": "string",
          "severity": "string",
          "sigName": "string",
          "signature": 0,
          "vendor": {
            "id": 0,
            "name": "string"
          },
          "version": "string",
          "versionData": {
            "countBehind": 0,
            "feedId": 0,
            "version": "string"
          },
          "vulnerabilities": [
            {
              "description": "string",
              "details": {
                "cpe": "string",
                "cve": "string",
                "cvss": {
```

```
"accessVector": "string",
                  "authentication": "string",
                  "availabilityImpact": "string",
                  "confidentialityImpact": "string",
                  "generatedOnEpoch": "string",
                  "integrityImpact": "string",
                  "score": "string",
                  "source": "string"
                },
                "cvss3": {
                  "attackComplexity": "string",
                  "attackVector": "string",
                  "availabilityImpact": "string",
                  "baseScore": "string",
                  "baseSeverity": "string",
                  "confidentialityImpact": "string",
                  "exploitabilityScore": "string",
                  "impactScore": "string",
                  "integrityImpact": "string",
                  "privilegesRequired": "string",
                  "scope": "string",
                  "userInteraction": "string",
                  "vectorString": "string"
                },
                "cwe": "string",
                "lastModifiedEpoch": "string",
                "publishedEpoch": "string",
                "references": [
                  "string"
                ]
              },
              "severity": "string",
              "severityIndex": 0,
              "staticId": 0
            }
          ]
        }
      }
    },
    "sessionCookie": "string",
    "fileTypeInfo": {
      "fileInfoExtended": {
        "extensions": "string",
        "groupId": "string",
        "typeId": "string"
      }
    },
    "scanInstanceUrl": "string",
    "scanWorkflowSnapshotId": "string",
    "coreEngines": [
      {
        "engineId": "string",
        "engineName": "string",
```

```
"dbVersion": "string",
        "definitionTime": "string"
      }
    ]
  },
  "processingState": 0,
  "exception": {
    "exceptionDetails": "string",
    "exceptionMessage": "string"
  },
  "created": "2019-08-24T14:15:22Z",
  "lastModified": "2019-08-24T14:15:22Z",
  "discoveryStarted": "2019-08-24T14:15:22Z",
  "fileProcessingDetails": {
    "streamingStartedTime": "2019-08-24T14:15:22Z",
    "streamingCompletedTime": "2019-08-24T14:15:22Z",
    "streamingTimeSpanMs": 0,
    "scanCompletedTime": "2019-08-24T14:15:22Z",
    "processingTimeSpanMs": 0
  },
  "coreEngines": [
    {
      "engineId": "string",
      "engineName": "string",
      "engineVersion": "string",
      "dbVersion": "string",
      "definitionTime": "string"
    }
  ],
  "remediation": {
    "isTagged": true,
    "isDeleted": true,
    "isMoved": true,
    "isCopied": true,
    "deepCdr": {
      "sanitizationDetails": {
        "failureCategory": "string",
        "details": "string"
      },
      "cdrObjectState": 0,
      "fileWasReplaced": true
    },
    "postActionFailedInformation": [
      {
        "objectActionType": 0,
        "exception": {
          "exceptionDetails": "string",
          "exceptionMessage": "string"
        }
      }
    ],
    "objectRemediationActions": [
      {
```

```
"objectActionState": 0,
        "cdrObjectState": 0,
        "exceptions": [
          "string"
        ],
        "id": "string",
        "fileId": "string",
        "fileScanCompletedTime": "2019-08-24T14:15:22Z",
        "remediationId": "string",
        "tenantId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z"
      }
    ]
  },
  "metaDefenderCoreUrl": "string",
  "discoveryStartedDaySec": 0,
  "fileDto": {
    "result": 0,
    "responseKey": "string",
    "responseMessage": "string",
    "responseMessageParams": {
      "property1": "string",
      "property2": "string"
    },
    "id": "string",
    "scanId": "string",
    "name": "string",
    "path": "string",
    "hash": "string",
    "fileOwner": {
      "userId": "string",
      "fullName": "string",
      "userName": "string",
      "emailAddress": "string"
    },
    "size": 0,
    "fileTypeCategory": 0,
    "extension": "string",
    "storageType": {
      "name": "string",
      "categoryType": 0,
      "protocolType": 0,
      "vendorType": 0
    },
    "storageName": "string",
    "storageLocation": "string",
    "scanName": "string",
    "progress": {
      "upload": 0,
      "scan": 0
    },
    "scanResult": {
      "extractedFiles": {
```

```
"scanResultI": 0,
        "detectedBy": 0,
        "filesExtractedCount": 0
      },
      "scanResults": {
        "scanDetails": [
          {
            "engineName": "string",
            "threatFound": "string",
            "scanResultI": 0,
            "defTime": "string",
            "scanTime": 0.1,
            "engId": "string",
            "waitTime": 0.1
          }
        ],
        "dataId": "string",
        "scanAllResultI": 0,
        "scanAllResultA": "string",
        "startTime": "string",
        "totalTime": 0.1,
        "totalAvs": 0,
        "totalDetections": 0,
        "progressPercentage": 0
      },
      "fileInfo": {
        "fileSize": 0,
        "uploadTimestamp": "string",
        "md5": "string",
        "sha1": "string",
        "sha256": "string",
        "fileTypeCategory": "string",
        "fileTypeDescription": "string",
        "fileTypeExtension": "string",
        "displayName": "string",
        "fileTypeId": "string"
      },
      "fileScanInfo": {
        "finalVerdict": {
          "blocked": true,
          "verdict": "string",
          "confidence": 0.1,
          "threatLevel": 0.1
        },
        "overviewReport": {
          "analysisState": "string",
          "signalGroups": [
            {
              "description": "string",
              "verdict": "string",
              "signals": [
                {
                  "originType": "string",
```

```
"strength": 0.1
                }
              ]
            }
          ]
        },
        "errorCategory": "string",
        "errorCode": 0,
        "errorDetails": "string"
      },
      "processInfo": {
        "postProcessing": {
          "actionsRan": "string",
          "actionsFailed": "string",
          "convertedTo": "string",
          "copyMoveDestination": "string",
          "convertedDestination": "string",
          "sanitizationDetails": {
            "failureCategory": "string",
            "details": "string"
          }
        },
        "progressPercentage": 0,
        "userAgent": "string",
        "profile": "string",
        "result": "string",
        "blockedReason": "string",
        "fileTypeSkippedScan": true,
        "processingTime": 0.1,
        "queueTime": 0.1,
        "resultI": 0
      },
      "dataId": "string",
      "dlpInfo": {
        "hits": null,
        "verdict": 0
      },
      "sbomInfo": {
        "finalVerdict": {
          "blocked": true,
          "licensingPackageCounts": {
            "allowed": 0,
            "blocked": 0,
            "unknown": 0,
            "unsupported": 0
          },
          "severity": 0,
          "totalPackageCount": 0,
          "verdict": "string",
          "vulnerabilityCounts": {
            "critical": 0,
            "high": 0,
            "medium": 0,
```

```
"unknown": 0
          }
        },
        "overviewReport": {
          "isPartialReport": true,
          "osInfo": {
            "osName": "string",
            "osVersion": "string"
          },
          "scanMode": "string",
          "steps": [
            {
              "command": "string",
              "packages": [
                {
                  "authors": "string",
                  "releaseDate": "2019-08-24T14:15:22Z",
                  "ecosystem": "string",
                  "group": "string",
                  "archive": true,
                  "executable": true,
                  "uid": "string",
                  "licenses": {
                    "allowed": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            "string"
                          ],
                          "conditions": [
                            "string"
                          ],
                          "limitations": [
                            "string"
                          ]
                        }
                      }
                    ],
                    "blocked": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            "string"
                          ],
                          "conditions": [
                            "string"
                          ],
```

```
"string"
                            ]
                          }
                        }
                      ]
                    },
                    "packageName": "string",
                    "packageVersion": "string",
                    "hashes": [
                      {
                        "sha256": "string",
                        "fileName": "string"
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    },
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "arch": "string",
                    "osName": "string",
                    "osVersion": "string",
                    "dependencies": [
                      {
                        "ecosystem": "string",
                        "group": "string",
                        "packageName": "string",
                        "packageVersion": "string",
                        "vulnerabilities": [
                          {
                            "id": "string",
                            "severity": 0,
                            "source": "string",
                            "fixedVersions": "string",
                            "cwes": [
                              "string"
                            ]
                          }
                        ],
                        "vulnerabilityCounts": {
```

```
"high": 0,
                        "medium": 0,
                        "low": 0,
                        "unknown": 0
                      }
                    }
                  ]
                }
              ],
              "licenseStatus": "string",
              "licensingPackageCounts": {
                "allowed": 0,
                "blocked": 0,
                "unknown": 0,
                "unsupported": 0
              },
              "step": 0,
              "vulnerabilityCount": 0,
              "sha256": "string",
              "severity": 0,
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              }
            }
          ],
          "packages": [
            {
              "authors": "string",
              "releaseDate": "2019-08-24T14:15:22Z",
              "ecosystem": "string",
              "group": "string",
              "archive": true,
              "executable": true,
              "uid": "string",
              "licenses": {
                "allowed": [
                  {
                    "id": "string",
                    "url": "string",
                    "category": "string",
                    "usageRestriction": {
                      "permissions": [
                        "string"
                      ],
                      "conditions": [
                        "string"
                      ],
                      "limitations": [
                        "string"
```

```
}
                    }
                  ],
                  "blocked": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ]
                },
                "packageName": "string",
                "packageVersion": "string",
                "target": "string",
                "hashes": [
                  {
                    "sha256": "string",
                    "fileName": "string"
                  }
                ],
                "vulnerabilities": [
                  {
                    "id": "string",
                    "severity": 0,
                    "source": "string",
                    "fixedVersions": "string",
                    "cwes": [
                      "string"
                    ]
                  }
                ],
                "vulnerabilityCounts": {
                  "critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
                },
                "dependencies": [
                  {
                    "ecosystem": "string",
                    "group": "string",
```

```
"packageVersion": "string",
                  "vulnerabilities": [
                    {
                      "id": "string",
                      "severity": 0,
                      "source": "string",
                      "fixedVersions": "string",
                      "cwes": [
                        "string"
                      ]
                    }
                  ],
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  }
                }
              ]
            }
          ],
          "dependenciesPackageCount": 0,
          "dependenciesVulnerabilityTotal": 0,
          "dependenciesVulnerabilityCounts": {
            "critical": 0,
            "high": 0,
            "medium": 0,
            "low": 0,
            "unknown": 0
          }
        },
        "resultTemplateHash": "string"
      },
      "vulnerabilityInfo": {
        "vulnerabilityInfoResult": {
          "code": 0,
          "hash": "string",
          "method": 0,
          "timeStamp": "string",
          "timing": 0,
          "detectedProduct": {
            "hasVulnerability": true,
            "isCurrent": true,
            "product": {
              "id": 0,
              "name": "string"
            },
            "remediationLink": "string",
            "severity": "string",
            "sigName": "string",
            "signature": 0,
```

```
"id": 0,
              "name": "string"
            },
            "version": "string",
            "versionData": {
              "countBehind": 0,
              "feedId": 0,
              "version": "string"
            },
            "vulnerabilities": [
              {
                "description": "string",
                "details": {
                  "cpe": "string",
                  "cve": "string",
                  "cvss": {
                    "accessComplexity": "string",
                    "accessVector": "string",
                    "authentication": "string",
                    "availabilityImpact": "string",
                    "confidentialityImpact": "string",
                    "generatedOnEpoch": "string",
                    "integrityImpact": "string",
                    "score": "string",
                    "source": "string"
                  },
                  "cvss3": {
                    "attackComplexity": "string",
                    "attackVector": "string",
                    "availabilityImpact": "string",
                    "baseScore": "string",
                    "baseSeverity": "string",
                    "confidentialityImpact": "string",
                    "exploitabilityScore": "string",
                    "impactScore": "string",
                    "integrityImpact": "string",
                    "privilegesRequired": "string",
                    "scope": "string",
                    "userInteraction": "string",
                    "vectorString": "string"
                  },
                  "cwe": "string",
                  "lastModifiedEpoch": "string",
                  "publishedEpoch": "string",
                  "references": [
                    "string"
                  ]
                },
                "severity": "string",
                "severityIndex": 0,
                "staticId": 0
              }
            ]
```

```
}
      },
      "sessionCookie": "string",
      "fileTypeInfo": {
        "fileInfoExtended": {
          "extensions": "string",
          "groupId": "string",
          "typeId": "string"
        }
      },
      "scanInstanceUrl": "string",
      "scanWorkflowSnapshotId": "string",
      "coreEngines": [
        {
          "engineId": "string",
          "engineName": "string",
          "engineVersion": "string",
          "dbVersion": "string",
          "definitionTime": "string"
        }
      ]
    },
    "processingState": 0,
    "exception": {
      "exceptionDetails": "string",
      "exceptionMessage": "string"
    },
    "created": "2019-08-24T14:15:22Z",
    "lastModified": "2019-08-24T14:15:22Z",
    "discoveryStarted": "2019-08-24T14:15:22Z",
    "fileProcessingDetails": {
      "streamingStartedTime": "2019-08-24T14:15:22Z",
      "streamingCompletedTime": "2019-08-24T14:15:22Z",
      "streamingTimeSpanMs": 0,
      "scanCompletedTime": "2019-08-24T14:15:22Z",
      "processingTimeSpanMs": 0
    },
    "coreEngines": [
      {
        "engineId": "string",
        "engineName": "string",
        "engineVersion": "string",
        "dbVersion": "string",
        "definitionTime": "string"
      }
    ],
    "remediation": {
      "isTagged": true,
      "isDeleted": true,
      "isMoved": true,
      "isCopied": true,
      "deepCdr": {
        "sanitizationDetails": {
```

```
"details": "string"
        },
        "cdrObjectState": 0,
        "fileWasReplaced": true
      },
      "postActionFailedInformation": [
        {
          "objectActionType": 0,
          "exception": {
            "exceptionDetails": "string",
            "exceptionMessage": "string"
          }
        }
      ],
      "objectRemediationActions": [
        {
          "objectActionType": 0,
          "objectActionState": 0,
          "cdrObjectState": 0,
          "exceptions": [
            "string"
          ],
          "id": "string",
          "fileId": "string",
          "fileScanCompletedTime": "2019-08-24T14:15:22Z",
          "remediationId": "string",
          "tenantId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z"
        }
      ]
    },
    "metaDefenderCoreUrl": "string",
    "discoveryStartedDaySec": 0,
    "objectStorageMetadata": "string",
    "scanWorkflowSnapshotId": "string",
    "scanWorkflowSnapshot": {
      "id": "string",
      "scanId": "string",
      "workflow": {
        "id": "string",
        "scanPoolId": "string",
        "scanProcessFlow": 0,
        "isIdentityScanningEnabled": true,
        "scanIntervalDays": 0,
        "rules": [
          "string"
        ],
        "type": 0,
        "workflowSupportedTechnologies": {
          "dlp": {
            "status": 0
          },
          "deepCdr": {
```

```
},
          "sandBox": {
            "status": 0
          },
          "metascan": {
            "status": 0
          }
        },
        "name": "string",
        "userAgent": "string",
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanPoolId": "string",
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "filters": {
          "id": "string",
          "minimumSize": 0,
          "maximumSize": 0,
          "minimumDateTime": "2019-08-24T14:15:22Z",
          "maximumDateTime": "2019-08-24T14:15:22Z"
        },
        "fileDefinition": {
          "blockingStrategy": 0,
          "configurableBlockingStrategy": {
```

```
"shouldTreatFilesWithVulnerabilitiesAsBlocked": true
          }
        },
        "remediations": [
          {
            "id": "string",
            "workflowId": "string",
            "isEnabled": true,
            "configuration": "string",
            "type": 0,
            "handlingType": 0
          }
        ],
        "tenantId": "string"
      },
      "startTimestamp": 0,
      "stopTimestamp": 0,
      "scanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "failOverScanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
```

```
"timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "failOverScanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "storage": {
        "id": "string",
        "name": "string",
        "groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      },
      "workflowStatistics": {
        "taggingStatistics": {
          "total": 0
        },
        "keepAllowedStatistics": {
          "total": 0
        },
        "keepBlockedStatistics": {
          "total": 0
        },
        "keepSanitizedStatistics": {
          "total": 0
        },
        "movedAllowedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
```

```
"name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "movedSanitizedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "movedBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
```

```
"source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "copyAllowedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "copyBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
```

```
"categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "deleteAllowedStatistics": {
          "total": 0
        },
        "deleteBlockedStatistics": {
          "total": 0
        },
        "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
      },
      "account": {
        "id": "string",
        "name": "string",
        "description": "string",
        "credentials": "string",
        "protocolType": 0,
        "vendorType": 0,
        "autoDiscovery": true,
        "autodiscoveryDestinationGroupId": "string"
      },
      "realTimeStartDate": "2019-08-24T14:15:22Z"
    },
    "versions": [
      {
        "result": 0,
        "responseKey": "string",
        "responseMessage": "string",
        "responseMessageParams": {
          "property1": "string",
          "property2": "string"
        },
        "id": "string",
        "scanId": "string",
        "name": "string",
        "path": "string",
        "hash": "string",
        "fileOwner": {
          "userId": "string",
          "fullName": "string",
          "userName": "string",
          "emailAddress": "string"
        },
```

```
"fileTypeCategory": 0,
        "extension": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageName": "string",
        "storageLocation": "string",
        "scanName": "string",
        "progress": {
          "upload": 0,
          "scan": 0
        },
        "scanResult": {
          "extractedFiles": {
            "dataId": "string",
            "scanResultI": 0,
            "detectedBy": 0,
            "filesExtractedCount": 0
          },
          "scanResults": {
            "scanDetails": [
              {
                "engineName": "string",
                "threatFound": "string",
                "scanResultI": 0,
                "defTime": "string",
                "scanTime": 0.1,
                "engId": "string",
                "waitTime": 0.1
              }
            ],
            "dataId": "string",
            "scanAllResultI": 0,
            "scanAllResultA": "string",
            "startTime": "string",
            "totalTime": 0.1,
            "totalAvs": 0,
            "totalDetections": 0,
            "progressPercentage": 0
          },
          "fileInfo": {
            "fileSize": 0,
            "uploadTimestamp": "string",
            "md5": "string",
            "sha1": "string",
            "sha256": "string",
            "fileTypeCategory": "string",
            "fileTypeDescription": "string",
            "fileTypeExtension": "string",
            "displayName": "string",
```

```
},
          "fileScanInfo": {
            "finalVerdict": {
              "blocked": true,
              "verdict": "string",
              "confidence": 0.1,
              "threatLevel": 0.1
            },
            "overviewReport": {
              "analysisState": "string",
              "signalGroups": [
                {
                  "description": "string",
                  "verdict": "string",
                  "signals": [
                    {
                      "originType": "string",
                      "signalReadable": "string",
                      "strength": 0.1
                    }
                  ]
                }
              ]
            },
            "errorCategory": "string",
            "errorCode": 0,
            "errorDetails": "string"
          },
          "processInfo": {
            "postProcessing": {
              "actionsRan": "string",
              "actionsFailed": "string",
              "convertedTo": "string",
              "copyMoveDestination": "string",
              "convertedDestination": "string",
              "sanitizationDetails": {
                "failureCategory": "string",
                "details": "string"
              }
            },
            "progressPercentage": 0,
            "userAgent": "string",
            "profile": "string",
            "result": "string",
            "blockedReason": "string",
            "fileTypeSkippedScan": true,
            "processingTime": 0.1,
            "queueTime": 0.1,
            "resultI": 0
          },
          "dataId": "string",
          "dlpInfo": {
            "hits": null,
```

```
},
          "sbomInfo": {
            "finalVerdict": {
              "blocked": true,
              "licensingPackageCounts": {
                "allowed": 0,
                "blocked": 0,
                "unknown": 0,
                "unsupported": 0
              },
              "severity": 0,
              "totalPackageCount": 0,
              "verdict": "string",
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              }
            },
            "overviewReport": {
              "isPartialReport": true,
              "osInfo": {
                "osName": "string",
                "osVersion": "string"
              },
              "scanMode": "string",
              "steps": [
                {
                  "command": "string",
                  "packages": [
                    {
                      "authors": "string",
                      "releaseDate": "2019-08-24T14:15:22Z",
                      "ecosystem": "string",
                      "group": "string",
                      "archive": true,
                      "executable": true,
                      "uid": "string",
                      "licenses": {
                        "allowed": [
                          {
                            "id": "string",
                            "url": "string",
                            "category": "string",
                            "usageRestriction": {
                              "permissions": [],
                              "conditions": [],
                              "limitations": []
                            }
                          }
                        ],
```

```
{
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [],
                          "conditions": [],
                          "limitations": []
                        }
                      }
                    ]
                  },
                  "packageName": "string",
                  "packageVersion": "string",
                  "hashes": [
                    {
                      "sha256": "string",
                      "fileName": "string"
                    }
                  ],
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  },
                  "vulnerabilities": [
                    {
                      "id": "string",
                      "severity": 0,
                      "source": "string",
                      "fixedVersions": "string",
                      "cwes": [
                        "string"
                      ]
                    }
                  ],
                  "arch": "string",
                  "osName": "string",
                  "osVersion": "string",
                  "dependencies": [
                    {
                      "ecosystem": "string",
                      "group": "string",
                      "packageName": "string",
                      "packageVersion": "string",
                      "vulnerabilities": [
                        {
                          "cwes": []
                        }
                      ],
                      "vulnerabilityCounts": {
```

```
"high": 0,
                            "medium": 0,
                            "low": 0,
                            "unknown": 0
                          }
                        }
                      ]
                    }
                  ],
                  "licenseStatus": "string",
                  "licensingPackageCounts": {
                    "allowed": 0,
                    "blocked": 0,
                    "unknown": 0,
                    "unsupported": 0
                  },
                  "step": 0,
                  "vulnerabilityCount": 0,
                  "sha256": "string",
                  "severity": 0,
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  }
                }
              ],
              "packages": [
                {
                  "authors": "string",
                  "releaseDate": "2019-08-24T14:15:22Z",
                  "ecosystem": "string",
                  "group": "string",
                  "archive": true,
                  "executable": true,
                  "uid": "string",
                  "licenses": {
                    "allowed": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            "string"
                          ],
                          "conditions": [
                            "string"
                          ],
                          "limitations": [
                            "string"
```

```
}
                        }
                      ],
                      "blocked": [
                        {
                          "id": "string",
                          "url": "string",
                          "category": "string",
                          "usageRestriction": {
                            "permissions": [
                              "string"
                            ],
                            "conditions": [
                              "string"
                            ],
                            "limitations": [
                              "string"
                            ]
                          }
                        }
                      ]
                    },
                    "packageName": "string",
                    "packageVersion": "string",
                    "target": "string",
                    "hashes": [
                      {
                        "sha256": "string",
                        "fileName": "string"
                      }
                    ],
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    },
                    "dependencies": [
                      {
                        "ecosystem": "string",
                        "group": "string",
```

```
"packageVersion": "string",
                      "vulnerabilities": [
                        {
                          "id": "string",
                          "severity": 0,
                          "source": "string",
                          "fixedVersions": "string",
                          "cwes": [
                            "string"
                          ]
                        }
                      ],
                      "vulnerabilityCounts": {
                        "critical": 0,
                        "high": 0,
                        "medium": 0,
                        "low": 0,
                        "unknown": 0
                      }
                    }
                  ]
                }
              ],
              "dependenciesPackageCount": 0,
              "dependenciesVulnerabilityTotal": 0,
              "dependenciesVulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              }
            },
            "resultTemplateHash": "string"
          },
          "vulnerabilityInfo": {
            "vulnerabilityInfoResult": {
              "code": 0,
              "hash": "string",
              "method": 0,
              "timeStamp": "string",
              "timing": 0,
              "detectedProduct": {
                "hasVulnerability": true,
                "isCurrent": true,
                "product": {
                  "id": 0,
                  "name": "string"
                },
                "remediationLink": "string",
                "severity": "string",
                "sigName": "string",
                "signature": 0,
```

```
"id": 0,
                  "name": "string"
                },
                "version": "string",
                "versionData": {
                  "countBehind": 0,
                  "feedId": 0,
                  "version": "string"
                },
                "vulnerabilities": [
                  {
                    "description": "string",
                    "details": {
                      "cpe": "string",
                      "cve": "string",
                      "cvss": {
                        "accessComplexity": "string",
                        "accessVector": "string",
                        "authentication": "string",
                        "availabilityImpact": "string",
                        "confidentialityImpact": "string",
                        "generatedOnEpoch": "string",
                        "integrityImpact": "string",
                        "score": "string",
                        "source": "string"
                      },
                      "cvss3": {
                        "attackComplexity": "string",
                        "attackVector": "string",
                        "availabilityImpact": "string",
                        "baseScore": "string",
                        "baseSeverity": "string",
                        "confidentialityImpact": "string",
                        "exploitabilityScore": "string",
                        "impactScore": "string",
                        "integrityImpact": "string",
                        "privilegesRequired": "string",
                        "scope": "string",
                        "userInteraction": "string",
                        "vectorString": "string"
                      },
                      "cwe": "string",
                      "lastModifiedEpoch": "string",
                      "publishedEpoch": "string",
                      "references": [
                        "string"
                      ]
                    },
                    "severity": "string",
                    "severityIndex": 0,
                    "staticId": 0
                  }
                ]
```

```
}
          },
          "sessionCookie": "string",
          "fileTypeInfo": {
            "fileInfoExtended": {
              "extensions": "string",
              "groupId": "string",
              "typeId": "string"
            }
          },
          "scanInstanceUrl": "string",
          "scanWorkflowSnapshotId": "string",
          "coreEngines": [
            {
              "engineId": "string",
              "engineName": "string",
              "engineVersion": "string",
              "dbVersion": "string",
              "definitionTime": "string"
            }
          ]
        },
        "processingState": 0,
        "exception": {
          "exceptionDetails": "string",
          "exceptionMessage": "string"
        },
        "created": "2019-08-24T14:15:22Z",
        "lastModified": "2019-08-24T14:15:22Z",
        "discoveryStarted": "2019-08-24T14:15:22Z",
        "fileProcessingDetails": {
          "streamingStartedTime": "2019-08-24T14:15:22Z",
          "streamingCompletedTime": "2019-08-24T14:15:22Z",
          "streamingTimeSpanMs": 0,
          "scanCompletedTime": "2019-08-24T14:15:22Z",
          "processingTimeSpanMs": 0
        },
        "coreEngines": [
          {
            "engineId": "string",
            "engineName": "string",
            "engineVersion": "string",
            "dbVersion": "string",
            "definitionTime": "string"
          }
        ],
        "remediation": {
          "isTagged": true,
          "isDeleted": true,
          "isMoved": true,
          "isCopied": true,
          "deepCdr": {
            "sanitizationDetails": {
```

```
"details": "string"
            },
            "cdrObjectState": 0,
            "fileWasReplaced": true
          },
          "postActionFailedInformation": [
            {
              "objectActionType": 0,
              "exception": {
                "exceptionDetails": "string",
                "exceptionMessage": "string"
              }
            }
          ],
          "objectRemediationActions": [
            {
              "objectActionType": 0,
              "objectActionState": 0,
              "cdrObjectState": 0,
              "exceptions": [
                "string"
              ],
              "id": "string",
              "fileId": "string",
              "fileScanCompletedTime": "2019-08-24T14:15:22Z",
              "remediationId": "string",
              "tenantId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z"
            }
          ]
        },
        "metaDefenderCoreUrl": "string",
        "discoveryStartedDaySec": 0,
        "objectStorageMetadata": "string",
        "scanWorkflowSnapshotId": "string",
        "scanWorkflowSnapshot": {
          "id": "string",
          "scanId": "string",
          "workflow": {
            "id": "string",
            "scanPoolId": "string",
            "scanProcessFlow": 0,
            "isIdentityScanningEnabled": true,
            "scanIntervalDays": 0,
            "rules": [
              "string"
            ],
            "type": 0,
            "workflowSupportedTechnologies": {
              "dlp": {
                "status": 0
              },
              "deepCdr": {
```

```
},
              "sandBox": {
                "status": 0
              },
              "metascan": {
                "status": 0
              }
            },
            "name": "string",
            "userAgent": "string",
            "scanPool": {
              "id": "string",
              "name": "string",
              "scanPoolType": 0,
              "isDefault": true,
              "scanInstances": [
                {
                  "id": "string",
                  "scanPoolId": "string",
                  "url": "string",
                  "apiKey": "string",
                  "timeout": "string",
                  "scanInstanceType": 0
                }
              ]
            },
            "failOverScanPoolId": "string",
            "failOverScanPool": {
              "id": "string",
              "name": "string",
              "scanPoolType": 0,
              "isDefault": true,
              "scanInstances": [
                {
                  "id": "string",
                  "scanPoolId": "string",
                  "url": "string",
                  "apiKey": "string",
                  "timeout": "string",
                  "scanInstanceType": 0
                }
              ]
            },
            "filters": {
              "id": "string",
              "minimumSize": 0,
              "maximumSize": 0,
              "minimumDateTime": "2019-08-24T14:15:22Z",
              "maximumDateTime": "2019-08-24T14:15:22Z"
            },
            "fileDefinition": {
              "blockingStrategy": 0,
              "configurableBlockingStrategy": {
```

```
"shouldTreatFilesWithVulnerabilitiesAsBlocked": true
              }
            },
            "remediations": [
              {
                "id": "string",
                "workflowId": "string",
                "isEnabled": true,
                "configuration": "string",
                "type": 0,
                "handlingType": 0
              }
            ],
            "tenantId": "string"
          },
          "startTimestamp": 0,
          "stopTimestamp": 0,
          "scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ],
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
```

```
"timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ],
          "storage": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          },
          "workflowStatistics": {
            "taggingStatistics": {
              "total": 0
            },
            "keepAllowedStatistics": {
              "total": 0
            },
            "keepBlockedStatistics": {
              "total": 0
            },
            "keepSanitizedStatistics": {
              "total": 0
            },
            "movedAllowedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
```

```
"name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "movedSanitizedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "movedBlockedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
```

```
"source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "copyAllowedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "copyBlockedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
```

```
"categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "deleteAllowedStatistics": {
              "total": 0
            },
            "deleteBlockedStatistics": {
              "total": 0
            },
            "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
          },
          "account": {
            "id": "string",
            "name": "string",
            "description": "string",
            "credentials": "string",
            "protocolType": 0,
            "vendorType": 0,
            "autoDiscovery": true,
            "autodiscoveryDestinationGroupId": "string"
          },
          "realTimeStartDate": "2019-08-24T14:15:22Z"
        },
        "versions": [],
        "hasBlockedVersions": true,
        "versionsCount": 0,
        "objectType": 0,
        "skippedScan": true,
        "$type": "string",
        "url": "string",
        "fileId": "string"
      }
    ],
    "hasBlockedVersions": true,
    "versionsCount": 0,
    "objectType": 0,
    "skippedScan": true,
    "$type": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	GetFileResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Group

Manage your groups

Add a group

`POST /api/group`

> Body parameter

```
{
  "name": "string",
  "description": "string",
  "storageIds": [
    "string"
  ]
}
```

Parameters

Name	In	Type	Required	Description
body	body	AddGroupParameters	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "group": {
    "id": "string",
    "name": "string",
    "description": "string",
    "storages": [
      {
        "id": "string",
        "name": "string",
        "groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      }
    ]
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	AddGroupCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch all groups

GET /api/group

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "groups": [
    {
      "id": "string",
      "name": "string",
      "description": "string",
      "storages": [
        {
          "id": "string",
          "groupId": "string",
          "accountId": "string",
          "name": "string",
          "storageClientId": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "scansInProgress": true,
          "isRtpInProgress": true,
          "isScheduledScanInProgress": true,
          "storageFunctionality": "string",
          "$type": "string"
        }
      ]
    }
  ]
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	GetGroups

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update a group

`PATCH /api/group`

> Body parameter

```
{
  "name": "string",
  "description": "string",
  "groupId": "string"
}
```

Parameters

Name	In	Type	Required	Description
body	body	UpdateGroupParameters	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "updatedGroup": {
    "id": "string",
    "name": "string",
    "description": "string",
    "storages": [
      {
        "id": "string",
        "name": "string",
        "groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      }
    ]
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateGroupCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Delete a group

`DELETE /api/group/{groupId}`

Parameters

Name	In	Type	Required	Description
groupId	path	string	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	DeleteGroupCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch group by ID

`GET /api/group/{groupId}`

Parameters

Name	In	Type	Required	Description
groupId	path	string	true	none

> Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "group": {
    "id": "string",
    "name": "string",
    "description": "string",
    "storages": [
      {
        "id": "string",
        "groupId": "string",
        "accountId": "string",
        "name": "string",
        "storageClientId": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "scansInProgress": true,
        "isRtpInProgress": true,
        "isScheduledScanInProgress": true,
        "storageFunctionality": "string",
        "$type": "string"
      }
    ]
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetGroupById

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch the number of groups

`GET /api/group/count`

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "groupCount": {
    "count": 0
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetGroupsCount

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Health Status

API that responds with 200 OK if application is running.

Get health status

`GET /api/health`

Parameters

Name	In	Type	Required	Description
verbose	query	boolean	false	whether to include extra details

> Example responses

> 200 Response


```
{
  "apiStatus": "string"
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	ApiHealthModel

This operation does not require authentication

Onboarding

Manage onboarding

Fetch onboarding configuration

`GET /api/onboarding`

Parameters

Name	In	Type	Required	Description
newConfigSetup	query	boolean	false	none
newDefaultWorkflow	query	boolean	false	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "hasUserConfigured": true,
  "hasStorageConfigured": true,
  "hasScanInstanceConfigured": true,
  "isOnboardingDone": true,
  "isEulaAgreed": true,
  "isCloudType": true
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetOnboarding

This operation does not require authentication

Finish onboarding

`POST /api/onboarding`

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Accept Eula

`POST /api/onboarding/eula`

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Remediations

Remediations information

Get Remediations by workflow id

`GET /api/remediations/workflow/{id}`

Parameters

Name	In	Type	Required	Description
id	path	string	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "workflowId": "string",
      "isEnabled": true,
      "configuration": "string",
      "type": 0,
      "handlingType": 0
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetRemediationByWorkflowIdCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get Remediation by id

`GET /api/remediations/{id}`

Parameters

Name	In	Type	Required	Description
id	path	string	true	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entry": {
    "id": "string",
    "workflowId": "string",
    "isEnabled": true,
    "configuration": "string",
    "type": 0,
    "handlingType": 0
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetRemediationByIdCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Delete Remediation

`DELETE /api/remediations/{id}`

Parameters

Name	In	Type	Required	Description
id	path	string	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	DeleteRemediationCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Add Remediation

`POST /api/remediations`

> Body parameter


```
{
  "workflowId": "string",
  "isEnabled": true,
  "configuration": "string",
  "type": 0,
  "handlingType": 0
}
```


Parameters

Name	In	Type	Required	Description
body	body	AddRemediationParameters	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	AddRemediationCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update Remediation

`PATCH /api/remediations`

> Body parameter


```
{
  "id": "string",
  "workflowId": "string",
  "isEnabled": true,
  "configuration": "string",
  "type": 0,
  "handlingType": 0
}
```


Parameters

Name	In	Type	Required	Description
body	body	UpdateRemediationParameters	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateRemediationCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Report

Generate reports

Get scans report

`GET /api/report`

Parameters

Name	In	Type	Required	Description
storageId	query	string	false	none
name	query	string	false	none
scanTypesFilter	query	array[integer]	false	none
scanStatesFilter	query	array[integer]	false	none
realTimeHandlingType	query	any	false	none
startDate	query	string	false	none
endDate	query	string	false	none
sortBy	query	array[integer]	false	none
sortOrder	query	array[integer]	false	none
startIndex	query	integer(int32)	false	none
count	query	integer(int32)	false	none
workflowId	query	string	false	none

Enumerated Values

Parameter	Value
scanTypesFilter	0
scanTypesFilter	1
scanTypesFilter	2
scanStatesFilter	0
scanStatesFilter	1
scanStatesFilter	2
scanStatesFilter	3
scanStatesFilter	4
sortBy	0
sortBy	1
sortBy	2
sortOrder	0
sortOrder	1

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "result": 0,
      "responseKey": "string",
      "responseMessage": "string",
      "responseMessageParams": {
        "property1": "string",
        "property2": "string"
      },
      "id": "string",
      "name": "string",
      "startTime": "2019-08-24T14:15:22Z",
      "stopTime": "2019-08-24T14:15:22Z",
      "scanType": "string",
      "scanningState": 0,
      "scanStatusDto": {
        "pendingFilesCount": 0,
        "pendingFilesSize": 0,
        "allowedFilesCount": 0,
        "allowedFilesSize": 0,
        "blockedFilesCount": 0,
        "blockedFilesSize": 0,
        "sanitizedFilesCount": 0,
        "failedSanitizedFilesCount": 0,
        "remediatedFilesCount": 0,
        "totalFilesCount": 0,
        "totalFilesSize": 0,
        "processingFailedCount": 0,
        "processingFailedSize": 0,
        "processingFailedRemediationCount": 0,
        "processedFilesCount": 0,
        "processedFilesSize": 0,
        "canceledFilesCount": 0,
        "canceledFilesSize": 0
      },
      "scanStatsDto": {
        "infectedFilesCount": 0,
        "filesWithVulnerabilitiesCount": 0,
        "filesWithSensitiveDataCount": 0,
        "filesBlockedBySandboxCount": 0,
        "scannedFilesCount": 0,
        "skippedFilesScanCount": 0,
        "scanFilesDistribution": {
          "documentsEncryptedFilesCount": 0,
```

```
"audioVideoFilesCount": 0,
          "openSSlEncryptedFilesCount": 0,
          "pdfFilesCount": 0,
          "graphicalFilesCount": 0,
          "diskImageFilesCount": 0,
          "textFilesCount": 0,
          "mailFilesCount": 0,
          "applicationFilesCount": 0,
          "documentsFilesCount": 0,
          "otherFilesCount": 0,
          "archivesFilesCount": 0
        }
      },
      "realTimeStartDate": "2019-08-24T14:15:22Z",
      "scanSequenceDto": {
        "previousScanId": "string",
        "nextScanId": "string"
      },
      "scanPartition": "string",
      "workflowId": "string",
      "scanConfigurationId": "string",
      "errors": [
        "string"
      ],
      "priority": 0,
      "scanWorkflowSnapshotDtos": [
        {
          "id": "string",
          "scanId": "string",
          "workflow": {
            "id": "string",
            "scanPoolId": "string",
            "scanProcessFlow": 0,
            "isIdentityScanningEnabled": true,
            "scanIntervalDays": 0,
            "rules": [
              "string"
            ],
            "type": 0,
            "workflowSupportedTechnologies": {
              "dlp": {
                "status": 0
              },
              "deepCdr": {
                "status": 0
              },
              "sandBox": {
                "status": 0
              },
              "metascan": {
                "status": 0
              }
            },
```

```
"userAgent": "string",
            "scanPool": {
              "id": "string",
              "name": "string",
              "scanPoolType": 0,
              "isDefault": true,
              "scanInstances": [
                {
                  "id": "string",
                  "scanPoolId": "string",
                  "url": "string",
                  "apiKey": "string",
                  "timeout": "string",
                  "scanInstanceType": 0
                }
              ]
            },
            "failOverScanPoolId": "string",
            "failOverScanPool": {
              "id": "string",
              "name": "string",
              "scanPoolType": 0,
              "isDefault": true,
              "scanInstances": [
                {
                  "id": "string",
                  "scanPoolId": "string",
                  "url": "string",
                  "apiKey": "string",
                  "timeout": "string",
                  "scanInstanceType": 0
                }
              ]
            },
            "filters": {
              "id": "string",
              "minimumSize": 0,
              "maximumSize": 0,
              "minimumDateTime": "2019-08-24T14:15:22Z",
              "maximumDateTime": "2019-08-24T14:15:22Z"
            },
            "fileDefinition": {
              "blockingStrategy": 0,
              "configurableBlockingStrategy": {
                "shouldTreatFilesWithSensitiveDataAsBlocked": true,
                "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
              }
            },
            "remediations": [
              {
                "id": "string",
                "workflowId": "string",
                "isEnabled": true,
```

```
"type": 0,
                "handlingType": 0
              }
            ],
            "tenantId": "string"
          },
          "startTimestamp": 0,
          "stopTimestamp": 0,
          "scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ],
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanInstances": [
            {
              "id": "string",
```

```
"url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ],
          "storage": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          },
          "workflowStatistics": {
            "taggingStatistics": {
              "total": 0
            },
            "keepAllowedStatistics": {
              "total": 0
            },
            "keepBlockedStatistics": {
              "total": 0
            },
            "keepSanitizedStatistics": {
              "total": 0
            },
            "movedAllowedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
```

```
"protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "movedSanitizedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "movedBlockedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
```

```
"message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "copyAllowedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "copyBlockedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
```

```
"lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "deleteAllowedStatistics": {
              "total": 0
            },
            "deleteBlockedStatistics": {
              "total": 0
            },
            "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
          },
          "account": {
            "id": "string",
            "name": "string",
            "description": "string",
            "credentials": "string",
            "protocolType": 0,
            "vendorType": 0,
            "autoDiscovery": true,
            "autodiscoveryDestinationGroupId": "string"
          },
          "realTimeStartDate": "2019-08-24T14:15:22Z"
        }
      ],
      "storageId": "string",
      "$type": "string"
    }
  ],
  "totalCount": 0
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetScansReportQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Download PDF Report

```
GET /api/report/download/{scanId}
```

Parameters

Name	In	Type	Required	Description
TypeOfReport	query	TypeOfReport	false	none
StartIndex	query	integer(int32)	false	starting position for pagination (0-based index)
Count	query	integer(int32)	false	none
StartDate	query	string(date-time)	false	filter files by scan date
EndDate	query	string(date-time)	false	filter files by scan date
Search	query	string	false	text filter
ScanType	query	array[integer]	false	scan type such as (on-demand, scheduled, real time)
StorageId	query	array[string]	false	filter based on a list of possible storage id
ScanId	query	array[string]	false	filter based on a list of possible scan id
ScanWorkflowSnapshotId	query	array[string]	false	filter based on a list of possible scan workflow snapshot id
ProcessingState	query	array[integer]	false	none
Result	query	array[integer]	false	filter by scan result (0 = Clean, 1 = Blocked, 2 = Processing)
ScanAllResultI	query	array[integer]	false	filter by MD Core scan result (0 = NoThreatDetected, 1 = Infected, 2 = Suspicious, etc.)
ScanResultICount	query	array[integer]	false	the number of engine results that are not “No threat detected”
SandboxVerdict	query	array[integer]	false	filter by Sandbox verdict (0 = NoThreat, 1 = Malicious, 2 = Suspicious)
VulnerabilityCount	query	array[integer]	false	the number of vulnerabilities detected
HasVulnerabilities	query	boolean	false	filter by presence of vulnerabilities
DlpDetection	query	boolean	false	DLP detection

HasBlockedFileVersions	query	boolean	false	filter by presence of blocked file versions
Remediations	query	array[integer]	false	filter by remediation performed on the file
SkippedScan	query	boolean	false	filter by whether the scan was skipped
SortBy	query	array[integer]	false	none
SortOrder	query	array[integer]	false	none
Expand	query	array[string]	false	Possible values for expand: workflow_snapshot
scanId	path	string	true	none

Enumerated Values

Parameter	Value
TypeOfReport	0
TypeOfReport	1
ScanType	0
ScanType	1
ScanType	2
ProcessingState	0
ProcessingState	1
ProcessingState	2
ProcessingState	3
ProcessingState	4
Result	0
Result	1
Result	2
ScanAllResultI	0
ScanAllResultI	1
ScanAllResultI	2
ScanAllResultI	3
ScanAllResultI	7
ScanAllResultI	8
ScanAllResultI	9
ScanAllResultI	10
ScanAllResultI	12
ScanAllResultI	13
ScanAllResultI	14
ScanAllResultI	15
ScanAllResultI	16
ScanAllResultI	17

ScanAllResultI	18
ScanAllResultI	19
ScanAllResultI	20
ScanAllResultI	21
ScanAllResultI	22
ScanAllResultI	23
ScanAllResultI	24
ScanAllResultI	25
ScanAllResultI	26
ScanAllResultI	27
ScanAllResultI	28
ScanAllResultI	29
ScanAllResultI	30
ScanAllResultI	31
ScanAllResultI	32
ScanAllResultI	33
ScanAllResultI	34
ScanAllResultI	36
ScanAllResultI	38
ScanAllResultI	39
ScanAllResultI	40
ScanAllResultI	41
ScanAllResultI	42
ScanAllResultI	43
ScanAllResultI	255
ScanAllResultI	1014
ScanAllResultI	9999
SandboxVerdict	0
SandboxVerdict	1

SandboxVerdict	2
SandboxVerdict	3
SandboxVerdict	4
SandboxVerdict	5
SandboxVerdict	999
Remediations	0
Remediations	5
Remediations	6
Remediations	7
Remediations	8
Remediations	9
Remediations	10
Remediations	11
Remediations	12
Remediations	13
Remediations	14
Remediations	15
Remediations	16
Remediations	17
Remediations	18
Remediations	19
Remediations	20
Remediations	21
Remediations	22
Remediations	23
Remediations	24
Remediations	25
Remediations	100
Remediations	101

Remediations	102
SortBy	0
SortBy	1
SortBy	2
SortBy	3
SortBy	4
SortBy	5
SortBy	6
SortBy	7
SortBy	8
SortOrder	0
SortOrder	1

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Scan

Manage your scans

Get scan by ID

`GET /api/scan/{scanId}`

Parameters

Name	In	Type	Required	Description
scanId	path	string	true	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scan": {
    "result": 0,
    "responseKey": "string",
    "responseMessage": "string",
    "responseMessageParams": {
      "property1": "string",
      "property2": "string"
    },
    "id": "string",
    "name": "string",
    "startTime": "2019-08-24T14:15:22Z",
    "stopTime": "2019-08-24T14:15:22Z",
    "scanType": "string",
    "scanningState": 0,
    "scanStatusDto": {
      "pendingFilesCount": 0,
      "pendingFilesSize": 0,
      "allowedFilesCount": 0,
      "allowedFilesSize": 0,
      "blockedFilesCount": 0,
      "blockedFilesSize": 0,
      "sanitizedFilesCount": 0,
      "failedSanitizedFilesCount": 0,
      "remediatedFilesCount": 0,
      "totalFilesCount": 0,
      "totalFilesSize": 0,
      "processingFailedCount": 0,
      "processingFailedSize": 0,
      "processingFailedRemediationCount": 0,
      "processedFilesCount": 0,
      "processedFilesSize": 0,
      "canceledFilesCount": 0,
      "canceledFilesSize": 0
    },
    "scanStatsDto": {
      "infectedFilesCount": 0,
      "filesWithVulnerabilitiesCount": 0,
      "filesWithSensitiveDataCount": 0,
      "filesBlockedBySandboxCount": 0,
      "scannedFilesCount": 0,
      "skippedFilesScanCount": 0,
      "scanFilesDistribution": {
        "documentsEncryptedFilesCount": 0,
        "executablesFilesCount": 0,
```

```
"openSSlEncryptedFilesCount": 0,
        "pdfFilesCount": 0,
        "graphicalFilesCount": 0,
        "diskImageFilesCount": 0,
        "textFilesCount": 0,
        "mailFilesCount": 0,
        "applicationFilesCount": 0,
        "documentsFilesCount": 0,
        "otherFilesCount": 0,
        "archivesFilesCount": 0
      }
    },
    "realTimeStartDate": "2019-08-24T14:15:22Z",
    "scanSequenceDto": {
      "previousScanId": "string",
      "nextScanId": "string"
    },
    "scanPartition": "string",
    "workflowId": "string",
    "scanConfigurationId": "string",
    "errors": [
      "string"
    ],
    "priority": 0,
    "scanWorkflowSnapshotDtos": [
      {
        "id": "string",
        "scanId": "string",
        "workflow": {
          "id": "string",
          "scanPoolId": "string",
          "scanProcessFlow": 0,
          "isIdentityScanningEnabled": true,
          "scanIntervalDays": 0,
          "rules": [
            "string"
          ],
          "type": 0,
          "workflowSupportedTechnologies": {
            "dlp": {
              "status": 0
            },
            "deepCdr": {
              "status": 0
            },
            "sandBox": {
              "status": 0
            },
            "metascan": {
              "status": 0
            }
          },
          "name": "string",
```

```
"scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanPoolId": "string",
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "filters": {
            "id": "string",
            "minimumSize": 0,
            "maximumSize": 0,
            "minimumDateTime": "2019-08-24T14:15:22Z",
            "maximumDateTime": "2019-08-24T14:15:22Z"
          },
          "fileDefinition": {
            "blockingStrategy": 0,
            "configurableBlockingStrategy": {
              "shouldTreatFilesWithSensitiveDataAsBlocked": true,
              "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
            }
          },
          "remediations": [
            {
              "id": "string",
              "workflowId": "string",
              "isEnabled": true,
              "configuration": "string",
```

```
"handlingType": 0
            }
          ],
          "tenantId": "string"
        },
        "startTimestamp": 0,
        "stopTimestamp": 0,
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
```

```
"apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "storage": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        },
        "workflowStatistics": {
          "taggingStatistics": {
            "total": 0
          },
          "keepAllowedStatistics": {
            "total": 0
          },
          "keepBlockedStatistics": {
            "total": 0
          },
          "keepSanitizedStatistics": {
            "total": 0
          },
          "movedAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
```

```
"vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedSanitizedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
```

```
"statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
```

```
"tenantId": "string"
            }
          },
          "deleteAllowedStatistics": {
            "total": 0
          },
          "deleteBlockedStatistics": {
            "total": 0
          },
          "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
        },
        "account": {
          "id": "string",
          "name": "string",
          "description": "string",
          "credentials": "string",
          "protocolType": 0,
          "vendorType": 0,
          "autoDiscovery": true,
          "autodiscoveryDestinationGroupId": "string"
        },
        "realTimeStartDate": "2019-08-24T14:15:22Z"
      }
    ],
    "storageId": "string",
    "$type": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetScanByIdQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Start a scan

`POST /api/scan/start`

To scan a specific folder using the optional Folder parameter, provide the absolute folder path: For Amazon S3 / S3 Compatible Types, Azure Blob, Azure Files, Google Cloud, Alibaba Cloud and Oracle Native: Including the Folder Location integrated in MDSS and excluding the Bucket Name, Container, etc. For Box: With or without the “All Files” folder For Sharepoint and Onedrive: Excluding the Document Library, Site, or Group For OPSWAT MFT Storage: The desired Folder Path when integrating with user For NFS / SMB / SFTP / FTP / SharePoint OnPrem: Only the folder path beyond your configured storage root (do not

include the base path set during integration] Provide the absolute folder path in the following format "{"Folder":"PATH_TO_SCAN"}

Body parameter


```
{
  "storageId": "string",
  "name": "string",
  "folder": "string",
  "scanPartition": "string",
  "priority": 0,
  "storageIds": [
    "string"
  ],
  "workflowId": "string",
  "mdssBaseUrl": "string",
  "scanConfigDto": {
    "realTimeScanConfig": {
      "isEnabled": true,
      "startDate": "2019-08-24T14:15:22Z",
      "handlingType": 0
    }
  }
}
```


Parameters

Name	In	Type	Required	Description
body	body	AddScanParameters	true	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scanId": "string",
  "scanIds": [
    "string"
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	StartScanCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Stop a scan

`POST /api/scan/stop/{scanId}`

Parameters

Name	In	Type	Required	Description
scanId	path	string	true	none

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Delete scans

`DELETE /api/scan/{storageId}`

> Body parameter

```
[
  "string"
]
```

Parameters

Name	In	Type	Required	Description
storageId	path	string	true	none
forceDeleteRtp	query	boolean	false	none
body	body	array[string]	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "deletedScans": [
    {
      "id": "string",
      "storageId": "string",
      "workflowId": "string",
      "scanScheduleId": "string",
      "executionId": "string",
      "name": "string",
      "userId": "string",
      "scanConfigDto": {
        "realTimeScanConfig": {
          "isEnabled": true,
          "startDate": "2019-08-24T14:15:22Z",
          "handlingType": 0
        }
      },
      "discoveryState": 0,
      "scanningState": 0,
      "scanType": 0,
      "startTime": "2019-08-24T14:15:22Z",
      "stopTime": "2019-08-24T14:15:22Z",
      "scanPartition": "string",
      "priority": 0,
      "additionalProperties": "string",
      "tenantId": "string",
      "errors": [
        "string"
      ]
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	DeleteScansCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch last completed scan

`GET /api/scan/last_completed/{storageId}`

Parameters

Name	In	Type	Required	Description
storageId	path	string	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scan": {
    "result": 0,
    "responseKey": "string",
    "responseMessage": "string",
    "responseMessageParams": {
      "property1": "string",
      "property2": "string"
    },
    "id": "string",
    "name": "string",
    "startTime": "2019-08-24T14:15:22Z",
    "stopTime": "2019-08-24T14:15:22Z",
    "scanType": "string",
    "scanningState": 0,
    "scanStatusDto": {
      "pendingFilesCount": 0,
      "pendingFilesSize": 0,
      "allowedFilesCount": 0,
      "allowedFilesSize": 0,
      "blockedFilesCount": 0,
      "blockedFilesSize": 0,
      "sanitizedFilesCount": 0,
      "failedSanitizedFilesCount": 0,
      "remediatedFilesCount": 0,
      "totalFilesCount": 0,
      "totalFilesSize": 0,
      "processingFailedCount": 0,
      "processingFailedSize": 0,
      "processingFailedRemediationCount": 0,
      "processedFilesCount": 0,
      "processedFilesSize": 0,
      "canceledFilesCount": 0,
      "canceledFilesSize": 0
    },
    "scanStatsDto": {
      "infectedFilesCount": 0,
      "filesWithVulnerabilitiesCount": 0,
      "filesWithSensitiveDataCount": 0,
      "filesBlockedBySandboxCount": 0,
      "scannedFilesCount": 0,
      "skippedFilesScanCount": 0,
      "scanFilesDistribution": {
        "documentsEncryptedFilesCount": 0,
        "executablesFilesCount": 0,
```

```
"openSSlEncryptedFilesCount": 0,
        "pdfFilesCount": 0,
        "graphicalFilesCount": 0,
        "diskImageFilesCount": 0,
        "textFilesCount": 0,
        "mailFilesCount": 0,
        "applicationFilesCount": 0,
        "documentsFilesCount": 0,
        "otherFilesCount": 0,
        "archivesFilesCount": 0
      }
    },
    "realTimeStartDate": "2019-08-24T14:15:22Z",
    "scanSequenceDto": {
      "previousScanId": "string",
      "nextScanId": "string"
    },
    "scanPartition": "string",
    "workflowId": "string",
    "scanConfigurationId": "string",
    "errors": [
      "string"
    ],
    "priority": 0,
    "scanWorkflowSnapshotDtos": [
      {
        "id": "string",
        "scanId": "string",
        "workflow": {
          "id": "string",
          "scanPoolId": "string",
          "scanProcessFlow": 0,
          "isIdentityScanningEnabled": true,
          "scanIntervalDays": 0,
          "rules": [
            "string"
          ],
          "type": 0,
          "workflowSupportedTechnologies": {
            "dlp": {
              "status": 0
            },
            "deepCdr": {
              "status": 0
            },
            "sandBox": {
              "status": 0
            },
            "metascan": {
              "status": 0
            }
          },
          "name": "string",
```

```
"scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanPoolId": "string",
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "filters": {
            "id": "string",
            "minimumSize": 0,
            "maximumSize": 0,
            "minimumDateTime": "2019-08-24T14:15:22Z",
            "maximumDateTime": "2019-08-24T14:15:22Z"
          },
          "fileDefinition": {
            "blockingStrategy": 0,
            "configurableBlockingStrategy": {
              "shouldTreatFilesWithSensitiveDataAsBlocked": true,
              "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
            }
          },
          "remediations": [
            {
              "id": "string",
              "workflowId": "string",
              "isEnabled": true,
              "configuration": "string",
```

```
"handlingType": 0
            }
          ],
          "tenantId": "string"
        },
        "startTimestamp": 0,
        "stopTimestamp": 0,
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
```

```
"apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "storage": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        },
        "workflowStatistics": {
          "taggingStatistics": {
            "total": 0
          },
          "keepAllowedStatistics": {
            "total": 0
          },
          "keepBlockedStatistics": {
            "total": 0
          },
          "keepSanitizedStatistics": {
            "total": 0
          },
          "movedAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
```

```
"vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedSanitizedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
```

```
"statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
```

```
"tenantId": "string"
            }
          },
          "deleteAllowedStatistics": {
            "total": 0
          },
          "deleteBlockedStatistics": {
            "total": 0
          },
          "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
        },
        "account": {
          "id": "string",
          "name": "string",
          "description": "string",
          "credentials": "string",
          "protocolType": 0,
          "vendorType": 0,
          "autoDiscovery": true,
          "autodiscoveryDestinationGroupId": "string"
        },
        "realTimeStartDate": "2019-08-24T14:15:22Z"
      }
    ],
    "storageId": "string",
    "$type": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetLastCompletedScanQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get an active scan by scan ID

`GET /api/scan/active/{scanId}`

Parameters

Name	In	Type	Required	Description
scanId	path	string	true	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scan": {
    "result": 0,
    "responseKey": "string",
    "responseMessage": "string",
    "responseMessageParams": {
      "property1": "string",
      "property2": "string"
    },
    "id": "string",
    "name": "string",
    "startTime": "2019-08-24T14:15:22Z",
    "stopTime": "2019-08-24T14:15:22Z",
    "scanType": "string",
    "scanningState": 0,
    "scanStatusDto": {
      "pendingFilesCount": 0,
      "pendingFilesSize": 0,
      "allowedFilesCount": 0,
      "allowedFilesSize": 0,
      "blockedFilesCount": 0,
      "blockedFilesSize": 0,
      "sanitizedFilesCount": 0,
      "failedSanitizedFilesCount": 0,
      "remediatedFilesCount": 0,
      "totalFilesCount": 0,
      "totalFilesSize": 0,
      "processingFailedCount": 0,
      "processingFailedSize": 0,
      "processingFailedRemediationCount": 0,
      "processedFilesCount": 0,
      "processedFilesSize": 0,
      "canceledFilesCount": 0,
      "canceledFilesSize": 0
    },
    "scanStatsDto": {
      "infectedFilesCount": 0,
      "filesWithVulnerabilitiesCount": 0,
      "filesWithSensitiveDataCount": 0,
      "filesBlockedBySandboxCount": 0,
      "scannedFilesCount": 0,
      "skippedFilesScanCount": 0,
      "scanFilesDistribution": {
        "documentsEncryptedFilesCount": 0,
        "executablesFilesCount": 0,
```

```
"openSSlEncryptedFilesCount": 0,
        "pdfFilesCount": 0,
        "graphicalFilesCount": 0,
        "diskImageFilesCount": 0,
        "textFilesCount": 0,
        "mailFilesCount": 0,
        "applicationFilesCount": 0,
        "documentsFilesCount": 0,
        "otherFilesCount": 0,
        "archivesFilesCount": 0
      }
    },
    "realTimeStartDate": "2019-08-24T14:15:22Z",
    "scanSequenceDto": {
      "previousScanId": "string",
      "nextScanId": "string"
    },
    "scanPartition": "string",
    "workflowId": "string",
    "scanConfigurationId": "string",
    "errors": [
      "string"
    ],
    "priority": 0,
    "scanWorkflowSnapshotDtos": [
      {
        "id": "string",
        "scanId": "string",
        "workflow": {
          "id": "string",
          "scanPoolId": "string",
          "scanProcessFlow": 0,
          "isIdentityScanningEnabled": true,
          "scanIntervalDays": 0,
          "rules": [
            "string"
          ],
          "type": 0,
          "workflowSupportedTechnologies": {
            "dlp": {
              "status": 0
            },
            "deepCdr": {
              "status": 0
            },
            "sandBox": {
              "status": 0
            },
            "metascan": {
              "status": 0
            }
          },
          "name": "string",
```

```
"scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanPoolId": "string",
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "filters": {
            "id": "string",
            "minimumSize": 0,
            "maximumSize": 0,
            "minimumDateTime": "2019-08-24T14:15:22Z",
            "maximumDateTime": "2019-08-24T14:15:22Z"
          },
          "fileDefinition": {
            "blockingStrategy": 0,
            "configurableBlockingStrategy": {
              "shouldTreatFilesWithSensitiveDataAsBlocked": true,
              "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
            }
          },
          "remediations": [
            {
              "id": "string",
              "workflowId": "string",
              "isEnabled": true,
              "configuration": "string",
```

```
"handlingType": 0
            }
          ],
          "tenantId": "string"
        },
        "startTimestamp": 0,
        "stopTimestamp": 0,
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
```

```
"apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "storage": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        },
        "workflowStatistics": {
          "taggingStatistics": {
            "total": 0
          },
          "keepAllowedStatistics": {
            "total": 0
          },
          "keepBlockedStatistics": {
            "total": 0
          },
          "keepSanitizedStatistics": {
            "total": 0
          },
          "movedAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
```

```
"vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedSanitizedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
```

```
"statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
```

```
"tenantId": "string"
            }
          },
          "deleteAllowedStatistics": {
            "total": 0
          },
          "deleteBlockedStatistics": {
            "total": 0
          },
          "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
        },
        "account": {
          "id": "string",
          "name": "string",
          "description": "string",
          "credentials": "string",
          "protocolType": 0,
          "vendorType": 0,
          "autoDiscovery": true,
          "autodiscoveryDestinationGroupId": "string"
        },
        "realTimeStartDate": "2019-08-24T14:15:22Z"
      }
    ],
    "storageId": "string",
    "$type": "string",
    "serviceName": "string",
    "storageName": "string",
    "currentProcessingFilePath": "string",
    "currentProcessingFileProgress": "string",
    "discoveryState": 0,
    "statusMessages": [
      {
        "message": "string",
        "severity": 0,
        "params": {
          "property1": "string",
          "property2": "string"
        }
      }
    ]
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetActiveScanQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get Real-Time scan by storage Id

`GET /api/scan/realtime/{storageId}`

Parameters

Name	In	Type	Required	Description
storageId	path	string	true	none
scanPartition	query	string	false	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scan": {
    "result": 0,
    "responseKey": "string",
    "responseMessage": "string",
    "responseMessageParams": {
      "property1": "string",
      "property2": "string"
    },
    "id": "string",
    "name": "string",
    "startTime": "2019-08-24T14:15:22Z",
    "stopTime": "2019-08-24T14:15:22Z",
    "scanType": "string",
    "scanningState": 0,
    "scanStatusDto": {
      "pendingFilesCount": 0,
      "pendingFilesSize": 0,
      "allowedFilesCount": 0,
      "allowedFilesSize": 0,
      "blockedFilesCount": 0,
      "blockedFilesSize": 0,
      "sanitizedFilesCount": 0,
      "failedSanitizedFilesCount": 0,
      "remediatedFilesCount": 0,
      "totalFilesCount": 0,
      "totalFilesSize": 0,
      "processingFailedCount": 0,
      "processingFailedSize": 0,
      "processingFailedRemediationCount": 0,
      "processedFilesCount": 0,
      "processedFilesSize": 0,
      "canceledFilesCount": 0,
      "canceledFilesSize": 0
    },
    "scanStatsDto": {
      "infectedFilesCount": 0,
      "filesWithVulnerabilitiesCount": 0,
      "filesWithSensitiveDataCount": 0,
      "filesBlockedBySandboxCount": 0,
      "scannedFilesCount": 0,
      "skippedFilesScanCount": 0,
      "scanFilesDistribution": {
        "documentsEncryptedFilesCount": 0,
        "executablesFilesCount": 0,
```

```
"openSSlEncryptedFilesCount": 0,
        "pdfFilesCount": 0,
        "graphicalFilesCount": 0,
        "diskImageFilesCount": 0,
        "textFilesCount": 0,
        "mailFilesCount": 0,
        "applicationFilesCount": 0,
        "documentsFilesCount": 0,
        "otherFilesCount": 0,
        "archivesFilesCount": 0
      }
    },
    "realTimeStartDate": "2019-08-24T14:15:22Z",
    "scanSequenceDto": {
      "previousScanId": "string",
      "nextScanId": "string"
    },
    "scanPartition": "string",
    "workflowId": "string",
    "scanConfigurationId": "string",
    "errors": [
      "string"
    ],
    "priority": 0,
    "scanWorkflowSnapshotDtos": [
      {
        "id": "string",
        "scanId": "string",
        "workflow": {
          "id": "string",
          "scanPoolId": "string",
          "scanProcessFlow": 0,
          "isIdentityScanningEnabled": true,
          "scanIntervalDays": 0,
          "rules": [
            "string"
          ],
          "type": 0,
          "workflowSupportedTechnologies": {
            "dlp": {
              "status": 0
            },
            "deepCdr": {
              "status": 0
            },
            "sandBox": {
              "status": 0
            },
            "metascan": {
              "status": 0
            }
          },
          "name": "string",
```

```
"scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanPoolId": "string",
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "filters": {
            "id": "string",
            "minimumSize": 0,
            "maximumSize": 0,
            "minimumDateTime": "2019-08-24T14:15:22Z",
            "maximumDateTime": "2019-08-24T14:15:22Z"
          },
          "fileDefinition": {
            "blockingStrategy": 0,
            "configurableBlockingStrategy": {
              "shouldTreatFilesWithSensitiveDataAsBlocked": true,
              "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
            }
          },
          "remediations": [
            {
              "id": "string",
              "workflowId": "string",
              "isEnabled": true,
              "configuration": "string",
```

```
"handlingType": 0
            }
          ],
          "tenantId": "string"
        },
        "startTimestamp": 0,
        "stopTimestamp": 0,
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
```

```
"apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "storage": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        },
        "workflowStatistics": {
          "taggingStatistics": {
            "total": 0
          },
          "keepAllowedStatistics": {
            "total": 0
          },
          "keepBlockedStatistics": {
            "total": 0
          },
          "keepSanitizedStatistics": {
            "total": 0
          },
          "movedAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
```

```
"vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedSanitizedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
```

```
"statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
```

```
"tenantId": "string"
            }
          },
          "deleteAllowedStatistics": {
            "total": 0
          },
          "deleteBlockedStatistics": {
            "total": 0
          },
          "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
        },
        "account": {
          "id": "string",
          "name": "string",
          "description": "string",
          "credentials": "string",
          "protocolType": 0,
          "vendorType": 0,
          "autoDiscovery": true,
          "autodiscoveryDestinationGroupId": "string"
        },
        "realTimeStartDate": "2019-08-24T14:15:22Z"
      }
    ],
    "storageId": "string",
    "$type": "string",
    "isRealTimeEnabled": true,
    "handlingType": 0
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetRealTimeProcessingScanQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

ScanInstance

List, add, update and delete Scan Instances

Get Scan Instances

`GET /api/scaninstance`

Parameters

Name	In	Type	Required	Description
scanPoolId	query	string	false	none

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "scanPoolId": "string",
      "url": "string",
      "apiKey": "string",
      "timeout": "string",
      "scanInstanceType": 0
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetAllScanInstancesCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Add a new Scan Instance

`POST /api/scaninstance`

> Body parameter

```
{
  "scanPoolId": "string",
  "url": "http://example.com",
  "apiKey": "string",
  "timeout": "14:15:22Z",
  "scanInstanceType": 0
}
```


Parameters

Name	In	Type	Required	Description
body	body	AddScanInstanceParameters	true	none

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scanInstanceId": "string"
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	AddScanInstanceCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update an existing Scan Instance

`PUT /api/scaninstance`

> Body parameter

```
{
  "id": "string",
  "url": "http://example.com",
  "apiKey": "string",
  "timeout": "14:15:22Z"
}
```


Parameters

Name	In	Type	Required	Description
body	body	UpdateScanInstanceParameters	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateScanInstanceCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get Scan Instance by ID

`GET /api/scaninstance/{id}`

Parameters

Name	In	Type	Required	Description
id	path	string	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entry": {
    "id": "string",
    "scanPoolId": "string",
    "url": "string",
    "apiKey": "string",
    "timeout": "string",
    "scanInstanceType": 0
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetScanInstanceByIdCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Delete a Scan Instance

```
DELETE /api/scaninstance/{id}
```

Parameters

Name	In	Type	Required	Description
id	path	string	true	none

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	DeleteScanInstanceCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

ScanInstance_TestScanInstance

`POST /api/scaninstance/Test`

> Body parameter

```
{
  "url": "string",
  "rules": [
    "string"
  ],
  "apiKey": "string",
  "userAgent": "string",
  "isCloudInstance": true
}
```


Parameters

Name	In	Type	Required	Description
body	body	TestScanInstanceParameters	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	TestScanInstanceCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

ScanPool

List, add, update and delete Scan Pools

Get Scan Pools

`GET /api/scanpool`

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetAllScanPoolsCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Add a new Scan Pool

`POST /api/scanpool`

> Body parameter

```
{
  "name": "string",
  "scanPoolType": 0
}
```

Parameters

Name	In	Type	Required	Description
body	body	AddScanPoolParameters	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scanPoolId": "string"
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	AddScanPoolCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update an existing Scan Pool

`PUT /api/scanpool`

> Body parameter

```
{
  "id": "string",
  "name": "string"
}
```

Parameters

Name	In	Type	Required	Description
body	body	UpdateScanPoolParameters	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateScanPoolCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get Scan Pool by ID

`GET /api/scanpool/{id}`

Parameters

Name	In	Type	Required	Description
id	path	string	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entry": {
    "id": "string",
    "name": "string",
    "scanPoolType": 0,
    "isDefault": true,
    "scanInstances": [
      {
        "id": "string",
        "scanPoolId": "string",
        "url": "string",
        "apiKey": "string",
        "timeout": "string",
        "scanInstanceType": 0
      }
    ]
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetScanPoolByIdCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Delete a Scan Pool

`DELETE /api/scanpool/{id}`

Parameters

Name	In	Type	Required	Description
id	path	string	true	none

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	DeleteScanPoolCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

ScanWorkflowSnapshot

List scan workflow snapshots

Get all scan workflow snapshots by scan ID

`GET /api/scanworkflowsnapshot/getAllFromScan/{scanId}`

Parameters

Name	In	Type	Required	Description
scanId	path	string	true	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scanWorkflowSnapshots": [
    {
      "id": "string",
      "scanId": "string",
      "workflow": {
        "id": "string",
        "scanPoolId": "string",
        "scanProcessFlow": 0,
        "isIdentityScanningEnabled": true,
        "scanIntervalDays": 0,
        "rules": [
          "string"
        ],
        "type": 0,
        "workflowSupportedTechnologies": {
          "dlp": {
            "status": 0
          },
          "deepCdr": {
            "status": 0
          },
          "sandBox": {
            "status": 0
          },
          "metascan": {
            "status": 0
          }
        },
        "name": "string",
        "userAgent": "string",
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
```

```
},
        "failOverScanPoolId": "string",
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "filters": {
          "id": "string",
          "minimumSize": 0,
          "maximumSize": 0,
          "minimumDateTime": "2019-08-24T14:15:22Z",
          "maximumDateTime": "2019-08-24T14:15:22Z"
        },
        "fileDefinition": {
          "blockingStrategy": 0,
          "configurableBlockingStrategy": {
            "shouldTreatFilesWithSensitiveDataAsBlocked": true,
            "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
          }
        },
        "remediations": [
          {
            "id": "string",
            "workflowId": "string",
            "isEnabled": true,
            "configuration": "string",
            "type": 0,
            "handlingType": 0
          }
        ],
        "tenantId": "string"
      },
      "startTimestamp": 0,
      "stopTimestamp": 0,
      "scanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
```

```
"scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "failOverScanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "failOverScanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "storage": {
        "id": "string",
        "name": "string",
        "groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
```

```
"categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      },
      "workflowStatistics": {
        "taggingStatistics": {
          "total": 0
        },
        "keepAllowedStatistics": {
          "total": 0
        },
        "keepBlockedStatistics": {
          "total": 0
        },
        "keepSanitizedStatistics": {
          "total": 0
        },
        "movedAllowedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "movedSanitizedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
```

```
"name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "movedBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "copyAllowedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
```

```
"source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "copyBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "deleteAllowedStatistics": {
          "total": 0
        },
        "deleteBlockedStatistics": {
          "total": 0
        },
        "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
      },
      "account": {
        "id": "string",
        "name": "string",
```

```
"credentials": "string",
        "protocolType": 0,
        "vendorType": 0,
        "autoDiscovery": true,
        "autodiscoveryDestinationGroupId": "string"
      },
      "realTimeStartDate": "2019-08-24T14:15:22Z"
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetScanWorkflowSnapshotsFromScanQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Settings

List or update your settings

Update notifications configuration

`PUT /api/settings/notifications`

> Body parameter

```
{
  "emailNotificationsConfig": {
    "generatedReportNotificationsConfig": {
      "isEnabled": true,
      "recipientsType": 0
    },
    "userRequestNotificationsConfig": {
      "isEnabled": true
    },
    "blockedFileNotificationsConfig": {
      "isEnabled": true,
      "recipientsType": 0
    }
  },
  "webhookNotificationsConfig": {
    "scanCompletedNotificationsConfig": {
      "isEnabled": true
    },
    "fileScannedNotificationConfig": {
      "isEnabled": true,
      "notificationScope": 0
    },
    "destinationAddress": "string",
    "headers": {
      "property1": "string",
      "property2": "string"
    },
    "requestType": 0
  },
  "rabbitMqNotificationsConfig": {
    "fileScannedNotificationConfig": {
      "isEnabled": true,
      "notificationScope": 0
    },
    "rabbitMqUrl": "string",
    "headers": {
      "property1": "string",
      "property2": "string"
    }
  }
}
```


Parameters

Name	In	Type	Required	Description
body	body	NotificationsConfigParameters	true	none

> Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "newNotificationsConfig": {
    "id": "string",
    "emailNotificationsConfig": {
      "userRequestNotificationsConfig": {
        "isEnabled": true
      },
      "generatedReportNotificationsConfig": {
        "isEnabled": true,
        "recipientsType": 0
      },
      "blockedFileNotificationsConfig": {
        "isEnabled": true,
        "recipientsType": 0
      }
    },
    "webhookNotificationsConfig": {
      "scanCompletedNotificationsConfig": {
        "isEnabled": true
      },
      "fileScannedNotificationConfig": {
        "isEnabled": true,
        "notificationScope": 0
      },
      "destinationAddress": "string",
      "headers": {
        "property1": "string",
        "property2": "string"
      },
      "requestType": 0
    },
    "rabbitMqNotificationsConfig": {
      "fileScannedNotificationConfig": {
        "isEnabled": true,
        "notificationScope": 0
      },
      "rabbitMqUrl": "string",
      "headers": {
        "property1": "string",
        "property2": "string"
      }
    }
  },
  "initialNotificationsConfig": {
    "id": "string",
```

```
"userRequestNotificationsConfig": {
        "isEnabled": true
      },
      "generatedReportNotificationsConfig": {
        "isEnabled": true,
        "recipientsType": 0
      },
      "blockedFileNotificationsConfig": {
        "isEnabled": true,
        "recipientsType": 0
      }
    },
    "webhookNotificationsConfig": {
      "scanCompletedNotificationsConfig": {
        "isEnabled": true
      },
      "fileScannedNotificationConfig": {
        "isEnabled": true,
        "notificationScope": 0
      },
      "destinationAddress": "string",
      "headers": {
        "property1": "string",
        "property2": "string"
      },
      "requestType": 0
    },
    "rabbitMqNotificationsConfig": {
      "fileScannedNotificationConfig": {
        "isEnabled": true,
        "notificationScope": 0
      },
      "rabbitMqUrl": "string",
      "headers": {
        "property1": "string",
        "property2": "string"
      }
    }
  },
  "typeResults": [
    {
      "type": 0,
      "status": 0,
      "errorKey": "string",
      "errorMessage": "string"
    }
  ]
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateNotificationsConfigResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get notifications configuration

`GET /api/settings/notifications`

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "emailNotificationsConfig": {
    "generatedReportNotificationsConfig": {
      "isEnabled": true,
      "recipientsType": 0
    },
    "userRequestNotificationsConfig": {
      "isEnabled": true
    },
    "blockedFileNotificationsConfig": {
      "isEnabled": true,
      "recipientsType": 0
    }
  },
  "webhookNotificationsConfig": {
    "scanCompletedNotificationsConfig": {
      "isEnabled": true
    },
    "fileScannedNotificationConfig": {
      "isEnabled": true,
      "notificationScope": 0
    },
    "destinationAddress": "string",
    "headers": {
      "property1": "string",
      "property2": "string"
    },
    "requestType": 0
  },
  "rabbitMqNotificationsConfig": {
    "fileScannedNotificationConfig": {
      "isEnabled": true,
      "notificationScope": 0
    },
    "rabbitMqUrl": "string",
    "headers": {
      "property1": "string",
      "property2": "string"
    }
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	GetNotificationsConfigResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update SMTP configuration

`PATCH /api/settings/smtp`

> Body parameter

```
{
  "host": "string",
  "port": 0,
  "baseUrl": "http://example.com",
  "senderAddress": "string",
  "senderName": "string",
  "domain": "string",
  "username": "string",
  "password": "string",
  "secureSocketOption": 0,
  "ignoreCertWarnings": true,
  "isSmtpEnabled": true
}
```

Parameters

Name	In	Type	Required	Description
body	body	UpdateSmtpConfigParameters	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "newSmtpConfig": {
    "host": "string",
    "baseUrl": "string",
    "port": 0,
    "senderAddress": "string",
    "senderName": "string",
    "domain": "string",
    "username": "string",
    "secureSocketOption": 0,
    "ignoreCertWarnings": true,
    "isSmtpEnabled": true
  },
  "oldSmtpConfig": {
    "host": "string",
    "baseUrl": "string",
    "port": 0,
    "senderAddress": "string",
    "senderName": "string",
    "domain": "string",
    "username": "string",
    "secureSocketOption": 0,
    "ignoreCertWarnings": true,
    "isSmtpEnabled": true
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateSmtpConfigCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch SMTP configuration

`GET /api/settings/smtp`

> Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "host": "string",
  "port": 0,
  "baseUrl": "string",
  "senderAddress": "string",
  "senderName": "string",
  "domain": "string",
  "username": "string",
  "secureSocketOption": 0,
  "ignoreCertWarnings": true,
  "isSmtpEnabled": true
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetSmtpConfigResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Online license activation

`POST /api/settings/admin/license/activate/online`

Body parameter


```
{
  "key": "string"
}
```

Parameters

Name	In	Type	Required	Description
body	body	OnlineActivationParameters	true	none

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	Activation

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Offline license activation

`POST /api/settings/admin/license/activate/offline`

> Body parameter

```
LicenseContent: string
```

Parameters

Name	In	Type	Required	Description
body	body	object	false	none
» LicenseContent	body	string(binary)¦null	false	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	Activation

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get license details

`GET /api/settings/admin/license`

> Example responses

> 200 Response

```
{
  "apiStatus": "string"
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	ApiHealthModel

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Deactivate license

`POST /api/settings/admin/license/deactivate`

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	Deactivation

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update retention configuration

`PUT /api/settings/retention`

> Body parameter

```
{
  "retentionAuditDays": 1,
  "reportRetention": {
    "isEnabled": true,
    "retentionDays": 1
  },
  "rtpHistoryRetention": {
    "isEnabled": true,
    "retentionDays": 1
  }
}
```


Parameters

Name	In	Type	Required	Description
body	body	RetentionConfigParameters	true	none

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	RestResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch retention configuration

`GET /api/settings/retention`

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "retentionAuditDays": 0,
  "reportRetentionConfiguration": {
    "isEnabled": true,
    "retentionDays": 0
  },
  "rtpHistoryRetentionConfiguration": {
    "isEnabled": true,
    "retentionDays": 0
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetRetentionConfig

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Generate encryption key

`POST /api/settings/encryption/generate`

Generate a new encryption key, replacing the old one if it exists.

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GenerateEncryptionKeyCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get encryption key creation date

`GET /api/settings/encryption/creation-date`

Get the creation date of the encryption key.

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "creationDate": "2019-08-24T14:15:22Z"
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetEncryptionKeyCreationDateQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update SSL configuration

`POST /api/settings/ssl`

Update SSL configuration of MDSS by uploading a certificate and a key to enable SSL.

> Body parameter

```
SslCertificate: string
SslKey: string
IsEnabled: true
```

Parameters

Name	In	Type	Required	Description
body	body	object	false	none
» SslCertificate	body	string(binary)¦null	false	none
» SslKey	body	string(binary)¦null	false	none
» IsEnabled	body	boolean¦null	false	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	GenerateEncryptionKeyCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get SSL configuration

`GET /api/settings/ssl`

Get current SSL configuration status.

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "creationDate": "2019-08-24T14:15:22Z"
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetEncryptionKeyCreationDateQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update Single User Session

`POST /api/settings/single-user-session`

Update Simultaneous Sessions allowed for a Single User.

Body parameter

```
{
  "isSingleUserSession": true
}
```

Parameters

Name	In	Type	Required	Description
body	body	PostSingleUserSessionParameters	true	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateSingleUserSessionCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get Single User Session

`GET /api/settings/single-user-session`

Get Simultaneous Sessions allowed for a Single User.

Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "creationDate": "2019-08-24T14:15:22Z"
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetEncryptionKeyCreationDateQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get Telemetry configuration

`GET /api/settings/telemetry`

Get MDSS Open Telemetry configuration.

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "telemetryConfig": {
    "isEnabled": true,
    "sendTelemetryToOpswat": true,
    "autoInstrumentation": true,
    "optelCollectors": [
      {
        "metricEndpoint": {
          "endpointUrl": "string",
          "endpointApiKey": "string",
          "protocol": 0
        },
        "traceEndpoint": {
          "endpointUrl": "string",
          "endpointApiKey": "string",
          "protocol": 0
        },
        "isOpswatCollector": true
      }
    ]
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetTelemetryConfigQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update Telemetry configuration

`POST /api/settings/telemetry`

Update MDSS Open Telemetry configuration.

> Body parameter

```
{
  "telemetryConfig": {
    "isEnabled": true,
    "sendTelemetryToOpswat": true,
    "autoInstrumentation": true,
    "optelCollectors": [
      {
        "metricEndpoint": {
          "endpointUrl": "string",
          "endpointApiKey": "string",
          "protocol": 0
        },
        "traceEndpoint": {
          "endpointUrl": "string",
          "endpointApiKey": "string",
          "protocol": 0
        },
        "isOpswatCollector": true
      }
    ]
  },
  "userInfo": {
    "userId": "string",
    "userName": "string"
  }
}
```


Parameters

Name	In	Type	Required	Description
body	body	UpdateTelemetryConfigCommandRequest	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateTelemetryConfigCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Sso

Sso Authentication and Sso configuration update

Get Sso Configuration

`GET /api/sso/config`

Allows retrieval of the current SSO configuration.

> Example responses

> 200 Response


```
{
  "isEnabled": true,
  "ssoProvider": 0,
  "ssoType": 0,
  "baseUrl": "string",
  "clientId": "string",
  "clientSecret": "string",
  "authority": "string",
  "identityProviderCertificate": "string",
  "identityProviderIssuer": "string",
  "loginUrl": "string",
  "logoutUrl": "string",
  "administratorRoleName": "string",
  "readonlyRoleName": "string"
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	SsoConfig

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update Sso Configuration

`PATCH /api/sso`

Allows updating the current Sso Configuration.

> Body parameter

```
{
  "clientId": "string",
  "clientSecret": "string",
  "authority": "string",
  "baseUrl": "string",
  "identityProviderIssuer": "string",
  "identityProviderLoginUrl": "string",
  "identityProviderLogoutUrl": "string",
  "identityProviderCertificate": {
    "content": "string"
  },
  "isEnabled": true,
  "ssoType": 0,
  "ssoProvider": 0,
  "administratorRoleName": "string",
  "readonlyRoleName": "string"
}
```

Parameters

Name	In	Type	Required	Description
body	body	UpdateSsoConfigParameters	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "clientId": "string",
  "authority": "string",
  "baseUrl": "string",
  "isEnabled": true,
  "ssoType": 0,
  "ssoProvider": 0,
  "identityProviderIssuer": "string",
  "identityProviderLoginUrl": "string",
  "identityProviderLogoutUrl": "string",
  "administratorRoleName": "string",
  "readonlyRoleName": "string",
  "ssoConfig": {
    "isEnabled": true,
    "ssoProvider": 0,
    "ssoType": 0,
    "baseUrl": "string",
    "clientId": "string",
    "clientSecret": "string",
    "authority": "string",
    "identityProviderCertificate": "string",
    "identityProviderIssuer": "string",
    "loginUrl": "string",
    "logoutUrl": "string",
    "administratorRoleName": "string",
    "readonlyRoleName": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	SsoConfigResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Storage

Manage your storage units

Fetch all storage units

`GET /api/storage/all`

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "groupId": "string",
      "accountId": "string",
      "name": "string",
      "storageClientId": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "scansInProgress": true,
      "isRtpInProgress": true,
      "isScheduledScanInProgress": true,
      "storageFunctionality": "string",
      "$type": "string"
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	EnumerateStorages

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch storage by ID

`GET /api/storage/{storageId}`

Parameters

Name	In	Type	Required	Description
storageId	path	string	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "storage": {
    "id": "string",
    "groupId": "string",
    "accountId": "string",
    "name": "string",
    "storageClientId": "string",
    "storageType": {
      "name": "string",
      "categoryType": 0,
      "protocolType": 0,
      "vendorType": 0
    },
    "storageStatus": {
      "message": "string",
      "statusType": 0
    },
    "scansInProgress": true,
    "isRtpInProgress": true,
    "isScheduledScanInProgress": true,
    "storageFunctionality": "string",
    "$type": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetStorage

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update a storage

`PATCH /api/storage/{storageId}`

Note! The following are templates for what is expected in the Credentials, CredentialsFile and Source fields. Please provide the correct values for your storage integration instead of null/false

Alibaba Cloud storage units:

Credentials:

“{"Endpoint":null,"AccessKeyId":null,"AccessKeySecret":null,"UseRamRole":false}”

Source: “{"BucketName":null,"FolderLocation":null}”

Amazon S3 / S3 Compatible storage units:

Credentials:

“{"ServiceUrl":null,"AccessKeyId":null,"SecretAccessKey":null,"RegionEndpoint":null,"AssumeRoleArn":null,"UseIa

Source: “{"BucketName":null,"FolderLocation":null}”

Azure Blob storage units:

Credentials: “{"TenantId":null,"ClientId":null,"ClientSecret":null,"StorageAccount":null}”

Source: “{"Container":null}”

Azure Files storage units:

Credentials: “{"AccountName":null,"AccountKey":null,"ShareName":null}”

Source: “{"FolderLocation":null}”

Box storage units:

CredentialsFile: upload the credentials file

Source: “{"FolderLocation":null}”

Dell Isilon / SMB Compatible storage units:

Credentials: “{"User":null,"Password":null,"Server":null}”

Source: “{"SharePath":null}”

Google Cloud storage units:

CredentialsFile: upload the credentials file

Credentials: “{"UseAdc":false}”

Source: “{"BucketName":null,"FolderLocation":null}”

Graph storage units:

Credentials: “{"TenantId":null,"ClientId":null,"ClientSecret":null}”

Source: “{"Group":null}”

> Body parameter

```
Name: string
StorageVendorType: string
StorageProtocolType: string
CredentialsFile: string
Credentials: string
Source: string
GroupId: string
AccountId: string
WorkflowId: string
TransferMetadataAndChecksum: true
```

Parameters

Name	In	Type	Required	Description
storageId	path	string	true	none
body	body	object	false	none
» Name	body	string¦null	false	none
» StorageVendorType	body	string¦null	false	none
» StorageProtocolType	body	string¦null	false	none
» CredentialsFile	body	string(binary)¦null	false	none
» Credentials	body	string¦null	false	none
» Source	body	string¦null	false	none
» GroupId	body	string¦null	false	none
» AccountId	body	string¦null	false	none
» WorkflowId	body	string¦null	false	none
» TransferMetadataAndChecksum	body	boolean¦null	false	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "updatedStorage": {
    "id": "string",
    "name": "string",
    "groupId": "string",
    "clientId": "string",
    "accountId": "string",
    "source": "string",
    "storageFunctionality": "string",
    "storageType": {
      "name": "string",
      "categoryType": 0,
      "protocolType": 0,
      "vendorType": 0
    },
    "storageStatus": {
      "message": "string",
      "statusType": 0
    },
    "workflowId": "string",
    "lastUpdated": "2019-08-24T14:15:22Z",
    "tenantId": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateNewStorageCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Delete a storage

```
DELETE /api/storage/{storageId}
```

Parameters

Name	In	Type	Required	Description
storageId	path	string	true	none
forceDelete	query	boolean	false	none

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch storage users

`GET /api/storage/{storageId}/users`

Parameters

Name	In	Type	Required	Description
storageId	path	string	true	none
filter	query	string	false	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "fullName": "string"
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	EnumerateStorageUsersQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Add a storage

`POST /api/storage`

Note! The following are templates for what is expected in the Credentials, CredentialsFile and Source fields. Please provide the correct values for your storage integration instead of null/false

Alibaba Cloud storage units:
Credentials:
“{"Endpoint":null,"AccessKeyId":null,"AccessKeySecret":null,"UseRamRole":false}”
Source: “{"BucketName":null,"FolderLocation":null}”

Amazon S3 / S3 Compatible storage units:
Credentials:
“{"ServiceUrl":null,"AccessKeyId":null,"SecretAccessKey":null,"RegionEndpoint":null,"AssumeRoleArn":null,"Usela
Source: “{"BucketName":null,"FolderLocation":null}”

Azure Blob storage units:
Credentials: “{"TenantId":null,"ClientId":null,"ClientSecret":null,"StorageAccount":null}”

Source: “{"Container":null}”

Azure Files storage units:
Credentials: “{"AccountName":null,"AccountKey":null,"ShareName":null}”
Source: “{"FolderLocation":null}”

Box storage units:
CredentialsFile: upload the credentials file
Source: “{"FolderLocation":null}”

Dell Isilon / SMB Compatible storage units:
Credentials: “{"User":null,"Password":null,"Server":null}”
Source: “{"SharePath":null}”

Google Cloud storage units:
CredentialsFile: upload the credentials file
Credentials: “{"UseAdc":false}”
Source: “{"BucketName":null,"FolderLocation":null}”

Graph storage units:
Credentials: “{"TenantId":null,"ClientId":null,"ClientSecret":null}”
Source: “{"Group":null}”

> Body parameter

```
Name: string
StorageVendorType: string
StorageProtocolType: string
CredentialsFile: string
Credentials: string
Source: string
GroupId: string
AccountId: string
WorkflowId: string
TransferMetadataAndChecksum: true
```

Parameters

Name	In	Type	Required	Description
body	body	object	false	none
» Name	body	string	false	none
» StorageVendorType	body	string¦null	false	none
» StorageProtocolType	body	string¦null	false	none
» CredentialsFile	body	string(binary)¦null	false	none
» Credentials	body	string¦null	false	none
» Source	body	string	false	none
» GroupId	body	string¦null	false	none
» AccountId	body	string¦null	false	none
» WorkflowId	body	string¦null	false	none
» TransferMetadataAndChecksum	body	boolean¦null	false	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "storageId": "string"
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	AddNewStorageCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Add batch of storages

`POST /api/storage/batch`

Note! The following are templates for what is expected in the Credentials, CredentialsFile and Source fields. Please provide the correct values for your storage integration instead of null/false

Alibaba Cloud storage units:
Credentials:
“{"Endpoint":null,"AccessKeyId":null,"AccessKeySecret":null,"UseRamRole":false}”
Source: “{"BucketName":null,"FolderLocation":null}”

Amazon S3 / S3 Compatible storage units:
Credentials:
“{"ServiceUrl":null,"AccessKeyId":null,"SecretAccessKey":null,"RegionEndpoint":null,"AssumeRoleArn":null,"UseIa
Source: “{"BucketName":null,"FolderLocation":null}”

Azure Blob storage units:
Credentials: “{"TenantId":null,"ClientId":null,"ClientSecret":null,"StorageAccount":null}”
Source: “{"Container":null}”

Azure Files storage units:
Credentials: “{"AccountName":null,"AccountKey":null,"ShareName":null}”
Source: “{"FolderLocation":null}”

Box storage units:
CredentialsFile: upload the credentials file
Source: “{"FolderLocation":null}”

Dell Isilon / SMB Compatible storage units:
Credentials: “{"User":null,"Password":null,"Server":null}”
Source: “{"SharePath":null}”

Google Cloud storage units:
CredentialsFile: upload the credentials file
Credentials: “{"UseAdc":false}”
Source: “{"BucketName":null,"FolderLocation":null}”

Graph storage units:

Credentials: “{"TenantId":null,"ClientId":null,"ClientSecret":null}”
Source: “{"Group":null}”

> Body parameter

```
{
  "addStorageParametersList": [
    {
      "name": "string",
      "storageVendorType": 0,
      "storageProtocolType": 0,
      "credentialsFile": "string",
      "credentials": "string",
      "source": "string",
      "groupId": "string",
      "accountId": "string",
      "workflowId": "string",
      "transferMetadataAndChecksum": true
    }
  ]
}
```

Parameters

Name	In	Type	Required	Description
body	body	AddBatchStorageParameters	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "storageId": "string"
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	AddNewStorageCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get batch of add storages results

`GET /api/storage/results`

Parameters

Name	In	Type	Required	Description
token	query	string	false	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "addStorageResultInfos": [
    {
      "result": 0,
      "responseKey": "string",
      "responseMessage": "string",
      "responseMessageParams": {
        "property1": "string",
        "property2": "string"
      },
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "source": "string",
      "name": "string",
      "vendorType": 0,
      "protocolType": 0
    }
  ],
  "isCompleted": true
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetBatchResultsResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch all storage units by filters (storage type or status)

`GET /api/storage/enumerate/{startIndex}/{count}`

Parameters

Name	In	Type	Required	Description
startIndex	path	string	true	none
count	path	string	true	none
sortingOrder	query	string	false	none
search	query	string	false	none
storageType	query	integer(int32)	false	none
storageStatus	query	integer(int32)	false	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "groupId": "string",
      "accountId": "string",
      "name": "string",
      "storageClientId": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "scansInProgress": true,
      "isRtpInProgress": true,
      "isScheduledScanInProgress": true,
      "storageFunctionality": "string",
      "$type": "string"
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	EnumerateStorages2

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Add a scan schedule

`PUT /api/storage/{storageId}/scan_schedule`

> Body parameter

```
{
  "name": "string",
  "workflowId": "string",
  "recurrencePattern": {
    "type": "string",
    "minutelyRecurrence": {
      "everyNumberOfMinutes": 5
    },
    "hourlyRecurrence": {
      "everyNumberOfHours": 1
    },
    "dailyRecurrence": {
      "everyNumberOfDays": 1
    },
    "weeklyRecurrence": {
      "daysOfWeek": [
        0
      ],
      "everyNumberOfWeeks": 0
    },
    "monthlyRecurrence": {
      "everyNumberOfMonths": 1,
      "dayOfMonth": 1
    },
    "startDate": "2019-08-24T14:15:22Z",
    "timeToStart": "2019-08-24T14:15:22Z",
    "avoidConcurrentScheduleRuns": true
  },
  "scanPartition": "string",
  "priority": 0,
  "$type": "string"
}
```


Parameters

Name	In	Type	Required	Description
storageId	path	string	true	none
body	body	AddScanScheduleParameters	true	none

Example responses

200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update a scan schedule

`PUT /api/storage/scan_schedule/update/{storageId}`

> Body parameter

```
{
  "name": "string",
  "workflowId": "string",
  "recurrencePattern": {
    "type": "string",
    "minutelyRecurrence": {
      "everyNumberOfMinutes": 5
    },
    "hourlyRecurrence": {
      "everyNumberOfHours": 1
    },
    "dailyRecurrence": {
      "everyNumberOfDays": 1
    },
    "weeklyRecurrence": {
      "daysOfWeek": [
        0
      ],
      "everyNumberOfWeeks": 0
    },
    "monthlyRecurrence": {
      "everyNumberOfMonths": 1,
      "dayOfMonth": 1
    },
    "startDate": "2019-08-24T14:15:22Z",
    "timeToStart": "2019-08-24T14:15:22Z",
    "avoidConcurrentScheduleRuns": true
  },
  "scanPartition": "string",
  "priority": 0,
  "$type": "string",
  "id": "string"
}
```

Parameters

Name	In	Type	Required	Description
storageId	path	string	true	none
body	body	UpdateScanScheduleParameters	true	none

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateScanScheduleCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch scan schedules

`GET /api/storage/scan_schedules/{sortDirection}/{startIndex}/{count}`

Parameters

Name	In	Type	Required	Description
sortDirection	path	SortDirectionFilter	true	none
startIndex	path	integer(int32)	true	none
count	path	integer(int32)	true	none
name	query	string	false	none
user	query	string	false	none
type	query	string	false	none
storageId	query	string	false	none

Enumerated Values

Parameter	Value
sortDirection	0
sortDirection	1

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "name": "string",
      "workflowId": "string",
      "scanConfigurationId": "string",
      "type": "string",
      "scanPartition": "string",
      "createdAt": "2019-08-24T14:15:22Z",
      "lastScanTime": "2019-08-24T14:15:22Z",
      "nextScanTime": "2019-08-24T14:15:22Z",
      "recurrencePatternDto": {
        "type": 0,
        "minutelyRecurrence": {
          "everyNumberOfMinutes": 0
        },
        "hourlyRecurrence": {
          "everyNumberOfHours": 0
        },
        "dailyRecurrence": {
          "everyNumberOfDays": 0
        },
        "weeklyRecurrence": {
          "daysOfWeek": [
            0
          ],
          "everyNumberOfWeeks": 0
        },
        "monthlyRecurrence": {
          "everyNumberOfMonths": 0,
          "dayOfMonth": 0
        },
        "startDate": "2019-08-24T14:15:22Z",
        "timeToStart": "2019-08-24T14:15:22Z",
        "avoidConcurrentScheduleRuns": true
      },
      "createdBy": "string",
      "priority": 0,
      "storageId": "string"
    }
  ],
  "totalCount": 0
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	EnumerateScanSchedules

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Delete scan schedules

`DELETE /api/storage/scan_schedule`

> Body parameter

```
[
  "string"
]
```

Parameters

Name	In	Type	Required	Description
body	body	array[string]	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "areAllDeleted": true
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	DeleteScanSchedulesCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Retrieve storage user name

`GET /api/storage/{storageId}/{userId}`

Parameters

Name	In	Type	Required	Description
storageId	path	string	true	none
userId	path	string	true	none

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "userName": "string"
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	RetrieveUserNameFromStorageQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

User

Create, list and update user accounts

Fetch active users

`GET /api/user/active/{startIndex}/{count}`

Parameters

Name	In	Type	Required	Description
startIndex	path	string	true	none
count	path	string	true	none
searchTerm	query	string	false	none
orderType	query	any	false	none
orderBy	query	any	false	none
searchText	query	string	false	none

> Example responses

> 200 Response

```
{
  "entries": [
    {
      "id": "string",
      "userName": "string",
      "fullName": "string",
      "email": "string",
      "roles": [
        {
          "bindingId": "string",
          "roleId": "string",
          "roleName": "string",
          "expiresAt": "2019-08-24T14:15:22Z"
        }
      ],
      "directPermissions": [
        {
          "id": "string",
          "permissionName": "string",
          "expiresAt": "2019-08-24T14:15:22Z"
        }
      ]
    }
  ],
  "totalCount": 0
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	EnumerateActiveUsers

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Register a user

`POST /api/user/register`

> Body parameter

```
{
  "fullName": "string",
  "userName": "string",
  "password": "string",
  "email": "string",
  "ssoUserId": "string"
}
```

Parameters

Name	In	Type	Required	Description
body	body	RegisterUserParameters	true	none

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

This operation does not require authentication

Create a user

`POST /api/user/create`

> Body parameter

```
{
  "fullName": "string",
  "userName": "string",
  "password": "string",
  "email": "string",
  "ssoUserId": "string",
  "role": 0
}
```

Parameters

Name	In	Type	Required	Description
body	body	CreateUserParameters	true	none

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Authenticate with a username and password

`POST /api/user/authenticate`

Authenticate with a username and password to obtain a JWT token,Most of the APIs require authentication in the form of providing a JWT token. Call this API in order to receive a token but please note that it's only valid for an hour so it should be periodically refreshed.

> Body parameter

```
{
  "userName": "string",
  "password": "string"
}
```

Parameters

Name	In	Type	Required	Description
body	body	AuthenticateUserCommandRequest	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "fullName": "string",
  "email": "string",
  "userName": "string",
  "role": 0,
  "user": {
    "id": "string",
    "userName": "string",
    "fullName": "string",
    "email": "string",
    "role": 0,
    "roles": [
      "string"
    ],
    "tenantId": "string",
    "ssoUserId": "string",
    "passwordHash": "string",
    "passwordSalt": "string",
    "passwordHashAlgorithm": 0
  },
  "userId": "string"
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	AuthenticateUserCommandResponseWithoutTokens

This operation does not require authentication

Fetch current user information

`GET /api/user/me`

Retrieve information about the currently authenticated user.

Example responses

200 Response


```
{
  "username": "string",
  "email": "string",
  "authProvider": "string",
  "role": "string",
  "id": "string",
  "roles": [
    {
      "bindingId": "string",
      "roleId": "string",
      "roleName": "string",
      "expiresAt": "2019-08-24T14:15:22Z"
    }
  ],
  "directPermissions": [
    {
      "id": "string",
      "permissionName": "string",
      "expiresAt": "2019-08-24T14:15:22Z"
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	CurrentUserInfo

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Refresh user token

`POST /api/user/refreshToken`

Refresh an expired accessToken using a valid refreshToken

Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

This operation does not require authentication

Logout

`POST /api/user/logout`

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update user role

`PUT /api/user/role`

> Body parameter

```
{
  "userId": "string",
  "role": 0
}
```

Parameters

Name	In	Type	Required	Description
body	body	UpdateUserRoleParameters	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "user": {
    "id": "string",
    "userName": "string",
    "fullName": "string",
    "email": "string",
    "role": 0,
    "roles": [
      "string"
    ],
    "tenantId": "string",
    "ssoUserId": "string",
    "passwordHash": "string",
    "passwordSalt": "string",
    "passwordHashAlgorithm": 0
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateUserCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Fetch users

`GET /api/user/requests`

Parameters

Name	In	Type	Required	Description
startIndex	query	integer(int32)	false	none
count	query	integer(int32)	false	none
orderType	query	any	false	none
orderBy	query	any	false	none
searchText	query	string	false	none

Example responses

200 Response


```
{
  "entries": [
    {
      "id": "string",
      "userName": "string",
      "fullName": "string",
      "email": "string",
      "role": 0,
      "roles": [
        "string"
      ],
      "tenantId": "string",
      "ssoUserId": "string",
      "passwordHash": "string",
      "passwordSalt": "string",
      "passwordHashAlgorithm": 0
    }
  ],
  "totalCount": 0
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	EnumerateUsers

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Delete a user

`DELETE /api/user/{userId}`

Parameters

Name	In	Type	Required	Description
userId	path	string	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "user": {
    "id": "string",
    "userName": "string",
    "fullName": "string",
    "email": "string",
    "role": 0,
    "roles": [
      "string"
    ],
    "tenantId": "string",
    "ssoUserId": "string",
    "passwordHash": "string",
    "passwordSalt": "string",
    "passwordHashAlgorithm": 0
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	UserCleanupCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update current user

`PUT /api/user`

> Body parameter

```
{
  "userId": "string",
  "role": 0,
  "userName": "string",
  "fullName": "string",
  "currentPassword": "string",
  "newPassword": "string",
  "email": "string"
}
```

Parameters

Name	In	Type	Required	Description
body	body	UpdateUserParameters	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "user": {
    "id": "string",
    "userName": "string",
    "fullName": "string",
    "email": "string",
    "role": 0,
    "roles": [
      "string"
    ],
    "tenantId": "string",
    "ssoUserId": "string",
    "passwordHash": "string",
    "passwordSalt": "string",
    "passwordHashAlgorithm": 0
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateUserCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Request password reset

`POST /api/user/password/reset`

> Body parameter


```
{
  "userInput": "string"
}
```

Parameters

Name	In	Type	Required	Description
body	body	RequestResetPasswordCommandRequest	true	none

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	RequestResetPasswordCommandResponse

This operation does not require authentication

Reset user password

`POST /api/user/password`

> Body parameter


```
{
  "newPassword": "stringstring",
  "newPasswordConfirmation": "stringstring",
  "secureToken": "string"
}
```

Parameters

Name	In	Type	Required	Description
body	body	UpdatePasswordCommandRequest	true	none

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "user": {
    "id": "string",
    "userName": "string",
    "fullName": "string",
    "email": "string",
    "role": 0,
    "roles": [
      "string"
    ],
    "tenantId": "string",
    "ssoUserId": "string",
    "passwordHash": "string",
    "passwordSalt": "string",
    "passwordHashAlgorithm": 0
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	UpdatePasswordCommandResponse

This operation does not require authentication

Get User Permissions

```
GET /api/user/{userId}/permissions
```

Parameters

Name	In	Type	Required	Description
userId	path	string	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "roles": [
    {
      "bindingId": "string",
      "roleId": "string",
      "roleName": "string",
      "expiresAt": "2019-08-24T14:15:22Z"
    }
  ],
  "directPermissions": [
    {
      "id": "string",
      "permissionName": "string",
      "expiresAt": "2019-08-24T14:15:22Z"
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetUserPermissionsQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

User Tour

Finalize and list user tours

Fetch tours completed by a user

`GET /api/usertour/{userId}`

Parameters

Name	In	Type	Required	Description
userId	path	string	true	none

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "userId": "string",
  "userTours": [
    {
      "tourType": 0,
      "finalizedDate": "2019-08-24T14:15:22Z"
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetUserTourQueryResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Mark a specific Tour as completed by a user

`POST /api/usertour/finalizeUserTour`

> Body parameter

```
{
  "userId": "string",
  "userTourType": 0
}
```

Parameters

Name	In	Type	Required	Description
body	body	FinalizeUserTour	true	none

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Webhook

File event trigger of real time processing

`POST /api/webhook/realtime`

This endpoint takes the parameters and tries to find the file in the specified storage and applies the workflow for the specified file

> Body parameter

```
{
  "metadata": "string",
  "storageClientId": "string",
  "scanId": "string"
}
```


Parameters

Name	In	Type	Required	Description
body	body	WebhookEventBasedRtpParameters	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "correlationId": "string"
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	WebhookEventBasedRtpCommandResponse
400	Bad Request	none	WebhookEventBasedRtpCommandResponse
404	Not Found	none	WebhookEventBasedRtpCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

File event trigger of real time processing by storage client ID

`POST /api/webhook/realtime/{storageClientId}/{scanId}`

This endpoint takes the parameter and tries to find the file in the specified storage and applies the workflow for the specified file

> Body parameter

```
null
```

Parameters

Name	In	Type	Required	Description
storageClientId	path	string	true	none
scanId	path	string	true	none
body	body	any	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "correlationId": "string"
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	WebhookEventBasedRtpCommandResponse
400	Bad Request	none	WebhookEventBasedRtpCommandResponse
404	Not Found	none	WebhookEventBasedRtpCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

File event trigger of real time processing for Box

`POST /api/webhook/realtime/public/{tenantId}/{storageClientId}/{scanId}`

This endpoint takes the parameter and tries to find the file in the specified Box storage and applies the workflow for the specified file

> Body parameter

```
null
```

Parameters

Name	In	Type	Required	Description
storageClientId	path	string	true	none
scanId	path	string	true	none
tenantId	path	string	true	none
body	body	any	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "correlationId": "string"
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	WebhookEventBasedRtpCommandResponse
400	Bad Request	none	WebhookEventBasedRtpCommandResponse
404	Not Found	none	WebhookEventBasedRtpCommandResponse

This operation does not require authentication

Webhook_EventHandlerGraphNotification

`POST /api/webhook/realtime/graph/{storageClientId}/{scanId}`

> Body parameter

```
null
```

Parameters

Name	In	Type	Required	Description
storageClientId	path	string	true	none
scanId	path	string	true	none
validationToken	query	string	false	none
body	body	any	false	none

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

This operation does not require authentication

Webhook_EventHandlerGraphNotification2

`POST /api/webhook/realtime/graph/{tenantId}/{storageClientId}/{scanId}`

> Body parameter

```
null
```

Parameters

Name	In	Type	Required	Description
storageClientId	path	string	true	none
scanId	path	string	true	none
validationToken	query	string	false	none
tenantId	path	string	true	none
body	body	any	false	none

> Example responses

> 200 Response

Responses

Status	Meaning	Description	Schema
200	OK	none	string

This operation does not require authentication

Workflow

List, add, update and delete Workflows

Get Workflows

`GET /api/workflow`

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "scanPoolId": "string",
      "scanProcessFlow": 0,
      "isIdentityScanningEnabled": true,
      "scanIntervalDays": 0,
      "rules": [
        "string"
      ],
      "type": 0,
      "workflowSupportedTechnologies": {
        "dlp": {
          "status": 0
        },
        "deepCdr": {
          "status": 0
        },
        "sandBox": {
          "status": 0
        },
        "metascan": {
          "status": 0
        }
      },
      "name": "string",
      "userAgent": "string",
      "scanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "failOverScanPoolId": "string",
```

```
"id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "filters": {
        "id": "string",
        "minimumSize": 0,
        "maximumSize": 0,
        "minimumDateTime": "2019-08-24T14:15:22Z",
        "maximumDateTime": "2019-08-24T14:15:22Z"
      },
      "fileDefinition": {
        "blockingStrategy": 0,
        "configurableBlockingStrategy": {
          "shouldTreatFilesWithSensitiveDataAsBlocked": true,
          "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
        }
      },
      "remediations": [
        {
          "id": "string",
          "workflowId": "string",
          "isEnabled": true,
          "configuration": "string",
          "type": 0,
          "handlingType": 0
        }
      ],
      "tenantId": "string"
    }
  ]
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetAllWorkflowsCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Add a new Workflow

`POST /api/workflow`

> Body parameter

```
{
  "scanPoolId": "string",
  "rules": [
    "string"
  ],
  "name": "string",
  "userAgent": "string",
  "failOverScanPoolId": "string",
  "filters": {
    "minimumSize": 0,
    "maximumSize": 0,
    "minimumDateTime": "2019-08-24T14:15:22Z",
    "maximumDateTime": "2019-08-24T14:15:22Z"
  },
  "scanProcessFlow": 0,
  "isIdentityScanningEnabled": true,
  "scanIntervalDays": 0
}
```

Parameters

Name	In	Type	Required	Description
body	body	AddWorkflowParameters	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "workflowId": "string"
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	AddWorkflowCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Update an existing Workflow

`PATCH /api/workflow`

> Body parameter

```
{
  "id": "string",
  "scanPoolId": "string",
  "scanProcessFlow": 0,
  "rules": [
    "string"
  ],
  "name": "string",
  "userAgent": "string",
  "failOverScanPoolId": "string",
  "failOverDisabled": true,
  "filters": {
    "minimumSize": 0,
    "maximumSize": 0,
    "minimumDateTime": "2019-08-24T14:15:22Z",
    "maximumDateTime": "2019-08-24T14:15:22Z"
  },
  "fileDefinition": {
    "blockingStrategy": 0,
    "configurableBlockingStrategy": {
      "shouldTreatFilesWithSensitiveDataAsBlocked": true,
      "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
    }
  },
  "isIdentityScanningEnabled": true,
  "scanIntervalDays": 0
}
```

Parameters

Name	In	Type	Required	Description
body	body	UpdateWorkflowParameters	true	none

> Example responses

> 200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	UpdateWorkflowCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Get Workflow by ID

`GET /api/workflow/{id}`

Parameters

Name	In	Type	Required	Description
id	path	string	true	none

> Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entry": {
    "id": "string",
    "scanPoolId": "string",
    "scanProcessFlow": 0,
    "isIdentityScanningEnabled": true,
    "scanIntervalDays": 0,
    "rules": [
      "string"
    ],
    "type": 0,
    "workflowSupportedTechnologies": {
      "dlp": {
        "status": 0
      },
      "deepCdr": {
        "status": 0
      },
      "sandBox": {
        "status": 0
      },
      "metascan": {
        "status": 0
      }
    },
    "name": "string",
    "userAgent": "string",
    "scanPool": {
      "id": "string",
      "name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    },
    "failOverScanPoolId": "string",
    "failOverScanPool": {
```

```
"name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    },
    "filters": {
      "id": "string",
      "minimumSize": 0,
      "maximumSize": 0,
      "minimumDateTime": "2019-08-24T14:15:22Z",
      "maximumDateTime": "2019-08-24T14:15:22Z"
    },
    "fileDefinition": {
      "blockingStrategy": 0,
      "configurableBlockingStrategy": {
        "shouldTreatFilesWithSensitiveDataAsBlocked": true,
        "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
      }
    },
    "remediations": [
      {
        "id": "string",
        "workflowId": "string",
        "isEnabled": true,
        "configuration": "string",
        "type": 0,
        "handlingType": 0
      }
    ],
    "tenantId": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	GetWorkflowByIdCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Delete a Workflow

`DELETE /api/workflow/{id}`

Parameters

Name	In	Type	Required	Description
id	path	string	true	none

> Example responses

> 200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	none	DeleteWorkflowCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Set a Workflow as default

`POST /api/workflow/setDefault/{id}`

Parameters

Name	In	Type	Required	Description
id	path	string	true	none

Example responses

200 Response


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	SetDefaultWorkflowCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Refresh supported technologies for a Workflow and return updated Workflow

`POST /api/workflow/enabledTechnologies`

Body parameter


```
{
  "workflowId": "string"
}
```


Parameters

Name	In	Type	Required	Description
body	body	RefreshSupportedTechnologiesCommandRequest	true	none

Example responses

200 Response

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "workflowDto": {
    "id": "string",
    "scanPoolId": "string",
    "scanProcessFlow": 0,
    "isIdentityScanningEnabled": true,
    "scanIntervalDays": 0,
    "rules": [
      "string"
    ],
    "type": 0,
    "workflowSupportedTechnologies": {
      "dlp": {
        "status": 0
      },
      "deepCdr": {
        "status": 0
      },
      "sandBox": {
        "status": 0
      },
      "metascan": {
        "status": 0
      }
    },
    "name": "string",
    "userAgent": "string",
    "scanPool": {
      "id": "string",
      "name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    },
    "failOverScanPoolId": "string",
    "failOverScanPool": {
```

```
"name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    },
    "filters": {
      "id": "string",
      "minimumSize": 0,
      "maximumSize": 0,
      "minimumDateTime": "2019-08-24T14:15:22Z",
      "maximumDateTime": "2019-08-24T14:15:22Z"
    },
    "fileDefinition": {
      "blockingStrategy": 0,
      "configurableBlockingStrategy": {
        "shouldTreatFilesWithSensitiveDataAsBlocked": true,
        "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
      }
    },
    "remediations": [
      {
        "id": "string",
        "workflowId": "string",
        "isEnabled": true,
        "configuration": "string",
        "type": 0,
        "handlingType": 0
      }
    ],
    "tenantId": "string"
  }
}
```


Responses

Status	Meaning	Description	Schema
200	OK	none	RefreshSupportedTechnologiesCommandResponse

To perform this operation, you must be authenticated by means of one of the following methods: Bearer

Schemas

GetRolesQueryResponse


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "roles": [
    {
      "id": "string",
      "name": "string",
      "permissions": [
        {
          "id": "string",
          "action": "string",
          "resource": "string"
        }
      ]
    }
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» roles	[Role]¦null	false	none	none

Role

```
{
  "id": "string",
  "name": "string",
  "permissions": [
    {
      "id": "string",
      "action": "string",
      "resource": "string"
    }
  ]
}
```


Properties

Name	Type	Required	Restrictions	Description
id	string(guid)	false	none	none
name	string	false	none	none
permissions	[Permission]¦null	false	none	none

Permission


```
{
  "id": "string",
  "action": "string",
  "resource": "string"
}
```


Properties

Name	Type	Required	Restrictions	Description
id	string(guid)	false	none	none
action	string	false	none	none
resource	string	false	none	none

RestResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
result	ResultType	false	none	- 0: Success - 1: Error - 2: InternalError - 3: NotFound - 4: Forbidden
responseKey	string¦null	false	none	none
responseMessage	string¦null	false	none	none
responseMessageParams	object¦null	false	none	none
» **additionalProperties**	string	false	none	none

ResultType

```
0
```

- 0: Success
- 1: Error
- 2: InternalError
- 3: NotFound
- 4: Forbidden

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Success - 1: Error - 2: InternalError - 3: NotFound - 4: Forbidden

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4

GetRoleByIdQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "role": {
    "id": "string",
    "name": "string",
    "permissions": [
      {
        "id": "string",
        "action": "string",
        "resource": "string"
      }
    ]
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» role	Role¦null	false	none	none

CreateRoleCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "role": {
    "id": "string",
    "name": "string",
    "permissions": [
      {
        "id": "string",
        "action": "string",
        "resource": "string"
      }
    ]
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» role	Role¦null	false	none	none

CreateRoleParameters

```
{
  "name": "string",
  "permissionIds": [
    "string"
  ],
  "templateRoleId": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
name	string	true	none	none
permissionIds	[string]¦null	false	none	none
templateRoleId	string(guid)¦null	false	none	none

UpdateRoleCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

UpdateRoleParameters

```
{
  "name": "string",
  "permissionIds": [
    "string"
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
name	string	true	none	none
permissionIds	[string]¦null	false	none	none

DeleteRoleCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

GetPermissionsQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "permissions": [
    {
      "id": "string",
      "action": "string",
      "resource": "string"
    }
  ]
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» permissions	[Permission]¦null	false	none	none

CreatePermissionCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "permission": {
    "id": "string",
    "action": "string",
    "resource": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» permission	Permission¦null	false	none	none

CreatePermissionParameters

```
{
  "action": "string",
  "resource": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
action	string	true	none	none
resource	string	true	none	none

DeletePermissionCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

AssignPermissionToRoleCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

RolePermissionParameters


```
{
  "permissionId": "string"
}
```


Properties

Name	Type	Required	Restrictions	Description
permissionId	string(guid)	true	none	none

RemovePermissionFromRoleCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

AssignRoleToUserCommandResponse


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "roleBindingId": "string"
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» roleBindingId	string(guid)¦null	false	none	none

AssignRoleToUserParameters

```
{
  "roleId": "string",
  "expiresAt": "2019-08-24T14:15:22Z"
}
```

Properties

Name	Type	Required	Restrictions	Description
roleId	string(guid)	true	none	none
expiresAt	string(date-time)¦null	false	none	none

ReplaceUserRolesCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

ReplaceUserRolesParameters

```
{
  "roleIds": [
    "string"
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
roleIds	[string]	true	none	none

RemoveRoleFromUserCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

AssignPermissionToUserCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "userPermissionId": "string"
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» userPermissionId	string(guid)¦null	false	none	none

AssignPermissionToUserParameters

```
{
  "permissionId": "string",
  "expiresAt": "2019-08-24T14:15:22Z"
}
```

Properties

Name	Type	Required	Restrictions	Description
permissionId	string(guid)	true	none	none
expiresAt	string(date-time)¦null	false	none	none

RemovePermissionFromUserCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

AddAccountCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "account": {
    "id": "string",
    "name": "string",
    "description": "string",
    "credentials": "string",
    "protocolType": 0,
    "vendorType": 0,
    "autoDiscovery": true,
    "autodiscoveryDestinationGroupId": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» account	Account¦null	false	none	none

Account

```
{
  "id": "string",
  "name": "string",
  "description": "string",
  "credentials": "string",
  "protocolType": 0,
  "vendorType": 0,
  "autoDiscovery": true,
  "autodiscoveryDestinationGroupId": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
name	string	false	none	none
description	string	false	none	none
credentials	string	false	none	none
protocolType	StorageProtocolType	false	none	- 0: Smb - 1: GraphSdk - 2: BoxSdk - 3: AwsSdk - 4: AzureBlobSdk - 5: AlibabaCloudSdk - 6: GoogleCloudSdk - 7: AzureFilesSdk - 8: Sftp - 9: Nfs - 10: Mft - 11: SharepointOnPrem - 12: Ftp - 13: OracleSdk - 100: GitHubSdk - 101: DockerHubSdk - 102: BitbucketSdk - 103: AmazonEcrSdk - 104: QuaySdk - 105: JFrogContainerSdk - 106: JFrogBinarySdk - 107: AzureAcrSdk - 108: GitLabSourceSdk - 109: GitLabContainerSdk - 110: SvnSdk - 111: AzureDevOpsSourceSdk - 112: AzureDevOpsBinarySdk - 113: DirectFile - 114: NexusBinarySdk - 115: GiteaSdk
vendorType	StorageVendorType¦null	false	none	- 0: AmazonS3 - 1: OneDrive - 2: Box - 3: DellIsilon - 4: AzureFiles - 5: SmbCompatible - 6: S3Compatible - 7: AzureBlob - 8: AlibabaCloud

				- 9: GoogleCloud - 10: Sftp - 11: SharePoint - 12: Nfs - 13: Wasabi - 14: OracleCloud - 15: NetAppOntap - 16: Cubbit - 17: Mft - 18: SharepointOnPrem - 19: Ftp - 20: Oracle - 21: NfsSisl - 22: ScalityRing - 100: GitHub - 101: DockerHub - 102: Bitbucket - 103: AmazonEcr - 104: Quay - 105: JFrogContainer - 106: JFrogBinary - 107: AzureAcr - 108: GitLabSource - 109: GitLabContainer - 110: Svn - 111: AzureDevOpsSource - 112: AzureDevOpsBinary - 113: DirectFile - 114: NexusBinary - 115: Gitea
autoDiscovery	boolean	false	none	none
autodiscoveryDestinationGroupId	string	false	none	none

StorageProtocolType

```
0
```

- 0: Smb
- 1: GraphSdk
- 2: BoxSdk
- 3: AwsSdk
- 4: AzureBlobSdk
- 5: AlibabaCloudSdk

- 6: GoogleCloudSdk
- 7: AzureFilesSdk
- 8: Sftp
- 9: Nfs
- 10: Mft
- 11: SharepointOnPrem
- 12: Ftp
- 13: OracleSdk
- 100: GitHubSdk
- 101: DockerHubSdk
- 102: BitbucketSdk
- 103: AmazonEcrSdk
- 104: QuaySdk
- 105: JFrogContainerSdk
- 106: JFrogBinarySdk
- 107: AzureAcrSdk
- 108: GitLabSourceSdk
- 109: GitLabContainerSdk
- 110: SvnSdk
- 111: AzureDevOpsSourceSdk
- 112: AzureDevOpsBinarySdk
- 113: DirectFile
- 114: NexusBinarySdk
- 115: GiteaSdk

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Smb - 1: GraphSdk - 2: BoxSdk - 3: AwsSdk - 4: AzureBlobSdk - 5: AlibabaCloudSdk - 6: GoogleCloudSdk - 7: AzureFilesSdk - 8: Sftp - 9: Nfs - 10: Mft - 11: SharepointOnPrem - 12: Ftp - 13: OracleSdk - 100: GitHubSdk - 101: DockerHubSdk - 102: BitbucketSdk - 103: AmazonEcrSdk - 104: QuaySdk - 105: JFrogContainerSdk - 106: JFrogBinarySdk - 107: AzureAcrSdk - 108: GitLabSourceSdk - 109: GitLabContainerSdk - 110: SvnSdk - 111: AzureDevOpsSourceSdk - 112: AzureDevOpsBinarySdk - 113: DirectFile - 114: NexusBinarySdk - 115: GiteaSdk

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4
anonymous	5
anonymous	6
anonymous	7
anonymous	8
anonymous	9
anonymous	10
anonymous	11
anonymous	12
anonymous	13
anonymous	100
anonymous	101
anonymous	102
anonymous	103
anonymous	104
anonymous	105
anonymous	106
anonymous	107
anonymous	108
anonymous	109
anonymous	110
anonymous	111
anonymous	112

anonymous	113
anonymous	114
anonymous	115

StorageVendorType

```
0
```

- 0: AmazonS3
- 1: OneDrive
- 2: Box
- 3: DellIsilon
- 4: AzureFiles
- 5: SmbCompatible
- 6: S3Compatible
- 7: AzureBlob
- 8: AlibabaCloud
- 9: GoogleCloud
- 10: Sftp
- 11: SharePoint
- 12: Nfs
- 13: Wasabi
- 14: OracleCloud
- 15: NetAppOntap
- 16: Cubbit
- 17: Mft
- 18: SharepointOnPrem
- 19: Ftp
- 20: Oracle
- 21: NfsSisl
- 22: ScalityRing
- 100: GitHub
- 101: DockerHub
- 102: Bitbucket
- 103: AmazonEcr

- 104: Quay
- 105: JFrogContainer
- 106: JFrogBinary
- 107: AzureAcr
- 108: GitLabSource
- 109: GitLabContainer
- 110: Svn
- 111: AzureDevOpsSource
- 112: AzureDevOpsBinary
- 113: DirectFile
- 114: NexusBinary
- 115: Gitea

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: AmazonS3 - 1: OneDrive - 2: Box - 3: DellIsilon - 4: AzureFiles - 5: SmbCompatible - 6: S3Compatible - 7: AzureBlob - 8: AlibabaCloud - 9: GoogleCloud - 10: Sftp - 11: SharePoint - 12: Nfs - 13: Wasabi - 14: OracleCloud - 15: NetAppOntap - 16: Cubbit - 17: Mft - 18: SharepointOnPrem - 19: Ftp - 20: Oracle - 21: NfsSisl - 22: ScalityRing - 100: GitHub - 101: DockerHub - 102: Bitbucket - 103: AmazonEcr - 104: Quay - 105: JFrogContainer - 106: JFrogBinary - 107: AzureAcr - 108: GitLabSource - 109: GitLabContainer - 110: Svn - 111: AzureDevOpsSource - 112: AzureDevOpsBinary - 113: DirectFile - 114: NexusBinary - 115: Gitea

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4
anonymous	5
anonymous	6
anonymous	7
anonymous	8
anonymous	9
anonymous	10
anonymous	11
anonymous	12
anonymous	13
anonymous	14
anonymous	15
anonymous	16
anonymous	17
anonymous	18
anonymous	19
anonymous	20
anonymous	21
anonymous	22
anonymous	100
anonymous	101
anonymous	102
anonymous	103

anonymous	104
anonymous	105
anonymous	106
anonymous	107
anonymous	108
anonymous	109
anonymous	110
anonymous	111
anonymous	112
anonymous	113
anonymous	114
anonymous	115

StorageProtocolType2

```
0
```

- 0: Smb
- 1: GraphSdk
- 2: BoxSdk
- 3: AwsSdk
- 4: AzureBlobSdk
- 5: AlibabaCloudSdk
- 6: GoogleCloudSdk
- 7: AzureFilesSdk
- 8: Sftp
- 9: Nfs
- 10: Mft
- 11: SharepointOnPrem
- 12: Ftp
- 13: OracleSdk

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Smb - 1: GraphSdk - 2: BoxSdk - 3: AwsSdk - 4: AzureBlobSdk - 5: AlibabaCloudSdk - 6: GoogleCloudSdk - 7: AzureFilesSdk - 8: Sftp - 9: Nfs - 10: Mft - 11: SharepointOnPrem - 12: Ftp - 13: OracleSdk

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4
anonymous	5
anonymous	6
anonymous	7
anonymous	8
anonymous	9
anonymous	10
anonymous	11
anonymous	12
anonymous	13

StorageVendorType2

```
0
```

- 0: AmazonS3
- 1: OneDrive
- 2: Box
- 3: DellIsilon
- 4: AzureFiles
- 5: SmbCompatible
- 6: S3Compatible
- 7: AzureBlob
- 8: AlibabaCloud
- 9: GoogleCloud
- 10: Sftp
- 11: SharePoint
- 12: Nfs
- 13: Wasabi
- 14: OracleCloud
- 15: NetAppOntap
- 16: Cubbit
- 17: Mft
- 18: SharepointOnPrem
- 19: Ftp
- 20: Oracle
- 21: NfsSisl
- 22: ScalityRing

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: AmazonS3 - 1: OneDrive - 2: Box - 3: Delllsilon - 4: AzureFiles - 5: SmbCompatible - 6: S3Compatible - 7: AzureBlob - 8: AlibabaCloud - 9: GoogleCloud - 10: Sftp - 11: SharePoint - 12: Nfs - 13: Wasabi - 14: OracleCloud - 15: NetAppOntap - 16: Cubbit - 17: Mft - 18: SharepointOnPrem - 19: Ftp - 20: Oracle - 21: NfsSisl - 22: ScalityRing

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4
anonymous	5
anonymous	6
anonymous	7
anonymous	8
anonymous	9
anonymous	10
anonymous	11
anonymous	12
anonymous	13
anonymous	14
anonymous	15
anonymous	16
anonymous	17
anonymous	18
anonymous	19
anonymous	20
anonymous	21
anonymous	22

DeleteAccountCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "accountId": "string"
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» accountId	string¦null	false	none	none

GetAccountById

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "account": {
    "id": "string",
    "name": "string",
    "description": "string",
    "storages": [
      {
        "id": "string",
        "groupId": "string",
        "accountId": "string",
        "name": "string",
        "storageClientId": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "scansInProgress": true,
        "isRtpInProgress": true,
        "isScheduledScanInProgress": true,
        "storageFunctionality": "string",
        "$type": "string"
      }
    ],
    "credentials": "string",
    "autoDiscovery": true,
    "groupId": "string",
    "protocolType": 0,
    "vendorType": 0
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» account	AccountWithStorage	false	none	none

AccountWithStorage


```
{
  "id": "string",
  "name": "string",
  "description": "string",
  "storages": [
    {
      "id": "string",
      "groupId": "string",
      "accountId": "string",
      "name": "string",
      "storageClientId": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "scansInProgress": true,
      "isRtpInProgress": true,
      "isScheduledScanInProgress": true,
      "storageFunctionality": "string",
      "$type": "string"
    }
  ],
  "credentials": "string",
  "autoDiscovery": true,
  "groupId": "string",
  "protocolType": 0,
  "vendorType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
name	string	false	none	none
description	string	false	none	none
storages	[Storage]¦null	false	none	none
credentials	string	false	none	none
autoDiscovery	boolean	false	none	none
groupId	string	false	none	none
protocolType	integer(int32)	false	none	none
vendorType	integer(int32)¦null	false	none	none

Storage


```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
groupId	string	false	none	none
accountId	string	false	none	none
name	string	false	none	none
storageClientId	string	false	none	none
storageType	StorageType	false	none	none
storageStatus	StorageStatus	false	none	none
scansInProgress	boolean	false	none	none
isRtpInProgress	boolean	false	none	none
isScheduledScanInProgress	boolean	false	none	none
storageFunctionality	string	false	none	none
$type	string	true	none	none

StorageType

```
{
  "name": "string",
  "categoryType": 0,
  "protocolType": 0,
  "vendorType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
name	string	false	none	none
categoryType	StorageCategoryType	false	none	- 0: Local - 1: Cloud - 2: Object
protocolType	StorageProtocolType2	false	none	- 0: Smb - 1: GraphSdk - 2: BoxSdk - 3: AwsSdk - 4: AzureBlobSdk - 5: AlibabaCloudSdk - 6: GoogleCloudSdk - 7: AzureFilesSdk - 8: Sftp - 9: Nfs - 10: Mft - 11: SharepointOnPrem - 12: Ftp - 13: OracleSdk
vendorType	StorageVendorType2	false	none	- 0: AmazonS3 - 1: OneDrive - 2: Box - 3: Delllsilon - 4: AzureFiles - 5: SmbCompatible - 6: S3Compatible - 7: AzureBlob - 8: AlibabaCloud - 9: GoogleCloud - 10: Sftp - 11: SharePoint - 12: Nfs - 13: Wasabi - 14: OracleCloud - 15: NetAppOntap - 16: Cubbit - 17: Mft - 18: SharepointOnPrem - 19: Ftp - 20: Oracle - 21: NfsSisl - 22: ScalityRing

StorageCategoryType

```
0
```

- 0: Local
- 1: Cloud
- 2: Object

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Local - 1: Cloud - 2: Object

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

StorageStatus

```
{
  "message": "string",
  "statusType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
message	string¦null	false	none	none
statusType	integer(int32)	false	none	none

AmazonSdkStorage


```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string",
  "serviceUrl": {
    "property1": "string",
    "property2": "string"
  },
  "bucketName": "string",
  "folderLocation": "string"
}
```


Properties

allOf - discriminator: Storage.$type

Name	Type	Required	Restrictions	Description
anonymous	Storage	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» serviceUrl	object	false	none	none
»» **additionalProperties**	string	false	none	none
» bucketName	string	false	none	none
» folderLocation	string	false	none	none

AlibabaCloudStorage


```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string",
  "bucketName": "string",
  "folderLocation": "string"
}
```


Properties

allOf - discriminator: Storage.$type

Name	Type	Required	Restrictions	Description
anonymous	Storage	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» bucketName	string	false	none	none
» folderLocation	string	false	none	none

AzureBlobStorage


```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string",
  "container": "string"
}
```


Properties

allOf - discriminator: Storage.$type

Name	Type	Required	Restrictions	Description
anonymous	Storage	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» container	string	false	none	none

AzureFilesStorage


```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string",
  "folderLocation": "string",
  "shareName": "string"
}
```


Properties

allOf - discriminator: Storage.$type

Name	Type	Required	Restrictions	Description
anonymous	Storage	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» folderLocation	string	false	none	none
» shareName	string	false	none	none

BoxStorage


```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string",
  "userName": "string",
  "groupName": "string",
  "userId": "string",
  "folderPath": "string"
}
```


Properties

allOf - discriminator: Storage.$type

Name	Type	Required	Restrictions	Description
anonymous	Storage	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» userName	string	false	none	none
» groupName	string	false	none	none
» userId	string	false	none	none
» folderPath	string	false	none	none

GoogleCloudStorage


```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string",
  "bucketName": "string",
  "folderLocation": "string"
}
```


Properties

allOf - discriminator: Storage.$type

Name	Type	Required	Restrictions	Description
anonymous	Storage	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» bucketName	string	false	none	none
» folderLocation	string	false	none	none

GraphStorage


```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string",
  "group": "string",
  "groupUrl": "string",
  "member": {
    "id": "string",
    "name": "string",
    "nameUrl": "string"
  },
  "folderLocation": "string",
  "userInfo": {
    "upn": "string",
    "userId": "string"
  }
}
```


Properties

allOf - discriminator: Storage.$type

Name	Type	Required	Restrictions	Description
anonymous	Storage	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» group	string¦null	false	none	none
» groupUrl	string¦null	false	none	none
» member	StorageMember¦null	false	none	none
» folderLocation	string¦null	false	none	none
» userInfo	GraphUserInfo¦null	false	none	none

StorageMember

```
{
  "id": "string",
  "name": "string",
  "nameUrl": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string¦null	false	none	none
name	string¦null	false	none	none
nameUrl	string¦null	false	none	none

GraphUserInfo

```
{
  "upn": "string",
  "userId": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
upn	string¦null	false	none	none
userId	string¦null	false	none	none

SftpStorage


```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string",
  "sharePath": "string"
}
```


Properties

allOf - discriminator: Storage.$type

Name	Type	Required	Restrictions	Description
anonymous	Storage	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» sharePath	string¦null	false	none	none

SmbStorage


```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string",
  "sharePath": "string"
}
```


Properties

allOf - discriminator: Storage.$type

Name	Type	Required	Restrictions	Description
anonymous	Storage	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» sharePath	string	false	none	none

NfsStorage


```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string",
  "user": "string",
  "sharePath": "string"
}
```


Properties

allOf - discriminator: Storage.$type

Name	Type	Required	Restrictions	Description
anonymous	Storage	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» user	string	false	none	none
» sharePath	string	false	none	none

MftStorage

```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string",
  "folderLocation": "string",
  "scanUsers": "string"
}
```


Properties

allOf - discriminator: Storage.$type

Name	Type	Required	Restrictions	Description
anonymous	Storage	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» folderLocation	string	false	none	none
» scanUsers	string	false	none	none

SharepointOnPremStorage

```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string",
  "documentLibrary": "string",
  "documentLibraryUrl": "string",
  "folderLocation": "string"
}
```


Properties

allOf - discriminator: Storage.$type

Name	Type	Required	Restrictions	Description
anonymous	Storage	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» documentLibrary	string	false	none	none
» documentLibraryUrl	string	false	none	none
» folderLocation	string	false	none	none

FtpStorage

```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string",
  "folderPath": "string"
}
```


Properties

allOf - discriminator: Storage.$type

Name	Type	Required	Restrictions	Description
anonymous	Storage	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» folderPath	string¦null	false	none	none

OracleSdkStorage

```
{
  "id": "string",
  "groupId": "string",
  "accountId": "string",
  "name": "string",
  "storageClientId": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "scansInProgress": true,
  "isRtpInProgress": true,
  "isScheduledScanInProgress": true,
  "storageFunctionality": "string",
  "$type": "string",
  "bucketName": "string",
  "folderLocation": "string"
}
```


Properties

allOf - discriminator: Storage.$type

Name	Type	Required	Restrictions	Description
anonymous	Storage	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» bucketName	string¦null	false	none	none
» folderLocation	string¦null	false	none	none

EnumerateAccountStoragesQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "autodiscoverySource": "string",
      "autodiscoverySourceUrl": "string"
    }
  ],
  "protocolType": 0,
  "vendorType": 0
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entries	[AccountStorage]	false	none	none
» protocolType	integer(int32)¦null	false	none	none
» vendorType	integer(int32)¦null	false	none	none

AccountStorage


```
{
  "autodiscoverySource": "string",
  "autodiscoverySourceUrl": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
autodiscoverySource	string¦null	false	none	none
autodiscoverySourceUrl	string¦null	false	none	none

GetAccounts

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "accounts": [
    {
      "id": "string",
      "name": "string",
      "description": "string",
      "storages": [
        {
          "id": "string",
          "groupId": "string",
          "accountId": "string",
          "name": "string",
          "storageClientId": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "scansInProgress": true,
          "isRtpInProgress": true,
          "isScheduledScanInProgress": true,
          "storageFunctionality": "string",
          "$type": "string"
        }
      ],
      "credentials": "string",
      "autoDiscovery": true,
      "groupId": "string",
      "protocolType": 0,
      "vendorType": 0
    }
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» accounts	[AccountWithStorage]	false	none	none

GetAccountsCount

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "accountCount": {
    "count": 0
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» accountCount	CountResult	false	none	none

CountResult

```
{
  "count": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
count	integer(int64)	false	none	none

UpdateAccountCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "accountId": "string"
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» accountId	string¦null	false	none	none

EnumerateAccountSourcesQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    "string"
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entries	[string]¦null	false	none	none

EnumerateLogsQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "logLevel": 0,
      "details": "string",
      "logType": 0,
      "userId": "string",
      "category": 0,
      "timeStamp": "2019-08-24T14:15:22Z",
      "userName": "string"
    }
  ],
  "filterCount": 0,
  "totalCount": 0
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» result	ResultType2	false	none	- 0: Success - 1: Error - 2: InternalError
» entries	[AuditLog]¦null	false	none	none
» filterCount	integer(int64)	false	none	none
» totalCount	integer(int64)	false	none	none

ResultType2

```
0
```

- 0: Success
- 1: Error
- 2: InternalError

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Success - 1: Error - 2: InternalError

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

AuditLog

```
{
  "id": "string",
  "logLevel": 0,
  "details": "string",
  "logType": 0,
  "userId": "string",
  "category": 0,
  "timeStamp": "2019-08-24T14:15:22Z",
  "userName": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string¦null	false	none	none
logLevel	AuditLogLevel	false	none	- 0: Debug - 1: Info - 2: Warn - 3: Error
details	string¦null	false	none	none
logType	LogType¦null	false	none	- 0: FileDiscovered - 1: FileScanned - 2: FileDiscoveryFailed - 3: FileCancelled - 100: ProcessStarted - 101: ProcessCompleted - 102: ProcessCancelled - 103: ProcessFailed - 200: FileTagged - 201: FileSanitizedKeptOriginal - 203: FileSanitizedDeletedOriginal - 204: FileDecryption - 205: BlockedFileMoved - 206: BlockedFileDeleted - 207: SanitizedFileMoved - 208: AllowedFileMoved - 209: BlockedFileCopied - 210: SanitizedFileCopied - 211: AllowedFileCopied - 212: BlockedFileKept - 213: AllowedFileKept - 214: SanitizedFileKept - 215: AllowedFileDeleted - 216: FileVersionsScanned - 217: DiscoveredFileDeleted - 218: DiscoveredFileMoved - 219: DiscoveredFileCopied - 220: DiscoveredFileKept - 221: SanitizedFileDeleted - 300: StorageAdded - 301: StorageUpdated - 302: StorageRemoved - 320: GroupAdded - 321: GroupUpdated - 322: GroupRemoved - 330: AccountAdded - 331: AccountUpdated - 332: AccountRemoved

				- 350: SecurityChecklistItem - 351: SecurityChecklistVerified - 400: Rtp - 410: FileTagging - 411: DeepCdrRemediation - 412: SanitizedFilesRemediation - 413: BlockedFilesRemediation - 414: TreatFilesWithSensitiveDataAsBlocked - 415: TreatFilesWithVulnerabilitiesAsBlocked - 416: AllowedFilesRemediation - 417: StrictBlockFilesStrategy - 418: ConfigurableBlockFilesStrategy - 419: DeletedEmptyFolders - 600: LoggedIn - 601: LoggedOut - 602: UserCreated - 603: UserRegistered - 604: UserRoleUpdated - 605: UserUpdated - 606: UserRemoved - 700: ScanInstanceAdd - 701: ScanInstanceUpdate - 702: ScanInstanceDelete - 703: ScanInstanceImport - 750: ScanPoolAdd - 751: ScanPoolUpdate - 752: ScanPoolDelete - 753: ScanPoolImport - 800: OnlineLicenseActivated - 801: OfflineLicenseActivated - 802: LicenseDeactivated - 900: ConfigurationImported - 901: ConfigurationExported - 950: Notifications - 975: Smtp - 1000: ExternalLoggerAdd - 1001: ExternalLoggerUpdate - 1002: ExternalLoggerDelete - 1100: DataRetention - 1200: ApiKeyAdd - 1201: ApiKeyUpdate - 1202: ApiKeyDelete - 1300: SsoConfiguration - 1400: WorkflowAdd - 1401: WorkflowUpdate - 1402: WorkflowDelete - 1403: WorkflowSetDefault - 1404: WorkflowImport

				- 1410: WorkflowTechnologies - 1500: ScanAdd - 1501: ScanUpdate - 1502: ScanDelete - 1503: ScanImport - 1600: OnDemandScanRequest - 1700: ReportGenerated - 1800: TourFinalized - 1900: ScanScheduleAdd - 1901: ScanScheduleUpdate - 1902: ScanScheduleDelete - 1903: ScanScheduleImport - 1904: ScanScheduleSkip - 2000: Encryption - 2101: RemediationAdd - 2102: RemediationUpdate - 2103: RemediationDelete - 2200: Certificates - 2300: Telemetry - 2400: AbacRoleAssigned - 2401: AbacRoleRemoved - 2402: AbacRoleCreated - 2403: AbacRoleDeleted - 2410: AbacPermissionAssigned - 2411: AbacPermissionRemoved - 2412: AbacRolesReplaced
userId	string¦null	false	none	none
category	AuditLogCategoryType	false	none	- 0: Processes - 1: PostActions - 2: StorageUnits - 3: Authentication - 4: Settings - 6: Users - 8: Notifications - 9: GroupUnits - 10: AccountUnits
timeStamp	string(date-time)¦null	false	none	none
userName	string¦null	false	none	none

AuditLogLevel

```
0
```

- 0: Debug
- 1: Info
- 2: Warn
- 3: Error

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Debug - 1: Info - 2: Warn - 3: Error

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3

LogType

```
0
```

- 0: FileDiscovered
- 1: FileScanned
- 2: FileDiscoveryFailed
- 3: FileCancelled
- 100: ProcessStarted
- 101: ProcessCompleted
- 102: ProcessCancelled
- 103: ProcessFailed
- 200: FileTagged
- 201: FileSanitizedKeptOriginal
- 203: FileSanitizedDeletedOriginal

- 204: FileDecryption
- 205: BlockedFileMoved
- 206: BlockedFileDeleted
- 207: SanitizedFileMoved
- 208: AllowedFileMoved
- 209: BlockedFileCopied
- 210: SanitizedFileCopied
- 211: AllowedFileCopied
- 212: BlockedFileKept
- 213: AllowedFileKept
- 214: SanitizedFileKept
- 215: AllowedFileDeleted
- 216: FileVersionsScanned
- 217: DiscoveredFileDeleted
- 218: DiscoveredFileMoved
- 219: DiscoveredFileCopied
- 220: DiscoveredFileKept
- 221: SanitizedFileDeleted
- 300: StorageAdded
- 301: StorageUpdated
- 302: StorageRemoved
- 320: GroupAdded
- 321: GroupUpdated
- 322: GroupRemoved
- 330: AccountAdded
- 331: AccountUpdated
- 332: AccountRemoved
- 350: SecurityChecklistItem
- 351: SecurityChecklistVerified
- 400: Rtp
- 410: FileTagging
- 411: DeepCdrRemediation
- 412: SanitizedFilesRemediation
- 413: BlockedFilesRemediation
- 414: TreatFilesWithSensitiveDataAsBlocked
- 415: TreatFilesWithVulnerabilitiesAsBlocked
- 416: AllowedFilesRemediation

- 417: StrictBlockFilesStrategy
- 418: ConfigurableBlockFilesStrategy
- 419: DeletedEmptyFolders
- 600: LoggedIn
- 601: LoggedOut
- 602: UserCreated
- 603: UserRegistered
- 604: UserRoleUpdated
- 605: UserUpdated
- 606: UserRemoved
- 700: ScanInstanceAdd
- 701: ScanInstanceUpdate
- 702: ScanInstanceDelete
- 703: ScanInstanceImport
- 750: ScanPoolAdd
- 751: ScanPoolUpdate
- 752: ScanPoolDelete
- 753: ScanPoolImport
- 800: OnlineLicenseActivated
- 801: OfflineLicenseActivated
- 802: LicenseDeactivated
- 900: ConfigurationImported
- 901: ConfigurationExported
- 950: Notifications
- 975: Smtp
- 1000: ExternalLoggerAdd
- 1001: ExternalLoggerUpdate
- 1002: ExternalLoggerDelete
- 1100: DataRetention
- 1200: ApiKeyAdd
- 1201: ApiKeyUpdate
- 1202: ApiKeyDelete
- 1300: SsoConfiguration
- 1400: WorkflowAdd
- 1401: WorkflowUpdate
- 1402: WorkflowDelete
- 1403: WorkflowSetDefault

- 1404: WorkflowImport
- 1410: WorkflowTechnologies
- 1500: ScanAdd
- 1501: ScanUpdate
- 1502: ScanDelete
- 1503: ScanImport
- 1600: OnDemandScanRequest
- 1700: ReportGenerated
- 1800: TourFinalized
- 1900: ScanScheduleAdd
- 1901: ScanScheduleUpdate
- 1902: ScanScheduleDelete
- 1903: ScanScheduleImport
- 1904: ScanScheduleSkip
- 2000: Encryption
- 2101: RemediationAdd
- 2102: RemediationUpdate
- 2103: RemediationDelete
- 2200: Certificates
- 2300: Telemetry
- 2400: AbacRoleAssigned
- 2401: AbacRoleRemoved
- 2402: AbacRoleCreated
- 2403: AbacRoleDeleted
- 2410: AbacPermissionAssigned
- 2411: AbacPermissionRemoved
- 2412: AbacRolesReplaced

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: FileDiscovered - 1: FileScanned - 2: FileDiscoveryFailed - 3: FileCancelled - 100: ProcessStarted - 101: ProcessCompleted - 102: ProcessCancelled - 103: ProcessFailed - 200: FileTagged - 201: FileSanitizedKeptOriginal - 203: FileSanitizedDeletedOriginal - 204: FileDecryption - 205: BlockedFileMoved - 206: BlockedFileDeleted - 207: SanitizedFileMoved - 208: AllowedFileMoved - 209: BlockedFileCopied - 210: SanitizedFileCopied - 211: AllowedFileCopied - 212: BlockedFileKept - 213: AllowedFileKept - 214: SanitizedFileKept - 215: AllowedFileDeleted - 216: FileVersionsScanned - 217: DiscoveredFileDeleted - 218: DiscoveredFileMoved - 219: DiscoveredFileCopied - 220: DiscoveredFileKept - 221: SanitizedFileDeleted - 300: StorageAdded - 301: StorageUpdated - 302: StorageRemoved - 320: GroupAdded - 321: GroupUpdated - 322: GroupRemoved - 330: AccountAdded - 331: AccountUpdated - 332: AccountRemoved - 350: SecurityChecklistItem - 351: SecurityChecklistVerified - 400: Rtp - 410: FileTagging - 411: DeepCdrRemediation - 412: SanitizedFilesRemediation - 413: BlockedFilesRemediation - 414: TreatFilesWithSensitiveDataAsBlocked

				- 415: TreatFilesWithVulnerabilitiesAsBlocked - 416: AllowedFilesRemediation - 417: StrictBlockFilesStrategy - 418: ConfigurableBlockFilesStrategy - 419: DeletedEmptyFolders - 600: LoggedIn - 601: LoggedOut - 602: UserCreated - 603: UserRegistered - 604: UserRoleUpdated - 605: UserUpdated - 606: UserRemoved - 700: ScanInstanceAdd - 701: ScanInstanceUpdate - 702: ScanInstanceDelete - 703: ScanInstanceImport - 750: ScanPoolAdd - 751: ScanPoolUpdate - 752: ScanPoolDelete - 753: ScanPoolImport - 800: OnlineLicenseActivated - 801: OfflineLicenseActivated - 802: LicenseDeactivated - 900: ConfigurationImported - 901: ConfigurationExported - 950: Notifications - 975: Smtp - 1000: ExternalLoggerAdd - 1001: ExternalLoggerUpdate - 1002: ExternalLoggerDelete - 1100: DataRetention - 1200: ApiKeyAdd - 1201: ApiKeyUpdate - 1202: ApiKeyDelete - 1300: SsoConfiguration - 1400: WorkflowAdd - 1401: WorkflowUpdate - 1402: WorkflowDelete - 1403: WorkflowSetDefault - 1404: WorkflowImport - 1410: WorkflowTechnologies - 1500: ScanAdd - 1501: ScanUpdate - 1502: ScanDelete - 1503: ScanImport - 1600: OnDemandScanRequest - 1700: ReportGenerated - 1800: TourFinalized

				- 1900: ScanScheduleAdd - 1901: ScanScheduleUpdate - 1902: ScanScheduleDelete - 1903: ScanScheduleImport - 1904: ScanScheduleSkip - 2000: Encryption - 2101: RemediationAdd - 2102: RemediationUpdate - 2103: RemediationDelete - 2200: Certificates - 2300: Telemetry - 2400: AbacRoleAssigned - 2401: AbacRoleRemoved - 2402: AbacRoleCreated - 2403: AbacRoleDeleted - 2410: AbacPermissionAssigned - 2411: AbacPermissionRemoved - 2412: AbacRolesReplaced

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	100
anonymous	101
anonymous	102
anonymous	103
anonymous	200
anonymous	201
anonymous	203
anonymous	204
anonymous	205
anonymous	206
anonymous	207
anonymous	208
anonymous	209
anonymous	210
anonymous	211
anonymous	212
anonymous	213
anonymous	214
anonymous	215
anonymous	216
anonymous	217
anonymous	218
anonymous	219

anonymous	220
anonymous	221
anonymous	300
anonymous	301
anonymous	302
anonymous	320
anonymous	321
anonymous	322
anonymous	330
anonymous	331
anonymous	332
anonymous	350
anonymous	351
anonymous	400
anonymous	410
anonymous	411
anonymous	412
anonymous	413
anonymous	414
anonymous	415
anonymous	416
anonymous	417
anonymous	418
anonymous	419
anonymous	600
anonymous	601
anonymous	602
anonymous	603
anonymous	604

anonymous	605
anonymous	606
anonymous	700
anonymous	701
anonymous	702
anonymous	703
anonymous	750
anonymous	751
anonymous	752
anonymous	753
anonymous	800
anonymous	801
anonymous	802
anonymous	900
anonymous	901
anonymous	950
anonymous	975
anonymous	1000
anonymous	1001
anonymous	1002
anonymous	1100
anonymous	1200
anonymous	1201
anonymous	1202
anonymous	1300
anonymous	1400
anonymous	1401
anonymous	1402
anonymous	1403

anonymous	1404
anonymous	1410
anonymous	1500
anonymous	1501
anonymous	1502
anonymous	1503
anonymous	1600
anonymous	1700
anonymous	1800
anonymous	1900
anonymous	1901
anonymous	1902
anonymous	1903
anonymous	1904
anonymous	2000
anonymous	2101
anonymous	2102
anonymous	2103
anonymous	2200
anonymous	2300
anonymous	2400
anonymous	2401
anonymous	2402
anonymous	2403
anonymous	2410
anonymous	2411
anonymous	2412

AuditLogCategoryType

```
0
```

- 0: Processes
- 1: PostActions
- 2: StorageUnits
- 3: Authentication
- 4: Settings
- 6: Users
- 8: Notifications
- 9: GroupUnits
- 10: AccountUnits

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Processes - 1: PostActions - 2: StorageUnits - 3: Authentication - 4: Settings - 6: Users - 8: Notifications - 9: GroupUnits - 10: AccountUnits

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4
anonymous	6
anonymous	8
anonymous	9
anonymous	10

ImportSettingsCommandResponse


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "areSmtpSettingsImported": true,
  "isLicenseImported": true,
  "isRetentionImported": true,
  "areNotificationsImported": true,
  "hasNotificationsPartialFailure": true,
  "activationDto": {
    "result": 0,
    "responseKey": "string",
    "responseMessage": "string",
    "responseMessageParams": {
      "property1": "string",
      "property2": "string"
    }
  },
  "newLocalUserId": "string",
  "areUsersImported": true
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» result	ResultType3	false	none	- 0: Success - 1: FailedToUpgradeConfigFile - 2: FailedRequestBodyIsEmpty - 3: FailedToActivateLicense - 4: FailedToDeserializeConfigFile - 5: FailedOutdatedVersion - 6: FailedInvalidConfigFile - 7: FailedMissingPassword - 8: FailedWrongPassword
» areSmtpSettingsImported	boolean	false	none	none
» isLicenseImported	boolean	false	none	none
» isRetentionImported	boolean	false	none	none
» areNotificationsImported	boolean	false	none	none
» hasNotificationsPartialFailure	boolean	false	none	none
» activationDto	Activation¦null	false	none	none
» newLocalUserId	string¦null	false	none	none
» areUsersImported	boolean	false	none	none

ResultType3

```
0
```

- 0: Success
- 1: FailedToUpgradeConfigFile
- 2: FailedRequestBodyIsEmpty
- 3: FailedToActivateLicense
- 4: FailedToDeserializeConfigFile
- 5: FailedOutdatedVersion
- 6: FailedInvalidConfigFile
- 7: FailedMissingPassword
- 8: FailedWrongPassword

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Success - 1: FailedToUpgradeConfigFile - 2: FailedRequestBodyIsEmpty - 3: FailedToActivateLicense - 4: FailedToDeserializeConfigFile - 5: FailedOutdatedVersion - 6: FailedInvalidConfigFile - 7: FailedMissingPassword - 8: FailedWrongPassword

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4
anonymous	5
anonymous	6
anonymous	7
anonymous	8

Activation


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

GetEnabledModules


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "enabledModules": [
    0
  ]
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» enabledModules	[integer]¦null	false	none	none

EnumerateExternalLoggersQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "isEnabled": true,
      "loggerType": 0,
      "connectionSettings": {
        "$type": "string"
      }
    }
  ]
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» result	ResultType4	false	none	- 0: Success - 1: Error - 2: InternalError
» entries	[ExternalLoggerConfig]¦null	false	none	none

ResultType4

```
0
```

- 0: Success
- 1: Error
- 2: InternalError

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Success - 1: Error - 2: InternalError

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

ExternalLoggerConfig

```
{
  "id": "string",
  "isEnabled": true,
  "loggerType": 0,
  "connectionSettings": {
    "$type": "string"
  }
}
```


Properties

Name	Type	Required	Restrictions	Description
id	string¦null	false	none	none
isEnabled	boolean	false	none	none
loggerType	ExternalLoggerType	false	none	- 0: SyslogUdp - 1: Kafka - 2: TcpSyslog - 3: TcpSyslogWithTls
connectionSettings	ExternalLoggerConnectionSettings¦null	false	none	none

ExternalLoggerType


```
0
```


- 0: SyslogUdp
- 1: Kafka
- 2: TcpSyslog
- 3: TcpSyslogWithTls

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: SyslogUdp - 1: Kafka - 2: TcpSyslog - 3: TcpSyslogWithTls

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3

ExternalLoggerConnectionSettings

```
{
  "$type": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
$type	string	true	none	none

SyslogUdpConnectionSettings

```
{
  "$type": "string",
  "serverAddress": "string",
  "port": 0,
  "facility": 0,
  "format": 0
}
```

Properties

allOf - discriminator: ExternalLoggerConnectionSettings.$type

Name	Type	Required	Restrictions	Description
anonymous	ExternalLoggerConnectionSettings	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» serverAddress	string¦null	false	none	none
» port	integer(int32)	false	none	none
» facility	SyslogFacility	false	none	- 0: KernelMessages - 1: UserLevelMessages - 2: MailSystem - 3: SystemDaemons - 4: SecurityOrAuthorizationMessages1 - 5: InternalMessages - 6: LinePrinterSubsystem - 7: NetworkNewsSubsystem - 8: UUCPSubsystem - 9: ClockDaemon1 - 10: SecurityOrAuthorizationMessages2 - 11: FTPDaemon - 12: NTPSubsystem - 13: LogAudit - 14: LogAlert - 15: ClockDaemon2 - 16: LocalUse0 - 17: LocalUse1 - 18: LocalUse2 - 19: LocalUse3 - 20: LocalUse4 - 21: LocalUse5 - 22: LocalUse6 - 23: LocalUse7
» format	SyslogOutputFormat	false	none	- 0: RFC5424

SyslogFacility

```
0
```

- 0: KernelMessages
- 1: UserLevelMessages
- 2: MailSystem
- 3: SystemDaemons

- 4: SecurityOrAuthorizationMessages1
- 5: InternalMessages
- 6: LinePrinterSubsystem
- 7: NetworkNewsSubsystem
- 8: UUCPSubsystem
- 9: ClockDaemon1
- 10: SecurityOrAuthorizationMessages2
- 11: FTPDaemon
- 12: NTPSubsystem
- 13: LogAudit
- 14: LogAlert
- 15: ClockDaemon2
- 16: LocalUse0
- 17: LocalUse1
- 18: LocalUse2
- 19: LocalUse3
- 20: LocalUse4
- 21: LocalUse5
- 22: LocalUse6
- 23: LocalUse7

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: KernelMessages - 1: UserLevelMessages - 2: MailSystem - 3: SystemDaemons - 4: SecurityOrAuthorizationMessages1 - 5: InternalMessages - 6: LinePrinterSubsystem - 7: NetworkNewsSubsystem - 8: UUCPSubsystem - 9: ClockDaemon1 - 10: SecurityOrAuthorizationMessages2 - 11: FTPDaemon - 12: NTPSubsystem - 13: LogAudit - 14: LogAlert - 15: ClockDaemon2 - 16: LocalUse0 - 17: LocalUse1 - 18: LocalUse2 - 19: LocalUse3 - 20: LocalUse4 - 21: LocalUse5 - 22: LocalUse6 - 23: LocalUse7

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4
anonymous	5
anonymous	6
anonymous	7
anonymous	8
anonymous	9
anonymous	10
anonymous	11
anonymous	12
anonymous	13
anonymous	14
anonymous	15
anonymous	16
anonymous	17
anonymous	18
anonymous	19
anonymous	20
anonymous	21
anonymous	22
anonymous	23

SyslogOutputFormat

```
0
```

- 0: RFC5424

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: RFC5424

Enumerated Values

Property	Value
anonymous	0

SyslogTcpConnectionSettings

```
{
  "$type": "string",
  "serverAddress": "string",
  "port": 0,
  "facility": 0,
  "format": 0
}
```

Properties

allOf - discriminator: ExternalLoggerConnectionSettings.$type

Name	Type	Required	Restrictions	Description
anonymous	ExternalLoggerConnectionSettings	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» serverAddress	string¦null	false	none	none
» port	integer(int32)	false	none	none
» facility	SyslogFacility	false	none	- 0: KernelMessages - 1: UserLevelMessages - 2: MailSystem - 3: SystemDaemons - 4: SecurityOrAuthorizationMessages1 - 5: InternalMessages - 6: LinePrinterSubsystem - 7: NetworkNewsSubsystem - 8: UUCPSubsystem - 9: ClockDaemon1 - 10: SecurityOrAuthorizationMessages2 - 11: FTPDaemon - 12: NTPSubsystem - 13: LogAudit - 14: LogAlert - 15: ClockDaemon2 - 16: LocalUse0 - 17: LocalUse1 - 18: LocalUse2 - 19: LocalUse3 - 20: LocalUse4 - 21: LocalUse5 - 22: LocalUse6 - 23: LocalUse7
» format	SyslogOutputFormat	false	none	- 0: RFC5424

KafkaConnectionSettings

```
{
  "$type": "string",
  "serverAddressWithPortList": "string",
  "topic": "string",
  "configuration": "string"
}
```

Properties

allOf - discriminator: ExternalLoggerConnectionSettings.$type

Name	Type	Required	Restrictions	Description
anonymous	ExternalLoggerConnectionSettings	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» serverAddressWithPortList	string¦null	false	none	none
» topic	string¦null	false	none	none
» configuration	string¦null	false	none	none

UpdateExternalLoggerStatusParameters

```
{
  "id": "string",
  "isEnabled": true
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	true	none	none
isEnabled	boolean	true	none	none

AddUdpSyslogExternalLoggerParameters

```
{
  "isEnabled": true,
  "serverAddress": "string",
  "port": 0,
  "loggerType": 0,
  "facility": 0,
  "format": 0,
  "$type": "string"
}
```


Properties

allOf - discriminator: AddUpdateSyslogExternalLoggersParameters.$type

Name	Type	Required	Restrictions	Description
anonymous	AddUpdateSyslogExternalLoggersParameters	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

AddUpdateSyslogExternalLoggersParameters


```
{
  "isEnabled": true,
  "serverAddress": "string",
  "port": 0,
  "loggerType": 0,
  "facility": 0,
  "format": 0,
  "$type": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
isEnabled	boolean	true	none	none
serverAddress	string	true	none	none
port	integer(int32)	true	none	none
loggerType	ExternalLoggerType	false	none	- 0: SyslogUdp - 1: Kafka - 2: TcpSyslog - 3: TcpSyslogWithTls
facility	SyslogFacility	false	none	- 0: KernelMessages - 1: UserLevelMessages - 2: MailSystem - 3: SystemDaemons - 4: SecurityOrAuthorizationMessages1 - 5: InternalMessages - 6: LinePrinterSubsystem - 7: NetworkNewsSubsystem - 8: UUCPSubsystem - 9: ClockDaemon1 - 10: SecurityOrAuthorizationMessages2 - 11: FTPDaemon - 12: NTPSubsystem - 13: LogAudit - 14: LogAlert - 15: ClockDaemon2 - 16: LocalUse0 - 17: LocalUse1 - 18: LocalUse2 - 19: LocalUse3 - 20: LocalUse4 - 21: LocalUse5 - 22: LocalUse6 - 23: LocalUse7
format	SyslogOutputFormat	false	none	- 0: RFC5424
$type	string	true	none	none

UpdateUdpSyslogExternalLoggerParameters

```
{
  "isEnabled": true,
  "serverAddress": "string",
  "port": 0,
  "loggerType": 0,
  "facility": 0,
  "format": 0,
  "$type": "string",
  "id": "string"
}
```


Properties

allOf - discriminator: AddUpdateSyslogExternalLoggersParameters.$type

Name	Type	Required	Restrictions	Description
anonymous	AddUpdateSyslogExternalLoggersParameters	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» id	string	true	none	none

AddKafkaExternalLoggerParameters


```
{
  "isEnabled": true,
  "serverAddressWithPortList": "string",
  "loggerType": 0,
  "topic": "string",
  "configuration": "string",
  "$type": "string"
}
```


Properties

allOf - discriminator: AddUpdateKafkaExternalLoggersParameters.$type

Name	Type	Required	Restrictions	Description
anonymous	AddUpdateKafkaExternalLoggersParameters	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

AddUpdateKafkaExternalLoggersParameters

```
{
  "isEnabled": true,
  "serverAddressWithPortList": "string",
  "loggerType": 0,
  "topic": "string",
  "configuration": "string",
  "$type": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
isEnabled	boolean	true	none	none
serverAddressWithPortList	string	true	none	none
loggerType	ExternalLoggerType	false	none	- 0: SyslogUdp - 1: Kafka - 2: TcpSyslog - 3: TcpSyslogWithTls
topic	string	true	none	none
configuration	string	true	none	none
$type	string	true	none	none

UpdateKafkaExternalLoggerParameters

```
{
  "isEnabled": true,
  "serverAddressWithPortList": "string",
  "loggerType": 0,
  "topic": "string",
  "configuration": "string",
  "$type": "string",
  "id": "string"
}
```


Properties

allOf - discriminator: AddUpdateKafkaExternalLoggersParameters.$type

Name	Type	Required	Restrictions	Description
anonymous	AddUpdateKafkaExternalLoggersParameters	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» id	string	true	none	none

OnDemandScanCommandResponse


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

OnDemandScanParameters

```
{
  "fileId": "string",
  "fileIds": [
    "string"
  ],
  "scanId": "string",
  "storageProtocolType": 0,
  "passwords": [
    "string"
  ],
  "workflowId": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
fileId	string¦null	false	none	ID of the file to rescan
fileIds	[string]¦null	false	none	IDs of the files to rescan
scanId	string	true	none	scan ID associated with the file
storageProtocolType	integer(int32)	true	none	protocol used by the storage (e.g., SMB, S3)
passwords	[string]	false	none	optional passwords for encrypted file scanning
workflowId	string	true	none	workflow ID to be used for the on-demand scan

OnDemandFilteredScanParameters

```
{
  "scanId": "string",
  "workflowId": "string",
  "passwords": [
    "string"
  ],
  "filterParameters": {
    "expand": [
      "string"
    ],
    "startDate": "2019-08-24T14:15:22Z",
    "endDate": "2019-08-24T14:15:22Z",
    "search": "string",
    "scanType": [
      0
    ],
    "storageId": [
      "string"
    ],
    "scanId": [
      "string"
    ],
    "scanWorkflowSnapshotId": [
      "string"
    ],
    "processingState": [
      0
    ],
    "result": [
      0
    ],
    "scanAllResultI": [
      0
    ],
    "scanResultICount": [
      0
    ],
    "sandboxVerdict": [
      0
    ],
    "vulnerabilityCount": [
      0
    ],
    "hasVulnerabilities": true,
    "dlpDetection": true,
    "hasBlockedFileVersions": true,
    "remediations": [
      0
    ],
    "skippedScan": true,
    "sortBy": [
      0
```

```
"sortOrder": [
      0
    ]
  },
  "storageProtocolType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
scanId	string	true	none	scan ID associated with the file
workflowId	string	true	none	workflow ID to use when re-scanning
passwords	[string]¦null	false	none	optional passwords for encrypted file scanning
filterParameters	FileFilterSortParameters	true	none	Filters to select files to be re-scanned
storageProtocolType	integer(int32)	true	none	storage protocol type to use when re-scanning

FileFilterSortParameters

```
{
  "expand": [
    "string"
  ],
  "startDate": "2019-08-24T14:15:22Z",
  "endDate": "2019-08-24T14:15:22Z",
  "search": "string",
  "scanType": [
    0
  ],
  "storageId": [
    "string"
  ],
  "scanId": [
    "string"
  ],
  "scanWorkflowSnapshotId": [
    "string"
  ],
  "processingState": [
    0
  ],
  "result": [
    0
  ],
  "scanAllResultI": [
    0
  ],
  "scanResultICount": [
    0
  ],
  "sandboxVerdict": [
    0
  ],
  "vulnerabilityCount": [
    0
  ],
  "hasVulnerabilities": true,
  "dlpDetection": true,
  "hasBlockedFileVersions": true,
  "remediations": [
    0
  ],
  "skippedScan": true,
  "sortBy": [
    0
  ],
  "sortOrder": [
    0
  ]
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	ExpandableParameters	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» startDate	string(date-time)	true	none	filter files by scan date
» endDate	string(date-time)	true	none	filter files by scan date
» search	string¦null	false	none	text filter
» scanType	[ScanType]¦null	false	none	scan type such as (on-demand, scheduled, real time)
» storageId	[string]¦null	false	none	filter based on a list of possible storage id
» scanId	[string]¦null	false	none	filter based on a list of possible scan id
» scanWorkflowSnapshotId	[string]¦null	false	none	filter based on a list of possible scan workflow snapshot id
» processingState	[ProcessingState]¦null	false	none	[- 0: InProgress - 1: Failed - 2: Canceled - 3: Available - 4: FailedRemediation]
» result	[ResultI]¦null	false	none	filter by scan result (0 = Clean, 1 = Blocked, 2 = Processing)
» scanAllResultI	[ScanInstanceResultI]¦null	false	none	filter by MD Core scan result (0 = NoThreatDetected, 1 = Infected, 2 = Suspicious, etc.)
» scanResultICount	[integer]¦null	false	none	the number of engine results that

				are not “No threat detected”
» sandboxVerdict	[SandboxVerdict]¦null	false	none	filter by Sandbox verdict (0 = NoThreat, 1 = Malicious, 2 = Suspicious)
» vulnerabilityCount	[integer]¦null	false	none	the number of vulnerabilities detected
» hasVulnerabilities	boolean¦null	false	none	filter by presence of vulnerabilities
» dlpDetection	boolean¦null	false	none	DLP detection
» hasBlockedFileVersions	boolean¦null	false	none	filter by presence of blocked file versions
» remediations	[ObjectActionType]¦null	false	none	filter by remediation performed on the file
» skippedScan	boolean¦null	false	none	filter by whether the scan was skipped
» sortBy	[FileSortOption]¦null	false	none	[- 0: DiscoveryStarted - 1: ScanId - 2: ProcessingState - 3: Result - 4: ScanAllResultI - 5: ScanResultICount - 6: VulnerabilityCount - 7: SandboxVerdict - 8: SkippedScan]
» sortOrder	[SortingOrder]¦null	false	none	[- 0: Ascending - 1: Descending]

ScanType

```
0
```

- 0: Instant
- 1: Scheduled
- 2: RealTime

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Instant - 1: Scheduled - 2: RealTime

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

ProcessingState

```
0
```

- 0: InProgress
- 1: Failed
- 2: Canceled
- 3: Available
- 4: FailedRemediation

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: InProgress - 1: Failed - 2: Canceled - 3: Available - 4: FailedRemediation

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4

ResultI

```
0
```

- 0: Allowed
- 1: Blocked
- 2: Processing

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Allowed - 1: Blocked - 2: Processing

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

ScanInstanceResultI

```
0
```

- 0: NoThreatDetected
- 1: Infected
- 2: Suspicious
- 3: Failed
- 7: Allowlisted
- 8: Blocklisted
- 9: ExceededArchiveDepth
- 10: NotScanned
- 12: EncryptedArchive
- 13: ExceededArchiveSize
- 14: ExceededArchiveFileNumber
- 15: PasswordProtectedDocument
- 16: ExceededArchiveTimeout
- 17: Mismatch
- 18: PotentiallyVulnerableFile
- 19: Cancelled
- 20: SensitiveDataFound
- 21: YaraRuleMatched
- 22: PotentiallyUnwanted
- 23: UnsupportedFileType
- 24: ExtractionFailed
- 25: ScanFailed
- 26: SuspiciousVerdictbySandbox
- 27: LikelyMaliciousVerdictbySandbox
- 28: MaliciousVerdictbySandbox
- 29: BlockedVerdictbySandbox
- 30: BlockedVerdictbyDeepCdr
- 31: GlobalTimeoutExceeded
- 32: VulnerableVerdictbySbom
- 33: NonVulnerableVerdictbySbom
- 34: BlockedVerdictbySbom
- 36: BlockedbyPostAction
- 38: KnownBad
- 39: KnownGood

- 40: Unknown
- 41: AllowedVerdictbyCoo
- 42: BlockedVerdictbyCoo
- 43: UnknownVerdictbyCoo
- 255: InProgress
- 1014: SkipProcessingFastSymlink
- 9999: NullResult

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: NoThreatDetected - 1: Infected - 2: Suspicious - 3: Failed - 7: Allowlisted - 8: Blocklisted - 9: ExceededArchiveDepth - 10: NotScanned - 12: EncryptedArchive - 13: ExceededArchiveSize - 14: ExceededArchiveFileNumber - 15: PasswordProtectedDocument - 16: ExceededArchiveTimeout - 17: Mismatch - 18: PotentiallyVulnerableFile - 19: Cancelled - 20: SensitiveDataFound - 21: YaraRuleMatched - 22: PotentiallyUnwanted - 23: UnsupportedFileType - 24: ExtractionFailed - 25: ScanFailed - 26: SuspiciousVerdictbySandbox - 27: LikelyMaliciousVerdictbySandbox - 28: MaliciousVerdictbySandbox - 29: BlockedVerdictbySandbox - 30: BlockedVerdictbyDeepCdr - 31: GlobalTimeoutExceeded - 32: VulnerableVerdictbySbom - 33: NonVulnerableVerdictbySbom - 34: BlockedVerdictbySbom - 36: BlockedbyPostAction - 38: KnownBad - 39: KnownGood - 40: Unknown - 41: AllowedVerdictbyCoo - 42: BlockedVerdictbyCoo - 43: UnknownVerdictbyCoo - 255: InProgress - 1014: SkipProcessingFastSymlink - 9999: NullResult

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	7
anonymous	8
anonymous	9
anonymous	10
anonymous	12
anonymous	13
anonymous	14
anonymous	15
anonymous	16
anonymous	17
anonymous	18
anonymous	19
anonymous	20
anonymous	21
anonymous	22
anonymous	23
anonymous	24
anonymous	25
anonymous	26
anonymous	27
anonymous	28
anonymous	29
anonymous	30

anonymous	31
anonymous	32
anonymous	33
anonymous	34
anonymous	36
anonymous	38
anonymous	39
anonymous	40
anonymous	41
anonymous	42
anonymous	43
anonymous	255
anonymous	1014
anonymous	9999

SandboxVerdict

```
0
```

- 0: NOTHREAT
- 1: MALICIOUS
- 2: LIKELYMALICIOUS
- 3: SUSPICIOUS
- 4: BENIGN
- 5: UNKNOWN
- 999: FAILED

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: NOTHREAT - 1: MALICIOUS - 2: LIKELYMALICIOUS - 3: SUSPICIOUS - 4: BENIGN - 5: UNKNOWN - 999: FAILED

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4
anonymous	5
anonymous	999

ObjectActionType

```
0
```

- 0: Tagging
- 5: AllowedMoved
- 6: BlockedMoved
- 7: BlockedDelete
- 8: AllowedSanitizedMoved
- 9: AllowedCopied
- 10: BlockedCopied
- 11: AllowedSanitizedCopied
- 12: AllowedSanitizedDeleted

- 13: AllowedDeleted
- 14: AllowedKept
- 15: BlockedKept
- 16: AllowedSanitizedKept
- 17: BlockedSanitizedCopied
- 18: BlockedSanitizedMoved
- 19: BlockedSanitizedDeleted
- 20: BlockedSanitizedKept
- 21: ScanFileVersions
- 22: DiscoveredMoved
- 23: DiscoveredCopied
- 24: DiscoveredKept
- 25: DiscoveredDelete
- 100: SoftDelete
- 101: PackageDelete
- 102: Copy

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Tagging - 5: AllowedMoved - 6: BlockedMoved - 7: BlockedDelete - 8: AllowedSanitizedMoved - 9: AllowedCopied - 10: BlockedCopied - 11: AllowedSanitizedCopied - 12: AllowedSanitizedDeleted - 13: AllowedDeleted - 14: AllowedKept - 15: BlockedKept - 16: AllowedSanitizedKept - 17: BlockedSanitizedCopied - 18: BlockedSanitizedMoved - 19: BlockedSanitizedDeleted - 20: BlockedSanitizedKept - 21: ScanFileVersions - 22: DiscoveredMoved - 23: DiscoveredCopied - 24: DiscoveredKept - 25: DiscoveredDelete - 100: SoftDelete - 101: PackageDelete - 102: Copy

Enumerated Values

Property	Value
anonymous	0
anonymous	5
anonymous	6
anonymous	7
anonymous	8
anonymous	9
anonymous	10
anonymous	11
anonymous	12
anonymous	13
anonymous	14
anonymous	15
anonymous	16
anonymous	17
anonymous	18
anonymous	19
anonymous	20
anonymous	21
anonymous	22
anonymous	23
anonymous	24
anonymous	25
anonymous	100
anonymous	101
anonymous	102

FileSortOption

```
0
```

- 0: DiscoveryStarted
- 1: ScanId
- 2: ProcessingState
- 3: Result
- 4: ScanAllResultI
- 5: ScanResultICount
- 6: VulnerabilityCount
- 7: SandboxVerdict
- 8: SkippedScan

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: DiscoveryStarted - 1: ScanId - 2: ProcessingState - 3: Result - 4: ScanAllResultI - 5: ScanResultICount - 6: VulnerabilityCount - 7: SandboxVerdict - 8: SkippedScan

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4
anonymous	5
anonymous	6
anonymous	7
anonymous	8

SortingOrder

```
0
```

- 0: Ascending
- 1: Descending

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Ascending - 1: Descending

Enumerated Values

Property	Value
anonymous	0
anonymous	1

ExpandableParameters

```
{
  "expand": [
    "string"
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
expand	[string]¦null	false	none	Possible values for expand: workflow_snapshot

GetFileResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "id": "string",
  "scanId": "string",
  "name": "string",
  "path": "string",
  "hash": "string",
  "fileOwner": {
    "userId": "string",
    "fullName": "string",
    "userName": "string",
    "emailAddress": "string"
  },
  "size": 0,
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageName": "string",
  "storageLocation": "string",
  "scanName": "string",
  "progress": {
    "upload": 0,
    "scan": 0
  },
  "scanResult": {
    "extractedFiles": {
      "dataId": "string",
      "scanResultI": 0,
      "detectedBy": 0,
      "filesExtractedCount": 0
    },
    "scanResults": {
      "scanDetails": [
        {
          "engineName": "string",
          "threatFound": "string",
          "scanResultI": 0,
          "defTime": "string",
          "scanTime": 0.1,
          "engId": "string",
          "waitTime": 0.1
        }
      ],
```

```
"scanAllResultI": 0,
      "scanAllResultA": "string",
      "startTime": "string",
      "totalTime": 0.1,
      "totalAvs": 0,
      "totalDetections": 0,
      "progressPercentage": 0
    },
    "fileInfo": {
      "fileSize": 0,
      "uploadTimestamp": "string",
      "md5": "string",
      "sha1": "string",
      "sha256": "string",
      "fileTypeCategory": "string",
      "fileTypeDescription": "string",
      "fileTypeExtension": "string",
      "displayName": "string",
      "fileTypeId": "string"
    },
    "fileScanInfo": {
      "finalVerdict": {
        "blocked": true,
        "verdict": "string",
        "confidence": 0.1,
        "threatLevel": 0.1
      },
      "overviewReport": {
        "analysisState": "string",
        "signalGroups": [
          {
            "description": "string",
            "verdict": "string",
            "signals": [
              {
                "originType": "string",
                "signalReadable": "string",
                "strength": 0.1
              }
            ]
          }
        ]
      },
      "errorCategory": "string",
      "errorCode": 0,
      "errorDetails": "string"
    },
    "processInfo": {
      "postProcessing": {
        "actionsRan": "string",
        "actionsFailed": "string",
        "convertedTo": "string",
        "copyMoveDestination": "string",
```

```
"sanitizationDetails": {
          "failureCategory": "string",
          "details": "string"
        }
      },
      "progressPercentage": 0,
      "userAgent": "string",
      "profile": "string",
      "result": "string",
      "blockedReason": "string",
      "fileTypeSkippedScan": true,
      "processingTime": 0.1,
      "queueTime": 0.1,
      "resultI": 0
    },
    "dataId": "string",
    "dlpInfo": {
      "hits": null,
      "verdict": 0
    },
    "sbomInfo": {
      "finalVerdict": {
        "blocked": true,
        "licensingPackageCounts": {
          "allowed": 0,
          "blocked": 0,
          "unknown": 0,
          "unsupported": 0
        },
        "severity": 0,
        "totalPackageCount": 0,
        "verdict": "string",
        "vulnerabilityCounts": {
          "critical": 0,
          "high": 0,
          "medium": 0,
          "low": 0,
          "unknown": 0
        }
      },
      "overviewReport": {
        "isPartialReport": true,
        "osInfo": {
          "osName": "string",
          "osVersion": "string"
        },
        "scanMode": "string",
        "steps": [
          {
            "command": "string",
            "packages": [
              {
                "authors": "string",
```

```
"ecosystem": "string",
                "group": "string",
                "archive": true,
                "executable": true,
                "uid": "string",
                "licenses": {
                  "allowed": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ],
                  "blocked": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ]
                },
                "packageName": "string",
                "packageVersion": "string",
                "hashes": [
                  {
                    "sha256": "string",
                    "fileName": "string"
                  }
                ],
                "vulnerabilityCounts": {
                  "critical": 0,
```

```
"medium": 0,
                  "low": 0,
                  "unknown": 0
                },
                "vulnerabilities": [
                  {
                    "id": "string",
                    "severity": 0,
                    "source": "string",
                    "fixedVersions": "string",
                    "cwes": [
                      "string"
                    ]
                  }
                ],
                "arch": "string",
                "osName": "string",
                "osVersion": "string",
                "dependencies": [
                  {
                    "ecosystem": "string",
                    "group": "string",
                    "packageName": "string",
                    "packageVersion": "string",
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    }
                  }
                ]
              }
            ],
            "licenseStatus": "string",
            "licensingPackageCounts": {
              "allowed": 0,
              "blocked": 0,
              "unknown": 0,
              "unsupported": 0
            },
```

```
"vulnerabilityCount": 0,
            "sha256": "string",
            "severity": 0,
            "vulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            }
          }
        ],
        "packages": [
          {
            "authors": "string",
            "releaseDate": "2019-08-24T14:15:22Z",
            "ecosystem": "string",
            "group": "string",
            "archive": true,
            "executable": true,
            "uid": "string",
            "licenses": {
              "allowed": [
                {
                  "id": "string",
                  "url": "string",
                  "category": "string",
                  "usageRestriction": {
                    "permissions": [
                      "string"
                    ],
                    "conditions": [
                      "string"
                    ],
                    "limitations": [
                      "string"
                    ]
                  }
                }
              ],
              "blocked": [
                {
                  "id": "string",
                  "url": "string",
                  "category": "string",
                  "usageRestriction": {
                    "permissions": [
                      "string"
                    ],
                    "conditions": [
                      "string"
                    ],
                    "limitations": [
```

```
]
                    }
                  }
                ]
              },
              "packageName": "string",
              "packageVersion": "string",
              "target": "string",
              "hashes": [
                {
                  "sha256": "string",
                  "fileName": "string"
                }
              ],
              "vulnerabilities": [
                {
                  "id": "string",
                  "severity": 0,
                  "source": "string",
                  "fixedVersions": "string",
                  "cwes": [
                    "string"
                  ]
                }
              ],
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              },
              "dependencies": [
                {
                  "ecosystem": "string",
                  "group": "string",
                  "packageName": "string",
                  "packageVersion": "string",
                  "vulnerabilities": [
                    {
                      "id": "string",
                      "severity": 0,
                      "source": "string",
                      "fixedVersions": "string",
                      "cwes": [
                        "string"
                      ]
                    }
                  ],
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
```

```
"unknown": 0
                }
              }
            ]
          }
        ],
        "dependenciesPackageCount": 0,
        "dependenciesVulnerabilityTotal": 0,
        "dependenciesVulnerabilityCounts": {
          "critical": 0,
          "high": 0,
          "medium": 0,
          "low": 0,
          "unknown": 0
        }
      },
      "resultTemplateHash": "string"
    },
    "vulnerabilityInfo": {
      "vulnerabilityInfoResult": {
        "code": 0,
        "hash": "string",
        "method": 0,
        "timeStamp": "string",
        "timing": 0,
        "detectedProduct": {
          "hasVulnerability": true,
          "isCurrent": true,
          "product": {
            "id": 0,
            "name": "string"
          },
          "remediationLink": "string",
          "severity": "string",
          "sigName": "string",
          "signature": 0,
          "vendor": {
            "id": 0,
            "name": "string"
          },
          "version": "string",
          "versionData": {
            "countBehind": 0,
            "feedId": 0,
            "version": "string"
          },
          "vulnerabilities": [
            {
              "description": "string",
              "details": {
                "cpe": "string",
                "cve": "string",
                "cvss": {
```

```
"accessVector": "string",
                  "authentication": "string",
                  "availabilityImpact": "string",
                  "confidentialityImpact": "string",
                  "generatedOnEpoch": "string",
                  "integrityImpact": "string",
                  "score": "string",
                  "source": "string"
                },
                "cvss3": {
                  "attackComplexity": "string",
                  "attackVector": "string",
                  "availabilityImpact": "string",
                  "baseScore": "string",
                  "baseSeverity": "string",
                  "confidentialityImpact": "string",
                  "exploitabilityScore": "string",
                  "impactScore": "string",
                  "integrityImpact": "string",
                  "privilegesRequired": "string",
                  "scope": "string",
                  "userInteraction": "string",
                  "vectorString": "string"
                },
                "cwe": "string",
                "lastModifiedEpoch": "string",
                "publishedEpoch": "string",
                "references": [
                  "string"
                ]
              },
              "severity": "string",
              "severityIndex": 0,
              "staticId": 0
            }
          ]
        }
      }
    },
    "sessionCookie": "string",
    "fileTypeInfo": {
      "fileInfoExtended": {
        "extensions": "string",
        "groupId": "string",
        "typeId": "string"
      }
    },
    "scanInstanceUrl": "string",
    "scanWorkflowSnapshotId": "string",
    "coreEngines": [
      {
        "engineId": "string",
        "engineName": "string",
```

```
"dbVersion": "string",
        "definitionTime": "string"
      }
    ]
  },
  "processingState": 0,
  "exception": {
    "exceptionDetails": "string",
    "exceptionMessage": "string"
  },
  "created": "2019-08-24T14:15:22Z",
  "lastModified": "2019-08-24T14:15:22Z",
  "discoveryStarted": "2019-08-24T14:15:22Z",
  "fileProcessingDetails": {
    "streamingStartedTime": "2019-08-24T14:15:22Z",
    "streamingCompletedTime": "2019-08-24T14:15:22Z",
    "streamingTimeSpanMs": 0,
    "scanCompletedTime": "2019-08-24T14:15:22Z",
    "processingTimeSpanMs": 0
  },
  "coreEngines": [
    {
      "engineId": "string",
      "engineName": "string",
      "engineVersion": "string",
      "dbVersion": "string",
      "definitionTime": "string"
    }
  ],
  "remediation": {
    "isTagged": true,
    "isDeleted": true,
    "isMoved": true,
    "isCopied": true,
    "deepCdr": {
      "sanitizationDetails": {
        "failureCategory": "string",
        "details": "string"
      },
      "cdrObjectState": 0,
      "fileWasReplaced": true
    },
    "postActionFailedInformation": [
      {
        "objectActionType": 0,
        "exception": {
          "exceptionDetails": "string",
          "exceptionMessage": "string"
        }
      }
    ],
    "objectRemediationActions": [
      {
```

```
"objectActionState": 0,
        "cdrObjectState": 0,
        "exceptions": [
          "string"
        ],
        "id": "string",
        "fileId": "string",
        "fileScanCompletedTime": "2019-08-24T14:15:22Z",
        "remediationId": "string",
        "tenantId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z"
      }
    ]
  },
  "metaDefenderCoreUrl": "string",
  "discoveryStartedDaySec": 0,
  "fileDto": {
    "result": 0,
    "responseKey": "string",
    "responseMessage": "string",
    "responseMessageParams": {
      "property1": "string",
      "property2": "string"
    },
    "id": "string",
    "scanId": "string",
    "name": "string",
    "path": "string",
    "hash": "string",
    "fileOwner": {
      "userId": "string",
      "fullName": "string",
      "userName": "string",
      "emailAddress": "string"
    },
    "size": 0,
    "fileTypeCategory": 0,
    "extension": "string",
    "storageType": {
      "name": "string",
      "categoryType": 0,
      "protocolType": 0,
      "vendorType": 0
    },
    "storageName": "string",
    "storageLocation": "string",
    "scanName": "string",
    "progress": {
      "upload": 0,
      "scan": 0
    },
    "scanResult": {
      "extractedFiles": {
```

```
"scanResultI": 0,
        "detectedBy": 0,
        "filesExtractedCount": 0
      },
      "scanResults": {
        "scanDetails": [
          {
            "engineName": "string",
            "threatFound": "string",
            "scanResultI": 0,
            "defTime": "string",
            "scanTime": 0.1,
            "engId": "string",
            "waitTime": 0.1
          }
        ],
        "dataId": "string",
        "scanAllResultI": 0,
        "scanAllResultA": "string",
        "startTime": "string",
        "totalTime": 0.1,
        "totalAvs": 0,
        "totalDetections": 0,
        "progressPercentage": 0
      },
      "fileInfo": {
        "fileSize": 0,
        "uploadTimestamp": "string",
        "md5": "string",
        "sha1": "string",
        "sha256": "string",
        "fileTypeCategory": "string",
        "fileTypeDescription": "string",
        "fileTypeExtension": "string",
        "displayName": "string",
        "fileTypeId": "string"
      },
      "fileScanInfo": {
        "finalVerdict": {
          "blocked": true,
          "verdict": "string",
          "confidence": 0.1,
          "threatLevel": 0.1
        },
        "overviewReport": {
          "analysisState": "string",
          "signalGroups": [
            {
              "description": "string",
              "verdict": "string",
              "signals": [
                {
                  "originType": "string",
```

```
"strength": 0.1
                }
              ]
            }
          ]
        },
        "errorCategory": "string",
        "errorCode": 0,
        "errorDetails": "string"
      },
      "processInfo": {
        "postProcessing": {
          "actionsRan": "string",
          "actionsFailed": "string",
          "convertedTo": "string",
          "copyMoveDestination": "string",
          "convertedDestination": "string",
          "sanitizationDetails": {
            "failureCategory": "string",
            "details": "string"
          }
        },
        "progressPercentage": 0,
        "userAgent": "string",
        "profile": "string",
        "result": "string",
        "blockedReason": "string",
        "fileTypeSkippedScan": true,
        "processingTime": 0.1,
        "queueTime": 0.1,
        "resultI": 0
      },
      "dataId": "string",
      "dlpInfo": {
        "hits": null,
        "verdict": 0
      },
      "sbomInfo": {
        "finalVerdict": {
          "blocked": true,
          "licensingPackageCounts": {
            "allowed": 0,
            "blocked": 0,
            "unknown": 0,
            "unsupported": 0
          },
          "severity": 0,
          "totalPackageCount": 0,
          "verdict": "string",
          "vulnerabilityCounts": {
            "critical": 0,
            "high": 0,
            "medium": 0,
```

```
"unknown": 0
          }
        },
        "overviewReport": {
          "isPartialReport": true,
          "osInfo": {
            "osName": "string",
            "osVersion": "string"
          },
          "scanMode": "string",
          "steps": [
            {
              "command": "string",
              "packages": [
                {
                  "authors": "string",
                  "releaseDate": "2019-08-24T14:15:22Z",
                  "ecosystem": "string",
                  "group": "string",
                  "archive": true,
                  "executable": true,
                  "uid": "string",
                  "licenses": {
                    "allowed": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            "string"
                          ],
                          "conditions": [
                            "string"
                          ],
                          "limitations": [
                            "string"
                          ]
                        }
                      }
                    ],
                    "blocked": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            "string"
                          ],
                          "conditions": [
                            "string"
                          ],
```

```
"string"
                            ]
                          }
                        }
                      ]
                    },
                    "packageName": "string",
                    "packageVersion": "string",
                    "hashes": [
                      {
                        "sha256": "string",
                        "fileName": "string"
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    },
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "arch": "string",
                    "osName": "string",
                    "osVersion": "string",
                    "dependencies": [
                      {
                        "ecosystem": "string",
                        "group": "string",
                        "packageName": "string",
                        "packageVersion": "string",
                        "vulnerabilities": [
                          {
                            "id": "string",
                            "severity": 0,
                            "source": "string",
                            "fixedVersions": "string",
                            "cwes": [
                              "string"
                            ]
                          }
                        ],
                        "vulnerabilityCounts": {
```

```
"high": 0,
                        "medium": 0,
                        "low": 0,
                        "unknown": 0
                      }
                    }
                  ]
                }
              ],
              "licenseStatus": "string",
              "licensingPackageCounts": {
                "allowed": 0,
                "blocked": 0,
                "unknown": 0,
                "unsupported": 0
              },
              "step": 0,
              "vulnerabilityCount": 0,
              "sha256": "string",
              "severity": 0,
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              }
            }
          ],
          "packages": [
            {
              "authors": "string",
              "releaseDate": "2019-08-24T14:15:22Z",
              "ecosystem": "string",
              "group": "string",
              "archive": true,
              "executable": true,
              "uid": "string",
              "licenses": {
                "allowed": [
                  {
                    "id": "string",
                    "url": "string",
                    "category": "string",
                    "usageRestriction": {
                      "permissions": [
                        "string"
                      ],
                      "conditions": [
                        "string"
                      ],
                      "limitations": [
                        "string"
```

```
}
                    }
                  ],
                  "blocked": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ]
                },
                "packageName": "string",
                "packageVersion": "string",
                "target": "string",
                "hashes": [
                  {
                    "sha256": "string",
                    "fileName": "string"
                  }
                ],
                "vulnerabilities": [
                  {
                    "id": "string",
                    "severity": 0,
                    "source": "string",
                    "fixedVersions": "string",
                    "cwes": [
                      "string"
                    ]
                  }
                ],
                "vulnerabilityCounts": {
                  "critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
                },
                "dependencies": [
                  {
                    "ecosystem": "string",
                    "group": "string",
```

```
"packageVersion": "string",
                  "vulnerabilities": [
                    {
                      "id": "string",
                      "severity": 0,
                      "source": "string",
                      "fixedVersions": "string",
                      "cwes": [
                        "string"
                      ]
                    }
                  ],
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  }
                }
              ]
            }
          ],
          "dependenciesPackageCount": 0,
          "dependenciesVulnerabilityTotal": 0,
          "dependenciesVulnerabilityCounts": {
            "critical": 0,
            "high": 0,
            "medium": 0,
            "low": 0,
            "unknown": 0
          }
        },
        "resultTemplateHash": "string"
      },
      "vulnerabilityInfo": {
        "vulnerabilityInfoResult": {
          "code": 0,
          "hash": "string",
          "method": 0,
          "timeStamp": "string",
          "timing": 0,
          "detectedProduct": {
            "hasVulnerability": true,
            "isCurrent": true,
            "product": {
              "id": 0,
              "name": "string"
            },
            "remediationLink": "string",
            "severity": "string",
            "sigName": "string",
            "signature": 0,
```

```
"id": 0,
              "name": "string"
            },
            "version": "string",
            "versionData": {
              "countBehind": 0,
              "feedId": 0,
              "version": "string"
            },
            "vulnerabilities": [
              {
                "description": "string",
                "details": {
                  "cpe": "string",
                  "cve": "string",
                  "cvss": {
                    "accessComplexity": "string",
                    "accessVector": "string",
                    "authentication": "string",
                    "availabilityImpact": "string",
                    "confidentialityImpact": "string",
                    "generatedOnEpoch": "string",
                    "integrityImpact": "string",
                    "score": "string",
                    "source": "string"
                  },
                  "cvss3": {
                    "attackComplexity": "string",
                    "attackVector": "string",
                    "availabilityImpact": "string",
                    "baseScore": "string",
                    "baseSeverity": "string",
                    "confidentialityImpact": "string",
                    "exploitabilityScore": "string",
                    "impactScore": "string",
                    "integrityImpact": "string",
                    "privilegesRequired": "string",
                    "scope": "string",
                    "userInteraction": "string",
                    "vectorString": "string"
                  },
                  "cwe": "string",
                  "lastModifiedEpoch": "string",
                  "publishedEpoch": "string",
                  "references": [
                    "string"
                  ]
                },
                "severity": "string",
                "severityIndex": 0,
                "staticId": 0
              }
            ]
```

```
}
      },
      "sessionCookie": "string",
      "fileTypeInfo": {
        "fileInfoExtended": {
          "extensions": "string",
          "groupId": "string",
          "typeId": "string"
        }
      },
      "scanInstanceUrl": "string",
      "scanWorkflowSnapshotId": "string",
      "coreEngines": [
        {
          "engineId": "string",
          "engineName": "string",
          "engineVersion": "string",
          "dbVersion": "string",
          "definitionTime": "string"
        }
      ]
    },
    "processingState": 0,
    "exception": {
      "exceptionDetails": "string",
      "exceptionMessage": "string"
    },
    "created": "2019-08-24T14:15:22Z",
    "lastModified": "2019-08-24T14:15:22Z",
    "discoveryStarted": "2019-08-24T14:15:22Z",
    "fileProcessingDetails": {
      "streamingStartedTime": "2019-08-24T14:15:22Z",
      "streamingCompletedTime": "2019-08-24T14:15:22Z",
      "streamingTimeSpanMs": 0,
      "scanCompletedTime": "2019-08-24T14:15:22Z",
      "processingTimeSpanMs": 0
    },
    "coreEngines": [
      {
        "engineId": "string",
        "engineName": "string",
        "engineVersion": "string",
        "dbVersion": "string",
        "definitionTime": "string"
      }
    ],
    "remediation": {
      "isTagged": true,
      "isDeleted": true,
      "isMoved": true,
      "isCopied": true,
      "deepCdr": {
        "sanitizationDetails": {
```

```
"details": "string"
        },
        "cdrObjectState": 0,
        "fileWasReplaced": true
      },
      "postActionFailedInformation": [
        {
          "objectActionType": 0,
          "exception": {
            "exceptionDetails": "string",
            "exceptionMessage": "string"
          }
        }
      ],
      "objectRemediationActions": [
        {
          "objectActionType": 0,
          "objectActionState": 0,
          "cdrObjectState": 0,
          "exceptions": [
            "string"
          ],
          "id": "string",
          "fileId": "string",
          "fileScanCompletedTime": "2019-08-24T14:15:22Z",
          "remediationId": "string",
          "tenantId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z"
        }
      ]
    },
    "metaDefenderCoreUrl": "string",
    "discoveryStartedDaySec": 0,
    "objectStorageMetadata": "string",
    "scanWorkflowSnapshotId": "string",
    "scanWorkflowSnapshot": {
      "id": "string",
      "scanId": "string",
      "workflow": {
        "id": "string",
        "scanPoolId": "string",
        "scanProcessFlow": 0,
        "isIdentityScanningEnabled": true,
        "scanIntervalDays": 0,
        "rules": [
          "string"
        ],
        "type": 0,
        "workflowSupportedTechnologies": {
          "dlp": {
            "status": 0
          },
          "deepCdr": {
```

```
},
          "sandBox": {
            "status": 0
          },
          "metascan": {
            "status": 0
          }
        },
        "name": "string",
        "userAgent": "string",
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanPoolId": "string",
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "filters": {
          "id": "string",
          "minimumSize": 0,
          "maximumSize": 0,
          "minimumDateTime": "2019-08-24T14:15:22Z",
          "maximumDateTime": "2019-08-24T14:15:22Z"
        },
        "fileDefinition": {
          "blockingStrategy": 0,
          "configurableBlockingStrategy": {
```

```
"shouldTreatFilesWithVulnerabilitiesAsBlocked": true
          }
        },
        "remediations": [
          {
            "id": "string",
            "workflowId": "string",
            "isEnabled": true,
            "configuration": "string",
            "type": 0,
            "handlingType": 0
          }
        ],
        "tenantId": "string"
      },
      "startTimestamp": 0,
      "stopTimestamp": 0,
      "scanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "failOverScanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
```

```
"timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "failOverScanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "storage": {
        "id": "string",
        "name": "string",
        "groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      },
      "workflowStatistics": {
        "taggingStatistics": {
          "total": 0
        },
        "keepAllowedStatistics": {
          "total": 0
        },
        "keepBlockedStatistics": {
          "total": 0
        },
        "keepSanitizedStatistics": {
          "total": 0
        },
        "movedAllowedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
```

```
"name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "movedSanitizedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "movedBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
```

```
"source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "copyAllowedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "copyBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
```

```
"categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "deleteAllowedStatistics": {
          "total": 0
        },
        "deleteBlockedStatistics": {
          "total": 0
        },
        "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
      },
      "account": {
        "id": "string",
        "name": "string",
        "description": "string",
        "credentials": "string",
        "protocolType": 0,
        "vendorType": 0,
        "autoDiscovery": true,
        "autodiscoveryDestinationGroupId": "string"
      },
      "realTimeStartDate": "2019-08-24T14:15:22Z"
    },
    "versions": [
      {
        "result": 0,
        "responseKey": "string",
        "responseMessage": "string",
        "responseMessageParams": {
          "property1": "string",
          "property2": "string"
        },
        "id": "string",
        "scanId": "string",
        "name": "string",
        "path": "string",
        "hash": "string",
        "fileOwner": {
          "userId": "string",
          "fullName": "string",
          "userName": "string",
          "emailAddress": "string"
        },
```

```
"fileTypeCategory": 0,
        "extension": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageName": "string",
        "storageLocation": "string",
        "scanName": "string",
        "progress": {
          "upload": 0,
          "scan": 0
        },
        "scanResult": {
          "extractedFiles": {
            "dataId": "string",
            "scanResultI": 0,
            "detectedBy": 0,
            "filesExtractedCount": 0
          },
          "scanResults": {
            "scanDetails": [
              {
                "engineName": "string",
                "threatFound": "string",
                "scanResultI": 0,
                "defTime": "string",
                "scanTime": 0.1,
                "engId": "string",
                "waitTime": 0.1
              }
            ],
            "dataId": "string",
            "scanAllResultI": 0,
            "scanAllResultA": "string",
            "startTime": "string",
            "totalTime": 0.1,
            "totalAvs": 0,
            "totalDetections": 0,
            "progressPercentage": 0
          },
          "fileInfo": {
            "fileSize": 0,
            "uploadTimestamp": "string",
            "md5": "string",
            "sha1": "string",
            "sha256": "string",
            "fileTypeCategory": "string",
            "fileTypeDescription": "string",
            "fileTypeExtension": "string",
            "displayName": "string",
```

```
},
          "fileScanInfo": {
            "finalVerdict": {
              "blocked": true,
              "verdict": "string",
              "confidence": 0.1,
              "threatLevel": 0.1
            },
            "overviewReport": {
              "analysisState": "string",
              "signalGroups": [
                {
                  "description": "string",
                  "verdict": "string",
                  "signals": [
                    {
                      "originType": "string",
                      "signalReadable": "string",
                      "strength": 0.1
                    }
                  ]
                }
              ]
            },
            "errorCategory": "string",
            "errorCode": 0,
            "errorDetails": "string"
          },
          "processInfo": {
            "postProcessing": {
              "actionsRan": "string",
              "actionsFailed": "string",
              "convertedTo": "string",
              "copyMoveDestination": "string",
              "convertedDestination": "string",
              "sanitizationDetails": {
                "failureCategory": "string",
                "details": "string"
              }
            },
            "progressPercentage": 0,
            "userAgent": "string",
            "profile": "string",
            "result": "string",
            "blockedReason": "string",
            "fileTypeSkippedScan": true,
            "processingTime": 0.1,
            "queueTime": 0.1,
            "resultI": 0
          },
          "dataId": "string",
          "dlpInfo": {
            "hits": null,
```

```
},
          "sbomInfo": {
            "finalVerdict": {
              "blocked": true,
              "licensingPackageCounts": {
                "allowed": 0,
                "blocked": 0,
                "unknown": 0,
                "unsupported": 0
              },
              "severity": 0,
              "totalPackageCount": 0,
              "verdict": "string",
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              }
            },
            "overviewReport": {
              "isPartialReport": true,
              "osInfo": {
                "osName": "string",
                "osVersion": "string"
              },
              "scanMode": "string",
              "steps": [
                {
                  "command": "string",
                  "packages": [
                    {
                      "authors": "string",
                      "releaseDate": "2019-08-24T14:15:22Z",
                      "ecosystem": "string",
                      "group": "string",
                      "archive": true,
                      "executable": true,
                      "uid": "string",
                      "licenses": {
                        "allowed": [
                          {
                            "id": "string",
                            "url": "string",
                            "category": "string",
                            "usageRestriction": {
                              "permissions": [],
                              "conditions": [],
                              "limitations": []
                            }
                          }
                        ],
```

```
{
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [],
                          "conditions": [],
                          "limitations": []
                        }
                      }
                    ]
                  },
                  "packageName": "string",
                  "packageVersion": "string",
                  "hashes": [
                    {
                      "sha256": "string",
                      "fileName": "string"
                    }
                  ],
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  },
                  "vulnerabilities": [
                    {
                      "id": "string",
                      "severity": 0,
                      "source": "string",
                      "fixedVersions": "string",
                      "cwes": [
                        "string"
                      ]
                    }
                  ],
                  "arch": "string",
                  "osName": "string",
                  "osVersion": "string",
                  "dependencies": [
                    {
                      "ecosystem": "string",
                      "group": "string",
                      "packageName": "string",
                      "packageVersion": "string",
                      "vulnerabilities": [
                        {
                          "cwes": []
                        }
                      ],
                      "vulnerabilityCounts": {
```

```
"high": 0,
                            "medium": 0,
                            "low": 0,
                            "unknown": 0
                          }
                        }
                      ]
                    }
                  ],
                  "licenseStatus": "string",
                  "licensingPackageCounts": {
                    "allowed": 0,
                    "blocked": 0,
                    "unknown": 0,
                    "unsupported": 0
                  },
                  "step": 0,
                  "vulnerabilityCount": 0,
                  "sha256": "string",
                  "severity": 0,
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  }
                }
              ],
              "packages": [
                {
                  "authors": "string",
                  "releaseDate": "2019-08-24T14:15:22Z",
                  "ecosystem": "string",
                  "group": "string",
                  "archive": true,
                  "executable": true,
                  "uid": "string",
                  "licenses": {
                    "allowed": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            "string"
                          ],
                          "conditions": [
                            "string"
                          ],
                          "limitations": [
                            "string"
```

```
}
                    }
                  ],
                  "blocked": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ]
                },
                "packageName": "string",
                "packageVersion": "string",
                "target": "string",
                "hashes": [
                  {
                    "sha256": "string",
                    "fileName": "string"
                  }
                ],
                "vulnerabilities": [
                  {
                    "id": "string",
                    "severity": 0,
                    "source": "string",
                    "fixedVersions": "string",
                    "cwes": [
                      "string"
                    ]
                  }
                ],
                "vulnerabilityCounts": {
                  "critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
                },
                "dependencies": [
                  {
                    "ecosystem": "string",
                    "group": "string",
```

```
"packageVersion": "string",
                      "vulnerabilities": [
                        {
                          "id": "string",
                          "severity": 0,
                          "source": "string",
                          "fixedVersions": "string",
                          "cwes": [
                            "string"
                          ]
                        }
                      ],
                      "vulnerabilityCounts": {
                        "critical": 0,
                        "high": 0,
                        "medium": 0,
                        "low": 0,
                        "unknown": 0
                      }
                    }
                  ]
                }
              ],
              "dependenciesPackageCount": 0,
              "dependenciesVulnerabilityTotal": 0,
              "dependenciesVulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              }
            },
            "resultTemplateHash": "string"
          },
          "vulnerabilityInfo": {
            "vulnerabilityInfoResult": {
              "code": 0,
              "hash": "string",
              "method": 0,
              "timeStamp": "string",
              "timing": 0,
              "detectedProduct": {
                "hasVulnerability": true,
                "isCurrent": true,
                "product": {
                  "id": 0,
                  "name": "string"
                },
                "remediationLink": "string",
                "severity": "string",
                "sigName": "string",
                "signature": 0,
```

```
"id": 0,
                  "name": "string"
                },
                "version": "string",
                "versionData": {
                  "countBehind": 0,
                  "feedId": 0,
                  "version": "string"
                },
                "vulnerabilities": [
                  {
                    "description": "string",
                    "details": {
                      "cpe": "string",
                      "cve": "string",
                      "cvss": {
                        "accessComplexity": "string",
                        "accessVector": "string",
                        "authentication": "string",
                        "availabilityImpact": "string",
                        "confidentialityImpact": "string",
                        "generatedOnEpoch": "string",
                        "integrityImpact": "string",
                        "score": "string",
                        "source": "string"
                      },
                      "cvss3": {
                        "attackComplexity": "string",
                        "attackVector": "string",
                        "availabilityImpact": "string",
                        "baseScore": "string",
                        "baseSeverity": "string",
                        "confidentialityImpact": "string",
                        "exploitabilityScore": "string",
                        "impactScore": "string",
                        "integrityImpact": "string",
                        "privilegesRequired": "string",
                        "scope": "string",
                        "userInteraction": "string",
                        "vectorString": "string"
                      },
                      "cwe": "string",
                      "lastModifiedEpoch": "string",
                      "publishedEpoch": "string",
                      "references": [
                        "string"
                      ]
                    },
                    "severity": "string",
                    "severityIndex": 0,
                    "staticId": 0
                  }
                ]
```

```
}
          },
          "sessionCookie": "string",
          "fileTypeInfo": {
            "fileInfoExtended": {
              "extensions": "string",
              "groupId": "string",
              "typeId": "string"
            }
          },
          "scanInstanceUrl": "string",
          "scanWorkflowSnapshotId": "string",
          "coreEngines": [
            {
              "engineId": "string",
              "engineName": "string",
              "engineVersion": "string",
              "dbVersion": "string",
              "definitionTime": "string"
            }
          ]
        },
        "processingState": 0,
        "exception": {
          "exceptionDetails": "string",
          "exceptionMessage": "string"
        },
        "created": "2019-08-24T14:15:22Z",
        "lastModified": "2019-08-24T14:15:22Z",
        "discoveryStarted": "2019-08-24T14:15:22Z",
        "fileProcessingDetails": {
          "streamingStartedTime": "2019-08-24T14:15:22Z",
          "streamingCompletedTime": "2019-08-24T14:15:22Z",
          "streamingTimeSpanMs": 0,
          "scanCompletedTime": "2019-08-24T14:15:22Z",
          "processingTimeSpanMs": 0
        },
        "coreEngines": [
          {
            "engineId": "string",
            "engineName": "string",
            "engineVersion": "string",
            "dbVersion": "string",
            "definitionTime": "string"
          }
        ],
        "remediation": {
          "isTagged": true,
          "isDeleted": true,
          "isMoved": true,
          "isCopied": true,
          "deepCdr": {
            "sanitizationDetails": {
```

```
"details": "string"
            },
            "cdrObjectState": 0,
            "fileWasReplaced": true
          },
          "postActionFailedInformation": [
            {
              "objectActionType": 0,
              "exception": {
                "exceptionDetails": "string",
                "exceptionMessage": "string"
              }
            }
          ],
          "objectRemediationActions": [
            {
              "objectActionType": 0,
              "objectActionState": 0,
              "cdrObjectState": 0,
              "exceptions": [
                "string"
              ],
              "id": "string",
              "fileId": "string",
              "fileScanCompletedTime": "2019-08-24T14:15:22Z",
              "remediationId": "string",
              "tenantId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z"
            }
          ]
        },
        "metaDefenderCoreUrl": "string",
        "discoveryStartedDaySec": 0,
        "objectStorageMetadata": "string",
        "scanWorkflowSnapshotId": "string",
        "scanWorkflowSnapshot": {
          "id": "string",
          "scanId": "string",
          "workflow": {
            "id": "string",
            "scanPoolId": "string",
            "scanProcessFlow": 0,
            "isIdentityScanningEnabled": true,
            "scanIntervalDays": 0,
            "rules": [
              "string"
            ],
            "type": 0,
            "workflowSupportedTechnologies": {
              "dlp": {
                "status": 0
              },
              "deepCdr": {
```

```
},
              "sandBox": {
                "status": 0
              },
              "metascan": {
                "status": 0
              }
            },
            "name": "string",
            "userAgent": "string",
            "scanPool": {
              "id": "string",
              "name": "string",
              "scanPoolType": 0,
              "isDefault": true,
              "scanInstances": [
                {
                  "id": "string",
                  "scanPoolId": "string",
                  "url": "string",
                  "apiKey": "string",
                  "timeout": "string",
                  "scanInstanceType": 0
                }
              ]
            },
            "failOverScanPoolId": "string",
            "failOverScanPool": {
              "id": "string",
              "name": "string",
              "scanPoolType": 0,
              "isDefault": true,
              "scanInstances": [
                {
                  "id": "string",
                  "scanPoolId": "string",
                  "url": "string",
                  "apiKey": "string",
                  "timeout": "string",
                  "scanInstanceType": 0
                }
              ]
            },
            "filters": {
              "id": "string",
              "minimumSize": 0,
              "maximumSize": 0,
              "minimumDateTime": "2019-08-24T14:15:22Z",
              "maximumDateTime": "2019-08-24T14:15:22Z"
            },
            "fileDefinition": {
              "blockingStrategy": 0,
              "configurableBlockingStrategy": {
```

```
"shouldTreatFilesWithVulnerabilitiesAsBlocked": true
              }
            },
            "remediations": [
              {
                "id": "string",
                "workflowId": "string",
                "isEnabled": true,
                "configuration": "string",
                "type": 0,
                "handlingType": 0
              }
            ],
            "tenantId": "string"
          },
          "startTimestamp": 0,
          "stopTimestamp": 0,
          "scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ],
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
```

```
"timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ],
          "storage": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          },
          "workflowStatistics": {
            "taggingStatistics": {
              "total": 0
            },
            "keepAllowedStatistics": {
              "total": 0
            },
            "keepBlockedStatistics": {
              "total": 0
            },
            "keepSanitizedStatistics": {
              "total": 0
            },
            "movedAllowedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
```

```
"name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "movedSanitizedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "movedBlockedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
```

```
"source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "copyAllowedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "copyBlockedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
```

```
"categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "deleteAllowedStatistics": {
              "total": 0
            },
            "deleteBlockedStatistics": {
              "total": 0
            },
            "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
          },
          "account": {
            "id": "string",
            "name": "string",
            "description": "string",
            "credentials": "string",
            "protocolType": 0,
            "vendorType": 0,
            "autoDiscovery": true,
            "autodiscoveryDestinationGroupId": "string"
          },
          "realTimeStartDate": "2019-08-24T14:15:22Z"
        },
        "versions": [],
        "hasBlockedVersions": true,
        "versionsCount": 0,
        "objectType": 0,
        "skippedScan": true,
        "$type": "string",
        "url": "string",
        "fileId": "string"
      }
    ],
    "hasBlockedVersions": true,
    "versionsCount": 0,
    "objectType": 0,
    "skippedScan": true,
    "$type": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» id	string¦null	false	none	none
» scanId	string¦null	false	none	none
» name	string¦null	false	none	none
» path	string¦null	false	none	none
» hash	string¦null	false	none	none
» fileOwner	FileOwner¦null	false	none	none
» size	integer(int64)	false	none	none
» storageType	StorageType¦null	false	none	none
» storageName	string¦null	false	none	none
» storageLocation	string¦null	false	none	none
» scanName	string¦null	false	none	none
» progress	Progress¦null	false	none	none
» scanResult	CoreScanResult¦null	false	none	none
» processingState	ProcessingState2	false	none	- 0: InProgress - 1: Failed - 2: Canceled - 3: Available - 4: FailedRemediation
» exception	Exception¦null	false	none	none
» created	string(date-time)	false	none	none
» lastModified	string(date-time)	false	none	none
» discoveryStarted	string(date-time)	false	none	none
» fileProcessingDetails	FileProcessingDetails¦null	false	none	none
» coreEngines	[CoreEngine]¦null	false	none	none
» remediation	FileRemediation¦null	false	none	none
» metaDefenderCoreUrl	string¦null	false	none	none
» discoveryStartedDaySec	integer(int64)	false	none	none

» fileDto	File¦null	false	none	none

FileOwner


```
{
  "userId": "string",
  "fullName": "string",
  "userName": "string",
  "emailAddress": "string"
}
```


Properties

Name	Type	Required	Restrictions	Description
userId	string¦null	false	none	none
fullName	string¦null	false	none	none
userName	string¦null	false	none	none
emailAddress	string¦null	false	none	none

Progress


```
{
  "upload": 0,
  "scan": 0
}
```


Properties

Name	Type	Required	Restrictions	Description
upload	integer	false	none	none
scan	integer	false	none	none

CoreScanResult

```
{
  "extractedFiles": {
    "dataId": "string",
    "scanResultI": 0,
    "detectedBy": 0,
    "filesExtractedCount": 0
  },
  "scanResults": {
    "scanDetails": [
      {
        "engineName": "string",
        "threatFound": "string",
        "scanResultI": 0,
        "defTime": "string",
        "scanTime": 0.1,
        "engId": "string",
        "waitTime": 0.1
      }
    ],
    "dataId": "string",
    "scanAllResultI": 0,
    "scanAllResultA": "string",
    "startTime": "string",
    "totalTime": 0.1,
    "totalAvs": 0,
    "totalDetections": 0,
    "progressPercentage": 0
  },
  "fileInfo": {
    "fileSize": 0,
    "uploadTimestamp": "string",
    "md5": "string",
    "sha1": "string",
    "sha256": "string",
    "fileTypeCategory": "string",
    "fileTypeDescription": "string",
    "fileTypeExtension": "string",
    "displayName": "string",
    "fileTypeId": "string"
  },
  "fileScanInfo": {
    "finalVerdict": {
      "blocked": true,
      "verdict": "string",
      "confidence": 0.1,
      "threatLevel": 0.1
    },
    "overviewReport": {
      "analysisState": "string",
      "signalGroups": [
        {
          "description": "string",
```

```
"signals": [
            {
              "originType": "string",
              "signalReadable": "string",
              "strength": 0.1
            }
          ]
        }
      ]
    },
    "errorCategory": "string",
    "errorCode": 0,
    "errorDetails": "string"
  },
  "processInfo": {
    "postProcessing": {
      "actionsRan": "string",
      "actionsFailed": "string",
      "convertedTo": "string",
      "copyMoveDestination": "string",
      "convertedDestination": "string",
      "sanitizationDetails": {
        "failureCategory": "string",
        "details": "string"
      }
    },
    "progressPercentage": 0,
    "userAgent": "string",
    "profile": "string",
    "result": "string",
    "blockedReason": "string",
    "fileTypeSkippedScan": true,
    "processingTime": 0.1,
    "queueTime": 0.1,
    "resultI": 0
  },
  "dataId": "string",
  "dlpInfo": {
    "hits": null,
    "verdict": 0
  },
  "sbomInfo": {
    "finalVerdict": {
      "blocked": true,
      "licensingPackageCounts": {
        "allowed": 0,
        "blocked": 0,
        "unknown": 0,
        "unsupported": 0
      },
      "severity": 0,
      "totalPackageCount": 0,
      "verdict": "string",
```

```
"critical": 0,
        "high": 0,
        "medium": 0,
        "low": 0,
        "unknown": 0
      }
    },
    "overviewReport": {
      "isPartialReport": true,
      "osInfo": {
        "osName": "string",
        "osVersion": "string"
      },
      "scanMode": "string",
      "steps": [
        {
          "command": "string",
          "packages": [
            {
              "authors": "string",
              "releaseDate": "2019-08-24T14:15:22Z",
              "ecosystem": "string",
              "group": "string",
              "archive": true,
              "executable": true,
              "uid": "string",
              "licenses": {
                "allowed": [
                  {
                    "id": "string",
                    "url": "string",
                    "category": "string",
                    "usageRestriction": {
                      "permissions": [
                        "string"
                      ],
                      "conditions": [
                        "string"
                      ],
                      "limitations": [
                        "string"
                      ]
                    }
                  }
                ],
                "blocked": [
                  {
                    "id": "string",
                    "url": "string",
                    "category": "string",
                    "usageRestriction": {
                      "permissions": [
                        "string"
```

```
"conditions": [
                        "string"
                      ],
                      "limitations": [
                        "string"
                      ]
                    }
                  }
                ]
              },
              "packageName": "string",
              "packageVersion": "string",
              "hashes": [
                {
                  "sha256": "string",
                  "fileName": "string"
                }
              ],
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              },
              "vulnerabilities": [
                {
                  "id": "string",
                  "severity": 0,
                  "source": "string",
                  "fixedVersions": "string",
                  "cwes": [
                    "string"
                  ]
                }
              ],
              "arch": "string",
              "osName": "string",
              "osVersion": "string",
              "dependencies": [
                {
                  "ecosystem": "string",
                  "group": "string",
                  "packageName": "string",
                  "packageVersion": "string",
                  "vulnerabilities": [
                    {
                      "id": "string",
                      "severity": 0,
                      "source": "string",
                      "fixedVersions": "string",
                      "cwes": [
                        "string"
```

```
}
                  ],
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  }
                }
              ]
            }
          ],
          "licenseStatus": "string",
          "licensingPackageCounts": {
            "allowed": 0,
            "blocked": 0,
            "unknown": 0,
            "unsupported": 0
          },
          "step": 0,
          "vulnerabilityCount": 0,
          "sha256": "string",
          "severity": 0,
          "vulnerabilityCounts": {
            "critical": 0,
            "high": 0,
            "medium": 0,
            "low": 0,
            "unknown": 0
          }
        }
      ],
      "packages": [
        {
          "authors": "string",
          "releaseDate": "2019-08-24T14:15:22Z",
          "ecosystem": "string",
          "group": "string",
          "archive": true,
          "executable": true,
          "uid": "string",
          "licenses": {
            "allowed": [
              {
                "id": "string",
                "url": "string",
                "category": "string",
                "usageRestriction": {
                  "permissions": [
                    "string"
                  ],
                  "conditions": [
```

```
],
                  "limitations": [
                    "string"
                  ]
                }
              }
            ],
            "blocked": [
              {
                "id": "string",
                "url": "string",
                "category": "string",
                "usageRestriction": {
                  "permissions": [
                    "string"
                  ],
                  "conditions": [
                    "string"
                  ],
                  "limitations": [
                    "string"
                  ]
                }
              }
            ]
          },
          "packageName": "string",
          "packageVersion": "string",
          "target": "string",
          "hashes": [
            {
              "sha256": "string",
              "fileName": "string"
            }
          ],
          "vulnerabilities": [
            {
              "id": "string",
              "severity": 0,
              "source": "string",
              "fixedVersions": "string",
              "cwes": [
                "string"
              ]
            }
          ],
          "vulnerabilityCounts": {
            "critical": 0,
            "high": 0,
            "medium": 0,
            "low": 0,
            "unknown": 0
          },
```

```
{
              "ecosystem": "string",
              "group": "string",
              "packageName": "string",
              "packageVersion": "string",
              "vulnerabilities": [
                {
                  "id": "string",
                  "severity": 0,
                  "source": "string",
                  "fixedVersions": "string",
                  "cwes": [
                    "string"
                  ]
                }
              ],
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              }
            }
          ]
        }
      ],
      "dependenciesPackageCount": 0,
      "dependenciesVulnerabilityTotal": 0,
      "dependenciesVulnerabilityCounts": {
        "critical": 0,
        "high": 0,
        "medium": 0,
        "low": 0,
        "unknown": 0
      }
    },
    "resultTemplateHash": "string"
  },
  "vulnerabilityInfo": {
    "vulnerabilityInfoResult": {
      "code": 0,
      "hash": "string",
      "method": 0,
      "timeStamp": "string",
      "timing": 0,
      "detectedProduct": {
        "hasVulnerability": true,
        "isCurrent": true,
        "product": {
          "id": 0,
          "name": "string"
        },
```

```
"severity": "string",
        "sigName": "string",
        "signature": 0,
        "vendor": {
          "id": 0,
          "name": "string"
        },
        "version": "string",
        "versionData": {
          "countBehind": 0,
          "feedId": 0,
          "version": "string"
        },
        "vulnerabilities": [
          {
            "description": "string",
            "details": {
              "cpe": "string",
              "cve": "string",
              "cvss": {
                "accessComplexity": "string",
                "accessVector": "string",
                "authentication": "string",
                "availabilityImpact": "string",
                "confidentialityImpact": "string",
                "generatedOnEpoch": "string",
                "integrityImpact": "string",
                "score": "string",
                "source": "string"
              },
              "cvss3": {
                "attackComplexity": "string",
                "attackVector": "string",
                "availabilityImpact": "string",
                "baseScore": "string",
                "baseSeverity": "string",
                "confidentialityImpact": "string",
                "exploitabilityScore": "string",
                "impactScore": "string",
                "integrityImpact": "string",
                "privilegesRequired": "string",
                "scope": "string",
                "userInteraction": "string",
                "vectorString": "string"
              },
              "cwe": "string",
              "lastModifiedEpoch": "string",
              "publishedEpoch": "string",
              "references": [
                "string"
              ]
            },
            "severity": "string",
```

```
"staticId": 0
          }
        ]
      }
    }
  },
  "sessionCookie": "string",
  "fileTypeInfo": {
    "fileInfoExtended": {
      "extensions": "string",
      "groupId": "string",
      "typeId": "string"
    }
  },
  "scanInstanceUrl": "string",
  "scanWorkflowSnapshotId": "string",
  "coreEngines": [
    {
      "engineId": "string",
      "engineName": "string",
      "engineVersion": "string",
      "dbVersion": "string",
      "definitionTime": "string"
    }
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
extractedFiles	ExtractedFiles¦null	false	none	none
scanResults	ScanResults¦null	false	none	none
fileInfo	FileInfo¦null	false	none	none
fileScanInfo	FileScanInfo¦null	false	none	none
processInfo	ProcessInfo¦null	false	none	none
dataId	string	false	none	none
dlpInfo	DlpInfo¦null	false	none	none
sbomInfo	SbomInfo¦null	false	none	none
vulnerabilityInfo	VulnerabilityInfo¦null	false	none	none
sessionCookie	string¦null	false	none	none
fileTypeInfo	FileTypeInfo¦null	false	none	none
scanInstanceUrl	string¦null	false	none	none
scanWorkflowSnapshotId	string¦null	false	none	none
coreEngines	[CoreEngine]¦null	false	none	none

ExtractedFiles

```
{
  "dataId": "string",
  "scanResultI": 0,
  "detectedBy": 0,
  "filesExtractedCount": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
dataId	string	false	none	none
scanResultI	integer(int32)¦null	false	none	none
detectedBy	integer(int32)¦null	false	none	none
filesExtractedCount	integer(int32)¦null	false	none	none

ScanResults

```
{
  "scanDetails": [
    {
      "engineName": "string",
      "threatFound": "string",
      "scanResultI": 0,
      "defTime": "string",
      "scanTime": 0.1,
      "engId": "string",
      "waitTime": 0.1
    }
  ],
  "dataId": "string",
  "scanAllResultI": 0,
  "scanAllResultA": "string",
  "startTime": "string",
  "totalTime": 0.1,
  "totalAvs": 0,
  "totalDetections": 0,
  "progressPercentage": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
scanDetails	[EngineResult]¦null	false	none	none
dataId	string	false	none	none
scanAllResultI	integer(int32)	false	none	none
scanAllResultA	string	false	none	none
startTime	string	false	none	none
totalTime	number(double)	false	none	none
totalAvs	integer(int32)	false	none	none
totalDetections	integer(int32)¦null	false	none	none
progressPercentage	integer(int32)	false	none	none

EngineResult

```
{
  "engineName": "string",
  "threatFound": "string",
  "scanResultI": 0,
  "defTime": "string",
  "scanTime": 0.1,
  "engId": "string",
  "waitTime": 0.1
}
```

Properties

Name	Type	Required	Restrictions	Description
engineName	string	false	none	none
threatFound	string	false	none	none
scanResultI	integer(int32)	false	none	none
defTime	string	false	none	none
scanTime	number(double)	false	none	none
engId	string	false	none	none
waitTime	number(double)	false	none	none

FileInfo

```
{
  "fileSize": 0,
  "uploadTimestamp": "string",
  "md5": "string",
  "sha1": "string",
  "sha256": "string",
  "fileTypeCategory": "string",
  "fileTypeDescription": "string",
  "fileTypeExtension": "string",
  "displayName": "string",
  "fileTypeId": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
fileSize	integer(int64)	false	none	none
uploadTimestamp	string	false	none	none
md5	string	false	none	none
sha1	string	false	none	none
sha256	string	false	none	none
fileTypeCategory	string	false	none	none
fileTypeDescription	string	false	none	none
fileTypeExtension	string	false	none	none
displayName	string	false	none	none
fileTypeId	string¦null	false	none	none

FileScanInfo

```
{
  "finalVerdict": {
    "blocked": true,
    "verdict": "string",
    "confidence": 0.1,
    "threatLevel": 0.1
  },
  "overviewReport": {
    "analysisState": "string",
    "signalGroups": [
      {
        "description": "string",
        "verdict": "string",
        "signals": [
          {
            "originType": "string",
            "signalReadable": "string",
            "strength": 0.1
          }
        ]
      }
    ]
  },
  "errorCategory": "string",
  "errorCode": 0,
  "errorDetails": "string"
}
```


Properties

Name	Type	Required	Restrictions	Description
finalVerdict	FileScanInfoFinalVerdict¦null	false	none	none
overviewReport	FileScanInfoOverviewReport¦null	false	none	none
errorCategory	string¦null	false	none	none
errorCode	integer(int32)¦null	false	none	none
errorDetails	string¦null	false	none	none

FileScanInfoFinalVerdict

```
{
  "blocked": true,
  "verdict": "string",
  "confidence": 0.1,
  "threatLevel": 0.1
}
```


Properties

Name	Type	Required	Restrictions	Description
blocked	boolean|null	false	none	none
verdict	string	false	none	none
confidence	number(float)	false	none	none
threatLevel	number(float)	false	none	none

FileScanInfoOverviewReport


```
{
  "analysisState": "string",
  "signalGroups": [
    {
      "description": "string",
      "verdict": "string",
      "signals": [
        {
          "originType": "string",
          "signalReadable": "string",
          "strength": 0.1
        }
      ]
    }
  ]
}
```


Properties

Name	Type	Required	Restrictions	Description
analysisState	string	false	none	none
signalGroups	[FileScanInfoSignalGroup]	false	none	none

FileScanInfoSignalGroup

```
{
  "description": "string",
  "verdict": "string",
  "signals": [
    {
      "originType": "string",
      "signalReadable": "string",
      "strength": 0.1
    }
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
description	string	false	none	none
verdict	string	false	none	none
signals	[FileScanInfoSignal]	false	none	none

FileScanInfoSignal

```
{
  "originType": "string",
  "signalReadable": "string",
  "strength": 0.1
}
```

Properties

Name	Type	Required	Restrictions	Description
originType	string	false	none	none
signalReadable	string	false	none	none
strength	number(float)	false	none	none

ProcessInfo


```
{
  "postProcessing": {
    "actionsRan": "string",
    "actionsFailed": "string",
    "convertedTo": "string",
    "copyMoveDestination": "string",
    "convertedDestination": "string",
    "sanitizationDetails": {
      "failureCategory": "string",
      "details": "string"
    }
  },
  "progressPercentage": 0,
  "userAgent": "string",
  "profile": "string",
  "result": "string",
  "blockedReason": "string",
  "fileTypeSkippedScan": true,
  "processingTime": 0.1,
  "queueTime": 0.1,
  "resultI": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
postProcessing	PostProcessing¦null	false	none	none
progressPercentage	integer(int32)	false	none	none
userAgent	string	false	none	none
profile	string	false	none	none
result	string¦null	false	none	none
blockedReason	string	false	none	none
fileTypeSkippedScan	boolean	false	none	none
processingTime	number(double)	false	none	none
queueTime	number(double)	false	none	none
resultI	ResultI2	false	none	- 0: Allowed - 1: Blocked - 2: Processing

PostProcessing

```
{
  "actionsRan": "string",
  "actionsFailed": "string",
  "convertedTo": "string",
  "copyMoveDestination": "string",
  "convertedDestination": "string",
  "sanitizationDetails": {
    "failureCategory": "string",
    "details": "string"
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
actionsRan	string	false	none	none
actionsFailed	string	false	none	none
convertedTo	string¦null	false	none	none
copyMoveDestination	string¦null	false	none	none
convertedDestination	string¦null	false	none	none
sanitizationDetails	SanitizationDetails¦null	false	none	none

SanitizationDetails

```
{
  "failureCategory": "string",
  "details": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
failureCategory	string¦null	false	none	none
details	string¦null	false	none	none

ResultI2

```
0
```

- 0: Allowed
- 1: Blocked
- 2: Processing

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Allowed - 1: Blocked - 2: Processing

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

DlpInfo

```
{
  "hits": null,
  "verdict": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
hits	any	false	none	none
verdict	integer(int64)¦null	false	none	none

SbomInfo

```
{
  "finalVerdict": {
    "blocked": true,
    "licensingPackageCounts": {
      "allowed": 0,
      "blocked": 0,
      "unknown": 0,
      "unsupported": 0
    },
    "severity": 0,
    "totalPackageCount": 0,
    "verdict": "string",
    "vulnerabilityCounts": {
      "critical": 0,
      "high": 0,
      "medium": 0,
      "low": 0,
      "unknown": 0
    }
  },
  "overviewReport": {
    "isPartialReport": true,
    "osInfo": {
      "osName": "string",
      "osVersion": "string"
    },
    "scanMode": "string",
    "steps": [
      {
        "command": "string",
        "packages": [
          {
            "authors": "string",
            "releaseDate": "2019-08-24T14:15:22Z",
            "ecosystem": "string",
            "group": "string",
            "archive": true,
            "executable": true,
            "uid": "string",
            "licenses": {
              "allowed": [
                {
                  "id": "string",
                  "url": "string",
                  "category": "string",
                  "usageRestriction": {
                    "permissions": [
                      "string"
                    ],
                    "conditions": [
                      "string"
                    ],
```

```
"string"
                    ]
                  }
                }
              ],
              "blocked": [
                {
                  "id": "string",
                  "url": "string",
                  "category": "string",
                  "usageRestriction": {
                    "permissions": [
                      "string"
                    ],
                    "conditions": [
                      "string"
                    ],
                    "limitations": [
                      "string"
                    ]
                  }
                }
              ]
            },
            "packageName": "string",
            "packageVersion": "string",
            "hashes": [
              {
                "sha256": "string",
                "fileName": "string"
              }
            ],
            "vulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            },
            "vulnerabilities": [
              {
                "id": "string",
                "severity": 0,
                "source": "string",
                "fixedVersions": "string",
                "cwes": [
                  "string"
                ]
              }
            ],
            "arch": "string",
            "osName": "string",
            "osVersion": "string",
```

```
{
                "ecosystem": "string",
                "group": "string",
                "packageName": "string",
                "packageVersion": "string",
                "vulnerabilities": [
                  {
                    "id": "string",
                    "severity": 0,
                    "source": "string",
                    "fixedVersions": "string",
                    "cwes": [
                      "string"
                    ]
                  }
                ],
                "vulnerabilityCounts": {
                  "critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
                }
              }
            ]
          }
        ],
        "licenseStatus": "string",
        "licensingPackageCounts": {
          "allowed": 0,
          "blocked": 0,
          "unknown": 0,
          "unsupported": 0
        },
        "step": 0,
        "vulnerabilityCount": 0,
        "sha256": "string",
        "severity": 0,
        "vulnerabilityCounts": {
          "critical": 0,
          "high": 0,
          "medium": 0,
          "low": 0,
          "unknown": 0
        }
      }
    ],
    "packages": [
      {
        "authors": "string",
        "releaseDate": "2019-08-24T14:15:22Z",
        "ecosystem": "string",
        "group": "string",
```

```
"executable": true,
        "uid": "string",
        "licenses": {
          "allowed": [
            {
              "id": "string",
              "url": "string",
              "category": "string",
              "usageRestriction": {
                "permissions": [
                  "string"
                ],
                "conditions": [
                  "string"
                ],
                "limitations": [
                  "string"
                ]
              }
            }
          ],
          "blocked": [
            {
              "id": "string",
              "url": "string",
              "category": "string",
              "usageRestriction": {
                "permissions": [
                  "string"
                ],
                "conditions": [
                  "string"
                ],
                "limitations": [
                  "string"
                ]
              }
            }
          ]
        },
        "packageName": "string",
        "packageVersion": "string",
        "target": "string",
        "hashes": [
          {
            "sha256": "string",
            "fileName": "string"
          }
        ],
        "vulnerabilities": [
          {
            "id": "string",
            "severity": 0,
```

```
"fixedVersions": "string",
            "cwes": [
              "string"
            ]
          }
        ],
        "vulnerabilityCounts": {
          "critical": 0,
          "high": 0,
          "medium": 0,
          "low": 0,
          "unknown": 0
        },
        "dependencies": [
          {
            "ecosystem": "string",
            "group": "string",
            "packageName": "string",
            "packageVersion": "string",
            "vulnerabilities": [
              {
                "id": "string",
                "severity": 0,
                "source": "string",
                "fixedVersions": "string",
                "cwes": [
                  "string"
                ]
              }
            ],
            "vulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            }
          }
        ]
      }
    ],
    "dependenciesPackageCount": 0,
    "dependenciesVulnerabilityTotal": 0,
    "dependenciesVulnerabilityCounts": {
      "critical": 0,
      "high": 0,
      "medium": 0,
      "low": 0,
      "unknown": 0
    }
  },
  "resultTemplateHash": "string"
```

```
}
```

Properties

Name	Type	Required	Restrictions	Description
finalVerdict	FinalVerdict¦null	false	none	none
overviewReport	OverviewReport¦null	false	none	none
resultTemplateHash	string	false	none	none

FinalVerdict

```
{
  "blocked": true,
  "licensingPackageCounts": {
    "allowed": 0,
    "blocked": 0,
    "unknown": 0,
    "unsupported": 0
  },
  "severity": 0,
  "totalPackageCount": 0,
  "verdict": "string",
  "vulnerabilityCounts": {
    "critical": 0,
    "high": 0,
    "medium": 0,
    "low": 0,
    "unknown": 0
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
blocked	boolean	false	none	none
licensingPackageCounts	LicensingIssues¦null	false	none	none
severity	Severity	false	none	- 0: None - 1: Critical - 2: High - 3: Medium - 4: Low - 5: Unknown
totalPackageCount	integer(int32)	false	none	none
verdict	string	false	none	none
vulnerabilityCounts	VulnerabilityIssues¦null	false	none	none

LicensingIssues

```
{
  "allowed": 0,
  "blocked": 0,
  "unknown": 0,
  "unsupported": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
allowed	integer(int32)	false	none	none
blocked	integer(int32)	false	none	none
unknown	integer(int32)	false	none	none
unsupported	integer(int32)	false	none	none

Severity

```
0
```

- 0: None
- 1: Critical
- 2: High
- 3: Medium
- 4: Low
- 5: Unknown

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: None - 1: Critical - 2: High - 3: Medium - 4: Low - 5: Unknown

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4
anonymous	5

VulnerabilityIssues

```
{
  "critical": 0,
  "high": 0,
  "medium": 0,
  "low": 0,
  "unknown": 0
}
```


Properties

Name	Type	Required	Restrictions	Description
critical	integer(int32)	false	none	none
high	integer(int32)	false	none	none
medium	integer(int32)	false	none	none
low	integer(int32)	false	none	none
unknown	integer(int32)¦null	false	none	none

OverviewReport

```
{
  "isPartialReport": true,
  "osInfo": {
    "osName": "string",
    "osVersion": "string"
  },
  "scanMode": "string",
  "steps": [
    {
      "command": "string",
      "packages": [
        {
          "authors": "string",
          "releaseDate": "2019-08-24T14:15:22Z",
          "ecosystem": "string",
          "group": "string",
          "archive": true,
          "executable": true,
          "uid": "string",
          "licenses": {
            "allowed": [
              {
                "id": "string",
                "url": "string",
                "category": "string",
                "usageRestriction": {
                  "permissions": [
                    "string"
                  ],
                  "conditions": [
                    "string"
                  ],
                  "limitations": [
                    "string"
                  ]
                }
              }
            ],
            "blocked": [
              {
                "id": "string",
                "url": "string",
                "category": "string",
                "usageRestriction": {
                  "permissions": [
                    "string"
                  ],
                  "conditions": [
                    "string"
                  ],
                  "limitations": [
                    "string"
```

```
}
            }
          ]
        },
        "packageName": "string",
        "packageVersion": "string",
        "hashes": [
          {
            "sha256": "string",
            "fileName": "string"
          }
        ],
        "vulnerabilityCounts": {
          "critical": 0,
          "high": 0,
          "medium": 0,
          "low": 0,
          "unknown": 0
        },
        "vulnerabilities": [
          {
            "id": "string",
            "severity": 0,
            "source": "string",
            "fixedVersions": "string",
            "cwes": [
              "string"
            ]
          }
        ],
        "arch": "string",
        "osName": "string",
        "osVersion": "string",
        "dependencies": [
          {
            "ecosystem": "string",
            "group": "string",
            "packageName": "string",
            "packageVersion": "string",
            "vulnerabilities": [
              {
                "id": "string",
                "severity": 0,
                "source": "string",
                "fixedVersions": "string",
                "cwes": [
                  "string"
                ]
              }
            ],
            "vulnerabilityCounts": {
              "critical": 0,
              "high": 0,
```

```
"low": 0,
                "unknown": 0
              }
            }
          ]
        }
      ],
      "licenseStatus": "string",
      "licensingPackageCounts": {
        "allowed": 0,
        "blocked": 0,
        "unknown": 0,
        "unsupported": 0
      },
      "step": 0,
      "vulnerabilityCount": 0,
      "sha256": "string",
      "severity": 0,
      "vulnerabilityCounts": {
        "critical": 0,
        "high": 0,
        "medium": 0,
        "low": 0,
        "unknown": 0
      }
    }
  ],
  "packages": [
    {
      "authors": "string",
      "releaseDate": "2019-08-24T14:15:22Z",
      "ecosystem": "string",
      "group": "string",
      "archive": true,
      "executable": true,
      "uid": "string",
      "licenses": {
        "allowed": [
          {
            "id": "string",
            "url": "string",
            "category": "string",
            "usageRestriction": {
              "permissions": [
                "string"
              ],
              "conditions": [
                "string"
              ],
              "limitations": [
                "string"
              ]
            }
```

```
],
        "blocked": [
          {
            "id": "string",
            "url": "string",
            "category": "string",
            "usageRestriction": {
              "permissions": [
                "string"
              ],
              "conditions": [
                "string"
              ],
              "limitations": [
                "string"
              ]
            }
          }
        ]
      },
      "packageName": "string",
      "packageVersion": "string",
      "target": "string",
      "hashes": [
        {
          "sha256": "string",
          "fileName": "string"
        }
      ],
      "vulnerabilities": [
        {
          "id": "string",
          "severity": 0,
          "source": "string",
          "fixedVersions": "string",
          "cwes": [
            "string"
          ]
        }
      ],
      "vulnerabilityCounts": {
        "critical": 0,
        "high": 0,
        "medium": 0,
        "low": 0,
        "unknown": 0
      },
      "dependencies": [
        {
          "ecosystem": "string",
          "group": "string",
          "packageName": "string",
          "packageVersion": "string",
```

```
{
              "id": "string",
              "severity": 0,
              "source": "string",
              "fixedVersions": "string",
              "cwes": [
                "string"
              ]
            }
          ],
          "vulnerabilityCounts": {
            "critical": 0,
            "high": 0,
            "medium": 0,
            "low": 0,
            "unknown": 0
          }
        }
      ]
    }
  ],
  "dependenciesPackageCount": 0,
  "dependenciesVulnerabilityTotal": 0,
  "dependenciesVulnerabilityCounts": {
    "critical": 0,
    "high": 0,
    "medium": 0,
    "low": 0,
    "unknown": 0
  }
}
```


Properties

Name	Type	Required	Restrictions	Description
isPartialReport	boolean	false	none	none
osInfo	OsInfo¦null	false	none	none
scanMode	string	false	none	none
steps	[oneOf]¦null	false	none	none
packages	[oneOf]¦null	false	none	none
dependenciesPackageCount	integer(int32)¦null	false	none	none
dependenciesVulnerabilityTotal	integer(int32)¦null	false	none	none
dependenciesVulnerabilityCounts	VulnerabilityCounts¦null	false	none	none

OsInfo


```
{
  "osName": "string",
  "osVersion": "string"
}
```


Properties

Name	Type	Required	Restrictions	Description
osName	string	false	none	none
osVersion	string	false	none	none

SbomStep

```
{
  "command": "string",
  "packages": [
    {
      "authors": "string",
      "releaseDate": "2019-08-24T14:15:22Z",
      "ecosystem": "string",
      "group": "string",
      "archive": true,
      "executable": true,
      "uid": "string",
      "licenses": {
        "allowed": [
          {
            "id": "string",
            "url": "string",
            "category": "string",
            "usageRestriction": {
              "permissions": [
                "string"
              ],
              "conditions": [
                "string"
              ],
              "limitations": [
                "string"
              ]
            }
          }
        ],
        "blocked": [
          {
            "id": "string",
            "url": "string",
            "category": "string",
            "usageRestriction": {
              "permissions": [
                "string"
              ],
              "conditions": [
                "string"
              ],
              "limitations": [
                "string"
              ]
            }
          }
        ]
      },
      "packageName": "string",
      "packageVersion": "string",
      "hashes": [
```

```
"sha256": "string",
          "fileName": "string"
        }
      ],
      "vulnerabilityCounts": {
        "critical": 0,
        "high": 0,
        "medium": 0,
        "low": 0,
        "unknown": 0
      },
      "vulnerabilities": [
        {
          "id": "string",
          "severity": 0,
          "source": "string",
          "fixedVersions": "string",
          "cwes": [
            "string"
          ]
        }
      ],
      "arch": "string",
      "osName": "string",
      "osVersion": "string",
      "dependencies": [
        {
          "ecosystem": "string",
          "group": "string",
          "packageName": "string",
          "packageVersion": "string",
          "vulnerabilities": [
            {
              "id": "string",
              "severity": 0,
              "source": "string",
              "fixedVersions": "string",
              "cwes": [
                "string"
              ]
            }
          ],
          "vulnerabilityCounts": {
            "critical": 0,
            "high": 0,
            "medium": 0,
            "low": 0,
            "unknown": 0
          }
        }
      ]
    }
  ],
```

```
"licensingPackageCounts": {
    "allowed": 0,
    "blocked": 0,
    "unknown": 0,
    "unsupported": 0
  },
  "step": 0,
  "vulnerabilityCount": 0,
  "sha256": "string",
  "severity": 0,
  "vulnerabilityCounts": {
    "critical": 0,
    "high": 0,
    "medium": 0,
    "low": 0,
    "unknown": 0
  }
}
```


Properties

Name	Type	Required	Restrictions	Description
command	string	false	none	none
packages	[oneOf]¦null	false	none	none
licenseStatus	string	false	none	none
licensingPackageCounts	LicensingIssues¦null	false	none	none
step	integer(int32)	false	none	none
vulnerabilityCount	integer(int32)	false	none	none
sha256	string	false	none	none
severity	Severity	false	none	- 0: None - 1: Critical - 2: High - 3: Medium - 4: Low - 5: Unknown
vulnerabilityCounts	VulnerabilityCounts¦null	false	none	none

ContainerPackage

```
{
  "authors": "string",
  "releaseDate": "2019-08-24T14:15:22Z",
  "ecosystem": "string",
  "group": "string",
  "archive": true,
  "executable": true,
  "uid": "string",
  "licenses": {
    "allowed": [
      {
        "id": "string",
        "url": "string",
        "category": "string",
        "usageRestriction": {
          "permissions": [
            "string"
          ],
          "conditions": [
            "string"
          ],
          "limitations": [
            "string"
          ]
        }
      }
    ],
    "blocked": [
      {
        "id": "string",
        "url": "string",
        "category": "string",
        "usageRestriction": {
          "permissions": [
            "string"
          ],
          "conditions": [
            "string"
          ],
          "limitations": [
            "string"
          ]
        }
      }
    ]
  },
  "packageName": "string",
  "packageVersion": "string",
  "hashes": [
    {
      "sha256": "string",
      "fileName": "string"
```

```
],
  "vulnerabilityCounts": {
    "critical": 0,
    "high": 0,
    "medium": 0,
    "low": 0,
    "unknown": 0
  },
  "vulnerabilities": [
    {
      "id": "string",
      "severity": 0,
      "source": "string",
      "fixedVersions": "string",
      "cwes": [
        "string"
      ]
    }
  ],
  "arch": "string",
  "osName": "string",
  "osVersion": "string",
  "dependencies": [
    {
      "ecosystem": "string",
      "group": "string",
      "packageName": "string",
      "packageVersion": "string",
      "vulnerabilities": [
        {
          "id": "string",
          "severity": 0,
          "source": "string",
          "fixedVersions": "string",
          "cwes": [
            "string"
          ]
        }
      ],
      "vulnerabilityCounts": {
        "critical": 0,
        "high": 0,
        "medium": 0,
        "low": 0,
        "unknown": 0
      }
    }
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
authors	string¦null	false	none	none
releaseDate	string(date-time)¦null	false	none	none
ecosystem	string	false	none	none
group	string	false	none	none
archive	boolean¦null	false	none	none
executable	boolean¦null	false	none	none
uid	string	false	none	none
licenses	Licenses¦null	false	none	none
packageName	string	false	none	none
packageVersion	string	false	none	none
hashes	[Hash]¦null	false	none	none
vulnerabilityCounts	VulnerabilityCounts¦null	false	none	none
vulnerabilities	[oneOf]¦null	false	none	none
arch	string	false	none	none
osName	string	false	none	none
osVersion	string	false	none	none
dependencies	[oneOf]¦null	false	none	none

Licenses

```
{
  "allowed": [
    {
      "id": "string",
      "url": "string",
      "category": "string",
      "usageRestriction": {
        "permissions": [
          "string"
        ],
        "conditions": [
          "string"
        ],
        "limitations": [
          "string"
        ]
      }
    }
  ],
  "blocked": [
    {
      "id": "string",
      "url": "string",
      "category": "string",
      "usageRestriction": {
        "permissions": [
          "string"
        ],
        "conditions": [
          "string"
        ],
        "limitations": [
          "string"
        ]
      }
    }
  ]
}
```


Properties

Name	Type	Required	Restrictions	Description
allowed	[LicenseInfo]¦null	false	none	none
blocked	[LicenseInfo]¦null	false	none	none

LicenseInfo

```
{
  "id": "string",
  "url": "string",
  "category": "string",
  "usageRestriction": {
    "permissions": [
      "string"
    ],
    "conditions": [
      "string"
    ],
    "limitations": [
      "string"
    ]
  }
}
```


Properties

Name	Type	Required	Restrictions	Description
id	string¦null	false	none	none
url	string¦null	false	none	none
category	string¦null	false	none	none
usageRestriction	UsageRestriction¦null	false	none	none

UsageRestriction


```
{
  "permissions": [
    "string"
  ],
  "conditions": [
    "string"
  ],
  "limitations": [
    "string"
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
permissions	[string]¦null	false	none	none
conditions	[string]¦null	false	none	none
limitations	[string]¦null	false	none	none

Hash

```
{
  "sha256": "string",
  "fileName": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
sha256	string	false	none	none
fileName	string¦null	false	none	none

VulnerabilityCounts

```
{
  "critical": 0,
  "high": 0,
  "medium": 0,
  "low": 0,
  "unknown": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
critical	integer(int32)	false	none	none
high	integer(int32)	false	none	none
medium	integer(int32)	false	none	none
low	integer(int32)	false	none	none
unknown	integer(int32)	false	none	none

SbomVulnerability

```
{
  "id": "string",
  "severity": 0,
  "source": "string",
  "fixedVersions": "string",
  "cwes": [
    "string"
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string¦null	false	none	none
severity	Severity	false	none	- 0: None - 1: Critical - 2: High - 3: Medium - 4: Low - 5: Unknown
source	string¦null	false	none	none
fixedVersions	string¦null	false	none	none
cwes	[string]¦null	false	none	none

Dependency


```
{
  "ecosystem": "string",
  "group": "string",
  "packageName": "string",
  "packageVersion": "string",
  "vulnerabilities": [
    {
      "id": "string",
      "severity": 0,
      "source": "string",
      "fixedVersions": "string",
      "cwes": [
        "string"
      ]
    }
  ],
  "vulnerabilityCounts": {
    "critical": 0,
    "high": 0,
    "medium": 0,
    "low": 0,
    "unknown": 0
  }
}
```


Properties

Name	Type	Required	Restrictions	Description
ecosystem	string	false	none	none
group	string	false	none	none
packageName	string	false	none	none
packageVersion	string	false	none	none
vulnerabilities	[oneOf]¦null	false	none	none
vulnerabilityCounts	VulnerabilityCounts¦null	false	none	none

SourceCodePackage

```
{
  "authors": "string",
  "releaseDate": "2019-08-24T14:15:22Z",
  "ecosystem": "string",
  "group": "string",
  "archive": true,
  "executable": true,
  "uid": "string",
  "licenses": {
    "allowed": [
      {
        "id": "string",
        "url": "string",
        "category": "string",
        "usageRestriction": {
          "permissions": [
            "string"
          ],
          "conditions": [
            "string"
          ],
          "limitations": [
            "string"
          ]
        }
      }
    ],
    "blocked": [
      {
        "id": "string",
        "url": "string",
        "category": "string",
        "usageRestriction": {
          "permissions": [
            "string"
          ],
          "conditions": [
            "string"
          ],
          "limitations": [
            "string"
          ]
        }
      }
    ]
  },
  "packageName": "string",
  "packageVersion": "string",
  "target": "string",
  "hashes": [
    {
      "sha256": "string",
```

```
}
  ],
  "vulnerabilities": [
    {
      "id": "string",
      "severity": 0,
      "source": "string",
      "fixedVersions": "string",
      "cwes": [
        "string"
      ]
    }
  ],
  "vulnerabilityCounts": {
    "critical": 0,
    "high": 0,
    "medium": 0,
    "low": 0,
    "unknown": 0
  },
  "dependencies": [
    {
      "ecosystem": "string",
      "group": "string",
      "packageName": "string",
      "packageVersion": "string",
      "vulnerabilities": [
        {
          "id": "string",
          "severity": 0,
          "source": "string",
          "fixedVersions": "string",
          "cwes": [
            "string"
          ]
        }
      ],
      "vulnerabilityCounts": {
        "critical": 0,
        "high": 0,
        "medium": 0,
        "low": 0,
        "unknown": 0
      }
    }
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
authors	string¦null	false	none	none
releaseDate	string(date-time)¦null	false	none	none
ecosystem	string	false	none	none
group	string	false	none	none
archive	boolean¦null	false	none	none
executable	boolean¦null	false	none	none
uid	string	false	none	none
licenses	Licenses¦null	false	none	none
packageName	string	false	none	none
packageVersion	string	false	none	none
target	string	false	none	none
hashes	[Hash]¦null	false	none	none
vulnerabilities	[oneOf]¦null	false	none	none
vulnerabilityCounts	VulnerabilityCounts¦null	false	none	none
dependencies	[oneOf]¦null	false	none	none

VulnerabilityInfo

```
{
  "vulnerabilityInfoResult": {
    "code": 0,
    "hash": "string",
    "method": 0,
    "timeStamp": "string",
    "timing": 0,
    "detectedProduct": {
      "hasVulnerability": true,
      "isCurrent": true,
      "product": {
        "id": 0,
        "name": "string"
      },
      "remediationLink": "string",
      "severity": "string",
      "sigName": "string",
      "signature": 0,
      "vendor": {
        "id": 0,
        "name": "string"
      },
      "version": "string",
      "versionData": {
        "countBehind": 0,
        "feedId": 0,
        "version": "string"
      },
      "vulnerabilities": [
        {
          "description": "string",
          "details": {
            "cpe": "string",
            "cve": "string",
            "cvss": {
              "accessComplexity": "string",
              "accessVector": "string",
              "authentication": "string",
              "availabilityImpact": "string",
              "confidentialityImpact": "string",
              "generatedOnEpoch": "string",
              "integrityImpact": "string",
              "score": "string",
              "source": "string"
            },
            "cvss3": {
              "attackComplexity": "string",
              "attackVector": "string",
              "availabilityImpact": "string",
              "baseScore": "string",
              "baseSeverity": "string",
              "confidentialityImpact": "string",
```

```
"impactScore": "string",
              "integrityImpact": "string",
              "privilegesRequired": "string",
              "scope": "string",
              "userInteraction": "string",
              "vectorString": "string"
            },
            "cwe": "string",
            "lastModifiedEpoch": "string",
            "publishedEpoch": "string",
            "references": [
              "string"
            ]
          },
          "severity": "string",
          "severityIndex": 0,
          "staticId": 0
        }
      ]
    }
  }
}
```


Properties

Name	Type	Required	Restrictions	Description
vulnerabilityInfoResult	VulnerabilityInfoResult¦null	false	none	none

VulnerabilityInfoResult

```
{
  "code": 0,
  "hash": "string",
  "method": 0,
  "timeStamp": "string",
  "timing": 0,
  "detectedProduct": {
    "hasVulnerability": true,
    "isCurrent": true,
    "product": {
      "id": 0,
      "name": "string"
    },
    "remediationLink": "string",
    "severity": "string",
    "sigName": "string",
    "signature": 0,
    "vendor": {
      "id": 0,
      "name": "string"
    },
    "version": "string",
    "versionData": {
      "countBehind": 0,
      "feedId": 0,
      "version": "string"
    },
    "vulnerabilities": [
      {
        "description": "string",
        "details": {
          "cpe": "string",
          "cve": "string",
          "cvss": {
            "accessComplexity": "string",
            "accessVector": "string",
            "authentication": "string",
            "availabilityImpact": "string",
            "confidentialityImpact": "string",
            "generatedOnEpoch": "string",
            "integrityImpact": "string",
            "score": "string",
            "source": "string"
          },
          "cvss3": {
            "attackComplexity": "string",
            "attackVector": "string",
            "availabilityImpact": "string",
            "baseScore": "string",
            "baseSeverity": "string",
            "confidentialityImpact": "string",
            "exploitabilityScore": "string",
```

```
"integrityImpact": "string",
            "privilegesRequired": "string",
            "scope": "string",
            "userInteraction": "string",
            "vectorString": "string"
          },
          "cwe": "string",
          "lastModifiedEpoch": "string",
          "publishedEpoch": "string",
          "references": [
            "string"
          ]
        },
        "severity": "string",
        "severityIndex": 0,
        "staticId": 0
      }
    ]
  }
}
```


Properties

Name	Type	Required	Restrictions	Description
code	integer(int64)¦null	false	none	none
hash	string	false	none	none
method	integer(int64)¦null	false	none	none
timeStamp	string	false	none	none
timing	integer(int64)¦null	false	none	none
detectedProduct	DetectedProduct¦null	false	none	none

DetectedProduct

```
{
  "hasVulnerability": true,
  "isCurrent": true,
  "product": {
    "id": 0,
    "name": "string"
  },
  "remediationLink": "string",
  "severity": "string",
  "sigName": "string",
  "signature": 0,
  "vendor": {
    "id": 0,
    "name": "string"
  },
  "version": "string",
  "versionData": {
    "countBehind": 0,
    "feedId": 0,
    "version": "string"
  },
  "vulnerabilities": [
    {
      "description": "string",
      "details": {
        "cpe": "string",
        "cve": "string",
        "cvss": {
          "accessComplexity": "string",
          "accessVector": "string",
          "authentication": "string",
          "availabilityImpact": "string",
          "confidentialityImpact": "string",
          "generatedOnEpoch": "string",
          "integrityImpact": "string",
          "score": "string",
          "source": "string"
        },
        "cvss3": {
          "attackComplexity": "string",
          "attackVector": "string",
          "availabilityImpact": "string",
          "baseScore": "string",
          "baseSeverity": "string",
          "confidentialityImpact": "string",
          "exploitabilityScore": "string",
          "impactScore": "string",
          "integrityImpact": "string",
          "privilegesRequired": "string",
          "scope": "string",
          "userInteraction": "string",
          "vectorString": "string"
```

```
"cwe": "string",
        "lastModifiedEpoch": "string",
        "publishedEpoch": "string",
        "references": [
          "string"
        ]
      },
      "severity": "string",
      "severityIndex": 0,
      "staticId": 0
    }
  ]
}
```


Properties

Name	Type	Required	Restrictions	Description
hasVulnerability	boolean¦null	false	none	none
isCurrent	boolean¦null	false	none	none
product	Product¦null	false	none	none
remediationLink	string	false	none	none
severity	string	false	none	none
sigName	string	false	none	none
signature	integer(int64)¦null	false	none	none
vendor	Vendor¦null	false	none	none
version	string	false	none	none
versionData	VersionData¦null	false	none	none
vulnerabilities	[Vulnerability]¦null	false	none	none

Product


```
{
  "id": 0,
  "name": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
id	integer(int64)¦null	false	none	none
name	string	false	none	none

Vendor

```
{
  "id": 0,
  "name": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
id	integer(int64)¦null	false	none	none
name	string	false	none	none

VersionData

```
{
  "countBehind": 0,
  "feedId": 0,
  "version": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
countBehind	integer(int64)¦null	false	none	none
feedId	integer(int64)¦null	false	none	none
version	string	false	none	none

Vulnerability


```
{
  "description": "string",
  "details": {
    "cpe": "string",
    "cve": "string",
    "cvss": {
      "accessComplexity": "string",
      "accessVector": "string",
      "authentication": "string",
      "availabilityImpact": "string",
      "confidentialityImpact": "string",
      "generatedOnEpoch": "string",
      "integrityImpact": "string",
      "score": "string",
      "source": "string"
    },
    "cvss3": {
      "attackComplexity": "string",
      "attackVector": "string",
      "availabilityImpact": "string",
      "baseScore": "string",
      "baseSeverity": "string",
      "confidentialityImpact": "string",
      "exploitabilityScore": "string",
      "impactScore": "string",
      "integrityImpact": "string",
      "privilegesRequired": "string",
      "scope": "string",
      "userInteraction": "string",
      "vectorString": "string"
    },
    "cwe": "string",
    "lastModifiedEpoch": "string",
    "publishedEpoch": "string",
    "references": [
      "string"
    ]
  },
  "severity": "string",
  "severityIndex": 0,
  "staticId": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
description	string	false	none	none
details	VulnerabilityDetails¦null	false	none	none
severity	string	false	none	none
severityIndex	integer(int32)¦null	false	none	none
staticId	integer(int64)¦null	false	none	none

VulnerabilityDetails

```
{
  "cpe": "string",
  "cve": "string",
  "cvss": {
    "accessComplexity": "string",
    "accessVector": "string",
    "authentication": "string",
    "availabilityImpact": "string",
    "confidentialityImpact": "string",
    "generatedOnEpoch": "string",
    "integrityImpact": "string",
    "score": "string",
    "source": "string"
  },
  "cvss3": {
    "attackComplexity": "string",
    "attackVector": "string",
    "availabilityImpact": "string",
    "baseScore": "string",
    "baseSeverity": "string",
    "confidentialityImpact": "string",
    "exploitabilityScore": "string",
    "impactScore": "string",
    "integrityImpact": "string",
    "privilegesRequired": "string",
    "scope": "string",
    "userInteraction": "string",
    "vectorString": "string"
  },
  "cwe": "string",
  "lastModifiedEpoch": "string",
  "publishedEpoch": "string",
  "references": [
    "string"
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
cpe	string	false	none	none
cve	string	false	none	none
cvss	Cvss¦null	false	none	none
cvss3	Cvss3¦null	false	none	none
cwe	string	false	none	none
lastModifiedEpoch	string	false	none	none
publishedEpoch	string	false	none	none
references	[string]	false	none	none

Cvss

```
{
  "accessComplexity": "string",
  "accessVector": "string",
  "authentication": "string",
  "availabilityImpact": "string",
  "confidentialityImpact": "string",
  "generatedOnEpoch": "string",
  "integrityImpact": "string",
  "score": "string",
  "source": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
accessComplexity	string	false	none	none
accessVector	string	false	none	none
authentication	string	false	none	none
availabilityImpact	string	false	none	none
confidentialityImpact	string	false	none	none
generatedOnEpoch	string	false	none	none
integrityImpact	string	false	none	none
score	string	false	none	none
source	string	false	none	none

Cvss3

```
{
  "attackComplexity": "string",
  "attackVector": "string",
  "availabilityImpact": "string",
  "baseScore": "string",
  "baseSeverity": "string",
  "confidentialityImpact": "string",
  "exploitabilityScore": "string",
  "impactScore": "string",
  "integrityImpact": "string",
  "privilegesRequired": "string",
  "scope": "string",
  "userInteraction": "string",
  "vectorString": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
attackComplexity	string	false	none	none
attackVector	string	false	none	none
availabilityImpact	string	false	none	none
baseScore	string	false	none	none
baseSeverity	string	false	none	none
confidentialityImpact	string	false	none	none
exploitabilityScore	string	false	none	none
impactScore	string	false	none	none
integrityImpact	string	false	none	none
privilegesRequired	string	false	none	none
scope	string	false	none	none
userInteraction	string	false	none	none
vectorString	string	false	none	none

FileTypeInfo

```
{
  "fileInfoExtended": {
    "extensions": "string",
    "groupId": "string",
    "typeId": "string"
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
fileInfoExtended	FileInfoExtended¦null	false	none	none

FileInfoExtended

```
{
  "extensions": "string",
  "groupId": "string",
  "typeId": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
extensions	string¦null	false	none	none
groupId	string¦null	false	none	none
typeId	string¦null	false	none	none

CoreEngine

```
{
  "engineId": "string",
  "engineName": "string",
  "engineVersion": "string",
  "dbVersion": "string",
  "definitionTime": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
engineId	string¦null	false	none	none
engineName	string	false	none	none
engineVersion	string	false	none	none
dbVersion	string	false	none	none
definitionTime	string	false	none	none

ProcessingState2

```
0
```

- 0: InProgress
- 1: Failed
- 2: Canceled
- 3: Available
- 4: FailedRemediation

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: InProgress - 1: Failed - 2: Canceled - 3: Available - 4: FailedRemediation

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4

Exception

```
{
  "exceptionDetails": "string",
  "exceptionMessage": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
exceptionDetails	string¦null	false	none	none
exceptionMessage	string¦null	false	none	none

FileProcessingDetails

```
{
  "streamingStartedTime": "2019-08-24T14:15:22Z",
  "streamingCompletedTime": "2019-08-24T14:15:22Z",
  "streamingTimeSpanMs": 0,
  "scanCompletedTime": "2019-08-24T14:15:22Z",
  "processingTimeSpanMs": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
streamingStartedTime	string(date-time)	false	none	none
streamingCompletedTime	string(date-time)	false	none	none
streamingTimeSpanMs	integer(int64)	false	none	none
scanCompletedTime	string(date-time)	false	none	none
processingTimeSpanMs	integer(int64)	false	none	none

FileRemediation

```
{
  "isTagged": true,
  "isDeleted": true,
  "isMoved": true,
  "isCopied": true,
  "deepCdr": {
    "sanitizationDetails": {
      "failureCategory": "string",
      "details": "string"
    },
    "cdrObjectState": 0,
    "fileWasReplaced": true
  },
  "postActionFailedInformation": [
    {
      "objectActionType": 0,
      "exception": {
        "exceptionDetails": "string",
        "exceptionMessage": "string"
      }
    }
  ],
  "objectRemediationActions": [
    {
      "objectActionType": 0,
      "objectActionState": 0,
      "cdrObjectState": 0,
      "exceptions": [
        "string"
      ],
      "id": "string",
      "fileId": "string",
      "fileScanCompletedTime": "2019-08-24T14:15:22Z",
      "remediationId": "string",
      "tenantId": "string",
      "lastUpdated": "2019-08-24T14:15:22Z"
    }
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
isTagged	boolean	false	none	none
isDeleted	boolean	false	none	none
isMoved	boolean	false	none	none
isCopied	boolean	false	none	none
deepCdr	DeepCdr¦null	false	none	none
postActionFailedInformation	[PostActionFailedInformation]¦null	false	none	none
objectRemediationActions	[ObjectRemediationAction]¦null	false	none	none

DeepCdr

```
{
  "sanitizationDetails": {
    "failureCategory": "string",
    "details": "string"
  },
  "cdrObjectState": 0,
  "fileWasReplaced": true
}
```

Properties

Name	Type	Required	Restrictions	Description
sanitizationDetails	SanitizationDetails¦null	false	none	none
cdrObjectState	CdrObjectState¦null	false	none	- 0: Original - 1: CdrOriginal - 2: CdrSanitized
fileWasReplaced	boolean	false	none	none

CdrObjectState

```
0
```

- 0: Original
- 1: CdrOriginal
- 2: CdrSanitized

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Original - 1: CdrOriginal - 2: CdrSanitized

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

PostActionFailedInformation

```
{
  "objectActionType": 0,
  "exception": {
    "exceptionDetails": "string",
    "exceptionMessage": "string"
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
objectActionType	ObjectActionType2	false	none	- 0: Tagging - 5: AllowedMoved - 6: BlockedMoved - 7: BlockedDelete - 8: AllowedSanitizedMoved - 9: AllowedCopied - 10: BlockedCopied - 11: AllowedSanitizedCopied - 12: AllowedSanitizedDeleted - 13: AllowedDeleted - 14: AllowedKept - 15: BlockedKept - 16: AllowedSanitizedKept - 17: BlockedSanitizedCopied - 18: BlockedSanitizedMoved - 19: BlockedSanitizedDeleted - 20: BlockedSanitizedKept - 21: ScanFileVersions - 22: DiscoveredMoved - 23: DiscoveredCopied - 24: DiscoveredKept - 25: DiscoveredDelete - 100: SoftDelete - 101: PackageDelete - 102: Copy
exception	Exception¦null	false	none	none

ObjectActionType2

```
0
```

- 0: Tagging
- 5: AllowedMoved
- 6: BlockedMoved
- 7: BlockedDelete
- 8: AllowedSanitizedMoved
- 9: AllowedCopied
- 10: BlockedCopied

- 11: AllowedSanitizedCopied
- 12: AllowedSanitizedDeleted
- 13: AllowedDeleted
- 14: AllowedKept
- 15: BlockedKept
- 16: AllowedSanitizedKept
- 17: BlockedSanitizedCopied
- 18: BlockedSanitizedMoved
- 19: BlockedSanitizedDeleted
- 20: BlockedSanitizedKept
- 21: ScanFileVersions
- 22: DiscoveredMoved
- 23: DiscoveredCopied
- 24: DiscoveredKept
- 25: DiscoveredDelete
- 100: SoftDelete
- 101: PackageDelete
- 102: Copy

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Tagging - 5: AllowedMoved - 6: BlockedMoved - 7: BlockedDelete - 8: AllowedSanitizedMoved - 9: AllowedCopied - 10: BlockedCopied - 11: AllowedSanitizedCopied - 12: AllowedSanitizedDeleted - 13: AllowedDeleted - 14: AllowedKept - 15: BlockedKept - 16: AllowedSanitizedKept - 17: BlockedSanitizedCopied - 18: BlockedSanitizedMoved - 19: BlockedSanitizedDeleted - 20: BlockedSanitizedKept - 21: ScanFileVersions - 22: DiscoveredMoved - 23: DiscoveredCopied - 24: DiscoveredKept - 25: DiscoveredDelete - 100: SoftDelete - 101: PackageDelete - 102: Copy

Enumerated Values

Property	Value
anonymous	0
anonymous	5
anonymous	6
anonymous	7
anonymous	8
anonymous	9
anonymous	10
anonymous	11
anonymous	12
anonymous	13
anonymous	14
anonymous	15
anonymous	16
anonymous	17
anonymous	18
anonymous	19
anonymous	20
anonymous	21
anonymous	22
anonymous	23
anonymous	24
anonymous	25
anonymous	100
anonymous	101
anonymous	102

ObjectRemediationAction


```
{
  "objectActionType": 0,
  "objectActionState": 0,
  "cdrObjectState": 0,
  "exceptions": [
    "string"
  ],
  "id": "string",
  "fileId": "string",
  "fileScanCompletedTime": "2019-08-24T14:15:22Z",
  "remediationId": "string",
  "tenantId": "string",
  "lastUpdated": "2019-08-24T14:15:22Z"
}
```

Properties

Name	Type	Required	Restrictions	Description
objectActionType	ObjectActionType2	false	none	- 0: Tagging - 5: AllowedMoved - 6: BlockedMoved - 7: BlockedDelete - 8: AllowedSanitizedMoved - 9: AllowedCopied - 10: BlockedCopied - 11: AllowedSanitizedCopied - 12: AllowedSanitizedDeleted - 13: AllowedDeleted - 14: AllowedKept - 15: BlockedKept - 16: AllowedSanitizedKept - 17: BlockedSanitizedCopied - 18: BlockedSanitizedMoved - 19: BlockedSanitizedDeleted - 20: BlockedSanitizedKept - 21: ScanFileVersions - 22: DiscoveredMoved - 23: DiscoveredCopied - 24: DiscoveredKept - 25: DiscoveredDelete - 100: SoftDelete - 101: PackageDelete - 102: Copy
objectActionState	ObjectActionState	false	none	- 0: NotApplied - 1: Applied - 3: CannotBeApplied - 4: Error - 5: SanitizationSkipped
cdrObjectState	CdrObjectState¦null	false	none	- 0: Original - 1: CdrOriginal - 2: CdrSanitized
exceptions	[string]	false	none	none
id	string	false	none	none

fileId	string	false	none	none
fileScanCompletedTime	string(date-time)	false	none	none
remediationId	string¦null	false	none	none
tenantId	string¦null	false	none	none
lastUpdated	string(date-time)¦null	false	none	none

ObjectActionState

```
0
```

- 0: NotApplied
- 1: Applied
- 3: CannotBeApplied
- 4: Error
- 5: SanitizationSkipped

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: NotApplied - 1: Applied - 3: CannotBeApplied - 4: Error - 5: SanitizationSkipped

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	3
anonymous	4
anonymous	5

File

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "id": "string",
  "scanId": "string",
  "name": "string",
  "path": "string",
  "hash": "string",
  "fileOwner": {
    "userId": "string",
    "fullName": "string",
    "userName": "string",
    "emailAddress": "string"
  },
  "size": 0,
  "fileTypeCategory": 0,
  "extension": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageName": "string",
  "storageLocation": "string",
  "scanName": "string",
  "progress": {
    "upload": 0,
    "scan": 0
  },
  "scanResult": {
    "extractedFiles": {
      "dataId": "string",
      "scanResultI": 0,
      "detectedBy": 0,
      "filesExtractedCount": 0
    },
    "scanResults": {
      "scanDetails": [
        {
          "engineName": "string",
          "threatFound": "string",
          "scanResultI": 0,
          "defTime": "string",
          "scanTime": 0.1,
          "engId": "string",
          "waitTime": 0.1
```

```
],
      "dataId": "string",
      "scanAllResultI": 0,
      "scanAllResultA": "string",
      "startTime": "string",
      "totalTime": 0.1,
      "totalAvs": 0,
      "totalDetections": 0,
      "progressPercentage": 0
    },
    "fileInfo": {
      "fileSize": 0,
      "uploadTimestamp": "string",
      "md5": "string",
      "sha1": "string",
      "sha256": "string",
      "fileTypeCategory": "string",
      "fileTypeDescription": "string",
      "fileTypeExtension": "string",
      "displayName": "string",
      "fileTypeId": "string"
    },
    "fileScanInfo": {
      "finalVerdict": {
        "blocked": true,
        "verdict": "string",
        "confidence": 0.1,
        "threatLevel": 0.1
      },
      "overviewReport": {
        "analysisState": "string",
        "signalGroups": [
          {
            "description": "string",
            "verdict": "string",
            "signals": [
              {
                "originType": "string",
                "signalReadable": "string",
                "strength": 0.1
              }
            ]
          }
        ]
      },
      "errorCategory": "string",
      "errorCode": 0,
      "errorDetails": "string"
    },
    "processInfo": {
      "postProcessing": {
        "actionsRan": "string",
        "actionsFailed": "string",
```

```
"copyMoveDestination": "string",
        "convertedDestination": "string",
        "sanitizationDetails": {
          "failureCategory": "string",
          "details": "string"
        }
      },
      "progressPercentage": 0,
      "userAgent": "string",
      "profile": "string",
      "result": "string",
      "blockedReason": "string",
      "fileTypeSkippedScan": true,
      "processingTime": 0.1,
      "queueTime": 0.1,
      "resultI": 0
    },
    "dataId": "string",
    "dlpInfo": {
      "hits": null,
      "verdict": 0
    },
    "sbomInfo": {
      "finalVerdict": {
        "blocked": true,
        "licensingPackageCounts": {
          "allowed": 0,
          "blocked": 0,
          "unknown": 0,
          "unsupported": 0
        },
        "severity": 0,
        "totalPackageCount": 0,
        "verdict": "string",
        "vulnerabilityCounts": {
          "critical": 0,
          "high": 0,
          "medium": 0,
          "low": 0,
          "unknown": 0
        }
      },
      "overviewReport": {
        "isPartialReport": true,
        "osInfo": {
          "osName": "string",
          "osVersion": "string"
        },
        "scanMode": "string",
        "steps": [
          {
            "command": "string",
            "packages": [
```

```
"authors": "string",
                "releaseDate": "2019-08-24T14:15:22Z",
                "ecosystem": "string",
                "group": "string",
                "archive": true,
                "executable": true,
                "uid": "string",
                "licenses": {
                  "allowed": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ],
                  "blocked": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ]
                },
                "packageName": "string",
                "packageVersion": "string",
                "hashes": [
                  {
                    "sha256": "string",
                    "fileName": "string"
                  }
                ],
```

```
"critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
                },
                "vulnerabilities": [
                  {
                    "id": "string",
                    "severity": 0,
                    "source": "string",
                    "fixedVersions": "string",
                    "cwes": [
                      "string"
                    ]
                  }
                ],
                "arch": "string",
                "osName": "string",
                "osVersion": "string",
                "dependencies": [
                  {
                    "ecosystem": "string",
                    "group": "string",
                    "packageName": "string",
                    "packageVersion": "string",
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    }
                  }
                ]
              }
            ],
            "licenseStatus": "string",
            "licensingPackageCounts": {
              "allowed": 0,
              "blocked": 0,
              "unknown": 0,
```

```
},
            "step": 0,
            "vulnerabilityCount": 0,
            "sha256": "string",
            "severity": 0,
            "vulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            }
          }
        ],
        "packages": [
          {
            "authors": "string",
            "releaseDate": "2019-08-24T14:15:22Z",
            "ecosystem": "string",
            "group": "string",
            "archive": true,
            "executable": true,
            "uid": "string",
            "licenses": {
              "allowed": [
                {
                  "id": "string",
                  "url": "string",
                  "category": "string",
                  "usageRestriction": {
                    "permissions": [
                      "string"
                    ],
                    "conditions": [
                      "string"
                    ],
                    "limitations": [
                      "string"
                    ]
                  }
                }
              ],
              "blocked": [
                {
                  "id": "string",
                  "url": "string",
                  "category": "string",
                  "usageRestriction": {
                    "permissions": [
                      "string"
                    ],
                    "conditions": [
                      "string"
```

```
"limitations": [
                      "string"
                    ]
                  }
                }
              ]
            },
            "packageName": "string",
            "packageVersion": "string",
            "target": "string",
            "hashes": [
              {
                "sha256": "string",
                "fileName": "string"
              }
            ],
            "vulnerabilities": [
              {
                "id": "string",
                "severity": 0,
                "source": "string",
                "fixedVersions": "string",
                "cwes": [
                  "string"
                ]
              }
            ],
            "vulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            },
            "dependencies": [
              {
                "ecosystem": "string",
                "group": "string",
                "packageName": "string",
                "packageVersion": "string",
                "vulnerabilities": [
                  {
                    "id": "string",
                    "severity": 0,
                    "source": "string",
                    "fixedVersions": "string",
                    "cwes": [
                      "string"
                    ]
                  }
                ],
                "vulnerabilityCounts": {
                  "critical": 0,
```

```
"medium": 0,
                  "low": 0,
                  "unknown": 0
                }
              }
            ]
          }
        ],
        "dependenciesPackageCount": 0,
        "dependenciesVulnerabilityTotal": 0,
        "dependenciesVulnerabilityCounts": {
          "critical": 0,
          "high": 0,
          "medium": 0,
          "low": 0,
          "unknown": 0
        }
      },
      "resultTemplateHash": "string"
    },
    "vulnerabilityInfo": {
      "vulnerabilityInfoResult": {
        "code": 0,
        "hash": "string",
        "method": 0,
        "timeStamp": "string",
        "timing": 0,
        "detectedProduct": {
          "hasVulnerability": true,
          "isCurrent": true,
          "product": {
            "id": 0,
            "name": "string"
          },
          "remediationLink": "string",
          "severity": "string",
          "sigName": "string",
          "signature": 0,
          "vendor": {
            "id": 0,
            "name": "string"
          },
          "version": "string",
          "versionData": {
            "countBehind": 0,
            "feedId": 0,
            "version": "string"
          },
          "vulnerabilities": [
            {
              "description": "string",
              "details": {
                "cpe": "string",
```

```
"cvss": {
                  "accessComplexity": "string",
                  "accessVector": "string",
                  "authentication": "string",
                  "availabilityImpact": "string",
                  "confidentialityImpact": "string",
                  "generatedOnEpoch": "string",
                  "integrityImpact": "string",
                  "score": "string",
                  "source": "string"
                },
                "cvss3": {
                  "attackComplexity": "string",
                  "attackVector": "string",
                  "availabilityImpact": "string",
                  "baseScore": "string",
                  "baseSeverity": "string",
                  "confidentialityImpact": "string",
                  "exploitabilityScore": "string",
                  "impactScore": "string",
                  "integrityImpact": "string",
                  "privilegesRequired": "string",
                  "scope": "string",
                  "userInteraction": "string",
                  "vectorString": "string"
                },
                "cwe": "string",
                "lastModifiedEpoch": "string",
                "publishedEpoch": "string",
                "references": [
                  "string"
                ]
              },
              "severity": "string",
              "severityIndex": 0,
              "staticId": 0
            }
          ]
        }
      }
    },
    "sessionCookie": "string",
    "fileTypeInfo": {
      "fileInfoExtended": {
        "extensions": "string",
        "groupId": "string",
        "typeId": "string"
      }
    },
    "scanInstanceUrl": "string",
    "scanWorkflowSnapshotId": "string",
    "coreEngines": [
      {
```

```
"engineName": "string",
        "engineVersion": "string",
        "dbVersion": "string",
        "definitionTime": "string"
      }
    ]
  },
  "processingState": 0,
  "exception": {
    "exceptionDetails": "string",
    "exceptionMessage": "string"
  },
  "created": "2019-08-24T14:15:22Z",
  "lastModified": "2019-08-24T14:15:22Z",
  "discoveryStarted": "2019-08-24T14:15:22Z",
  "fileProcessingDetails": {
    "streamingStartedTime": "2019-08-24T14:15:22Z",
    "streamingCompletedTime": "2019-08-24T14:15:22Z",
    "streamingTimeSpanMs": 0,
    "scanCompletedTime": "2019-08-24T14:15:22Z",
    "processingTimeSpanMs": 0
  },
  "coreEngines": [
    {
      "engineId": "string",
      "engineName": "string",
      "engineVersion": "string",
      "dbVersion": "string",
      "definitionTime": "string"
    }
  ],
  "remediation": {
    "isTagged": true,
    "isDeleted": true,
    "isMoved": true,
    "isCopied": true,
    "deepCdr": {
      "sanitizationDetails": {
        "failureCategory": "string",
        "details": "string"
      },
      "cdrObjectState": 0,
      "fileWasReplaced": true
    },
    "postActionFailedInformation": [
      {
        "objectActionType": 0,
        "exception": {
          "exceptionDetails": "string",
          "exceptionMessage": "string"
        }
      }
    ],
```

```
{
        "objectActionType": 0,
        "objectActionState": 0,
        "cdrObjectState": 0,
        "exceptions": [
          "string"
        ],
        "id": "string",
        "fileId": "string",
        "fileScanCompletedTime": "2019-08-24T14:15:22Z",
        "remediationId": "string",
        "tenantId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z"
      }
    ]
  },
  "metaDefenderCoreUrl": "string",
  "discoveryStartedDaySec": 0,
  "objectStorageMetadata": "string",
  "scanWorkflowSnapshotId": "string",
  "scanWorkflowSnapshot": {
    "id": "string",
    "scanId": "string",
    "workflow": {
      "id": "string",
      "scanPoolId": "string",
      "scanProcessFlow": 0,
      "isIdentityScanningEnabled": true,
      "scanIntervalDays": 0,
      "rules": [
        "string"
      ],
      "type": 0,
      "workflowSupportedTechnologies": {
        "dlp": {
          "status": 0
        },
        "deepCdr": {
          "status": 0
        },
        "sandBox": {
          "status": 0
        },
        "metascan": {
          "status": 0
        }
      },
      "name": "string",
      "userAgent": "string",
      "scanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
```

```
"scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "failOverScanPoolId": "string",
      "failOverScanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "filters": {
        "id": "string",
        "minimumSize": 0,
        "maximumSize": 0,
        "minimumDateTime": "2019-08-24T14:15:22Z",
        "maximumDateTime": "2019-08-24T14:15:22Z"
      },
      "fileDefinition": {
        "blockingStrategy": 0,
        "configurableBlockingStrategy": {
          "shouldTreatFilesWithSensitiveDataAsBlocked": true,
          "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
        }
      },
      "remediations": [
        {
          "id": "string",
          "workflowId": "string",
          "isEnabled": true,
          "configuration": "string",
          "type": 0,
          "handlingType": 0
        }
      ],
      "tenantId": "string"
```

```
"startTimestamp": 0,
    "stopTimestamp": 0,
    "scanPool": {
      "id": "string",
      "name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    },
    "scanInstances": [
      {
        "id": "string",
        "scanPoolId": "string",
        "url": "string",
        "apiKey": "string",
        "timeout": "string",
        "scanInstanceType": 0
      }
    ],
    "failOverScanPool": {
      "id": "string",
      "name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    },
    "failOverScanInstances": [
      {
        "id": "string",
        "scanPoolId": "string",
        "url": "string",
        "apiKey": "string",
        "timeout": "string",
        "scanInstanceType": 0
      }
```

```
"storage": {
      "id": "string",
      "name": "string",
      "groupId": "string",
      "clientId": "string",
      "accountId": "string",
      "source": "string",
      "storageFunctionality": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "workflowId": "string",
      "lastUpdated": "2019-08-24T14:15:22Z",
      "tenantId": "string"
    },
    "workflowStatistics": {
      "taggingStatistics": {
        "total": 0
      },
      "keepAllowedStatistics": {
        "total": 0
      },
      "keepBlockedStatistics": {
        "total": 0
      },
      "keepSanitizedStatistics": {
        "total": 0
      },
      "movedAllowedStatistics": {
        "total": 0,
        "destinationStorageInformation": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
```

```
},
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        }
      },
      "movedSanitizedStatistics": {
        "total": 0,
        "destinationStorageInformation": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        }
      },
      "movedBlockedStatistics": {
        "total": 0,
        "destinationStorageInformation": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
```

```
}
      },
      "copyAllowedStatistics": {
        "total": 0,
        "destinationStorageInformation": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        }
      },
      "copyBlockedStatistics": {
        "total": 0,
        "destinationStorageInformation": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        }
      },
      "deleteAllowedStatistics": {
```

```
},
      "deleteBlockedStatistics": {
        "total": 0
      },
      "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
    },
    "account": {
      "id": "string",
      "name": "string",
      "description": "string",
      "credentials": "string",
      "protocolType": 0,
      "vendorType": 0,
      "autoDiscovery": true,
      "autodiscoveryDestinationGroupId": "string"
    },
    "realTimeStartDate": "2019-08-24T14:15:22Z"
  },
  "versions": [
    {
      "result": 0,
      "responseKey": "string",
      "responseMessage": "string",
      "responseMessageParams": {
        "property1": "string",
        "property2": "string"
      },
      "id": "string",
      "scanId": "string",
      "name": "string",
      "path": "string",
      "hash": "string",
      "fileOwner": {
        "userId": "string",
        "fullName": "string",
        "userName": "string",
        "emailAddress": "string"
      },
      "size": 0,
      "fileTypeCategory": 0,
      "extension": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageName": "string",
      "storageLocation": "string",
      "scanName": "string",
      "progress": {
        "upload": 0,
        "scan": 0
```

```
"scanResult": {
        "extractedFiles": {
          "dataId": "string",
          "scanResultI": 0,
          "detectedBy": 0,
          "filesExtractedCount": 0
        },
        "scanResults": {
          "scanDetails": [
            {
              "engineName": "string",
              "threatFound": "string",
              "scanResultI": 0,
              "defTime": "string",
              "scanTime": 0.1,
              "engId": "string",
              "waitTime": 0.1
            }
          ],
          "dataId": "string",
          "scanAllResultI": 0,
          "scanAllResultA": "string",
          "startTime": "string",
          "totalTime": 0.1,
          "totalAvs": 0,
          "totalDetections": 0,
          "progressPercentage": 0
        },
        "fileInfo": {
          "fileSize": 0,
          "uploadTimestamp": "string",
          "md5": "string",
          "sha1": "string",
          "sha256": "string",
          "fileTypeCategory": "string",
          "fileTypeDescription": "string",
          "fileTypeExtension": "string",
          "displayName": "string",
          "fileTypeId": "string"
        },
        "fileScanInfo": {
          "finalVerdict": {
            "blocked": true,
            "verdict": "string",
            "confidence": 0.1,
            "threatLevel": 0.1
          },
          "overviewReport": {
            "analysisState": "string",
            "signalGroups": [
              {
                "description": "string",
                "verdict": "string",
```

```
{
                    "originType": "string",
                    "signalReadable": "string",
                    "strength": 0.1
                  }
                ]
              }
            ]
          },
          "errorCategory": "string",
          "errorCode": 0,
          "errorDetails": "string"
        },
        "processInfo": {
          "postProcessing": {
            "actionsRan": "string",
            "actionsFailed": "string",
            "convertedTo": "string",
            "copyMoveDestination": "string",
            "convertedDestination": "string",
            "sanitizationDetails": {
              "failureCategory": "string",
              "details": "string"
            }
          },
          "progressPercentage": 0,
          "userAgent": "string",
          "profile": "string",
          "result": "string",
          "blockedReason": "string",
          "fileTypeSkippedScan": true,
          "processingTime": 0.1,
          "queueTime": 0.1,
          "resultI": 0
        },
        "dataId": "string",
        "dlpInfo": {
          "hits": null,
          "verdict": 0
        },
        "sbomInfo": {
          "finalVerdict": {
            "blocked": true,
            "licensingPackageCounts": {
              "allowed": 0,
              "blocked": 0,
              "unknown": 0,
              "unsupported": 0
            },
            "severity": 0,
            "totalPackageCount": 0,
            "verdict": "string",
            "vulnerabilityCounts": {
```

```
"high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            }
          },
          "overviewReport": {
            "isPartialReport": true,
            "osInfo": {
              "osName": "string",
              "osVersion": "string"
            },
            "scanMode": "string",
            "steps": [
              {
                "command": "string",
                "packages": [
                  {
                    "authors": "string",
                    "releaseDate": "2019-08-24T14:15:22Z",
                    "ecosystem": "string",
                    "group": "string",
                    "archive": true,
                    "executable": true,
                    "uid": "string",
                    "licenses": {
                      "allowed": [
                        {
                          "id": "string",
                          "url": "string",
                          "category": "string",
                          "usageRestriction": {
                            "permissions": [
                              null
                            ],
                            "conditions": [
                              null
                            ],
                            "limitations": [
                              null
                            ]
                          }
                        }
                      ],
                      "blocked": [
                        {
                          "id": "string",
                          "url": "string",
                          "category": "string",
                          "usageRestriction": {
                            "permissions": [
                              null
                            ],
```

```
null
                            ],
                            "limitations": [
                              null
                            ]
                          }
                        }
                      ]
                    },
                    "packageName": "string",
                    "packageVersion": "string",
                    "hashes": [
                      {
                        "sha256": "string",
                        "fileName": "string"
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    },
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "arch": "string",
                    "osName": "string",
                    "osVersion": "string",
                    "dependencies": [
                      {
                        "ecosystem": "string",
                        "group": "string",
                        "packageName": "string",
                        "packageVersion": "string",
                        "vulnerabilities": [
                          {
                            "id": "string",
                            "severity": 0,
                            "source": "string",
                            "fixedVersions": "string",
                            "cwes": [
                              null
                            ]
```

```
],
                        "vulnerabilityCounts": {
                          "critical": 0,
                          "high": 0,
                          "medium": 0,
                          "low": 0,
                          "unknown": 0
                        }
                      }
                    ]
                  }
                ],
                "licenseStatus": "string",
                "licensingPackageCounts": {
                  "allowed": 0,
                  "blocked": 0,
                  "unknown": 0,
                  "unsupported": 0
                },
                "step": 0,
                "vulnerabilityCount": 0,
                "sha256": "string",
                "severity": 0,
                "vulnerabilityCounts": {
                  "critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
                }
              }
            ],
            "packages": [
              {
                "authors": "string",
                "releaseDate": "2019-08-24T14:15:22Z",
                "ecosystem": "string",
                "group": "string",
                "archive": true,
                "executable": true,
                "uid": "string",
                "licenses": {
                  "allowed": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
```

```
"limitations": [
                        "string"
                      ]
                    }
                  }
                ],
                "blocked": [
                  {
                    "id": "string",
                    "url": "string",
                    "category": "string",
                    "usageRestriction": {
                      "permissions": [
                        "string"
                      ],
                      "conditions": [
                        "string"
                      ],
                      "limitations": [
                        "string"
                      ]
                    }
                  }
                ]
              },
              "packageName": "string",
              "packageVersion": "string",
              "target": "string",
              "hashes": [
                {
                  "sha256": "string",
                  "fileName": "string"
                }
              ],
              "vulnerabilities": [
                {
                  "id": "string",
                  "severity": 0,
                  "source": "string",
                  "fixedVersions": "string",
                  "cwes": [
                    "string"
                  ]
                }
              ],
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              },
              "dependencies": [
```

```
"ecosystem": "string",
                    "group": "string",
                    "packageName": "string",
                    "packageVersion": "string",
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    }
                  }
                ]
              }
            ],
            "dependenciesPackageCount": 0,
            "dependenciesVulnerabilityTotal": 0,
            "dependenciesVulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            }
          },
          "resultTemplateHash": "string"
        },
        "vulnerabilityInfo": {
          "vulnerabilityInfoResult": {
            "code": 0,
            "hash": "string",
            "method": 0,
            "timeStamp": "string",
            "timing": 0,
            "detectedProduct": {
              "hasVulnerability": true,
              "isCurrent": true,
              "product": {
                "id": 0,
                "name": "string"
              },
              "remediationLink": "string",
```

```
"sigName": "string",
            "signature": 0,
            "vendor": {
              "id": 0,
              "name": "string"
            },
            "version": "string",
            "versionData": {
              "countBehind": 0,
              "feedId": 0,
              "version": "string"
            },
            "vulnerabilities": [
              {
                "description": "string",
                "details": {
                  "cpe": "string",
                  "cve": "string",
                  "cvss": {
                    "accessComplexity": "string",
                    "accessVector": "string",
                    "authentication": "string",
                    "availabilityImpact": "string",
                    "confidentialityImpact": "string",
                    "generatedOnEpoch": "string",
                    "integrityImpact": "string",
                    "score": "string",
                    "source": "string"
                  },
                  "cvss3": {
                    "attackComplexity": "string",
                    "attackVector": "string",
                    "availabilityImpact": "string",
                    "baseScore": "string",
                    "baseSeverity": "string",
                    "confidentialityImpact": "string",
                    "exploitabilityScore": "string",
                    "impactScore": "string",
                    "integrityImpact": "string",
                    "privilegesRequired": "string",
                    "scope": "string",
                    "userInteraction": "string",
                    "vectorString": "string"
                  },
                  "cwe": "string",
                  "lastModifiedEpoch": "string",
                  "publishedEpoch": "string",
                  "references": [
                    "string"
                  ]
                },
                "severity": "string",
                "severityIndex": 0,
```

```
}
              ]
            }
          }
        },
        "sessionCookie": "string",
        "fileTypeInfo": {
          "fileInfoExtended": {
            "extensions": "string",
            "groupId": "string",
            "typeId": "string"
          }
        },
        "scanInstanceUrl": "string",
        "scanWorkflowSnapshotId": "string",
        "coreEngines": [
          {
            "engineId": "string",
            "engineName": "string",
            "engineVersion": "string",
            "dbVersion": "string",
            "definitionTime": "string"
          }
        ]
      },
      "processingState": 0,
      "exception": {
        "exceptionDetails": "string",
        "exceptionMessage": "string"
      },
      "created": "2019-08-24T14:15:22Z",
      "lastModified": "2019-08-24T14:15:22Z",
      "discoveryStarted": "2019-08-24T14:15:22Z",
      "fileProcessingDetails": {
        "streamingStartedTime": "2019-08-24T14:15:22Z",
        "streamingCompletedTime": "2019-08-24T14:15:22Z",
        "streamingTimeSpanMs": 0,
        "scanCompletedTime": "2019-08-24T14:15:22Z",
        "processingTimeSpanMs": 0
      },
      "coreEngines": [
        {
          "engineId": "string",
          "engineName": "string",
          "engineVersion": "string",
          "dbVersion": "string",
          "definitionTime": "string"
        }
      ],
      "remediation": {
        "isTagged": true,
        "isDeleted": true,
        "isMoved": true,
```

```
"deepCdr": {
          "sanitizationDetails": {
            "failureCategory": "string",
            "details": "string"
          },
          "cdrObjectState": 0,
          "fileWasReplaced": true
        },
        "postActionFailedInformation": [
          {
            "objectActionType": 0,
            "exception": {
              "exceptionDetails": "string",
              "exceptionMessage": "string"
            }
          }
        ],
        "objectRemediationActions": [
          {
            "objectActionType": 0,
            "objectActionState": 0,
            "cdrObjectState": 0,
            "exceptions": [
              "string"
            ],
            "id": "string",
            "fileId": "string",
            "fileScanCompletedTime": "2019-08-24T14:15:22Z",
            "remediationId": "string",
            "tenantId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z"
          }
        ]
      },
      "metaDefenderCoreUrl": "string",
      "discoveryStartedDaySec": 0,
      "objectStorageMetadata": "string",
      "scanWorkflowSnapshotId": "string",
      "scanWorkflowSnapshot": {
        "id": "string",
        "scanId": "string",
        "workflow": {
          "id": "string",
          "scanPoolId": "string",
          "scanProcessFlow": 0,
          "isIdentityScanningEnabled": true,
          "scanIntervalDays": 0,
          "rules": [
            "string"
          ],
          "type": 0,
          "workflowSupportedTechnologies": {
            "dlp": {
```

```
},
            "deepCdr": {
              "status": 0
            },
            "sandBox": {
              "status": 0
            },
            "metascan": {
              "status": 0
            }
          },
          "name": "string",
          "userAgent": "string",
          "scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanPoolId": "string",
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "filters": {
            "id": "string",
            "minimumSize": 0,
            "maximumSize": 0,
            "minimumDateTime": "2019-08-24T14:15:22Z",
            "maximumDateTime": "2019-08-24T14:15:22Z"
          },
```

```
"blockingStrategy": 0,
            "configurableBlockingStrategy": {
              "shouldTreatFilesWithSensitiveDataAsBlocked": true,
              "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
            }
          },
          "remediations": [
            {
              "id": "string",
              "workflowId": "string",
              "isEnabled": true,
              "configuration": "string",
              "type": 0,
              "handlingType": 0
            }
          ],
          "tenantId": "string"
        },
        "startTimestamp": 0,
        "stopTimestamp": 0,
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
```

```
"scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "storage": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        },
        "workflowStatistics": {
          "taggingStatistics": {
            "total": 0
          },
          "keepAllowedStatistics": {
            "total": 0
          },
          "keepBlockedStatistics": {
            "total": 0
          },
          "keepSanitizedStatistics": {
            "total": 0
          },
```

```
"total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedSanitizedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
```

```
"groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
```

```
"storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "deleteAllowedStatistics": {
            "total": 0
          },
          "deleteBlockedStatistics": {
            "total": 0
          },
          "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
        },
        "account": {
          "id": "string",
          "name": "string",
          "description": "string",
          "credentials": "string",
          "protocolType": 0,
          "vendorType": 0,
          "autoDiscovery": true,
          "autodiscoveryDestinationGroupId": "string"
        },
        "realTimeStartDate": "2019-08-24T14:15:22Z"
      },
      "versions": [],
      "hasBlockedVersions": true,
      "versionsCount": 0,
      "objectType": 0,
      "skippedScan": true,
      "$type": "string",
      "url": "string",
      "fileId": "string"
    }
  ],
  "hasBlockedVersions": true,
  "versionsCount": 0,
  "objectType": 0,
  "skippedScan": true,
  "$type": "string"
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	FileBase	false	none	none

and - discriminator: $type

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» storageType	StorageType¦null	false	none	none
» storageName	string¦null	false	none	none
» storageLocation	string¦null	false	none	none
» scanName	string¦null	false	none	none
» progress	Progress¦null	false	none	none
» scanResult	CoreScanResult¦null	false	none	none
» processingState	ProcessingState2	false	none	- 0: InProgress - 1: Failed - 2: Canceled - 3: Available - 4: FailedRemediation
» exception	Exception¦null	false	none	none
» created	string(date-time)	false	none	none
» lastModified	string(date-time)	false	none	none
» discoveryStarted	string(date-time)	false	none	none
» fileProcessingDetails	FileProcessingDetails¦null	false	none	none
» coreEngines	[CoreEngine]¦null	false	none	none
» remediation	FileRemediation¦null	false	none	none
» metaDefenderCoreUrl	string¦null	false	none	none
» discoveryStartedDaySec	integer(int64)	false	none	none
» objectStorageMetadata	string¦null	false	none	none
» scanWorkflowSnapshotId	string¦null	false	none	none
» scanWorkflowSnapshot	ScanWorkflowSnapshot¦null	false	none	none
» versions	[FileVersion]¦null	false	none	none
» hasBlockedVersions	boolean	false	none	none
» versionsCount	integer(int32)	false	none	none

» objectType	ObjectType	false	none	- 0: File - 1: FileVersion
» skippedScan	boolean	false	none	none
» $type	string	true	none	none

ScanWorkflowSnapshot

```
{
  "id": "string",
  "scanId": "string",
  "workflow": {
    "id": "string",
    "scanPoolId": "string",
    "scanProcessFlow": 0,
    "isIdentityScanningEnabled": true,
    "scanIntervalDays": 0,
    "rules": [
      "string"
    ],
    "type": 0,
    "workflowSupportedTechnologies": {
      "dlp": {
        "status": 0
      },
      "deepCdr": {
        "status": 0
      },
      "sandBox": {
        "status": 0
      },
      "metascan": {
        "status": 0
      }
    },
    "name": "string",
    "userAgent": "string",
    "scanPool": {
      "id": "string",
      "name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    },
    "failOverScanPoolId": "string",
    "failOverScanPool": {
      "id": "string",
      "name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
```

```
"id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    },
    "filters": {
      "id": "string",
      "minimumSize": 0,
      "maximumSize": 0,
      "minimumDateTime": "2019-08-24T14:15:22Z",
      "maximumDateTime": "2019-08-24T14:15:22Z"
    },
    "fileDefinition": {
      "blockingStrategy": 0,
      "configurableBlockingStrategy": {
        "shouldTreatFilesWithSensitiveDataAsBlocked": true,
        "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
      }
    },
    "remediations": [
      {
        "id": "string",
        "workflowId": "string",
        "isEnabled": true,
        "configuration": "string",
        "type": 0,
        "handlingType": 0
      }
    ],
    "tenantId": "string"
  },
  "startTimestamp": 0,
  "stopTimestamp": 0,
  "scanPool": {
    "id": "string",
    "name": "string",
    "scanPoolType": 0,
    "isDefault": true,
    "scanInstances": [
      {
        "id": "string",
        "scanPoolId": "string",
        "url": "string",
        "apiKey": "string",
        "timeout": "string",
        "scanInstanceType": 0
      }
    ]
  },
```

```
{
      "id": "string",
      "scanPoolId": "string",
      "url": "string",
      "apiKey": "string",
      "timeout": "string",
      "scanInstanceType": 0
    }
  ],
  "failOverScanPool": {
    "id": "string",
    "name": "string",
    "scanPoolType": 0,
    "isDefault": true,
    "scanInstances": [
      {
        "id": "string",
        "scanPoolId": "string",
        "url": "string",
        "apiKey": "string",
        "timeout": "string",
        "scanInstanceType": 0
      }
    ]
  },
  "failOverScanInstances": [
    {
      "id": "string",
      "scanPoolId": "string",
      "url": "string",
      "apiKey": "string",
      "timeout": "string",
      "scanInstanceType": 0
    }
  ],
  "storage": {
    "id": "string",
    "name": "string",
    "groupId": "string",
    "clientId": "string",
    "accountId": "string",
    "source": "string",
    "storageFunctionality": "string",
    "storageType": {
      "name": "string",
      "categoryType": 0,
      "protocolType": 0,
      "vendorType": 0
    },
    "storageStatus": {
      "message": "string",
      "statusType": 0
    },
```

```
"lastUpdated": "2019-08-24T14:15:22Z",
    "tenantId": "string"
  },
  "workflowStatistics": {
    "taggingStatistics": {
      "total": 0
    },
    "keepAllowedStatistics": {
      "total": 0
    },
    "keepBlockedStatistics": {
      "total": 0
    },
    "keepSanitizedStatistics": {
      "total": 0
    },
    "movedAllowedStatistics": {
      "total": 0,
      "destinationStorageInformation": {
        "id": "string",
        "name": "string",
        "groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      }
    },
    "movedSanitizedStatistics": {
      "total": 0,
      "destinationStorageInformation": {
        "id": "string",
        "name": "string",
        "groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
```

```
"protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      }
    },
    "movedBlockedStatistics": {
      "total": 0,
      "destinationStorageInformation": {
        "id": "string",
        "name": "string",
        "groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      }
    },
    "copyAllowedStatistics": {
      "total": 0,
      "destinationStorageInformation": {
        "id": "string",
        "name": "string",
        "groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
```

```
"message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      }
    },
    "copyBlockedStatistics": {
      "total": 0,
      "destinationStorageInformation": {
        "id": "string",
        "name": "string",
        "groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      }
    },
    "deleteAllowedStatistics": {
      "total": 0
    },
    "deleteBlockedStatistics": {
      "total": 0
    },
    "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
  },
  "account": {
    "id": "string",
    "name": "string",
    "description": "string",
    "credentials": "string",
    "protocolType": 0,
    "vendorType": 0,
    "autoDiscovery": true,
    "autodiscoveryDestinationGroupId": "string"
  },
  "realTimeStartDate": "2019-08-24T14:15:22Z"
```

```
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
scanId	string	false	none	none
workflow	Workflow	false	none	none
startTimestamp	integer(int64)	false	none	none
stopTimestamp	integer(int64)	false	none	none
scanPool	ScanPool¦null	false	none	none
scanInstances	[ScanInstance]¦null	false	none	none
failOverScanPool	ScanPool¦null	false	none	none
failOverScanInstances	[ScanInstance]¦null	false	none	none
storage	NewStorage¦null	false	none	none
workflowStatistics	WorkflowStatistics	false	none	none
account	Account¦null	false	none	none
realTimeStartDate	string(date-time)¦null	false	none	none

Workflow

```
{
  "id": "string",
  "scanPoolId": "string",
  "scanProcessFlow": 0,
  "isIdentityScanningEnabled": true,
  "scanIntervalDays": 0,
  "rules": [
    "string"
  ],
  "type": 0,
  "workflowSupportedTechnologies": {
    "dlp": {
      "status": 0
    },
    "deepCdr": {
      "status": 0
    },
    "sandBox": {
      "status": 0
    },
    "metascan": {
      "status": 0
    }
  },
  "name": "string",
  "userAgent": "string",
  "scanPool": {
    "id": "string",
    "name": "string",
    "scanPoolType": 0,
    "isDefault": true,
    "scanInstances": [
      {
        "id": "string",
        "scanPoolId": "string",
        "url": "string",
        "apiKey": "string",
        "timeout": "string",
        "scanInstanceType": 0
      }
    ]
  },
  "failOverScanPoolId": "string",
  "failOverScanPool": {
    "id": "string",
    "name": "string",
    "scanPoolType": 0,
    "isDefault": true,
    "scanInstances": [
      {
        "id": "string",
        "scanPoolId": "string",
```

```
"apiKey": "string",
        "timeout": "string",
        "scanInstanceType": 0
      }
    ]
  },
  "filters": {
    "id": "string",
    "minimumSize": 0,
    "maximumSize": 0,
    "minimumDateTime": "2019-08-24T14:15:22Z",
    "maximumDateTime": "2019-08-24T14:15:22Z"
  },
  "fileDefinition": {
    "blockingStrategy": 0,
    "configurableBlockingStrategy": {
      "shouldTreatFilesWithSensitiveDataAsBlocked": true,
      "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
    }
  },
  "remediations": [
    {
      "id": "string",
      "workflowId": "string",
      "isEnabled": true,
      "configuration": "string",
      "type": 0,
      "handlingType": 0
    }
  ],
  "tenantId": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
scanPoolId	string¦null	false	none	none
scanProcessFlow	ScanProcessFlow¦null	false	none	- 0: WithScanning - 1: WithoutScanning
isIdentityScanningEnabled	boolean	false	none	none
scanIntervalDays	integer(int32)¦null	false	none	none
rules	[string]¦null	false	none	none
type	WorkflowType	false	none	- 0: Default - 1: Basic
workflowSupportedTechnologies	WorkflowSupportedTechnologies¦null	false	none	none
name	string	false	none	none
userAgent	string¦null	false	none	none
scanPool	ScanPool¦null	false	none	none
failOverScanPoolId	string¦null	false	none	none
failOverScanPool	ScanPool¦null	false	none	none
filters	Filters	false	none	none
fileDefinition	FileDefinition¦null	false	none	none
remediations	[Remediation]	false	none	none
tenantId	string¦null	false	none	none

ScanProcessFlow

```
0
```

- 0: WithScanning
- 1: WithoutScanning

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: WithScanning - 1: WithoutScanning

Enumerated Values

Property	Value
anonymous	0
anonymous	1

WorkflowType

```
0
```

- 0: Default
- 1: Basic

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Default - 1: Basic

Enumerated Values

Property	Value
anonymous	0
anonymous	1

WorkflowSupportedTechnologies

```
{
  "dlp": {
    "status": 0
  },
  "deepCdr": {
    "status": 0
  },
  "sandBox": {
    "status": 0
  },
  "metascan": {
    "status": 0
  }
}
```


Properties

Name	Type	Required	Restrictions	Description
dlp	SupportedTechnology	false	none	none
deepCdr	SupportedTechnology	false	none	none
sandBox	SupportedTechnology	false	none	none
metascan	SupportedTechnology	false	none	none

SupportedTechnology


```
{
  "status": 0
}
```


Properties

Name	Type	Required	Restrictions	Description
status	TechnologyStatus	false	none	- 0: Unknown - 1: Disabled - 2: Enabled

TechnologyStatus

```
0
```

- 0: Unknown
- 1: Disabled
- 2: Enabled

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Unknown - 1: Disabled - 2: Enabled

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

ScanPool

```
{
  "id": "string",
  "name": "string",
  "scanPoolType": 0,
  "isDefault": true,
  "scanInstances": [
    {
      "id": "string",
      "scanPoolId": "string",
      "url": "string",
      "apiKey": "string",
      "timeout": "string",
      "scanInstanceType": 0
    }
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
name	string	false	none	none
scanPoolType	ScanInstanceType	false	none	- 0: MdCore - 1: MdCloud
isDefault	boolean	false	none	none
scanInstances	[ScanInstance]¦null	false	none	none

ScanInstanceType

```
0
```

- 0: MdCore
- 1: MdCloud

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: MdCore - 1: MdCloud

Enumerated Values

Property	Value
anonymous	0
anonymous	1

ScanInstance

```
{
  "id": "string",
  "scanPoolId": "string",
  "url": "string",
  "apiKey": "string",
  "timeout": "string",
  "scanInstanceType": 0
}
```


Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
scanPoolId	string	false	none	none
url	string	false	none	none
apiKey	string|null	false	none	none
timeout	string(duration)	false	none	none
scanInstanceType	ScanInstanceType	false	none	- 0: MdCore - 1: MdCloud

Filters


```
{
  "id": "string",
  "minimumSize": 0,
  "maximumSize": 0,
  "minimumDateTime": "2019-08-24T14:15:22Z",
  "maximumDateTime": "2019-08-24T14:15:22Z"
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
minimumSize	integer(int64)	false	none	none
maximumSize	integer(int64)	false	none	none
minimumDateTime	string(date-time)	false	none	none
maximumDateTime	string(date-time)	false	none	none

FileDefinition

```
{
  "blockingStrategy": 0,
  "configurableBlockingStrategy": {
    "shouldTreatFilesWithSensitiveDataAsBlocked": true,
    "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
blockingStrategy	FileBlockingStrategy	false	none	- 0: Strict - 1: Configurable
configurableBlockingStrategy	ConfigurableBlockingStrategy	false	none	none

FileBlockingStrategy

```
0
```

- 0: Strict
- 1: Configurable

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Strict - 1: Configurable

Enumerated Values

Property	Value
anonymous	0
anonymous	1

ConfigurableBlockingStrategy

```
{
  "shouldTreatFilesWithSensitiveDataAsBlocked": true,
  "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
}
```

Properties

Name	Type	Required	Restrictions	Description
shouldTreatFilesWithSensitiveDataAsBlocked	boolean	false	none	none
shouldTreatFilesWithVulnerabilitiesAsBlocked	boolean	false	none	none

Remediation

```
{
  "id": "string",
  "workflowId": "string",
  "isEnabled": true,
  "configuration": "string",
  "type": 0,
  "handlingType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
workflowId	string	false	none	none
isEnabled	boolean	false	none	none
configuration	string	false	none	none
type	RemediationType	false	none	- 0: FileTagging - 1: DeepCdr - 2: BlockedFileRemediation - 3: AllowedSanitizedFileRemediation - 4: AllowedFileRemediation - 5: BlockedSanitizedFileRemediation - 6: DeleteEmptyFoldersRemediation - 7: ScanFileVersions - 8: DiscoveredFileRemediation
handlingType	RemediationHandlingType	false	none	- 0: Delete - 1: Move - 2: Copy - 3: Keep - 999: NotApplicable

RemediationType

```
0
```

- 0: FileTagging
- 1: DeepCdr
- 2: BlockedFileRemediation
- 3: AllowedSanitizedFileRemediation
- 4: AllowedFileRemediation
- 5: BlockedSanitizedFileRemediation
- 6: DeleteEmptyFoldersRemediation
- 7: ScanFileVersions
- 8: DiscoveredFileRemediation

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: FileTagging - 1: DeepCdr - 2: BlockedFileRemediation - 3: AllowedSanitizedFileRemediation - 4: AllowedFileRemediation - 5: BlockedSanitizedFileRemediation - 6: DeleteEmptyFoldersRemediation - 7: ScanFileVersions - 8: DiscoveredFileRemediation

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4
anonymous	5
anonymous	6
anonymous	7
anonymous	8

RemediationHandlingType

```
0
```

- 0: Delete
- 1: Move
- 2: Copy
- 3: Keep
- 999: NotApplicable

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Delete - 1: Move - 2: Copy - 3: Keep - 999: NotApplicable

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	999

NewStorage

```
{
  "id": "string",
  "name": "string",
  "groupId": "string",
  "clientId": "string",
  "accountId": "string",
  "source": "string",
  "storageFunctionality": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "workflowId": "string",
  "lastUpdated": "2019-08-24T14:15:22Z",
  "tenantId": "string"
}
```


Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
name	string	false	none	none
groupId	string	false	none	none
clientId	string	false	none	none
accountId	string	false	none	none
source	string	false	none	none
storageFunctionality	string	false	none	none
storageType	NewStorageType	false	none	none
storageStatus	NewStorageStatus	false	none	none
workflowId	string¦null	false	none	none
lastUpdated	string(date-time)¦null	false	none	none
tenantId	string¦null	false	none	none

NewStorageType

```
{
  "name": "string",
  "categoryType": 0,
  "protocolType": 0,
  "vendorType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
name	string	false	none	none
categoryType	StorageCategoryType2	false	none	- 0: Local - 1: Cloud - 2: Object
protocolType	StorageProtocolType	false	none	- 0: Smb - 1: GraphSdk - 2: BoxSdk - 3: AwsSdk - 4: AzureBlobSdk - 5: AlibabaCloudSdk - 6: GoogleCloudSdk - 7: AzureFilesSdk - 8: Sftp - 9: Nfs - 10: Mft - 11: SharepointOnPrem - 12: Ftp - 13: OracleSdk - 100: GitHubSdk - 101: DockerHubSdk - 102: BitbucketSdk - 103: AmazonEcrSdk - 104: QuaySdk - 105: JFrogContainerSdk - 106: JFrogBinarySdk - 107: AzureAcrSdk - 108: GitLabSourceSdk - 109: GitLabContainerSdk - 110: SvnSdk - 111: AzureDevOpsSourceSdk - 112: AzureDevOpsBinarySdk - 113: DirectFile - 114: NexusBinarySdk - 115: GiteaSdk
vendorType	StorageVendorType	false	none	- 0: AmazonS3 - 1: OneDrive - 2: Box - 3: DellIsilon - 4: AzureFiles - 5: SmbCompatible - 6: S3Compatible - 7: AzureBlob - 8: AlibabaCloud - 9: GoogleCloud

				- 10: Sftp - 11: SharePoint - 12: Nfs - 13: Wasabi - 14: OracleCloud - 15: NetAppOntap - 16: Cubbit - 17: Mft - 18: SharepointOnPrem - 19: Ftp - 20: Oracle - 21: NfsSisl - 22: ScalityRing - 100: GitHub - 101: DockerHub - 102: Bitbucket - 103: AmazonEcr - 104: Quay - 105: JFrogContainer - 106: JFrogBinary - 107: AzureAcr - 108: GitLabSource - 109: GitLabContainer - 110: Svn - 111: AzureDevOpsSource - 112: AzureDevOpsBinary - 113: DirectFile - 114: NexusBinary - 115: Gitea

StorageCategoryType2

```
0
```

- 0: Local
- 1: Cloud
- 2: Object

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Local - 1: Cloud - 2: Object

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

NewStorageStatus

```
{
  "message": "string",
  "statusType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
message	string¦null	false	none	none
statusType	StorageStatusType	false	none	- 0: NotTested - 1: Testing - 2: Operational - 3: Error

StorageStatusType

```
0
```

- 0: NotTested
- 1: Testing
- 2: Operational
- 3: Error

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: NotTested - 1: Testing - 2: Operational - 3: Error

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3

WorkflowStatistics

```
{
  "taggingStatistics": {
    "total": 0
  },
  "keepAllowedStatistics": {
    "total": 0
  },
  "keepBlockedStatistics": {
    "total": 0
  },
  "keepSanitizedStatistics": {
    "total": 0
  },
  "movedAllowedStatistics": {
    "total": 0,
    "destinationStorageInformation": {
      "id": "string",
      "name": "string",
      "groupId": "string",
      "clientId": "string",
      "accountId": "string",
      "source": "string",
      "storageFunctionality": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "workflowId": "string",
      "lastUpdated": "2019-08-24T14:15:22Z",
      "tenantId": "string"
    }
  },
  "movedSanitizedStatistics": {
    "total": 0,
    "destinationStorageInformation": {
      "id": "string",
      "name": "string",
      "groupId": "string",
      "clientId": "string",
      "accountId": "string",
      "source": "string",
      "storageFunctionality": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
```

```
},
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "workflowId": "string",
      "lastUpdated": "2019-08-24T14:15:22Z",
      "tenantId": "string"
    }
  },
  "movedBlockedStatistics": {
    "total": 0,
    "destinationStorageInformation": {
      "id": "string",
      "name": "string",
      "groupId": "string",
      "clientId": "string",
      "accountId": "string",
      "source": "string",
      "storageFunctionality": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "workflowId": "string",
      "lastUpdated": "2019-08-24T14:15:22Z",
      "tenantId": "string"
    }
  },
  "copyAllowedStatistics": {
    "total": 0,
    "destinationStorageInformation": {
      "id": "string",
      "name": "string",
      "groupId": "string",
      "clientId": "string",
      "accountId": "string",
      "source": "string",
      "storageFunctionality": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageStatus": {
        "message": "string",
```

```
},
      "workflowId": "string",
      "lastUpdated": "2019-08-24T14:15:22Z",
      "tenantId": "string"
    }
  },
  "copyBlockedStatistics": {
    "total": 0,
    "destinationStorageInformation": {
      "id": "string",
      "name": "string",
      "groupId": "string",
      "clientId": "string",
      "accountId": "string",
      "source": "string",
      "storageFunctionality": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "workflowId": "string",
      "lastUpdated": "2019-08-24T14:15:22Z",
      "tenantId": "string"
    }
  },
  "deleteAllowedStatistics": {
    "total": 0
  },
  "deleteBlockedStatistics": {
    "total": 0
  },
  "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
}
```

Properties

Name	Type	Required	Restrictions	Description
taggingStatistics	TaggingStatistics¦null	false	none	none
keepAllowedStatistics	KeepStatistics¦null	false	none	none
keepBlockedStatistics	KeepStatistics¦null	false	none	none
keepSanitizedStatistics	KeepStatistics¦null	false	none	none
movedAllowedStatistics	MovedStatistics¦null	false	none	none
movedSanitizedStatistics	MovedStatistics¦null	false	none	none
movedBlockedStatistics	MovedStatistics¦null	false	none	none
copyAllowedStatistics	MovedStatistics¦null	false	none	none
copyBlockedStatistics	MovedStatistics¦null	false	none	none
deleteAllowedStatistics	DeleteStatistics¦null	false	none	none
deleteBlockedStatistics	DeleteStatistics¦null	false	none	none
lastCalculatedStatisticsTime	string(date-time)¦null	false	none	none

TaggingStatistics

```
{
  "total": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
total	integer(int64)	false	none	none

KeepStatistics

```
{
  "total": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
total	integer(int64)	false	none	none

MovedStatistics


```
{
  "total": 0,
  "destinationStorageInformation": {
    "id": "string",
    "name": "string",
    "groupId": "string",
    "clientId": "string",
    "accountId": "string",
    "source": "string",
    "storageFunctionality": "string",
    "storageType": {
      "name": "string",
      "categoryType": 0,
      "protocolType": 0,
      "vendorType": 0
    },
    "storageStatus": {
      "message": "string",
      "statusType": 0
    },
    "workflowId": "string",
    "lastUpdated": "2019-08-24T14:15:22Z",
    "tenantId": "string"
  }
}
```


Properties

Name	Type	Required	Restrictions	Description
total	integer(int64)	false	none	none
destinationStorageInformation	NewStorage	false	none	none

DeleteStatistics

```
{
  "total": 0
}
```


Properties

Name	Type	Required	Restrictions	Description
total	integer(int64)	false	none	none

FileVersion

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "id": "string",
  "scanId": "string",
  "name": "string",
  "path": "string",
  "hash": "string",
  "fileOwner": {
    "userId": "string",
    "fullName": "string",
    "userName": "string",
    "emailAddress": "string"
  },
  "size": 0,
  "fileTypeCategory": 0,
  "extension": "string",
  "storageType": {
    "name": "string",
    "categoryType": 0,
    "protocolType": 0,
    "vendorType": 0
  },
  "storageName": "string",
  "storageLocation": "string",
  "scanName": "string",
  "progress": {
    "upload": 0,
    "scan": 0
  },
  "scanResult": {
    "extractedFiles": {
      "dataId": "string",
      "scanResultI": 0,
      "detectedBy": 0,
      "filesExtractedCount": 0
    },
    "scanResults": {
      "scanDetails": [
        {
          "engineName": "string",
          "threatFound": "string",
          "scanResultI": 0,
          "defTime": "string",
          "scanTime": 0.1,
          "engId": "string",
          "waitTime": 0.1
```

```
],
      "dataId": "string",
      "scanAllResultI": 0,
      "scanAllResultA": "string",
      "startTime": "string",
      "totalTime": 0.1,
      "totalAvs": 0,
      "totalDetections": 0,
      "progressPercentage": 0
    },
    "fileInfo": {
      "fileSize": 0,
      "uploadTimestamp": "string",
      "md5": "string",
      "sha1": "string",
      "sha256": "string",
      "fileTypeCategory": "string",
      "fileTypeDescription": "string",
      "fileTypeExtension": "string",
      "displayName": "string",
      "fileTypeId": "string"
    },
    "fileScanInfo": {
      "finalVerdict": {
        "blocked": true,
        "verdict": "string",
        "confidence": 0.1,
        "threatLevel": 0.1
      },
      "overviewReport": {
        "analysisState": "string",
        "signalGroups": [
          {
            "description": "string",
            "verdict": "string",
            "signals": [
              {
                "originType": "string",
                "signalReadable": "string",
                "strength": 0.1
              }
            ]
          }
        ]
      },
      "errorCategory": "string",
      "errorCode": 0,
      "errorDetails": "string"
    },
    "processInfo": {
      "postProcessing": {
        "actionsRan": "string",
        "actionsFailed": "string",
```

```
"copyMoveDestination": "string",
        "convertedDestination": "string",
        "sanitizationDetails": {
          "failureCategory": "string",
          "details": "string"
        }
      },
      "progressPercentage": 0,
      "userAgent": "string",
      "profile": "string",
      "result": "string",
      "blockedReason": "string",
      "fileTypeSkippedScan": true,
      "processingTime": 0.1,
      "queueTime": 0.1,
      "resultI": 0
    },
    "dataId": "string",
    "dlpInfo": {
      "hits": null,
      "verdict": 0
    },
    "sbomInfo": {
      "finalVerdict": {
        "blocked": true,
        "licensingPackageCounts": {
          "allowed": 0,
          "blocked": 0,
          "unknown": 0,
          "unsupported": 0
        },
        "severity": 0,
        "totalPackageCount": 0,
        "verdict": "string",
        "vulnerabilityCounts": {
          "critical": 0,
          "high": 0,
          "medium": 0,
          "low": 0,
          "unknown": 0
        }
      },
      "overviewReport": {
        "isPartialReport": true,
        "osInfo": {
          "osName": "string",
          "osVersion": "string"
        },
        "scanMode": "string",
        "steps": [
          {
            "command": "string",
            "packages": [
```

```
"authors": "string",
                "releaseDate": "2019-08-24T14:15:22Z",
                "ecosystem": "string",
                "group": "string",
                "archive": true,
                "executable": true,
                "uid": "string",
                "licenses": {
                  "allowed": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ],
                  "blocked": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ]
                },
                "packageName": "string",
                "packageVersion": "string",
                "hashes": [
                  {
                    "sha256": "string",
                    "fileName": "string"
                  }
                ],
```

```
"critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
                },
                "vulnerabilities": [
                  {
                    "id": "string",
                    "severity": 0,
                    "source": "string",
                    "fixedVersions": "string",
                    "cwes": [
                      "string"
                    ]
                  }
                ],
                "arch": "string",
                "osName": "string",
                "osVersion": "string",
                "dependencies": [
                  {
                    "ecosystem": "string",
                    "group": "string",
                    "packageName": "string",
                    "packageVersion": "string",
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    }
                  }
                ]
              }
            ],
            "licenseStatus": "string",
            "licensingPackageCounts": {
              "allowed": 0,
              "blocked": 0,
              "unknown": 0,
```

```
},
            "step": 0,
            "vulnerabilityCount": 0,
            "sha256": "string",
            "severity": 0,
            "vulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            }
          }
        ],
        "packages": [
          {
            "authors": "string",
            "releaseDate": "2019-08-24T14:15:22Z",
            "ecosystem": "string",
            "group": "string",
            "archive": true,
            "executable": true,
            "uid": "string",
            "licenses": {
              "allowed": [
                {
                  "id": "string",
                  "url": "string",
                  "category": "string",
                  "usageRestriction": {
                    "permissions": [
                      "string"
                    ],
                    "conditions": [
                      "string"
                    ],
                    "limitations": [
                      "string"
                    ]
                  }
                }
              ],
              "blocked": [
                {
                  "id": "string",
                  "url": "string",
                  "category": "string",
                  "usageRestriction": {
                    "permissions": [
                      "string"
                    ],
                    "conditions": [
                      "string"
```

```
"limitations": [
                      "string"
                    ]
                  }
                }
              ]
            },
            "packageName": "string",
            "packageVersion": "string",
            "target": "string",
            "hashes": [
              {
                "sha256": "string",
                "fileName": "string"
              }
            ],
            "vulnerabilities": [
              {
                "id": "string",
                "severity": 0,
                "source": "string",
                "fixedVersions": "string",
                "cwes": [
                  "string"
                ]
              }
            ],
            "vulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            },
            "dependencies": [
              {
                "ecosystem": "string",
                "group": "string",
                "packageName": "string",
                "packageVersion": "string",
                "vulnerabilities": [
                  {
                    "id": "string",
                    "severity": 0,
                    "source": "string",
                    "fixedVersions": "string",
                    "cwes": [
                      "string"
                    ]
                  }
                ],
                "vulnerabilityCounts": {
                  "critical": 0,
```

```
"medium": 0,
                  "low": 0,
                  "unknown": 0
                }
              }
            ]
          }
        ],
        "dependenciesPackageCount": 0,
        "dependenciesVulnerabilityTotal": 0,
        "dependenciesVulnerabilityCounts": {
          "critical": 0,
          "high": 0,
          "medium": 0,
          "low": 0,
          "unknown": 0
        }
      },
      "resultTemplateHash": "string"
    },
    "vulnerabilityInfo": {
      "vulnerabilityInfoResult": {
        "code": 0,
        "hash": "string",
        "method": 0,
        "timeStamp": "string",
        "timing": 0,
        "detectedProduct": {
          "hasVulnerability": true,
          "isCurrent": true,
          "product": {
            "id": 0,
            "name": "string"
          },
          "remediationLink": "string",
          "severity": "string",
          "sigName": "string",
          "signature": 0,
          "vendor": {
            "id": 0,
            "name": "string"
          },
          "version": "string",
          "versionData": {
            "countBehind": 0,
            "feedId": 0,
            "version": "string"
          },
          "vulnerabilities": [
            {
              "description": "string",
              "details": {
                "cpe": "string",
```

```
"cvss": {
                  "accessComplexity": "string",
                  "accessVector": "string",
                  "authentication": "string",
                  "availabilityImpact": "string",
                  "confidentialityImpact": "string",
                  "generatedOnEpoch": "string",
                  "integrityImpact": "string",
                  "score": "string",
                  "source": "string"
                },
                "cvss3": {
                  "attackComplexity": "string",
                  "attackVector": "string",
                  "availabilityImpact": "string",
                  "baseScore": "string",
                  "baseSeverity": "string",
                  "confidentialityImpact": "string",
                  "exploitabilityScore": "string",
                  "impactScore": "string",
                  "integrityImpact": "string",
                  "privilegesRequired": "string",
                  "scope": "string",
                  "userInteraction": "string",
                  "vectorString": "string"
                },
                "cwe": "string",
                "lastModifiedEpoch": "string",
                "publishedEpoch": "string",
                "references": [
                  "string"
                ]
              },
              "severity": "string",
              "severityIndex": 0,
              "staticId": 0
            }
          ]
        }
      }
    },
    "sessionCookie": "string",
    "fileTypeInfo": {
      "fileInfoExtended": {
        "extensions": "string",
        "groupId": "string",
        "typeId": "string"
      }
    },
    "scanInstanceUrl": "string",
    "scanWorkflowSnapshotId": "string",
    "coreEngines": [
      {
```

```
"engineName": "string",
        "engineVersion": "string",
        "dbVersion": "string",
        "definitionTime": "string"
      }
    ]
  },
  "processingState": 0,
  "exception": {
    "exceptionDetails": "string",
    "exceptionMessage": "string"
  },
  "created": "2019-08-24T14:15:22Z",
  "lastModified": "2019-08-24T14:15:22Z",
  "discoveryStarted": "2019-08-24T14:15:22Z",
  "fileProcessingDetails": {
    "streamingStartedTime": "2019-08-24T14:15:22Z",
    "streamingCompletedTime": "2019-08-24T14:15:22Z",
    "streamingTimeSpanMs": 0,
    "scanCompletedTime": "2019-08-24T14:15:22Z",
    "processingTimeSpanMs": 0
  },
  "coreEngines": [
    {
      "engineId": "string",
      "engineName": "string",
      "engineVersion": "string",
      "dbVersion": "string",
      "definitionTime": "string"
    }
  ],
  "remediation": {
    "isTagged": true,
    "isDeleted": true,
    "isMoved": true,
    "isCopied": true,
    "deepCdr": {
      "sanitizationDetails": {
        "failureCategory": "string",
        "details": "string"
      },
      "cdrObjectState": 0,
      "fileWasReplaced": true
    },
    "postActionFailedInformation": [
      {
        "objectActionType": 0,
        "exception": {
          "exceptionDetails": "string",
          "exceptionMessage": "string"
        }
      }
    ],
```

```
{
        "objectActionType": 0,
        "objectActionState": 0,
        "cdrObjectState": 0,
        "exceptions": [
          "string"
        ],
        "id": "string",
        "fileId": "string",
        "fileScanCompletedTime": "2019-08-24T14:15:22Z",
        "remediationId": "string",
        "tenantId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z"
      }
    ]
  },
  "metaDefenderCoreUrl": "string",
  "discoveryStartedDaySec": 0,
  "objectStorageMetadata": "string",
  "scanWorkflowSnapshotId": "string",
  "scanWorkflowSnapshot": {
    "id": "string",
    "scanId": "string",
    "workflow": {
      "id": "string",
      "scanPoolId": "string",
      "scanProcessFlow": 0,
      "isIdentityScanningEnabled": true,
      "scanIntervalDays": 0,
      "rules": [
        "string"
      ],
      "type": 0,
      "workflowSupportedTechnologies": {
        "dlp": {
          "status": 0
        },
        "deepCdr": {
          "status": 0
        },
        "sandBox": {
          "status": 0
        },
        "metascan": {
          "status": 0
        }
      },
      "name": "string",
      "userAgent": "string",
      "scanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
```

```
"scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "failOverScanPoolId": "string",
      "failOverScanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "filters": {
        "id": "string",
        "minimumSize": 0,
        "maximumSize": 0,
        "minimumDateTime": "2019-08-24T14:15:22Z",
        "maximumDateTime": "2019-08-24T14:15:22Z"
      },
      "fileDefinition": {
        "blockingStrategy": 0,
        "configurableBlockingStrategy": {
          "shouldTreatFilesWithSensitiveDataAsBlocked": true,
          "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
        }
      },
      "remediations": [
        {
          "id": "string",
          "workflowId": "string",
          "isEnabled": true,
          "configuration": "string",
          "type": 0,
          "handlingType": 0
        }
      ],
      "tenantId": "string"
```

```
"startTimestamp": 0,
    "stopTimestamp": 0,
    "scanPool": {
      "id": "string",
      "name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    },
    "scanInstances": [
      {
        "id": "string",
        "scanPoolId": "string",
        "url": "string",
        "apiKey": "string",
        "timeout": "string",
        "scanInstanceType": 0
      }
    ],
    "failOverScanPool": {
      "id": "string",
      "name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    },
    "failOverScanInstances": [
      {
        "id": "string",
        "scanPoolId": "string",
        "url": "string",
        "apiKey": "string",
        "timeout": "string",
        "scanInstanceType": 0
      }
```

```
"storage": {
      "id": "string",
      "name": "string",
      "groupId": "string",
      "clientId": "string",
      "accountId": "string",
      "source": "string",
      "storageFunctionality": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "workflowId": "string",
      "lastUpdated": "2019-08-24T14:15:22Z",
      "tenantId": "string"
    },
    "workflowStatistics": {
      "taggingStatistics": {
        "total": 0
      },
      "keepAllowedStatistics": {
        "total": 0
      },
      "keepBlockedStatistics": {
        "total": 0
      },
      "keepSanitizedStatistics": {
        "total": 0
      },
      "movedAllowedStatistics": {
        "total": 0,
        "destinationStorageInformation": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
```

```
},
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        }
      },
      "movedSanitizedStatistics": {
        "total": 0,
        "destinationStorageInformation": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        }
      },
      "movedBlockedStatistics": {
        "total": 0,
        "destinationStorageInformation": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
```

```
}
      },
      "copyAllowedStatistics": {
        "total": 0,
        "destinationStorageInformation": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        }
      },
      "copyBlockedStatistics": {
        "total": 0,
        "destinationStorageInformation": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        }
      },
      "deleteAllowedStatistics": {
```

```
},
      "deleteBlockedStatistics": {
        "total": 0
      },
      "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
    },
    "account": {
      "id": "string",
      "name": "string",
      "description": "string",
      "credentials": "string",
      "protocolType": 0,
      "vendorType": 0,
      "autoDiscovery": true,
      "autodiscoveryDestinationGroupId": "string"
    },
    "realTimeStartDate": "2019-08-24T14:15:22Z"
  },
  "versions": [
    {
      "result": 0,
      "responseKey": "string",
      "responseMessage": "string",
      "responseMessageParams": {
        "property1": "string",
        "property2": "string"
      },
      "id": "string",
      "scanId": "string",
      "name": "string",
      "path": "string",
      "hash": "string",
      "fileOwner": {
        "userId": "string",
        "fullName": "string",
        "userName": "string",
        "emailAddress": "string"
      },
      "size": 0,
      "fileTypeCategory": 0,
      "extension": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageName": "string",
      "storageLocation": "string",
      "scanName": "string",
      "progress": {
        "upload": 0,
        "scan": 0
```

```
"scanResult": {
        "extractedFiles": {
          "dataId": "string",
          "scanResultI": 0,
          "detectedBy": 0,
          "filesExtractedCount": 0
        },
        "scanResults": {
          "scanDetails": [
            {
              "engineName": "string",
              "threatFound": "string",
              "scanResultI": 0,
              "defTime": "string",
              "scanTime": 0.1,
              "engId": "string",
              "waitTime": 0.1
            }
          ],
          "dataId": "string",
          "scanAllResultI": 0,
          "scanAllResultA": "string",
          "startTime": "string",
          "totalTime": 0.1,
          "totalAvs": 0,
          "totalDetections": 0,
          "progressPercentage": 0
        },
        "fileInfo": {
          "fileSize": 0,
          "uploadTimestamp": "string",
          "md5": "string",
          "sha1": "string",
          "sha256": "string",
          "fileTypeCategory": "string",
          "fileTypeDescription": "string",
          "fileTypeExtension": "string",
          "displayName": "string",
          "fileTypeId": "string"
        },
        "fileScanInfo": {
          "finalVerdict": {
            "blocked": true,
            "verdict": "string",
            "confidence": 0.1,
            "threatLevel": 0.1
          },
          "overviewReport": {
            "analysisState": "string",
            "signalGroups": [
              {
                "description": "string",
                "verdict": "string",
```

```
{
                    "originType": "string",
                    "signalReadable": "string",
                    "strength": 0.1
                  }
                ]
              }
            ]
          },
          "errorCategory": "string",
          "errorCode": 0,
          "errorDetails": "string"
        },
        "processInfo": {
          "postProcessing": {
            "actionsRan": "string",
            "actionsFailed": "string",
            "convertedTo": "string",
            "copyMoveDestination": "string",
            "convertedDestination": "string",
            "sanitizationDetails": {
              "failureCategory": "string",
              "details": "string"
            }
          },
          "progressPercentage": 0,
          "userAgent": "string",
          "profile": "string",
          "result": "string",
          "blockedReason": "string",
          "fileTypeSkippedScan": true,
          "processingTime": 0.1,
          "queueTime": 0.1,
          "resultI": 0
        },
        "dataId": "string",
        "dlpInfo": {
          "hits": null,
          "verdict": 0
        },
        "sbomInfo": {
          "finalVerdict": {
            "blocked": true,
            "licensingPackageCounts": {
              "allowed": 0,
              "blocked": 0,
              "unknown": 0,
              "unsupported": 0
            },
            "severity": 0,
            "totalPackageCount": 0,
            "verdict": "string",
            "vulnerabilityCounts": {
```

```
"high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            }
          },
          "overviewReport": {
            "isPartialReport": true,
            "osInfo": {
              "osName": "string",
              "osVersion": "string"
            },
            "scanMode": "string",
            "steps": [
              {
                "command": "string",
                "packages": [
                  {
                    "authors": "string",
                    "releaseDate": "2019-08-24T14:15:22Z",
                    "ecosystem": "string",
                    "group": "string",
                    "archive": true,
                    "executable": true,
                    "uid": "string",
                    "licenses": {
                      "allowed": [
                        {
                          "id": "string",
                          "url": "string",
                          "category": "string",
                          "usageRestriction": {
                            "permissions": [
                              null
                            ],
                            "conditions": [
                              null
                            ],
                            "limitations": [
                              null
                            ]
                          }
                        }
                      ],
                      "blocked": [
                        {
                          "id": "string",
                          "url": "string",
                          "category": "string",
                          "usageRestriction": {
                            "permissions": [
                              null
                            ],
```

```
null
                            ],
                            "limitations": [
                              null
                            ]
                          }
                        }
                      ]
                    },
                    "packageName": "string",
                    "packageVersion": "string",
                    "hashes": [
                      {
                        "sha256": "string",
                        "fileName": "string"
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    },
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "arch": "string",
                    "osName": "string",
                    "osVersion": "string",
                    "dependencies": [
                      {
                        "ecosystem": "string",
                        "group": "string",
                        "packageName": "string",
                        "packageVersion": "string",
                        "vulnerabilities": [
                          {
                            "id": "string",
                            "severity": 0,
                            "source": "string",
                            "fixedVersions": "string",
                            "cwes": [
                              null
                            ]
```

```
],
                        "vulnerabilityCounts": {
                          "critical": 0,
                          "high": 0,
                          "medium": 0,
                          "low": 0,
                          "unknown": 0
                        }
                      }
                    ]
                  }
                ],
                "licenseStatus": "string",
                "licensingPackageCounts": {
                  "allowed": 0,
                  "blocked": 0,
                  "unknown": 0,
                  "unsupported": 0
                },
                "step": 0,
                "vulnerabilityCount": 0,
                "sha256": "string",
                "severity": 0,
                "vulnerabilityCounts": {
                  "critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
                }
              }
            ],
            "packages": [
              {
                "authors": "string",
                "releaseDate": "2019-08-24T14:15:22Z",
                "ecosystem": "string",
                "group": "string",
                "archive": true,
                "executable": true,
                "uid": "string",
                "licenses": {
                  "allowed": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
```

```
"limitations": [
                        "string"
                      ]
                    }
                  }
                ],
                "blocked": [
                  {
                    "id": "string",
                    "url": "string",
                    "category": "string",
                    "usageRestriction": {
                      "permissions": [
                        "string"
                      ],
                      "conditions": [
                        "string"
                      ],
                      "limitations": [
                        "string"
                      ]
                    }
                  }
                ]
              },
              "packageName": "string",
              "packageVersion": "string",
              "target": "string",
              "hashes": [
                {
                  "sha256": "string",
                  "fileName": "string"
                }
              ],
              "vulnerabilities": [
                {
                  "id": "string",
                  "severity": 0,
                  "source": "string",
                  "fixedVersions": "string",
                  "cwes": [
                    "string"
                  ]
                }
              ],
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              },
              "dependencies": [
```

```
"ecosystem": "string",
                    "group": "string",
                    "packageName": "string",
                    "packageVersion": "string",
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    }
                  }
                ]
              }
            ],
            "dependenciesPackageCount": 0,
            "dependenciesVulnerabilityTotal": 0,
            "dependenciesVulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            }
          },
          "resultTemplateHash": "string"
        },
        "vulnerabilityInfo": {
          "vulnerabilityInfoResult": {
            "code": 0,
            "hash": "string",
            "method": 0,
            "timeStamp": "string",
            "timing": 0,
            "detectedProduct": {
              "hasVulnerability": true,
              "isCurrent": true,
              "product": {
                "id": 0,
                "name": "string"
              },
              "remediationLink": "string",
```

```
"sigName": "string",
            "signature": 0,
            "vendor": {
              "id": 0,
              "name": "string"
            },
            "version": "string",
            "versionData": {
              "countBehind": 0,
              "feedId": 0,
              "version": "string"
            },
            "vulnerabilities": [
              {
                "description": "string",
                "details": {
                  "cpe": "string",
                  "cve": "string",
                  "cvss": {
                    "accessComplexity": "string",
                    "accessVector": "string",
                    "authentication": "string",
                    "availabilityImpact": "string",
                    "confidentialityImpact": "string",
                    "generatedOnEpoch": "string",
                    "integrityImpact": "string",
                    "score": "string",
                    "source": "string"
                  },
                  "cvss3": {
                    "attackComplexity": "string",
                    "attackVector": "string",
                    "availabilityImpact": "string",
                    "baseScore": "string",
                    "baseSeverity": "string",
                    "confidentialityImpact": "string",
                    "exploitabilityScore": "string",
                    "impactScore": "string",
                    "integrityImpact": "string",
                    "privilegesRequired": "string",
                    "scope": "string",
                    "userInteraction": "string",
                    "vectorString": "string"
                  },
                  "cwe": "string",
                  "lastModifiedEpoch": "string",
                  "publishedEpoch": "string",
                  "references": [
                    "string"
                  ]
                },
                "severity": "string",
                "severityIndex": 0,
```

```
}
              ]
            }
          }
        },
        "sessionCookie": "string",
        "fileTypeInfo": {
          "fileInfoExtended": {
            "extensions": "string",
            "groupId": "string",
            "typeId": "string"
          }
        },
        "scanInstanceUrl": "string",
        "scanWorkflowSnapshotId": "string",
        "coreEngines": [
          {
            "engineId": "string",
            "engineName": "string",
            "engineVersion": "string",
            "dbVersion": "string",
            "definitionTime": "string"
          }
        ]
      },
      "processingState": 0,
      "exception": {
        "exceptionDetails": "string",
        "exceptionMessage": "string"
      },
      "created": "2019-08-24T14:15:22Z",
      "lastModified": "2019-08-24T14:15:22Z",
      "discoveryStarted": "2019-08-24T14:15:22Z",
      "fileProcessingDetails": {
        "streamingStartedTime": "2019-08-24T14:15:22Z",
        "streamingCompletedTime": "2019-08-24T14:15:22Z",
        "streamingTimeSpanMs": 0,
        "scanCompletedTime": "2019-08-24T14:15:22Z",
        "processingTimeSpanMs": 0
      },
      "coreEngines": [
        {
          "engineId": "string",
          "engineName": "string",
          "engineVersion": "string",
          "dbVersion": "string",
          "definitionTime": "string"
        }
      ],
      "remediation": {
        "isTagged": true,
        "isDeleted": true,
        "isMoved": true,
```

```
"deepCdr": {
          "sanitizationDetails": {
            "failureCategory": "string",
            "details": "string"
          },
          "cdrObjectState": 0,
          "fileWasReplaced": true
        },
        "postActionFailedInformation": [
          {
            "objectActionType": 0,
            "exception": {
              "exceptionDetails": "string",
              "exceptionMessage": "string"
            }
          }
        ],
        "objectRemediationActions": [
          {
            "objectActionType": 0,
            "objectActionState": 0,
            "cdrObjectState": 0,
            "exceptions": [
              "string"
            ],
            "id": "string",
            "fileId": "string",
            "fileScanCompletedTime": "2019-08-24T14:15:22Z",
            "remediationId": "string",
            "tenantId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z"
          }
        ]
      },
      "metaDefenderCoreUrl": "string",
      "discoveryStartedDaySec": 0,
      "objectStorageMetadata": "string",
      "scanWorkflowSnapshotId": "string",
      "scanWorkflowSnapshot": {
        "id": "string",
        "scanId": "string",
        "workflow": {
          "id": "string",
          "scanPoolId": "string",
          "scanProcessFlow": 0,
          "isIdentityScanningEnabled": true,
          "scanIntervalDays": 0,
          "rules": [
            "string"
          ],
          "type": 0,
          "workflowSupportedTechnologies": {
            "dlp": {
```

```
},
            "deepCdr": {
              "status": 0
            },
            "sandBox": {
              "status": 0
            },
            "metascan": {
              "status": 0
            }
          },
          "name": "string",
          "userAgent": "string",
          "scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanPoolId": "string",
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "filters": {
            "id": "string",
            "minimumSize": 0,
            "maximumSize": 0,
            "minimumDateTime": "2019-08-24T14:15:22Z",
            "maximumDateTime": "2019-08-24T14:15:22Z"
          },
```

```
"blockingStrategy": 0,
            "configurableBlockingStrategy": {
              "shouldTreatFilesWithSensitiveDataAsBlocked": true,
              "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
            }
          },
          "remediations": [
            {
              "id": "string",
              "workflowId": "string",
              "isEnabled": true,
              "configuration": "string",
              "type": 0,
              "handlingType": 0
            }
          ],
          "tenantId": "string"
        },
        "startTimestamp": 0,
        "stopTimestamp": 0,
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
```

```
"scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "storage": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        },
        "workflowStatistics": {
          "taggingStatistics": {
            "total": 0
          },
          "keepAllowedStatistics": {
            "total": 0
          },
          "keepBlockedStatistics": {
            "total": 0
          },
          "keepSanitizedStatistics": {
            "total": 0
          },
```

```
"total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedSanitizedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
```

```
"groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
```

```
"storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "deleteAllowedStatistics": {
            "total": 0
          },
          "deleteBlockedStatistics": {
            "total": 0
          },
          "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
        },
        "account": {
          "id": "string",
          "name": "string",
          "description": "string",
          "credentials": "string",
          "protocolType": 0,
          "vendorType": 0,
          "autoDiscovery": true,
          "autodiscoveryDestinationGroupId": "string"
        },
        "realTimeStartDate": "2019-08-24T14:15:22Z"
      },
      "versions": [],
      "hasBlockedVersions": true,
      "versionsCount": 0,
      "objectType": 0,
      "skippedScan": true,
      "$type": "string",
      "url": "string",
      "fileId": "string"
    }
  ],
  "hasBlockedVersions": true,
  "versionsCount": 0,
  "objectType": 0,
  "skippedScan": true,
  "$type": "string",
  "url": "string",
  "fileId": "string"
```

```
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	File	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» url	string¦null	false	none	none
» fileId	string¦null	false	none	none

ObjectType

```
0
```

- 0: File
- 1: FileVersion

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: File - 1: FileVersion

Enumerated Values

Property	Value
anonymous	0
anonymous	1

FileBase


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "id": "string",
  "scanId": "string",
  "name": "string",
  "path": "string",
  "hash": "string",
  "fileOwner": {
    "userId": "string",
    "fullName": "string",
    "userName": "string",
    "emailAddress": "string"
  },
  "size": 0,
  "fileTypeCategory": 0,
  "extension": "string"
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» id	string	false	none	none
» scanId	string	false	none	none
» name	string	false	none	none
» path	string	false	none	none
» hash	string	false	none	none
» fileOwner	FileOwner	false	none	none
» size	integer(int64)	false	none	none
» fileTypeCategory	FileTypeCategory¦null	false	none	- 0: Other - 1: Archive - 2: Application - 3: Document - 4: EncryptedDocument - 5: Executable - 6: Graphical - 7: DiskImage - 8: AudioVideo - 9: OpenSslEncrypted - 10: Pdf - 11: Text - 12: MailMessage
» extension	string¦null	false	none	none

FileTypeCategory

```
0
```

- 0: Other
- 1: Archive
- 2: Application
- 3: Document
- 4: EncryptedDocument
- 5: Executable
- 6: Graphical
- 7: DiskImage

- 8: AudioVideo
- 9: OpenSslEncrypted
- 10: Pdf
- 11: Text
- 12: MailMessage

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Other - 1: Archive - 2: Application - 3: Document - 4: EncryptedDocument - 5: Executable - 6: Graphical - 7: DiskImage - 8: AudioVideo - 9: OpenSslEncrypted - 10: Pdf - 11: Text - 12: MailMessage

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4
anonymous	5
anonymous	6
anonymous	7
anonymous	8
anonymous	9
anonymous	10
anonymous	11
anonymous	12

GetArchiveScanResultResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "archiveScanResults": [
    {
      "extractedFiles": {
        "dataId": "string",
        "scanResultI": 0,
        "detectedBy": 0,
        "filesExtractedCount": 0
      },
      "scanResults": {
        "scanDetails": [
          {
            "engineName": "string",
            "threatFound": "string",
            "scanResultI": 0,
            "defTime": "string",
            "scanTime": 0.1,
            "engId": "string",
            "waitTime": 0.1
          }
        ],
        "dataId": "string",
        "scanAllResultI": 0,
        "scanAllResultA": "string",
        "startTime": "string",
        "totalTime": 0.1,
        "totalAvs": 0,
        "totalDetections": 0,
        "progressPercentage": 0
      },
      "fileInfo": {
        "fileSize": 0,
        "uploadTimestamp": "string",
        "md5": "string",
        "sha1": "string",
        "sha256": "string",
        "fileTypeCategory": "string",
        "fileTypeDescription": "string",
        "fileTypeExtension": "string",
        "displayName": "string",
        "fileTypeId": "string"
      },
      "fileScanInfo": {
        "finalVerdict": {
          "blocked": true,
```

```
"confidence": 0.1,
          "threatLevel": 0.1
        },
        "overviewReport": {
          "analysisState": "string",
          "signalGroups": [
            {
              "description": "string",
              "verdict": "string",
              "signals": [
                {
                  "originType": "string",
                  "signalReadable": "string",
                  "strength": 0.1
                }
              ]
            }
          ]
        },
        "errorCategory": "string",
        "errorCode": 0,
        "errorDetails": "string"
      },
      "processInfo": {
        "postProcessing": {
          "actionsRan": "string",
          "actionsFailed": "string",
          "convertedTo": "string",
          "copyMoveDestination": "string",
          "convertedDestination": "string",
          "sanitizationDetails": {
            "failureCategory": "string",
            "details": "string"
          }
        },
        "progressPercentage": 0,
        "userAgent": "string",
        "profile": "string",
        "result": "string",
        "blockedReason": "string",
        "fileTypeSkippedScan": true,
        "processingTime": 0.1,
        "queueTime": 0.1,
        "resultI": 0
      },
      "dataId": "string",
      "dlpInfo": {
        "hits": null,
        "verdict": 0
      },
      "sbomInfo": {
        "finalVerdict": {
          "blocked": true,
```

```
"allowed": 0,
            "blocked": 0,
            "unknown": 0,
            "unsupported": 0
          },
          "severity": 0,
          "totalPackageCount": 0,
          "verdict": "string",
          "vulnerabilityCounts": {
            "critical": 0,
            "high": 0,
            "medium": 0,
            "low": 0,
            "unknown": 0
          }
        },
        "overviewReport": {
          "isPartialReport": true,
          "osInfo": {
            "osName": "string",
            "osVersion": "string"
          },
          "scanMode": "string",
          "steps": [
            {
              "command": "string",
              "packages": [
                {
                  "authors": "string",
                  "releaseDate": "2019-08-24T14:15:22Z",
                  "ecosystem": "string",
                  "group": "string",
                  "archive": true,
                  "executable": true,
                  "uid": "string",
                  "licenses": {
                    "allowed": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            "string"
                          ],
                          "conditions": [
                            "string"
                          ],
                          "limitations": [
                            "string"
                          ]
                        }
                      }
```

```
"blocked": [
                        {
                          "id": "string",
                          "url": "string",
                          "category": "string",
                          "usageRestriction": {
                            "permissions": [
                              "string"
                            ],
                            "conditions": [
                              "string"
                            ],
                            "limitations": [
                              "string"
                            ]
                          }
                        }
                      ]
                    },
                    "packageName": "string",
                    "packageVersion": "string",
                    "hashes": [
                      {
                        "sha256": "string",
                        "fileName": "string"
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    },
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "arch": "string",
                    "osName": "string",
                    "osVersion": "string",
                    "dependencies": [
                      {
                        "ecosystem": "string",
                        "group": "string",
                        "packageName": "string",
```

```
"vulnerabilities": [
                        {
                          "id": "string",
                          "severity": 0,
                          "source": "string",
                          "fixedVersions": "string",
                          "cwes": [
                            "string"
                          ]
                        }
                      ],
                      "vulnerabilityCounts": {
                        "critical": 0,
                        "high": 0,
                        "medium": 0,
                        "low": 0,
                        "unknown": 0
                      }
                    }
                  ]
                }
              ],
              "licenseStatus": "string",
              "licensingPackageCounts": {
                "allowed": 0,
                "blocked": 0,
                "unknown": 0,
                "unsupported": 0
              },
              "step": 0,
              "vulnerabilityCount": 0,
              "sha256": "string",
              "severity": 0,
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              }
            }
          ],
          "packages": [
            {
              "authors": "string",
              "releaseDate": "2019-08-24T14:15:22Z",
              "ecosystem": "string",
              "group": "string",
              "archive": true,
              "executable": true,
              "uid": "string",
              "licenses": {
                "allowed": [
```

```
"id": "string",
                    "url": "string",
                    "category": "string",
                    "usageRestriction": {
                      "permissions": [
                        "string"
                      ],
                      "conditions": [
                        "string"
                      ],
                      "limitations": [
                        "string"
                      ]
                    }
                  }
                ],
                "blocked": [
                  {
                    "id": "string",
                    "url": "string",
                    "category": "string",
                    "usageRestriction": {
                      "permissions": [
                        "string"
                      ],
                      "conditions": [
                        "string"
                      ],
                      "limitations": [
                        "string"
                      ]
                    }
                  }
                ]
              },
              "packageName": "string",
              "packageVersion": "string",
              "target": "string",
              "hashes": [
                {
                  "sha256": "string",
                  "fileName": "string"
                }
              ],
              "vulnerabilities": [
                {
                  "id": "string",
                  "severity": 0,
                  "source": "string",
                  "fixedVersions": "string",
                  "cwes": [
                    "string"
                  ]
```

```
],
              "vulnerabilityCounts": {
                "critical": 0,
                "high": 0,
                "medium": 0,
                "low": 0,
                "unknown": 0
              },
              "dependencies": [
                {
                  "ecosystem": "string",
                  "group": "string",
                  "packageName": "string",
                  "packageVersion": "string",
                  "vulnerabilities": [
                    {
                      "id": "string",
                      "severity": 0,
                      "source": "string",
                      "fixedVersions": "string",
                      "cwes": [
                        "string"
                      ]
                    }
                  ],
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  }
                }
              ]
            }
          ],
          "dependenciesPackageCount": 0,
          "dependenciesVulnerabilityTotal": 0,
          "dependenciesVulnerabilityCounts": {
            "critical": 0,
            "high": 0,
            "medium": 0,
            "low": 0,
            "unknown": 0
          }
        },
        "resultTemplateHash": "string"
      },
      "vulnerabilityInfo": {
        "vulnerabilityInfoResult": {
          "code": 0,
          "hash": "string",
          "method": 0,
```

```
"timing": 0,
          "detectedProduct": {
            "hasVulnerability": true,
            "isCurrent": true,
            "product": {
              "id": 0,
              "name": "string"
            },
            "remediationLink": "string",
            "severity": "string",
            "sigName": "string",
            "signature": 0,
            "vendor": {
              "id": 0,
              "name": "string"
            },
            "version": "string",
            "versionData": {
              "countBehind": 0,
              "feedId": 0,
              "version": "string"
            },
            "vulnerabilities": [
              {
                "description": "string",
                "details": {
                  "cpe": "string",
                  "cve": "string",
                  "cvss": {
                    "accessComplexity": "string",
                    "accessVector": "string",
                    "authentication": "string",
                    "availabilityImpact": "string",
                    "confidentialityImpact": "string",
                    "generatedOnEpoch": "string",
                    "integrityImpact": "string",
                    "score": "string",
                    "source": "string"
                  },
                  "cvss3": {
                    "attackComplexity": "string",
                    "attackVector": "string",
                    "availabilityImpact": "string",
                    "baseScore": "string",
                    "baseSeverity": "string",
                    "confidentialityImpact": "string",
                    "exploitabilityScore": "string",
                    "impactScore": "string",
                    "integrityImpact": "string",
                    "privilegesRequired": "string",
                    "scope": "string",
                    "userInteraction": "string",
                    "vectorString": "string"
```

```
"cwe": "string",
                  "lastModifiedEpoch": "string",
                  "publishedEpoch": "string",
                  "references": [
                    "string"
                  ]
                },
                "severity": "string",
                "severityIndex": 0,
                "staticId": 0
              }
            ]
          }
        }
      },
      "sessionCookie": "string",
      "fileTypeInfo": {
        "fileInfoExtended": {
          "extensions": "string",
          "groupId": "string",
          "typeId": "string"
        }
      },
      "scanInstanceUrl": "string",
      "scanWorkflowSnapshotId": "string",
      "coreEngines": [
        {
          "engineId": "string",
          "engineName": "string",
          "engineVersion": "string",
          "dbVersion": "string",
          "definitionTime": "string"
        }
      ]
    }
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» archiveScanResults	[CoreScanResult]	false	none	none

EnumerateFiles

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "result": 0,
      "responseKey": "string",
      "responseMessage": "string",
      "responseMessageParams": {
        "property1": "string",
        "property2": "string"
      },
      "id": "string",
      "scanId": "string",
      "name": "string",
      "path": "string",
      "hash": "string",
      "fileOwner": {
        "userId": "string",
        "fullName": "string",
        "userName": "string",
        "emailAddress": "string"
      },
      "size": 0,
      "fileTypeCategory": 0,
      "extension": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageName": "string",
      "storageLocation": "string",
      "scanName": "string",
      "progress": {
        "upload": 0,
        "scan": 0
      },
      "scanResult": {
        "extractedFiles": {
          "dataId": "string",
          "scanResultI": 0,
          "detectedBy": 0,
          "filesExtractedCount": 0
        },
        "scanResults": {
```

```
{
              "engineName": "string",
              "threatFound": "string",
              "scanResultI": 0,
              "defTime": "string",
              "scanTime": 0.1,
              "engId": "string",
              "waitTime": 0.1
            }
          ],
          "dataId": "string",
          "scanAllResultI": 0,
          "scanAllResultA": "string",
          "startTime": "string",
          "totalTime": 0.1,
          "totalAvs": 0,
          "totalDetections": 0,
          "progressPercentage": 0
        },
        "fileInfo": {
          "fileSize": 0,
          "uploadTimestamp": "string",
          "md5": "string",
          "sha1": "string",
          "sha256": "string",
          "fileTypeCategory": "string",
          "fileTypeDescription": "string",
          "fileTypeExtension": "string",
          "displayName": "string",
          "fileTypeId": "string"
        },
        "fileScanInfo": {
          "finalVerdict": {
            "blocked": true,
            "verdict": "string",
            "confidence": 0.1,
            "threatLevel": 0.1
          },
          "overviewReport": {
            "analysisState": "string",
            "signalGroups": [
              {
                "description": "string",
                "verdict": "string",
                "signals": [
                  {
                    "originType": "string",
                    "signalReadable": "string",
                    "strength": 0.1
                  }
                ]
              }
            ]
```

```
"errorCategory": "string",
          "errorCode": 0,
          "errorDetails": "string"
        },
        "processInfo": {
          "postProcessing": {
            "actionsRan": "string",
            "actionsFailed": "string",
            "convertedTo": "string",
            "copyMoveDestination": "string",
            "convertedDestination": "string",
            "sanitizationDetails": {
              "failureCategory": "string",
              "details": "string"
            }
          },
          "progressPercentage": 0,
          "userAgent": "string",
          "profile": "string",
          "result": "string",
          "blockedReason": "string",
          "fileTypeSkippedScan": true,
          "processingTime": 0.1,
          "queueTime": 0.1,
          "resultI": 0
        },
        "dataId": "string",
        "dlpInfo": {
          "hits": null,
          "verdict": 0
        },
        "sbomInfo": {
          "finalVerdict": {
            "blocked": true,
            "licensingPackageCounts": {
              "allowed": 0,
              "blocked": 0,
              "unknown": 0,
              "unsupported": 0
            },
            "severity": 0,
            "totalPackageCount": 0,
            "verdict": "string",
            "vulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            }
          },
          "overviewReport": {
            "isPartialReport": true,
```

```
"osName": "string",
            "osVersion": "string"
          },
          "scanMode": "string",
          "steps": [
            {
              "command": "string",
              "packages": [
                {
                  "authors": "string",
                  "releaseDate": "2019-08-24T14:15:22Z",
                  "ecosystem": "string",
                  "group": "string",
                  "archive": true,
                  "executable": true,
                  "uid": "string",
                  "licenses": {
                    "allowed": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            null
                          ],
                          "conditions": [
                            null
                          ],
                          "limitations": [
                            null
                          ]
                        }
                      }
                    ],
                    "blocked": [
                      {
                        "id": "string",
                        "url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            null
                          ],
                          "conditions": [
                            null
                          ],
                          "limitations": [
                            null
                          ]
                        }
                      }
                    ]
```

```
"packageName": "string",
                    "packageVersion": "string",
                    "hashes": [
                      {
                        "sha256": "string",
                        "fileName": "string"
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    },
                    "vulnerabilities": [
                      {
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "arch": "string",
                    "osName": "string",
                    "osVersion": "string",
                    "dependencies": [
                      {
                        "ecosystem": "string",
                        "group": "string",
                        "packageName": "string",
                        "packageVersion": "string",
                        "vulnerabilities": [
                          {
                            "id": "string",
                            "severity": 0,
                            "source": "string",
                            "fixedVersions": "string",
                            "cwes": [
                              null
                            ]
                          }
                        ],
                        "vulnerabilityCounts": {
                          "critical": 0,
                          "high": 0,
                          "medium": 0,
                          "low": 0,
                          "unknown": 0
                        }
```

```
]
                  }
                ],
                "licenseStatus": "string",
                "licensingPackageCounts": {
                  "allowed": 0,
                  "blocked": 0,
                  "unknown": 0,
                  "unsupported": 0
                },
                "step": 0,
                "vulnerabilityCount": 0,
                "sha256": "string",
                "severity": 0,
                "vulnerabilityCounts": {
                  "critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
                }
              }
            ],
            "packages": [
              {
                "authors": "string",
                "releaseDate": "2019-08-24T14:15:22Z",
                "ecosystem": "string",
                "group": "string",
                "archive": true,
                "executable": true,
                "uid": "string",
                "licenses": {
                  "allowed": [
                    {
                      "id": "string",
                      "url": "string",
                      "category": "string",
                      "usageRestriction": {
                        "permissions": [
                          "string"
                        ],
                        "conditions": [
                          "string"
                        ],
                        "limitations": [
                          "string"
                        ]
                      }
                    }
                  ],
                  "blocked": [
                    {
```

```
"url": "string",
                        "category": "string",
                        "usageRestriction": {
                          "permissions": [
                            "string"
                          ],
                          "conditions": [
                            "string"
                          ],
                          "limitations": [
                            "string"
                          ]
                        }
                      }
                    ]
                  },
                  "packageName": "string",
                  "packageVersion": "string",
                  "target": "string",
                  "hashes": [
                    {
                      "sha256": "string",
                      "fileName": "string"
                    }
                  ],
                  "vulnerabilities": [
                    {
                      "id": "string",
                      "severity": 0,
                      "source": "string",
                      "fixedVersions": "string",
                      "cwes": [
                        "string"
                      ]
                    }
                  ],
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  },
                  "dependencies": [
                    {
                      "ecosystem": "string",
                      "group": "string",
                      "packageName": "string",
                      "packageVersion": "string",
                      "vulnerabilities": [
                        {
                          "id": "string",
                          "severity": 0,
```

```
"fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    }
                  }
                ]
              }
            ],
            "dependenciesPackageCount": 0,
            "dependenciesVulnerabilityTotal": 0,
            "dependenciesVulnerabilityCounts": {
              "critical": 0,
              "high": 0,
              "medium": 0,
              "low": 0,
              "unknown": 0
            }
          },
          "resultTemplateHash": "string"
        },
        "vulnerabilityInfo": {
          "vulnerabilityInfoResult": {
            "code": 0,
            "hash": "string",
            "method": 0,
            "timeStamp": "string",
            "timing": 0,
            "detectedProduct": {
              "hasVulnerability": true,
              "isCurrent": true,
              "product": {
                "id": 0,
                "name": "string"
              },
              "remediationLink": "string",
              "severity": "string",
              "sigName": "string",
              "signature": 0,
              "vendor": {
                "id": 0,
                "name": "string"
              },
              "version": "string",
              "versionData": {
```

```
"feedId": 0,
                "version": "string"
              },
              "vulnerabilities": [
                {
                  "description": "string",
                  "details": {
                    "cpe": "string",
                    "cve": "string",
                    "cvss": {
                      "accessComplexity": "string",
                      "accessVector": "string",
                      "authentication": "string",
                      "availabilityImpact": "string",
                      "confidentialityImpact": "string",
                      "generatedOnEpoch": "string",
                      "integrityImpact": "string",
                      "score": "string",
                      "source": "string"
                    },
                    "cvss3": {
                      "attackComplexity": "string",
                      "attackVector": "string",
                      "availabilityImpact": "string",
                      "baseScore": "string",
                      "baseSeverity": "string",
                      "confidentialityImpact": "string",
                      "exploitabilityScore": "string",
                      "impactScore": "string",
                      "integrityImpact": "string",
                      "privilegesRequired": "string",
                      "scope": "string",
                      "userInteraction": "string",
                      "vectorString": "string"
                    },
                    "cwe": "string",
                    "lastModifiedEpoch": "string",
                    "publishedEpoch": "string",
                    "references": [
                      "string"
                    ]
                  },
                  "severity": "string",
                  "severityIndex": 0,
                  "staticId": 0
                }
              ]
            }
          }
        },
        "sessionCookie": "string",
        "fileTypeInfo": {
          "fileInfoExtended": {
```

```
"groupId": "string",
            "typeId": "string"
          }
        },
        "scanInstanceUrl": "string",
        "scanWorkflowSnapshotId": "string",
        "coreEngines": [
          {
            "engineId": "string",
            "engineName": "string",
            "engineVersion": "string",
            "dbVersion": "string",
            "definitionTime": "string"
          }
        ]
      },
      "processingState": 0,
      "exception": {
        "exceptionDetails": "string",
        "exceptionMessage": "string"
      },
      "created": "2019-08-24T14:15:22Z",
      "lastModified": "2019-08-24T14:15:22Z",
      "discoveryStarted": "2019-08-24T14:15:22Z",
      "fileProcessingDetails": {
        "streamingStartedTime": "2019-08-24T14:15:22Z",
        "streamingCompletedTime": "2019-08-24T14:15:22Z",
        "streamingTimeSpanMs": 0,
        "scanCompletedTime": "2019-08-24T14:15:22Z",
        "processingTimeSpanMs": 0
      },
      "coreEngines": [
        {
          "engineId": "string",
          "engineName": "string",
          "engineVersion": "string",
          "dbVersion": "string",
          "definitionTime": "string"
        }
      ],
      "remediation": {
        "isTagged": true,
        "isDeleted": true,
        "isMoved": true,
        "isCopied": true,
        "deepCdr": {
          "sanitizationDetails": {
            "failureCategory": "string",
            "details": "string"
          },
          "cdrObjectState": 0,
          "fileWasReplaced": true
        },
```

```
{
            "objectActionType": 0,
            "exception": {
              "exceptionDetails": "string",
              "exceptionMessage": "string"
            }
          }
        ],
        "objectRemediationActions": [
          {
            "objectActionType": 0,
            "objectActionState": 0,
            "cdrObjectState": 0,
            "exceptions": [
              "string"
            ],
            "id": "string",
            "fileId": "string",
            "fileScanCompletedTime": "2019-08-24T14:15:22Z",
            "remediationId": "string",
            "tenantId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z"
          }
        ]
      },
      "metaDefenderCoreUrl": "string",
      "discoveryStartedDaySec": 0,
      "objectStorageMetadata": "string",
      "scanWorkflowSnapshotId": "string",
      "scanWorkflowSnapshot": {
        "id": "string",
        "scanId": "string",
        "workflow": {
          "id": "string",
          "scanPoolId": "string",
          "scanProcessFlow": 0,
          "isIdentityScanningEnabled": true,
          "scanIntervalDays": 0,
          "rules": [
            "string"
          ],
          "type": 0,
          "workflowSupportedTechnologies": {
            "dlp": {
              "status": 0
            },
            "deepCdr": {
              "status": 0
            },
            "sandBox": {
              "status": 0
            },
            "metascan": {
```

```
}
          },
          "name": "string",
          "userAgent": "string",
          "scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanPoolId": "string",
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "filters": {
            "id": "string",
            "minimumSize": 0,
            "maximumSize": 0,
            "minimumDateTime": "2019-08-24T14:15:22Z",
            "maximumDateTime": "2019-08-24T14:15:22Z"
          },
          "fileDefinition": {
            "blockingStrategy": 0,
            "configurableBlockingStrategy": {
              "shouldTreatFilesWithSensitiveDataAsBlocked": true,
              "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
            }
          },
          "remediations": [
            {
```

```
"workflowId": "string",
              "isEnabled": true,
              "configuration": "string",
              "type": 0,
              "handlingType": 0
            }
          ],
          "tenantId": "string"
        },
        "startTimestamp": 0,
        "stopTimestamp": 0,
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
```

```
{
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "storage": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        },
        "workflowStatistics": {
          "taggingStatistics": {
            "total": 0
          },
          "keepAllowedStatistics": {
            "total": 0
          },
          "keepBlockedStatistics": {
            "total": 0
          },
          "keepSanitizedStatistics": {
            "total": 0
          },
          "movedAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
```

```
"storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedSanitizedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
```

```
"vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
```

```
"statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "deleteAllowedStatistics": {
            "total": 0
          },
          "deleteBlockedStatistics": {
            "total": 0
          },
          "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
        },
        "account": {
          "id": "string",
          "name": "string",
          "description": "string",
          "credentials": "string",
          "protocolType": 0,
          "vendorType": 0,
          "autoDiscovery": true,
          "autodiscoveryDestinationGroupId": "string"
        },
        "realTimeStartDate": "2019-08-24T14:15:22Z"
      },
      "versions": [
        {
          "result": 0,
          "responseKey": "string",
          "responseMessage": "string",
          "responseMessageParams": {
            "property1": "string",
            "property2": "string"
          },
          "id": "string",
          "scanId": "string",
          "name": "string",
          "path": "string",
          "hash": "string",
          "fileOwner": {
            "userId": "string",
            "fullName": "string",
            "userName": "string",
            "emailAddress": "string"
          },
          "size": 0,
          "fileTypeCategory": 0,
          "extension": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
```

```
"vendorType": 0
          },
          "storageName": "string",
          "storageLocation": "string",
          "scanName": "string",
          "progress": {
            "upload": 0,
            "scan": 0
          },
          "scanResult": {
            "extractedFiles": {
              "dataId": "string",
              "scanResultI": 0,
              "detectedBy": 0,
              "filesExtractedCount": 0
            },
            "scanResults": {
              "scanDetails": [
                {
                  "engineName": "string",
                  "threatFound": "string",
                  "scanResultI": 0,
                  "defTime": "string",
                  "scanTime": 0.1,
                  "engId": "string",
                  "waitTime": 0.1
                }
              ],
              "dataId": "string",
              "scanAllResultI": 0,
              "scanAllResultA": "string",
              "startTime": "string",
              "totalTime": 0.1,
              "totalAvs": 0,
              "totalDetections": 0,
              "progressPercentage": 0
            },
            "fileInfo": {
              "fileSize": 0,
              "uploadTimestamp": "string",
              "md5": "string",
              "sha1": "string",
              "sha256": "string",
              "fileTypeCategory": "string",
              "fileTypeDescription": "string",
              "fileTypeExtension": "string",
              "displayName": "string",
              "fileTypeId": "string"
            },
            "fileScanInfo": {
              "finalVerdict": {
                "blocked": true,
                "verdict": "string",
```

```
"threatLevel": 0.1
              },
              "overviewReport": {
                "analysisState": "string",
                "signalGroups": [
                  {
                    "description": "string",
                    "verdict": "string",
                    "signals": [
                      {
                        "originType": "string",
                        "signalReadable": "string",
                        "strength": 0.1
                      }
                    ]
                  }
                ]
              },
              "errorCategory": "string",
              "errorCode": 0,
              "errorDetails": "string"
            },
            "processInfo": {
              "postProcessing": {
                "actionsRan": "string",
                "actionsFailed": "string",
                "convertedTo": "string",
                "copyMoveDestination": "string",
                "convertedDestination": "string",
                "sanitizationDetails": {
                  "failureCategory": "string",
                  "details": "string"
                }
              },
              "progressPercentage": 0,
              "userAgent": "string",
              "profile": "string",
              "result": "string",
              "blockedReason": "string",
              "fileTypeSkippedScan": true,
              "processingTime": 0.1,
              "queueTime": 0.1,
              "resultI": 0
            },
            "dataId": "string",
            "dlpInfo": {
              "hits": null,
              "verdict": 0
            },
            "sbomInfo": {
              "finalVerdict": {
                "blocked": true,
                "licensingPackageCounts": {
```

```
"blocked": 0,
                  "unknown": 0,
                  "unsupported": 0
                },
                "severity": 0,
                "totalPackageCount": 0,
                "verdict": "string",
                "vulnerabilityCounts": {
                  "critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
                }
              },
              "overviewReport": {
                "isPartialReport": true,
                "osInfo": {
                  "osName": "string",
                  "osVersion": "string"
                },
                "scanMode": "string",
                "steps": [
                  {
                    "command": "string",
                    "packages": [
                      {
                        "authors": "string",
                        "releaseDate": "2019-08-24T14:15:22Z",
                        "ecosystem": "string",
                        "group": "string",
                        "archive": true,
                        "executable": true,
                        "uid": "string",
                        "licenses": {
                          "allowed": [
                            {}
                          ],
                          "blocked": [
                            {}
                          ]
                        },
                        "packageName": "string",
                        "packageVersion": "string",
                        "hashes": [
                          {
                            "sha256": "string",
                            "fileName": "string"
                          }
                        ],
                        "vulnerabilityCounts": {
                          "critical": 0,
                          "high": 0,
```

```
"low": 0,
                        "unknown": 0
                      },
                      "vulnerabilities": [
                        {
                          "id": "string",
                          "severity": 0,
                          "source": "string",
                          "fixedVersions": "string",
                          "cwes": [
                            null
                          ]
                        }
                      ],
                      "arch": "string",
                      "osName": "string",
                      "osVersion": "string",
                      "dependencies": [
                        {
                          "ecosystem": "string",
                          "group": "string",
                          "packageName": "string",
                          "packageVersion": "string",
                          "vulnerabilities": [
                            null
                          ],
                          "vulnerabilityCounts": {}
                        }
                      ]
                    }
                  ],
                  "licenseStatus": "string",
                  "licensingPackageCounts": {
                    "allowed": 0,
                    "blocked": 0,
                    "unknown": 0,
                    "unsupported": 0
                  },
                  "step": 0,
                  "vulnerabilityCount": 0,
                  "sha256": "string",
                  "severity": 0,
                  "vulnerabilityCounts": {
                    "critical": 0,
                    "high": 0,
                    "medium": 0,
                    "low": 0,
                    "unknown": 0
                  }
                }
              ],
              "packages": [
                {
```

```
"releaseDate": "2019-08-24T14:15:22Z",
                    "ecosystem": "string",
                    "group": "string",
                    "archive": true,
                    "executable": true,
                    "uid": "string",
                    "licenses": {
                      "allowed": [
                        {
                          "id": "string",
                          "url": "string",
                          "category": "string",
                          "usageRestriction": {
                            "permissions": [
                              null
                            ],
                            "conditions": [
                              null
                            ],
                            "limitations": [
                              null
                            ]
                          }
                        }
                      ],
                      "blocked": [
                        {
                          "id": "string",
                          "url": "string",
                          "category": "string",
                          "usageRestriction": {
                            "permissions": [
                              null
                            ],
                            "conditions": [
                              null
                            ],
                            "limitations": [
                              null
                            ]
                          }
                        }
                      ]
                    },
                    "packageName": "string",
                    "packageVersion": "string",
                    "target": "string",
                    "hashes": [
                      {
                        "sha256": "string",
                        "fileName": "string"
                      }
                    ],
```

```
{
                        "id": "string",
                        "severity": 0,
                        "source": "string",
                        "fixedVersions": "string",
                        "cwes": [
                          "string"
                        ]
                      }
                    ],
                    "vulnerabilityCounts": {
                      "critical": 0,
                      "high": 0,
                      "medium": 0,
                      "low": 0,
                      "unknown": 0
                    },
                    "dependencies": [
                      {
                        "ecosystem": "string",
                        "group": "string",
                        "packageName": "string",
                        "packageVersion": "string",
                        "vulnerabilities": [
                          {
                            "id": "string",
                            "severity": 0,
                            "source": "string",
                            "fixedVersions": "string",
                            "cwes": [
                              null
                            ]
                          }
                        ],
                        "vulnerabilityCounts": {
                          "critical": 0,
                          "high": 0,
                          "medium": 0,
                          "low": 0,
                          "unknown": 0
                        }
                      }
                    ]
                  }
                ],
                "dependenciesPackageCount": 0,
                "dependenciesVulnerabilityTotal": 0,
                "dependenciesVulnerabilityCounts": {
                  "critical": 0,
                  "high": 0,
                  "medium": 0,
                  "low": 0,
                  "unknown": 0
```

```
},
            "resultTemplateHash": "string"
          },
          "vulnerabilityInfo": {
            "vulnerabilityInfoResult": {
              "code": 0,
              "hash": "string",
              "method": 0,
              "timeStamp": "string",
              "timing": 0,
              "detectedProduct": {
                "hasVulnerability": true,
                "isCurrent": true,
                "product": {
                  "id": 0,
                  "name": "string"
                },
                "remediationLink": "string",
                "severity": "string",
                "sigName": "string",
                "signature": 0,
                "vendor": {
                  "id": 0,
                  "name": "string"
                },
                "version": "string",
                "versionData": {
                  "countBehind": 0,
                  "feedId": 0,
                  "version": "string"
                },
                "vulnerabilities": [
                  {
                    "description": "string",
                    "details": {
                      "cpe": "string",
                      "cve": "string",
                      "cvss": {
                        "accessComplexity": "string",
                        "accessVector": "string",
                        "authentication": "string",
                        "availabilityImpact": "string",
                        "confidentialityImpact": "string",
                        "generatedOnEpoch": "string",
                        "integrityImpact": "string",
                        "score": "string",
                        "source": "string"
                      },
                      "cvss3": {
                        "attackComplexity": "string",
                        "attackVector": "string",
                        "availabilityImpact": "string",
                        "baseScore": "string",
```

```
"confidentialityImpact": "string",
                          "exploitabilityScore": "string",
                          "impactScore": "string",
                          "integrityImpact": "string",
                          "privilegesRequired": "string",
                          "scope": "string",
                          "userInteraction": "string",
                          "vectorString": "string"
                        },
                        "cwe": "string",
                        "lastModifiedEpoch": "string",
                        "publishedEpoch": "string",
                        "references": [
                          "string"
                        ]
                      },
                      "severity": "string",
                      "severityIndex": 0,
                      "staticId": 0
                    }
                  ]
                }
              }
            },
            "sessionCookie": "string",
            "fileTypeInfo": {
              "fileInfoExtended": {
                "extensions": "string",
                "groupId": "string",
                "typeId": "string"
              }
            },
            "scanInstanceUrl": "string",
            "scanWorkflowSnapshotId": "string",
            "coreEngines": [
              {
                "engineId": "string",
                "engineName": "string",
                "engineVersion": "string",
                "dbVersion": "string",
                "definitionTime": "string"
              }
            ]
          },
          "processingState": 0,
          "exception": {
            "exceptionDetails": "string",
            "exceptionMessage": "string"
          },
          "created": "2019-08-24T14:15:22Z",
          "lastModified": "2019-08-24T14:15:22Z",
          "discoveryStarted": "2019-08-24T14:15:22Z",
          "fileProcessingDetails": {
```

```
"streamingCompletedTime": "2019-08-24T14:15:22Z",
            "streamingTimeSpanMs": 0,
            "scanCompletedTime": "2019-08-24T14:15:22Z",
            "processingTimeSpanMs": 0
          },
          "coreEngines": [
            {
              "engineId": "string",
              "engineName": "string",
              "engineVersion": "string",
              "dbVersion": "string",
              "definitionTime": "string"
            }
          ],
          "remediation": {
            "isTagged": true,
            "isDeleted": true,
            "isMoved": true,
            "isCopied": true,
            "deepCdr": {
              "sanitizationDetails": {
                "failureCategory": "string",
                "details": "string"
              },
              "cdrObjectState": 0,
              "fileWasReplaced": true
            },
            "postActionFailedInformation": [
              {
                "objectActionType": 0,
                "exception": {
                  "exceptionDetails": "string",
                  "exceptionMessage": "string"
                }
              }
            ],
            "objectRemediationActions": [
              {
                "objectActionType": 0,
                "objectActionState": 0,
                "cdrObjectState": 0,
                "exceptions": [
                  "string"
                ],
                "id": "string",
                "fileId": "string",
                "fileScanCompletedTime": "2019-08-24T14:15:22Z",
                "remediationId": "string",
                "tenantId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z"
              }
            ]
          },
```

```
"discoveryStartedDaySec": 0,
          "objectStorageMetadata": "string",
          "scanWorkflowSnapshotId": "string",
          "scanWorkflowSnapshot": {
            "id": "string",
            "scanId": "string",
            "workflow": {
              "id": "string",
              "scanPoolId": "string",
              "scanProcessFlow": 0,
              "isIdentityScanningEnabled": true,
              "scanIntervalDays": 0,
              "rules": [
                "string"
              ],
              "type": 0,
              "workflowSupportedTechnologies": {
                "dlp": {
                  "status": 0
                },
                "deepCdr": {
                  "status": 0
                },
                "sandBox": {
                  "status": 0
                },
                "metascan": {
                  "status": 0
                }
              },
              "name": "string",
              "userAgent": "string",
              "scanPool": {
                "id": "string",
                "name": "string",
                "scanPoolType": 0,
                "isDefault": true,
                "scanInstances": [
                  {
                    "id": "string",
                    "scanPoolId": "string",
                    "url": "string",
                    "apiKey": "string",
                    "timeout": "string",
                    "scanInstanceType": 0
                  }
                ]
              },
              "failOverScanPoolId": "string",
              "failOverScanPool": {
                "id": "string",
                "name": "string",
                "scanPoolType": 0,
```

```
"scanInstances": [
                  {
                    "id": "string",
                    "scanPoolId": "string",
                    "url": "string",
                    "apiKey": "string",
                    "timeout": "string",
                    "scanInstanceType": 0
                  }
                ]
              },
              "filters": {
                "id": "string",
                "minimumSize": 0,
                "maximumSize": 0,
                "minimumDateTime": "2019-08-24T14:15:22Z",
                "maximumDateTime": "2019-08-24T14:15:22Z"
              },
              "fileDefinition": {
                "blockingStrategy": 0,
                "configurableBlockingStrategy": {
                  "shouldTreatFilesWithSensitiveDataAsBlocked": true,
                  "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
                }
              },
              "remediations": [
                {
                  "id": "string",
                  "workflowId": "string",
                  "isEnabled": true,
                  "configuration": "string",
                  "type": 0,
                  "handlingType": 0
                }
              ],
              "tenantId": "string"
            },
            "startTimestamp": 0,
            "stopTimestamp": 0,
            "scanPool": {
              "id": "string",
              "name": "string",
              "scanPoolType": 0,
              "isDefault": true,
              "scanInstances": [
                {
                  "id": "string",
                  "scanPoolId": "string",
                  "url": "string",
                  "apiKey": "string",
                  "timeout": "string",
                  "scanInstanceType": 0
                }
```

```
},
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ],
            "failOverScanPool": {
              "id": "string",
              "name": "string",
              "scanPoolType": 0,
              "isDefault": true,
              "scanInstances": [
                {
                  "id": "string",
                  "scanPoolId": "string",
                  "url": "string",
                  "apiKey": "string",
                  "timeout": "string",
                  "scanInstanceType": 0
                }
              ]
            },
            "failOverScanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ],
            "storage": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
```

```
},
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            },
            "workflowStatistics": {
              "taggingStatistics": {
                "total": 0
              },
              "keepAllowedStatistics": {
                "total": 0
              },
              "keepBlockedStatistics": {
                "total": 0
              },
              "keepSanitizedStatistics": {
                "total": 0
              },
              "movedAllowedStatistics": {
                "total": 0,
                "destinationStorageInformation": {
                  "id": "string",
                  "name": "string",
                  "groupId": "string",
                  "clientId": "string",
                  "accountId": "string",
                  "source": "string",
                  "storageFunctionality": "string",
                  "storageType": {
                    "name": "string",
                    "categoryType": 0,
                    "protocolType": 0,
                    "vendorType": 0
                  },
                  "storageStatus": {
                    "message": "string",
                    "statusType": 0
                  },
                  "workflowId": "string",
                  "lastUpdated": "2019-08-24T14:15:22Z",
                  "tenantId": "string"
                }
              },
              "movedSanitizedStatistics": {
                "total": 0,
                "destinationStorageInformation": {
                  "id": "string",
                  "name": "string",
                  "groupId": "string",
                  "clientId": "string",
                  "accountId": "string",
                  "source": "string",
                  "storageFunctionality": "string",
```

```
"name": "string",
                    "categoryType": 0,
                    "protocolType": 0,
                    "vendorType": 0
                  },
                  "storageStatus": {
                    "message": "string",
                    "statusType": 0
                  },
                  "workflowId": "string",
                  "lastUpdated": "2019-08-24T14:15:22Z",
                  "tenantId": "string"
                }
              },
              "movedBlockedStatistics": {
                "total": 0,
                "destinationStorageInformation": {
                  "id": "string",
                  "name": "string",
                  "groupId": "string",
                  "clientId": "string",
                  "accountId": "string",
                  "source": "string",
                  "storageFunctionality": "string",
                  "storageType": {
                    "name": "string",
                    "categoryType": 0,
                    "protocolType": 0,
                    "vendorType": 0
                  },
                  "storageStatus": {
                    "message": "string",
                    "statusType": 0
                  },
                  "workflowId": "string",
                  "lastUpdated": "2019-08-24T14:15:22Z",
                  "tenantId": "string"
                }
              },
              "copyAllowedStatistics": {
                "total": 0,
                "destinationStorageInformation": {
                  "id": "string",
                  "name": "string",
                  "groupId": "string",
                  "clientId": "string",
                  "accountId": "string",
                  "source": "string",
                  "storageFunctionality": "string",
                  "storageType": {
                    "name": "string",
                    "categoryType": 0,
                    "protocolType": 0,
```

```
},
                  "storageStatus": {
                    "message": "string",
                    "statusType": 0
                  },
                  "workflowId": "string",
                  "lastUpdated": "2019-08-24T14:15:22Z",
                  "tenantId": "string"
                }
              },
              "copyBlockedStatistics": {
                "total": 0,
                "destinationStorageInformation": {
                  "id": "string",
                  "name": "string",
                  "groupId": "string",
                  "clientId": "string",
                  "accountId": "string",
                  "source": "string",
                  "storageFunctionality": "string",
                  "storageType": {
                    "name": "string",
                    "categoryType": 0,
                    "protocolType": 0,
                    "vendorType": 0
                  },
                  "storageStatus": {
                    "message": "string",
                    "statusType": 0
                  },
                  "workflowId": "string",
                  "lastUpdated": "2019-08-24T14:15:22Z",
                  "tenantId": "string"
                }
              },
              "deleteAllowedStatistics": {
                "total": 0
              },
              "deleteBlockedStatistics": {
                "total": 0
              },
              "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
            },
            "account": {
              "id": "string",
              "name": "string",
              "description": "string",
              "credentials": "string",
              "protocolType": 0,
              "vendorType": 0,
              "autoDiscovery": true,
              "autodiscoveryDestinationGroupId": "string"
            },
```

```
},
          "versions": [],
          "hasBlockedVersions": true,
          "versionsCount": 0,
          "objectType": 0,
          "skippedScan": true,
          "$type": "string",
          "url": "string",
          "fileId": "string"
        }
      ],
      "hasBlockedVersions": true,
      "versionsCount": 0,
      "objectType": 0,
      "skippedScan": true,
      "$type": "string"
    }
  ],
  "filterCount": 0,
  "totalCount": 0
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entries	[File]|null	false	none	none
» filterCount	integer(int64)	false	none	none
» totalCount	integer(int64)	false	none	none

CancelFileParameters

```
{
  "fileIds": [
    "string"
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
fileIds	[string]¦null	false	none	none

AddGroupCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "group": {
    "id": "string",
    "name": "string",
    "description": "string",
    "storages": [
      {
        "id": "string",
        "name": "string",
        "groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      }
    ]
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» group	Group¦null	false	none	none

Group


```
{
  "id": "string",
  "name": "string",
  "description": "string",
  "storages": [
    {
      "id": "string",
      "name": "string",
      "groupId": "string",
      "clientId": "string",
      "accountId": "string",
      "source": "string",
      "storageFunctionality": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "workflowId": "string",
      "lastUpdated": "2019-08-24T14:15:22Z",
      "tenantId": "string"
    }
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
name	string	false	none	none
description	string	false	none	none
storages	[NewStorage]¦null	false	none	none

AddGroupParameters

```
{
  "name": "string",
  "description": "string",
  "storageIds": [
    "string"
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
name	string	true	none	group name
description	string¦null	false	none	description of the group
storageIds	[string]	true	none	associated storages

DeleteGroupCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

GetGroupById

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "group": {
    "id": "string",
    "name": "string",
    "description": "string",
    "storages": [
      {
        "id": "string",
        "groupId": "string",
        "accountId": "string",
        "name": "string",
        "storageClientId": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "scansInProgress": true,
        "isRtpInProgress": true,
        "isScheduledScanInProgress": true,
        "storageFunctionality": "string",
        "$type": "string"
      }
    ]
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» group	GroupWithStorage	false	none	none

GroupWithStorage


```
{
  "id": "string",
  "name": "string",
  "description": "string",
  "storages": [
    {
      "id": "string",
      "groupId": "string",
      "accountId": "string",
      "name": "string",
      "storageClientId": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "scansInProgress": true,
      "isRtpInProgress": true,
      "isScheduledScanInProgress": true,
      "storageFunctionality": "string",
      "$type": "string"
    }
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
name	string	false	none	none
description	string	false	none	none
storages	[Storage]¦null	false	none	none

GetGroups

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "groups": [
    {
      "id": "string",
      "name": "string",
      "description": "string",
      "storages": [
        {
          "id": "string",
          "groupId": "string",
          "accountId": "string",
          "name": "string",
          "storageClientId": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "scansInProgress": true,
          "isRtpInProgress": true,
          "isScheduledScanInProgress": true,
          "storageFunctionality": "string",
          "$type": "string"
        }
      ]
    }
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» groups	[GroupWithStorage]	false	none	none

GetGroupsCount

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "groupCount": {
    "count": 0
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» groupCount	CountResult	false	none	none

UpdateGroupCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "updatedGroup": {
    "id": "string",
    "name": "string",
    "description": "string",
    "storages": [
      {
        "id": "string",
        "name": "string",
        "groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      }
    ]
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» updatedGroup	Group¦null	false	none	none

UpdateGroupParameters

```
{
  "name": "string",
  "description": "string",
  "groupId": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
name	string¦null	false	none	group name
description	string¦null	false	none	description of the group
groupId	string	true	none	group id to update

ApiHealthModel

```
{
  "apiStatus": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
apiStatus	string¦null	false	none	none

GetOnboarding

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "hasUserConfigured": true,
  "hasStorageConfigured": true,
  "hasScanInstanceConfigured": true,
  "isOnboardingDone": true,
  "isEulaAgreed": true,
  "isCloudType": true
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» hasUserConfigured	boolean	false	none	none
» hasStorageConfigured	boolean	false	none	none
» hasScanInstanceConfigured	boolean	false	none	none
» isOnboardingDone	boolean	false	none	none
» isEulaAgreed	boolean	false	none	none
» isCloudType	boolean	false	none	none

GetRemediationByWorkflowIdCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "workflowId": "string",
      "isEnabled": true,
      "configuration": "string",
      "type": 0,
      "handlingType": 0
    }
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entries	[Remediation]¦null	false	none	none

GetRemediationByIdCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entry": {
    "id": "string",
    "workflowId": "string",
    "isEnabled": true,
    "configuration": "string",
    "type": 0,
    "handlingType": 0
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entry	Remediation¦null	false	none	none

AddRemediationCommandResponse


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

AddRemediationParameters

```
{
  "workflowId": "string",
  "isEnabled": true,
  "configuration": "string",
  "type": 0,
  "handlingType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
workflowId	string	false	none	none
isEnabled	boolean	false	none	none
configuration	string	false	none	none
type	RemediationType	false	none	- 0: FileTagging - 1: DeepCdr - 2: BlockedFileRemediation - 3: AllowedSanitizedFileRemediation - 4: AllowedFileRemediation - 5: BlockedSanitizedFileRemediation - 6: DeleteEmptyFoldersRemediation - 7: ScanFileVersions - 8: DiscoveredFileRemediation
handlingType	RemediationHandlingType	false	none	- 0: Delete - 1: Move - 2: Copy - 3: Keep - 999: NotApplicable

UpdateRemediationCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

UpdateRemediationParameters

```
{
  "id": "string",
  "workflowId": "string",
  "isEnabled": true,
  "configuration": "string",
  "type": 0,
  "handlingType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
workflowId	string¦null	false	none	none
isEnabled	boolean¦null	false	none	none
configuration	string¦null	false	none	none
type	RemediationType¦null	false	none	- 0: FileTagging - 1: DeepCdr - 2: BlockedFileRemediation - 3: AllowedSanitizedFileRemediation - 4: AllowedFileRemediation - 5: BlockedSanitizedFileRemediation - 6: DeleteEmptyFoldersRemediation - 7: ScanFileVersions - 8: DiscoveredFileRemediation
handlingType	RemediationHandlingType¦null	false	none	- 0: Delete - 1: Move - 2: Copy - 3: Keep - 999: NotApplicable

DeleteRemediationCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

GetScansReportQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "result": 0,
      "responseKey": "string",
      "responseMessage": "string",
      "responseMessageParams": {
        "property1": "string",
        "property2": "string"
      },
      "id": "string",
      "name": "string",
      "startTime": "2019-08-24T14:15:22Z",
      "stopTime": "2019-08-24T14:15:22Z",
      "scanType": "string",
      "scanningState": 0,
      "scanStatusDto": {
        "pendingFilesCount": 0,
        "pendingFilesSize": 0,
        "allowedFilesCount": 0,
        "allowedFilesSize": 0,
        "blockedFilesCount": 0,
        "blockedFilesSize": 0,
        "sanitizedFilesCount": 0,
        "failedSanitizedFilesCount": 0,
        "remediatedFilesCount": 0,
        "totalFilesCount": 0,
        "totalFilesSize": 0,
        "processingFailedCount": 0,
        "processingFailedSize": 0,
        "processingFailedRemediationCount": 0,
        "processedFilesCount": 0,
        "processedFilesSize": 0,
        "canceledFilesCount": 0,
        "canceledFilesSize": 0
      },
      "scanStatsDto": {
        "infectedFilesCount": 0,
        "filesWithVulnerabilitiesCount": 0,
        "filesWithSensitiveDataCount": 0,
        "filesBlockedBySandboxCount": 0,
        "scannedFilesCount": 0,
        "skippedFilesScanCount": 0,
        "scanFilesDistribution": {
          "documentsEncryptedFilesCount": 0,
```

```
"audioVideoFilesCount": 0,
          "openSSlEncryptedFilesCount": 0,
          "pdfFilesCount": 0,
          "graphicalFilesCount": 0,
          "diskImageFilesCount": 0,
          "textFilesCount": 0,
          "mailFilesCount": 0,
          "applicationFilesCount": 0,
          "documentsFilesCount": 0,
          "otherFilesCount": 0,
          "archivesFilesCount": 0
        }
      },
      "realTimeStartDate": "2019-08-24T14:15:22Z",
      "scanSequenceDto": {
        "previousScanId": "string",
        "nextScanId": "string"
      },
      "scanPartition": "string",
      "workflowId": "string",
      "scanConfigurationId": "string",
      "errors": [
        "string"
      ],
      "priority": 0,
      "scanWorkflowSnapshotDtos": [
        {
          "id": "string",
          "scanId": "string",
          "workflow": {
            "id": "string",
            "scanPoolId": "string",
            "scanProcessFlow": 0,
            "isIdentityScanningEnabled": true,
            "scanIntervalDays": 0,
            "rules": [
              "string"
            ],
            "type": 0,
            "workflowSupportedTechnologies": {
              "dlp": {
                "status": 0
              },
              "deepCdr": {
                "status": 0
              },
              "sandBox": {
                "status": 0
              },
              "metascan": {
                "status": 0
              }
            },
```

```
"userAgent": "string",
            "scanPool": {
              "id": "string",
              "name": "string",
              "scanPoolType": 0,
              "isDefault": true,
              "scanInstances": [
                {
                  "id": "string",
                  "scanPoolId": "string",
                  "url": "string",
                  "apiKey": "string",
                  "timeout": "string",
                  "scanInstanceType": 0
                }
              ]
            },
            "failOverScanPoolId": "string",
            "failOverScanPool": {
              "id": "string",
              "name": "string",
              "scanPoolType": 0,
              "isDefault": true,
              "scanInstances": [
                {
                  "id": "string",
                  "scanPoolId": "string",
                  "url": "string",
                  "apiKey": "string",
                  "timeout": "string",
                  "scanInstanceType": 0
                }
              ]
            },
            "filters": {
              "id": "string",
              "minimumSize": 0,
              "maximumSize": 0,
              "minimumDateTime": "2019-08-24T14:15:22Z",
              "maximumDateTime": "2019-08-24T14:15:22Z"
            },
            "fileDefinition": {
              "blockingStrategy": 0,
              "configurableBlockingStrategy": {
                "shouldTreatFilesWithSensitiveDataAsBlocked": true,
                "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
              }
            },
            "remediations": [
              {
                "id": "string",
                "workflowId": "string",
                "isEnabled": true,
```

```
"type": 0,
                "handlingType": 0
              }
            ],
            "tenantId": "string"
          },
          "startTimestamp": 0,
          "stopTimestamp": 0,
          "scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ],
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanInstances": [
            {
              "id": "string",
```

```
"url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ],
          "storage": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          },
          "workflowStatistics": {
            "taggingStatistics": {
              "total": 0
            },
            "keepAllowedStatistics": {
              "total": 0
            },
            "keepBlockedStatistics": {
              "total": 0
            },
            "keepSanitizedStatistics": {
              "total": 0
            },
            "movedAllowedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
```

```
"protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "movedSanitizedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "movedBlockedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
```

```
"message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "copyAllowedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
                "workflowId": "string",
                "lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "copyBlockedStatistics": {
              "total": 0,
              "destinationStorageInformation": {
                "id": "string",
                "name": "string",
                "groupId": "string",
                "clientId": "string",
                "accountId": "string",
                "source": "string",
                "storageFunctionality": "string",
                "storageType": {
                  "name": "string",
                  "categoryType": 0,
                  "protocolType": 0,
                  "vendorType": 0
                },
                "storageStatus": {
                  "message": "string",
                  "statusType": 0
                },
```

```
"lastUpdated": "2019-08-24T14:15:22Z",
                "tenantId": "string"
              }
            },
            "deleteAllowedStatistics": {
              "total": 0
            },
            "deleteBlockedStatistics": {
              "total": 0
            },
            "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
          },
          "account": {
            "id": "string",
            "name": "string",
            "description": "string",
            "credentials": "string",
            "protocolType": 0,
            "vendorType": 0,
            "autoDiscovery": true,
            "autodiscoveryDestinationGroupId": "string"
          },
          "realTimeStartDate": "2019-08-24T14:15:22Z"
        }
      ],
      "storageId": "string",
      "$type": "string"
    }
  ],
  "totalCount": 0
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entries	[ScanResponse]¦null	false	none	none
» totalCount	integer(int64)¦null	false	none	none
» result	ResultType5	false	none	- 0: Success - 1: StorageNotFound - 2: Error

ScanResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "id": "string",
  "name": "string",
  "startTime": "2019-08-24T14:15:22Z",
  "stopTime": "2019-08-24T14:15:22Z",
  "scanType": "string",
  "scanningState": 0,
  "scanStatusDto": {
    "pendingFilesCount": 0,
    "pendingFilesSize": 0,
    "allowedFilesCount": 0,
    "allowedFilesSize": 0,
    "blockedFilesCount": 0,
    "blockedFilesSize": 0,
    "sanitizedFilesCount": 0,
    "failedSanitizedFilesCount": 0,
    "remediatedFilesCount": 0,
    "totalFilesCount": 0,
    "totalFilesSize": 0,
    "processingFailedCount": 0,
    "processingFailedSize": 0,
    "processingFailedRemediationCount": 0,
    "processedFilesCount": 0,
    "processedFilesSize": 0,
    "canceledFilesCount": 0,
    "canceledFilesSize": 0
  },
  "scanStatsDto": {
    "infectedFilesCount": 0,
    "filesWithVulnerabilitiesCount": 0,
    "filesWithSensitiveDataCount": 0,
    "filesBlockedBySandboxCount": 0,
    "scannedFilesCount": 0,
    "skippedFilesScanCount": 0,
    "scanFilesDistribution": {
      "documentsEncryptedFilesCount": 0,
      "executablesFilesCount": 0,
      "audioVideoFilesCount": 0,
      "openSSlEncryptedFilesCount": 0,
      "pdfFilesCount": 0,
      "graphicalFilesCount": 0,
      "diskImageFilesCount": 0,
      "textFilesCount": 0,
      "mailFilesCount": 0,
      "applicationFilesCount": 0,
```

```
"otherFilesCount": 0,
      "archivesFilesCount": 0
    }
  },
  "realTimeStartDate": "2019-08-24T14:15:22Z",
  "scanSequenceDto": {
    "previousScanId": "string",
    "nextScanId": "string"
  },
  "scanPartition": "string",
  "workflowId": "string",
  "scanConfigurationId": "string",
  "errors": [
    "string"
  ],
  "priority": 0,
  "scanWorkflowSnapshotDtos": [
    {
      "id": "string",
      "scanId": "string",
      "workflow": {
        "id": "string",
        "scanPoolId": "string",
        "scanProcessFlow": 0,
        "isIdentityScanningEnabled": true,
        "scanIntervalDays": 0,
        "rules": [
          "string"
        ],
        "type": 0,
        "workflowSupportedTechnologies": {
          "dlp": {
            "status": 0
          },
          "deepCdr": {
            "status": 0
          },
          "sandBox": {
            "status": 0
          },
          "metascan": {
            "status": 0
          }
        },
        "name": "string",
        "userAgent": "string",
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
```

```
"scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanPoolId": "string",
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "filters": {
          "id": "string",
          "minimumSize": 0,
          "maximumSize": 0,
          "minimumDateTime": "2019-08-24T14:15:22Z",
          "maximumDateTime": "2019-08-24T14:15:22Z"
        },
        "fileDefinition": {
          "blockingStrategy": 0,
          "configurableBlockingStrategy": {
            "shouldTreatFilesWithSensitiveDataAsBlocked": true,
            "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
          }
        },
        "remediations": [
          {
            "id": "string",
            "workflowId": "string",
            "isEnabled": true,
            "configuration": "string",
            "type": 0,
            "handlingType": 0
          }
        ],
        "tenantId": "string"
      },
      "startTimestamp": 0,
      "stopTimestamp": 0,
```

```
"id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "failOverScanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "failOverScanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "storage": {
        "id": "string",
```

```
"groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      },
      "workflowStatistics": {
        "taggingStatistics": {
          "total": 0
        },
        "keepAllowedStatistics": {
          "total": 0
        },
        "keepBlockedStatistics": {
          "total": 0
        },
        "keepSanitizedStatistics": {
          "total": 0
        },
        "movedAllowedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
```

```
"tenantId": "string"
          }
        },
        "movedSanitizedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "movedBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
```

```
"total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "copyBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "deleteAllowedStatistics": {
          "total": 0
        },
        "deleteBlockedStatistics": {
```

```
},
        "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
      },
      "account": {
        "id": "string",
        "name": "string",
        "description": "string",
        "credentials": "string",
        "protocolType": 0,
        "vendorType": 0,
        "autoDiscovery": true,
        "autodiscoveryDestinationGroupId": "string"
      },
      "realTimeStartDate": "2019-08-24T14:15:22Z"
    }
  ],
  "storageId": "string",
  "$type": "string"
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and - discriminator: $type

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» id	string	false	none	none
» name	string	false	none	none
» startTime	string(date-time)	false	none	none
» stopTime	string(date-time)	false	none	none
» scanType	string	false	none	none
» scanningState	ScanningState	false	none	- 0: NotStarted - 1: Running - 2: Cancelled - 3: Failed - 4: Completed
» scanStatusDto	ScanStatus	false	none	none
» scanStatsDto	ScanStats	false	none	none
» realTimeStartDate	string(date-time)¦null	false	none	none
» scanSequenceDto	ScanSequence¦null	false	none	none
» scanPartition	string¦null	false	none	none
» workflowId	string	true	none	WorkflowId required if ScanConfigurationId is null
» scanConfigurationId	string¦null	false	none	This property is obsolete and will be removed in the future. Use WorkflowId instead.
» errors	[string]	false	none	none
» priority	MessagePriority	false	none	- 0: Low - 1: Medium - 2: High
» scanWorkflowSnapshotDtos	[ScanWorkflowSnapshot]	false	none	none
» storageId	string	false	none	none
» $type	string	true	none	none

ScanningState

```
0
```

- 0: NotStarted
- 1: Running
- 2: Cancelled
- 3: Failed
- 4: Completed

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: NotStarted - 1: Running - 2: Cancelled - 3: Failed - 4: Completed

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4

ScanStatus

```
{
  "pendingFilesCount": 0,
  "pendingFilesSize": 0,
  "allowedFilesCount": 0,
  "allowedFilesSize": 0,
  "blockedFilesCount": 0,
  "blockedFilesSize": 0,
  "sanitizedFilesCount": 0,
  "failedSanitizedFilesCount": 0,
  "remediatedFilesCount": 0,
  "totalFilesCount": 0,
  "totalFilesSize": 0,
  "processingFailedCount": 0,
  "processingFailedSize": 0,
  "processingFailedRemediationCount": 0,
  "processedFilesCount": 0,
  "processedFilesSize": 0,
  "canceledFilesCount": 0,
  "canceledFilesSize": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
pendingFilesCount	integer(int64)	false	none	none
pendingFilesSize	integer(int64)	false	none	none
allowedFilesCount	integer(int64)	false	none	none
allowedFilesSize	integer(int64)	false	none	none
blockedFilesCount	integer(int64)	false	none	none
blockedFilesSize	integer(int64)	false	none	none
sanitizedFilesCount	integer(int64)	false	none	none
failedSanitizedFilesCount	integer(int64)	false	none	none
remediatedFilesCount	integer(int64)	false	none	none
totalFilesCount	integer(int64)	false	none	none
totalFilesSize	integer(int64)	false	none	none
processingFailedCount	integer(int64)	false	none	none
processingFailedSize	integer(int64)	false	none	none
processingFailedRemediationCount	integer(int64)	false	none	none
processedFilesCount	integer(int64)	false	none	none
processedFilesSize	integer(int64)	false	none	none
canceledFilesCount	integer(int64)	false	none	none
canceledFilesSize	integer(int64)	false	none	none

ScanStats

```
{
  "infectedFilesCount": 0,
  "filesWithVulnerabilitiesCount": 0,
  "filesWithSensitiveDataCount": 0,
  "filesBlockedBySandboxCount": 0,
  "scannedFilesCount": 0,
  "skippedFilesScanCount": 0,
  "scanFilesDistribution": {
    "documentsEncryptedFilesCount": 0,
    "executablesFilesCount": 0,
    "audioVideoFilesCount": 0,
    "openSSlEncryptedFilesCount": 0,
    "pdfFilesCount": 0,
    "graphicalFilesCount": 0,
    "diskImageFilesCount": 0,
    "textFilesCount": 0,
    "mailFilesCount": 0,
    "applicationFilesCount": 0,
    "documentsFilesCount": 0,
    "otherFilesCount": 0,
    "archivesFilesCount": 0
  }
}
```


Properties

Name	Type	Required	Restrictions	Description
infectedFilesCount	integer(int64)	false	none	none
filesWithVulnerabilitiesCount	integer(int64)	false	none	none
filesWithSensitiveDataCount	integer(int64)	false	none	none
filesBlockedBySandboxCount	integer(int64)	false	none	none
scannedFilesCount	integer(int64)	false	none	none
skippedFilesScanCount	integer(int64)	false	none	none
scanFilesDistribution	ScanFilesDistribution|null	false	none	none

ScanFilesDistribution

```
{
  "documentsEncryptedFilesCount": 0,
  "executablesFilesCount": 0,
  "audioVideoFilesCount": 0,
  "openSSlEncryptedFilesCount": 0,
  "pdfFilesCount": 0,
  "graphicalFilesCount": 0,
  "diskImageFilesCount": 0,
  "textFilesCount": 0,
  "mailFilesCount": 0,
  "applicationFilesCount": 0,
  "documentsFilesCount": 0,
  "otherFilesCount": 0,
  "archivesFilesCount": 0
}
```


Properties

Name	Type	Required	Restrictions	Description
documentsEncryptedFilesCount	integer(int64)¦null	false	none	none
executablesFilesCount	integer(int64)¦null	false	none	none
audioVideoFilesCount	integer(int64)¦null	false	none	none
openSSlEncryptedFilesCount	integer(int64)¦null	false	none	none
pdfFilesCount	integer(int64)¦null	false	none	none
graphicalFilesCount	integer(int64)¦null	false	none	none
diskImageFilesCount	integer(int64)¦null	false	none	none
textFilesCount	integer(int64)¦null	false	none	none
mailFilesCount	integer(int64)¦null	false	none	none
applicationFilesCount	integer(int64)¦null	false	none	none
documentsFilesCount	integer(int64)¦null	false	none	none
otherFilesCount	integer(int64)¦null	false	none	none
archivesFilesCount	integer(int64)¦null	false	none	none

ScanSequence

```
{
  "previousScanId": "string",
  "nextScanId": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
previousScanId	string¦null	false	none	none
nextScanId	string¦null	false	none	none

MessagePriority

```
0
```

- 0: Low
- 1: Medium
- 2: High

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Low - 1: Medium - 2: High

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

RealTimeScanResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "id": "string",
  "name": "string",
  "startTime": "2019-08-24T14:15:22Z",
  "stopTime": "2019-08-24T14:15:22Z",
  "scanType": "string",
  "scanningState": 0,
  "scanStatusDto": {
    "pendingFilesCount": 0,
    "pendingFilesSize": 0,
    "allowedFilesCount": 0,
    "allowedFilesSize": 0,
    "blockedFilesCount": 0,
    "blockedFilesSize": 0,
    "sanitizedFilesCount": 0,
    "failedSanitizedFilesCount": 0,
    "remediatedFilesCount": 0,
    "totalFilesCount": 0,
    "totalFilesSize": 0,
    "processingFailedCount": 0,
    "processingFailedSize": 0,
    "processingFailedRemediationCount": 0,
    "processedFilesCount": 0,
    "processedFilesSize": 0,
    "canceledFilesCount": 0,
    "canceledFilesSize": 0
  },
  "scanStatsDto": {
    "infectedFilesCount": 0,
    "filesWithVulnerabilitiesCount": 0,
    "filesWithSensitiveDataCount": 0,
    "filesBlockedBySandboxCount": 0,
    "scannedFilesCount": 0,
    "skippedFilesScanCount": 0,
    "scanFilesDistribution": {
      "documentsEncryptedFilesCount": 0,
      "executablesFilesCount": 0,
      "audioVideoFilesCount": 0,
      "openSSlEncryptedFilesCount": 0,
      "pdfFilesCount": 0,
      "graphicalFilesCount": 0,
      "diskImageFilesCount": 0,
      "textFilesCount": 0,
      "mailFilesCount": 0,
      "applicationFilesCount": 0,
```

```
"otherFilesCount": 0,
      "archivesFilesCount": 0
    }
  },
  "realTimeStartDate": "2019-08-24T14:15:22Z",
  "scanSequenceDto": {
    "previousScanId": "string",
    "nextScanId": "string"
  },
  "scanPartition": "string",
  "workflowId": "string",
  "scanConfigurationId": "string",
  "errors": [
    "string"
  ],
  "priority": 0,
  "scanWorkflowSnapshotDtos": [
    {
      "id": "string",
      "scanId": "string",
      "workflow": {
        "id": "string",
        "scanPoolId": "string",
        "scanProcessFlow": 0,
        "isIdentityScanningEnabled": true,
        "scanIntervalDays": 0,
        "rules": [
          "string"
        ],
        "type": 0,
        "workflowSupportedTechnologies": {
          "dlp": {
            "status": 0
          },
          "deepCdr": {
            "status": 0
          },
          "sandBox": {
            "status": 0
          },
          "metascan": {
            "status": 0
          }
        },
        "name": "string",
        "userAgent": "string",
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
```

```
"scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanPoolId": "string",
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "filters": {
          "id": "string",
          "minimumSize": 0,
          "maximumSize": 0,
          "minimumDateTime": "2019-08-24T14:15:22Z",
          "maximumDateTime": "2019-08-24T14:15:22Z"
        },
        "fileDefinition": {
          "blockingStrategy": 0,
          "configurableBlockingStrategy": {
            "shouldTreatFilesWithSensitiveDataAsBlocked": true,
            "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
          }
        },
        "remediations": [
          {
            "id": "string",
            "workflowId": "string",
            "isEnabled": true,
            "configuration": "string",
            "type": 0,
            "handlingType": 0
          }
        ],
        "tenantId": "string"
      },
      "startTimestamp": 0,
      "stopTimestamp": 0,
```

```
"id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "failOverScanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "failOverScanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "storage": {
        "id": "string",
```

```
"groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      },
      "workflowStatistics": {
        "taggingStatistics": {
          "total": 0
        },
        "keepAllowedStatistics": {
          "total": 0
        },
        "keepBlockedStatistics": {
          "total": 0
        },
        "keepSanitizedStatistics": {
          "total": 0
        },
        "movedAllowedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
```

```
"tenantId": "string"
          }
        },
        "movedSanitizedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "movedBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
```

```
"total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "copyBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "deleteAllowedStatistics": {
          "total": 0
        },
        "deleteBlockedStatistics": {
```

```
},
        "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
      },
      "account": {
        "id": "string",
        "name": "string",
        "description": "string",
        "credentials": "string",
        "protocolType": 0,
        "vendorType": 0,
        "autoDiscovery": true,
        "autodiscoveryDestinationGroupId": "string"
      },
      "realTimeStartDate": "2019-08-24T14:15:22Z"
    }
  ],
  "storageId": "string",
  "$type": "string",
  "isRealTimeEnabled": true,
  "handlingType": 0
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	ScanResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» isRealTimeEnabled	boolean	false	none	none
» handlingType	RealTimeHandlingType	false	none	- 0: Polling - 1: EventBased

RealTimeHandlingType


```
0
```


- 0: Polling
- 1: EventBased

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Polling - 1: EventBased

Enumerated Values

Property	Value
anonymous	0
anonymous	1

ResultType5

```
0
```

- 0: Success
- 1: StorageNotFound
- 2: Error

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Success - 1: StorageNotFound - 2: Error

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

ScanTypeFilter

```
0
```

- 0: Instant
- 1: Scheduled
- 2: RealTime

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Instant - 1: Scheduled - 2: RealTime

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

ScanStateFilter

```
0
```

- 0: NotStarted
- 1: Running
- 2: Cancelled
- 3: Failed
- 4: Completed

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: NotStarted - 1: Running - 2: Cancelled - 3: Failed - 4: Completed

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4

RealTimeHandlingType2

```
0
```

- 0: Polling
- 1: EventBased

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Polling - 1: EventBased

Enumerated Values

Property	Value
anonymous	0
anonymous	1

ScanSortOption

```
0
```

- 0: StartTime
- 1: Name
- 2: WorkflowId

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: StartTime - 1: Name - 2: WorkflowId

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

SortingOrder2

```
0
```

- 0: Ascending
- 1: Descending

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Ascending - 1: Descending

Enumerated Values

Property	Value
anonymous	0
anonymous	1

TypeOfReport

```
0
```

- 0: Executive
- 1: Summary

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Executive - 1: Summary

Enumerated Values

Property	Value
anonymous	0
anonymous	1

GetScanByIdQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scan": {
    "result": 0,
    "responseKey": "string",
    "responseMessage": "string",
    "responseMessageParams": {
      "property1": "string",
      "property2": "string"
    },
    "id": "string",
    "name": "string",
    "startTime": "2019-08-24T14:15:22Z",
    "stopTime": "2019-08-24T14:15:22Z",
    "scanType": "string",
    "scanningState": 0,
    "scanStatusDto": {
      "pendingFilesCount": 0,
      "pendingFilesSize": 0,
      "allowedFilesCount": 0,
      "allowedFilesSize": 0,
      "blockedFilesCount": 0,
      "blockedFilesSize": 0,
      "sanitizedFilesCount": 0,
      "failedSanitizedFilesCount": 0,
      "remediatedFilesCount": 0,
      "totalFilesCount": 0,
      "totalFilesSize": 0,
      "processingFailedCount": 0,
      "processingFailedSize": 0,
      "processingFailedRemediationCount": 0,
      "processedFilesCount": 0,
      "processedFilesSize": 0,
      "canceledFilesCount": 0,
      "canceledFilesSize": 0
    },
    "scanStatsDto": {
      "infectedFilesCount": 0,
      "filesWithVulnerabilitiesCount": 0,
      "filesWithSensitiveDataCount": 0,
      "filesBlockedBySandboxCount": 0,
      "scannedFilesCount": 0,
      "skippedFilesScanCount": 0,
      "scanFilesDistribution": {
        "documentsEncryptedFilesCount": 0,
        "executablesFilesCount": 0,
```

```
"openSSlEncryptedFilesCount": 0,
        "pdfFilesCount": 0,
        "graphicalFilesCount": 0,
        "diskImageFilesCount": 0,
        "textFilesCount": 0,
        "mailFilesCount": 0,
        "applicationFilesCount": 0,
        "documentsFilesCount": 0,
        "otherFilesCount": 0,
        "archivesFilesCount": 0
      }
    },
    "realTimeStartDate": "2019-08-24T14:15:22Z",
    "scanSequenceDto": {
      "previousScanId": "string",
      "nextScanId": "string"
    },
    "scanPartition": "string",
    "workflowId": "string",
    "scanConfigurationId": "string",
    "errors": [
      "string"
    ],
    "priority": 0,
    "scanWorkflowSnapshotDtos": [
      {
        "id": "string",
        "scanId": "string",
        "workflow": {
          "id": "string",
          "scanPoolId": "string",
          "scanProcessFlow": 0,
          "isIdentityScanningEnabled": true,
          "scanIntervalDays": 0,
          "rules": [
            "string"
          ],
          "type": 0,
          "workflowSupportedTechnologies": {
            "dlp": {
              "status": 0
            },
            "deepCdr": {
              "status": 0
            },
            "sandBox": {
              "status": 0
            },
            "metascan": {
              "status": 0
            }
          },
          "name": "string",
```

```
"scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanPoolId": "string",
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "filters": {
            "id": "string",
            "minimumSize": 0,
            "maximumSize": 0,
            "minimumDateTime": "2019-08-24T14:15:22Z",
            "maximumDateTime": "2019-08-24T14:15:22Z"
          },
          "fileDefinition": {
            "blockingStrategy": 0,
            "configurableBlockingStrategy": {
              "shouldTreatFilesWithSensitiveDataAsBlocked": true,
              "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
            }
          },
          "remediations": [
            {
              "id": "string",
              "workflowId": "string",
              "isEnabled": true,
              "configuration": "string",
```

```
"handlingType": 0
            }
          ],
          "tenantId": "string"
        },
        "startTimestamp": 0,
        "stopTimestamp": 0,
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
```

```
"apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "storage": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        },
        "workflowStatistics": {
          "taggingStatistics": {
            "total": 0
          },
          "keepAllowedStatistics": {
            "total": 0
          },
          "keepBlockedStatistics": {
            "total": 0
          },
          "keepSanitizedStatistics": {
            "total": 0
          },
          "movedAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
```

```
"vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedSanitizedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
```

```
"statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
```

```
"tenantId": "string"
            }
          },
          "deleteAllowedStatistics": {
            "total": 0
          },
          "deleteBlockedStatistics": {
            "total": 0
          },
          "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
        },
        "account": {
          "id": "string",
          "name": "string",
          "description": "string",
          "credentials": "string",
          "protocolType": 0,
          "vendorType": 0,
          "autoDiscovery": true,
          "autodiscoveryDestinationGroupId": "string"
        },
        "realTimeStartDate": "2019-08-24T14:15:22Z"
      }
    ],
    "storageId": "string",
    "$type": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» scan	ScanResponse|null	false	none	none

StartScanCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scanId": "string",
  "scanIds": [
    "string"
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» scanId	string¦null	false	none	none
» scanIds	[string]¦null	false	none	none

AddScanParameters

```
{
  "storageId": "string",
  "name": "string",
  "folder": "string",
  "scanPartition": "string",
  "priority": 0,
  "storageIds": [
    "string"
  ],
  "workflowId": "string",
  "mdssBaseUrl": "string",
  "scanConfigDto": {
    "realTimeScanConfig": {
      "isEnabled": true,
      "startDate": "2019-08-24T14:15:22Z",
      "handlingType": 0
    }
  }
}
```


Properties

Name	Type	Required	Restrictions	Description
storageId	string¦null	false	none	This property is obsolete and will be removed in the future. Use StorageIds instead.
name	string	false	none	user-defined name for the scan
folder	string¦null	false	none	This property is obsolete and will be removed in the future. Use ScanPartition instead.
scanPartition	string¦null	false	none	partiton of storage where the scan will run
priority	integer(int32)	false	none	priority such as (low, medium, high)
storageIds	[string]¦null	false	none	id's of storages to be scanned
workflowId	string	true	none	associated workflow to be executed
mdssBaseUrl	string¦null	false	none	base URL for MDSS, if not provided will use Request.Host
scanConfigDto	ScanConfig¦null	false	none	none

ScanConfig

```
{
  "realTimeScanConfig": {
    "isEnabled": true,
    "startDate": "2019-08-24T14:15:22Z",
    "handlingType": 0
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
realTimeScanConfig	RealTimeScanConfig¦null	false	none	none

RealTimeScanConfig

```
{
  "isEnabled": true,
  "startDate": "2019-08-24T14:15:22Z",
  "handlingType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
isEnabled	boolean	false	none	none
startDate	string(date-time)¦null	false	none	none
handlingType	RealTimeHandlingType	false	none	- 0: Polling - 1: EventBased

DeleteScansCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "deletedScans": [
    {
      "id": "string",
      "storageId": "string",
      "workflowId": "string",
      "scanScheduleId": "string",
      "executionId": "string",
      "name": "string",
      "userId": "string",
      "scanConfigDto": {
        "realTimeScanConfig": {
          "isEnabled": true,
          "startDate": "2019-08-24T14:15:22Z",
          "handlingType": 0
        }
      },
      "discoveryState": 0,
      "scanningState": 0,
      "scanType": 0,
      "startTime": "2019-08-24T14:15:22Z",
      "stopTime": "2019-08-24T14:15:22Z",
      "scanPartition": "string",
      "priority": 0,
      "additionalProperties": "string",
      "tenantId": "string",
      "errors": [
        "string"
      ]
    }
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» deletedScans	[Scan]	false	none	none

Scan


```
{
  "id": "string",
  "storageId": "string",
  "workflowId": "string",
  "scanScheduleId": "string",
  "executionId": "string",
  "name": "string",
  "userId": "string",
  "scanConfigDto": {
    "realTimeScanConfig": {
      "isEnabled": true,
      "startDate": "2019-08-24T14:15:22Z",
      "handlingType": 0
    }
  },
  "discoveryState": 0,
  "scanningState": 0,
  "scanType": 0,
  "startTime": "2019-08-24T14:15:22Z",
  "stopTime": "2019-08-24T14:15:22Z",
  "scanPartition": "string",
  "priority": 0,
  "additionalProperties": "string",
  "tenantId": "string",
  "errors": [
    "string"
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
storageId	string	false	none	none
workflowId	string¦null	false	none	none
scanScheduleId	string¦null	false	none	none
executionId	string¦null	false	none	none
name	string¦null	false	none	none
userId	string¦null	false	none	none
scanConfigDto	ScanConfig	false	none	none
discoveryState	DiscoveryState	false	none	- 0: NotStarted - 1: Running - 2: Cancelled - 3: Failed - 4: Completed
scanningState	ScanningState	false	none	- 0: NotStarted - 1: Running - 2: Cancelled - 3: Failed - 4: Completed
scanType	ScanType	false	none	- 0: Instant - 1: Scheduled - 2: RealTime
startTime	string(date-time)	false	none	none
stopTime	string(date-time)	false	none	none
scanPartition	string¦null	false	none	none
priority	MessagePriority	false	none	- 0: Low - 1: Medium - 2: High
additionalProperties	string¦null	false	none	none
tenantId	string¦null	false	none	none
errors	[string]¦null	false	none	none

DiscoveryState

```
0
```

- 0: NotStarted
- 1: Running
- 2: Cancelled
- 3: Failed
- 4: Completed

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: NotStarted - 1: Running - 2: Cancelled - 3: Failed - 4: Completed

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4

GetLastCompletedScanQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scan": {
    "result": 0,
    "responseKey": "string",
    "responseMessage": "string",
    "responseMessageParams": {
      "property1": "string",
      "property2": "string"
    },
    "id": "string",
    "name": "string",
    "startTime": "2019-08-24T14:15:22Z",
    "stopTime": "2019-08-24T14:15:22Z",
    "scanType": "string",
    "scanningState": 0,
    "scanStatusDto": {
      "pendingFilesCount": 0,
      "pendingFilesSize": 0,
      "allowedFilesCount": 0,
      "allowedFilesSize": 0,
      "blockedFilesCount": 0,
      "blockedFilesSize": 0,
      "sanitizedFilesCount": 0,
      "failedSanitizedFilesCount": 0,
      "remediatedFilesCount": 0,
      "totalFilesCount": 0,
      "totalFilesSize": 0,
      "processingFailedCount": 0,
      "processingFailedSize": 0,
      "processingFailedRemediationCount": 0,
      "processedFilesCount": 0,
      "processedFilesSize": 0,
      "canceledFilesCount": 0,
      "canceledFilesSize": 0
    },
    "scanStatsDto": {
      "infectedFilesCount": 0,
      "filesWithVulnerabilitiesCount": 0,
      "filesWithSensitiveDataCount": 0,
      "filesBlockedBySandboxCount": 0,
      "scannedFilesCount": 0,
      "skippedFilesScanCount": 0,
      "scanFilesDistribution": {
        "documentsEncryptedFilesCount": 0,
        "executablesFilesCount": 0,
```

```
"openSSlEncryptedFilesCount": 0,
        "pdfFilesCount": 0,
        "graphicalFilesCount": 0,
        "diskImageFilesCount": 0,
        "textFilesCount": 0,
        "mailFilesCount": 0,
        "applicationFilesCount": 0,
        "documentsFilesCount": 0,
        "otherFilesCount": 0,
        "archivesFilesCount": 0
      }
    },
    "realTimeStartDate": "2019-08-24T14:15:22Z",
    "scanSequenceDto": {
      "previousScanId": "string",
      "nextScanId": "string"
    },
    "scanPartition": "string",
    "workflowId": "string",
    "scanConfigurationId": "string",
    "errors": [
      "string"
    ],
    "priority": 0,
    "scanWorkflowSnapshotDtos": [
      {
        "id": "string",
        "scanId": "string",
        "workflow": {
          "id": "string",
          "scanPoolId": "string",
          "scanProcessFlow": 0,
          "isIdentityScanningEnabled": true,
          "scanIntervalDays": 0,
          "rules": [
            "string"
          ],
          "type": 0,
          "workflowSupportedTechnologies": {
            "dlp": {
              "status": 0
            },
            "deepCdr": {
              "status": 0
            },
            "sandBox": {
              "status": 0
            },
            "metascan": {
              "status": 0
            }
          },
          "name": "string",
```

```
"scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanPoolId": "string",
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "filters": {
            "id": "string",
            "minimumSize": 0,
            "maximumSize": 0,
            "minimumDateTime": "2019-08-24T14:15:22Z",
            "maximumDateTime": "2019-08-24T14:15:22Z"
          },
          "fileDefinition": {
            "blockingStrategy": 0,
            "configurableBlockingStrategy": {
              "shouldTreatFilesWithSensitiveDataAsBlocked": true,
              "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
            }
          },
          "remediations": [
            {
              "id": "string",
              "workflowId": "string",
              "isEnabled": true,
              "configuration": "string",
```

```
"handlingType": 0
            }
          ],
          "tenantId": "string"
        },
        "startTimestamp": 0,
        "stopTimestamp": 0,
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
```

```
"apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "storage": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        },
        "workflowStatistics": {
          "taggingStatistics": {
            "total": 0
          },
          "keepAllowedStatistics": {
            "total": 0
          },
          "keepBlockedStatistics": {
            "total": 0
          },
          "keepSanitizedStatistics": {
            "total": 0
          },
          "movedAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
```

```
"vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedSanitizedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
```

```
"statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
```

```
"tenantId": "string"
            }
          },
          "deleteAllowedStatistics": {
            "total": 0
          },
          "deleteBlockedStatistics": {
            "total": 0
          },
          "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
        },
        "account": {
          "id": "string",
          "name": "string",
          "description": "string",
          "credentials": "string",
          "protocolType": 0,
          "vendorType": 0,
          "autoDiscovery": true,
          "autodiscoveryDestinationGroupId": "string"
        },
        "realTimeStartDate": "2019-08-24T14:15:22Z"
      }
    ],
    "storageId": "string",
    "$type": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» scan	ScanResponse|null	false	none	none

GetActiveScanQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scan": {
    "result": 0,
    "responseKey": "string",
    "responseMessage": "string",
    "responseMessageParams": {
      "property1": "string",
      "property2": "string"
    },
    "id": "string",
    "name": "string",
    "startTime": "2019-08-24T14:15:22Z",
    "stopTime": "2019-08-24T14:15:22Z",
    "scanType": "string",
    "scanningState": 0,
    "scanStatusDto": {
      "pendingFilesCount": 0,
      "pendingFilesSize": 0,
      "allowedFilesCount": 0,
      "allowedFilesSize": 0,
      "blockedFilesCount": 0,
      "blockedFilesSize": 0,
      "sanitizedFilesCount": 0,
      "failedSanitizedFilesCount": 0,
      "remediatedFilesCount": 0,
      "totalFilesCount": 0,
      "totalFilesSize": 0,
      "processingFailedCount": 0,
      "processingFailedSize": 0,
      "processingFailedRemediationCount": 0,
      "processedFilesCount": 0,
      "processedFilesSize": 0,
      "canceledFilesCount": 0,
      "canceledFilesSize": 0
    },
    "scanStatsDto": {
      "infectedFilesCount": 0,
      "filesWithVulnerabilitiesCount": 0,
      "filesWithSensitiveDataCount": 0,
      "filesBlockedBySandboxCount": 0,
      "scannedFilesCount": 0,
      "skippedFilesScanCount": 0,
      "scanFilesDistribution": {
        "documentsEncryptedFilesCount": 0,
        "executablesFilesCount": 0,
```

```
"openSSlEncryptedFilesCount": 0,
        "pdfFilesCount": 0,
        "graphicalFilesCount": 0,
        "diskImageFilesCount": 0,
        "textFilesCount": 0,
        "mailFilesCount": 0,
        "applicationFilesCount": 0,
        "documentsFilesCount": 0,
        "otherFilesCount": 0,
        "archivesFilesCount": 0
      }
    },
    "realTimeStartDate": "2019-08-24T14:15:22Z",
    "scanSequenceDto": {
      "previousScanId": "string",
      "nextScanId": "string"
    },
    "scanPartition": "string",
    "workflowId": "string",
    "scanConfigurationId": "string",
    "errors": [
      "string"
    ],
    "priority": 0,
    "scanWorkflowSnapshotDtos": [
      {
        "id": "string",
        "scanId": "string",
        "workflow": {
          "id": "string",
          "scanPoolId": "string",
          "scanProcessFlow": 0,
          "isIdentityScanningEnabled": true,
          "scanIntervalDays": 0,
          "rules": [
            "string"
          ],
          "type": 0,
          "workflowSupportedTechnologies": {
            "dlp": {
              "status": 0
            },
            "deepCdr": {
              "status": 0
            },
            "sandBox": {
              "status": 0
            },
            "metascan": {
              "status": 0
            }
          },
          "name": "string",
```

```
"scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanPoolId": "string",
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "filters": {
            "id": "string",
            "minimumSize": 0,
            "maximumSize": 0,
            "minimumDateTime": "2019-08-24T14:15:22Z",
            "maximumDateTime": "2019-08-24T14:15:22Z"
          },
          "fileDefinition": {
            "blockingStrategy": 0,
            "configurableBlockingStrategy": {
              "shouldTreatFilesWithSensitiveDataAsBlocked": true,
              "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
            }
          },
          "remediations": [
            {
              "id": "string",
              "workflowId": "string",
              "isEnabled": true,
              "configuration": "string",
```

```
"handlingType": 0
            }
          ],
          "tenantId": "string"
        },
        "startTimestamp": 0,
        "stopTimestamp": 0,
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
```

```
"apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "storage": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        },
        "workflowStatistics": {
          "taggingStatistics": {
            "total": 0
          },
          "keepAllowedStatistics": {
            "total": 0
          },
          "keepBlockedStatistics": {
            "total": 0
          },
          "keepSanitizedStatistics": {
            "total": 0
          },
          "movedAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
```

```
"vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedSanitizedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
```

```
"statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
```

```
"tenantId": "string"
            }
          },
          "deleteAllowedStatistics": {
            "total": 0
          },
          "deleteBlockedStatistics": {
            "total": 0
          },
          "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
        },
        "account": {
          "id": "string",
          "name": "string",
          "description": "string",
          "credentials": "string",
          "protocolType": 0,
          "vendorType": 0,
          "autoDiscovery": true,
          "autodiscoveryDestinationGroupId": "string"
        },
        "realTimeStartDate": "2019-08-24T14:15:22Z"
      }
    ],
    "storageId": "string",
    "$type": "string",
    "serviceName": "string",
    "storageName": "string",
    "currentProcessingFilePath": "string",
    "currentProcessingFileProgress": "string",
    "discoveryState": 0,
    "statusMessages": [
      {
        "message": "string",
        "severity": 0,
        "params": {
          "property1": "string",
          "property2": "string"
        }
      }
    ]
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» scan	ActiveScanResponse¦null	false	none	none

ActiveScanResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "id": "string",
  "name": "string",
  "startTime": "2019-08-24T14:15:22Z",
  "stopTime": "2019-08-24T14:15:22Z",
  "scanType": "string",
  "scanningState": 0,
  "scanStatusDto": {
    "pendingFilesCount": 0,
    "pendingFilesSize": 0,
    "allowedFilesCount": 0,
    "allowedFilesSize": 0,
    "blockedFilesCount": 0,
    "blockedFilesSize": 0,
    "sanitizedFilesCount": 0,
    "failedSanitizedFilesCount": 0,
    "remediatedFilesCount": 0,
    "totalFilesCount": 0,
    "totalFilesSize": 0,
    "processingFailedCount": 0,
    "processingFailedSize": 0,
    "processingFailedRemediationCount": 0,
    "processedFilesCount": 0,
    "processedFilesSize": 0,
    "canceledFilesCount": 0,
    "canceledFilesSize": 0
  },
  "scanStatsDto": {
    "infectedFilesCount": 0,
    "filesWithVulnerabilitiesCount": 0,
    "filesWithSensitiveDataCount": 0,
    "filesBlockedBySandboxCount": 0,
    "scannedFilesCount": 0,
    "skippedFilesScanCount": 0,
    "scanFilesDistribution": {
      "documentsEncryptedFilesCount": 0,
      "executablesFilesCount": 0,
      "audioVideoFilesCount": 0,
      "openSSlEncryptedFilesCount": 0,
      "pdfFilesCount": 0,
      "graphicalFilesCount": 0,
      "diskImageFilesCount": 0,
      "textFilesCount": 0,
      "mailFilesCount": 0,
      "applicationFilesCount": 0,
```

```
"otherFilesCount": 0,
      "archivesFilesCount": 0
    }
  },
  "realTimeStartDate": "2019-08-24T14:15:22Z",
  "scanSequenceDto": {
    "previousScanId": "string",
    "nextScanId": "string"
  },
  "scanPartition": "string",
  "workflowId": "string",
  "scanConfigurationId": "string",
  "errors": [
    "string"
  ],
  "priority": 0,
  "scanWorkflowSnapshotDtos": [
    {
      "id": "string",
      "scanId": "string",
      "workflow": {
        "id": "string",
        "scanPoolId": "string",
        "scanProcessFlow": 0,
        "isIdentityScanningEnabled": true,
        "scanIntervalDays": 0,
        "rules": [
          "string"
        ],
        "type": 0,
        "workflowSupportedTechnologies": {
          "dlp": {
            "status": 0
          },
          "deepCdr": {
            "status": 0
          },
          "sandBox": {
            "status": 0
          },
          "metascan": {
            "status": 0
          }
        },
        "name": "string",
        "userAgent": "string",
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
```

```
"scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanPoolId": "string",
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "filters": {
          "id": "string",
          "minimumSize": 0,
          "maximumSize": 0,
          "minimumDateTime": "2019-08-24T14:15:22Z",
          "maximumDateTime": "2019-08-24T14:15:22Z"
        },
        "fileDefinition": {
          "blockingStrategy": 0,
          "configurableBlockingStrategy": {
            "shouldTreatFilesWithSensitiveDataAsBlocked": true,
            "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
          }
        },
        "remediations": [
          {
            "id": "string",
            "workflowId": "string",
            "isEnabled": true,
            "configuration": "string",
            "type": 0,
            "handlingType": 0
          }
        ],
        "tenantId": "string"
      },
      "startTimestamp": 0,
      "stopTimestamp": 0,
```

```
"id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "failOverScanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "failOverScanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "storage": {
        "id": "string",
```

```
"groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      },
      "workflowStatistics": {
        "taggingStatistics": {
          "total": 0
        },
        "keepAllowedStatistics": {
          "total": 0
        },
        "keepBlockedStatistics": {
          "total": 0
        },
        "keepSanitizedStatistics": {
          "total": 0
        },
        "movedAllowedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
```

```
"tenantId": "string"
          }
        },
        "movedSanitizedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "movedBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
```

```
"total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "copyBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "deleteAllowedStatistics": {
          "total": 0
        },
        "deleteBlockedStatistics": {
```

```
},
        "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
      },
      "account": {
        "id": "string",
        "name": "string",
        "description": "string",
        "credentials": "string",
        "protocolType": 0,
        "vendorType": 0,
        "autoDiscovery": true,
        "autodiscoveryDestinationGroupId": "string"
      },
      "realTimeStartDate": "2019-08-24T14:15:22Z"
    }
  ],
  "storageId": "string",
  "$type": "string",
  "serviceName": "string",
  "storageName": "string",
  "currentProcessingFilePath": "string",
  "currentProcessingFileProgress": "string",
  "discoveryState": 0,
  "statusMessages": [
    {
      "message": "string",
      "severity": 0,
      "params": {
        "property1": "string",
        "property2": "string"
      }
    }
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	ScanResponse	false	none	none

and - discriminator: $type

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» serviceName	string	false	none	none
» storageName	string	false	none	none
» currentProcessingFilePath	string	false	none	none
» currentProcessingFileProgress	string	false	none	none
» discoveryState	DiscoveryState	false	none	- 0: NotStarted - 1: Running - 2: Cancelled - 3: Failed - 4: Completed
» statusMessages	[MessageTemplate]	false	none	none
» $type	string	true	none	none

MessageTemplate

```
{
  "message": "string",
  "severity": 0,
  "params": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
message	string	false	none	none
severity	MessageSeverity	false	none	- 0: Information - 1: Warning - 2: Error
params	object¦null	false	none	none
» **additionalProperties**	string	false	none	none

MessageSeverity

```
0
```

- 0: Information
- 1: Warning
- 2: Error

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Information - 1: Warning - 2: Error

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

ActiveRealTimeScanResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "id": "string",
  "name": "string",
  "startTime": "2019-08-24T14:15:22Z",
  "stopTime": "2019-08-24T14:15:22Z",
  "scanType": "string",
  "scanningState": 0,
  "scanStatusDto": {
    "pendingFilesCount": 0,
    "pendingFilesSize": 0,
    "allowedFilesCount": 0,
    "allowedFilesSize": 0,
    "blockedFilesCount": 0,
    "blockedFilesSize": 0,
    "sanitizedFilesCount": 0,
    "failedSanitizedFilesCount": 0,
    "remediatedFilesCount": 0,
    "totalFilesCount": 0,
    "totalFilesSize": 0,
    "processingFailedCount": 0,
    "processingFailedSize": 0,
    "processingFailedRemediationCount": 0,
    "processedFilesCount": 0,
    "processedFilesSize": 0,
    "canceledFilesCount": 0,
    "canceledFilesSize": 0
  },
  "scanStatsDto": {
    "infectedFilesCount": 0,
    "filesWithVulnerabilitiesCount": 0,
    "filesWithSensitiveDataCount": 0,
    "filesBlockedBySandboxCount": 0,
    "scannedFilesCount": 0,
    "skippedFilesScanCount": 0,
    "scanFilesDistribution": {
      "documentsEncryptedFilesCount": 0,
      "executablesFilesCount": 0,
      "audioVideoFilesCount": 0,
      "openSSlEncryptedFilesCount": 0,
      "pdfFilesCount": 0,
      "graphicalFilesCount": 0,
      "diskImageFilesCount": 0,
      "textFilesCount": 0,
      "mailFilesCount": 0,
      "applicationFilesCount": 0,
```

```
"otherFilesCount": 0,
      "archivesFilesCount": 0
    }
  },
  "realTimeStartDate": "2019-08-24T14:15:22Z",
  "scanSequenceDto": {
    "previousScanId": "string",
    "nextScanId": "string"
  },
  "scanPartition": "string",
  "workflowId": "string",
  "scanConfigurationId": "string",
  "errors": [
    "string"
  ],
  "priority": 0,
  "scanWorkflowSnapshotDtos": [
    {
      "id": "string",
      "scanId": "string",
      "workflow": {
        "id": "string",
        "scanPoolId": "string",
        "scanProcessFlow": 0,
        "isIdentityScanningEnabled": true,
        "scanIntervalDays": 0,
        "rules": [
          "string"
        ],
        "type": 0,
        "workflowSupportedTechnologies": {
          "dlp": {
            "status": 0
          },
          "deepCdr": {
            "status": 0
          },
          "sandBox": {
            "status": 0
          },
          "metascan": {
            "status": 0
          }
        },
        "name": "string",
        "userAgent": "string",
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
```

```
"scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanPoolId": "string",
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "filters": {
          "id": "string",
          "minimumSize": 0,
          "maximumSize": 0,
          "minimumDateTime": "2019-08-24T14:15:22Z",
          "maximumDateTime": "2019-08-24T14:15:22Z"
        },
        "fileDefinition": {
          "blockingStrategy": 0,
          "configurableBlockingStrategy": {
            "shouldTreatFilesWithSensitiveDataAsBlocked": true,
            "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
          }
        },
        "remediations": [
          {
            "id": "string",
            "workflowId": "string",
            "isEnabled": true,
            "configuration": "string",
            "type": 0,
            "handlingType": 0
          }
        ],
        "tenantId": "string"
      },
      "startTimestamp": 0,
      "stopTimestamp": 0,
```

```
"id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "failOverScanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "failOverScanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "storage": {
        "id": "string",
```

```
"groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
          "name": "string",
          "categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      },
      "workflowStatistics": {
        "taggingStatistics": {
          "total": 0
        },
        "keepAllowedStatistics": {
          "total": 0
        },
        "keepBlockedStatistics": {
          "total": 0
        },
        "keepSanitizedStatistics": {
          "total": 0
        },
        "movedAllowedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
```

```
"tenantId": "string"
          }
        },
        "movedSanitizedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "movedBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
```

```
"total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "copyBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "deleteAllowedStatistics": {
          "total": 0
        },
        "deleteBlockedStatistics": {
```

```
},
        "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
      },
      "account": {
        "id": "string",
        "name": "string",
        "description": "string",
        "credentials": "string",
        "protocolType": 0,
        "vendorType": 0,
        "autoDiscovery": true,
        "autodiscoveryDestinationGroupId": "string"
      },
      "realTimeStartDate": "2019-08-24T14:15:22Z"
    }
  ],
  "storageId": "string",
  "$type": "string",
  "serviceName": "string",
  "storageName": "string",
  "currentProcessingFilePath": "string",
  "currentProcessingFileProgress": "string",
  "discoveryState": 0,
  "statusMessages": [
    {
      "message": "string",
      "severity": 0,
      "params": {
        "property1": "string",
        "property2": "string"
      }
    }
  ],
  "isRealTimeEnabled": true,
  "handlingType": 0
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	ActiveScanResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» isRealTimeEnabled	boolean	false	none	none
» handlingType	RealTimeHandlingType	false	none	- 0: Polling - 1: EventBased

GetRealTimeProcessingScanQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scan": {
    "result": 0,
    "responseKey": "string",
    "responseMessage": "string",
    "responseMessageParams": {
      "property1": "string",
      "property2": "string"
    },
    "id": "string",
    "name": "string",
    "startTime": "2019-08-24T14:15:22Z",
    "stopTime": "2019-08-24T14:15:22Z",
    "scanType": "string",
    "scanningState": 0,
    "scanStatusDto": {
      "pendingFilesCount": 0,
      "pendingFilesSize": 0,
      "allowedFilesCount": 0,
      "allowedFilesSize": 0,
      "blockedFilesCount": 0,
      "blockedFilesSize": 0,
      "sanitizedFilesCount": 0,
      "failedSanitizedFilesCount": 0,
      "remediatedFilesCount": 0,
      "totalFilesCount": 0,
      "totalFilesSize": 0,
      "processingFailedCount": 0,
      "processingFailedSize": 0,
      "processingFailedRemediationCount": 0,
      "processedFilesCount": 0,
      "processedFilesSize": 0,
      "canceledFilesCount": 0,
      "canceledFilesSize": 0
    },
    "scanStatsDto": {
      "infectedFilesCount": 0,
      "filesWithVulnerabilitiesCount": 0,
      "filesWithSensitiveDataCount": 0,
      "filesBlockedBySandboxCount": 0,
      "scannedFilesCount": 0,
      "skippedFilesScanCount": 0,
      "scanFilesDistribution": {
        "documentsEncryptedFilesCount": 0,
        "executablesFilesCount": 0,
```

```
"openSSlEncryptedFilesCount": 0,
        "pdfFilesCount": 0,
        "graphicalFilesCount": 0,
        "diskImageFilesCount": 0,
        "textFilesCount": 0,
        "mailFilesCount": 0,
        "applicationFilesCount": 0,
        "documentsFilesCount": 0,
        "otherFilesCount": 0,
        "archivesFilesCount": 0
      }
    },
    "realTimeStartDate": "2019-08-24T14:15:22Z",
    "scanSequenceDto": {
      "previousScanId": "string",
      "nextScanId": "string"
    },
    "scanPartition": "string",
    "workflowId": "string",
    "scanConfigurationId": "string",
    "errors": [
      "string"
    ],
    "priority": 0,
    "scanWorkflowSnapshotDtos": [
      {
        "id": "string",
        "scanId": "string",
        "workflow": {
          "id": "string",
          "scanPoolId": "string",
          "scanProcessFlow": 0,
          "isIdentityScanningEnabled": true,
          "scanIntervalDays": 0,
          "rules": [
            "string"
          ],
          "type": 0,
          "workflowSupportedTechnologies": {
            "dlp": {
              "status": 0
            },
            "deepCdr": {
              "status": 0
            },
            "sandBox": {
              "status": 0
            },
            "metascan": {
              "status": 0
            }
          },
          "name": "string",
```

```
"scanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "failOverScanPoolId": "string",
          "failOverScanPool": {
            "id": "string",
            "name": "string",
            "scanPoolType": 0,
            "isDefault": true,
            "scanInstances": [
              {
                "id": "string",
                "scanPoolId": "string",
                "url": "string",
                "apiKey": "string",
                "timeout": "string",
                "scanInstanceType": 0
              }
            ]
          },
          "filters": {
            "id": "string",
            "minimumSize": 0,
            "maximumSize": 0,
            "minimumDateTime": "2019-08-24T14:15:22Z",
            "maximumDateTime": "2019-08-24T14:15:22Z"
          },
          "fileDefinition": {
            "blockingStrategy": 0,
            "configurableBlockingStrategy": {
              "shouldTreatFilesWithSensitiveDataAsBlocked": true,
              "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
            }
          },
          "remediations": [
            {
              "id": "string",
              "workflowId": "string",
              "isEnabled": true,
              "configuration": "string",
```

```
"handlingType": 0
            }
          ],
          "tenantId": "string"
        },
        "startTimestamp": 0,
        "stopTimestamp": 0,
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "failOverScanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
```

```
"apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ],
        "storage": {
          "id": "string",
          "name": "string",
          "groupId": "string",
          "clientId": "string",
          "accountId": "string",
          "source": "string",
          "storageFunctionality": "string",
          "storageType": {
            "name": "string",
            "categoryType": 0,
            "protocolType": 0,
            "vendorType": 0
          },
          "storageStatus": {
            "message": "string",
            "statusType": 0
          },
          "workflowId": "string",
          "lastUpdated": "2019-08-24T14:15:22Z",
          "tenantId": "string"
        },
        "workflowStatistics": {
          "taggingStatistics": {
            "total": 0
          },
          "keepAllowedStatistics": {
            "total": 0
          },
          "keepBlockedStatistics": {
            "total": 0
          },
          "keepSanitizedStatistics": {
            "total": 0
          },
          "movedAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
```

```
"vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedSanitizedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "movedBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
```

```
"statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyAllowedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
              "lastUpdated": "2019-08-24T14:15:22Z",
              "tenantId": "string"
            }
          },
          "copyBlockedStatistics": {
            "total": 0,
            "destinationStorageInformation": {
              "id": "string",
              "name": "string",
              "groupId": "string",
              "clientId": "string",
              "accountId": "string",
              "source": "string",
              "storageFunctionality": "string",
              "storageType": {
                "name": "string",
                "categoryType": 0,
                "protocolType": 0,
                "vendorType": 0
              },
              "storageStatus": {
                "message": "string",
                "statusType": 0
              },
              "workflowId": "string",
```

```
"tenantId": "string"
            }
          },
          "deleteAllowedStatistics": {
            "total": 0
          },
          "deleteBlockedStatistics": {
            "total": 0
          },
          "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
        },
        "account": {
          "id": "string",
          "name": "string",
          "description": "string",
          "credentials": "string",
          "protocolType": 0,
          "vendorType": 0,
          "autoDiscovery": true,
          "autodiscoveryDestinationGroupId": "string"
        },
        "realTimeStartDate": "2019-08-24T14:15:22Z"
      }
    ],
    "storageId": "string",
    "$type": "string",
    "isRealTimeEnabled": true,
    "handlingType": 0
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» scan	RealTimeScanResponse¦null	false	none	none

GetAllScanInstancesCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "scanPoolId": "string",
      "url": "string",
      "apiKey": "string",
      "timeout": "string",
      "scanInstanceType": 0
    }
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entries	[ScanInstance]¦null	false	none	none

GetScanInstanceByIdCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entry": {
    "id": "string",
    "scanPoolId": "string",
    "url": "string",
    "apiKey": "string",
    "timeout": "string",
    "scanInstanceType": 0
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entry	ScanInstance¦null	false	none	none

AddScanInstanceCommandResponse


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scanInstanceId": "string"
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» scanInstanceId	string¦null	false	none	none

AddScanInstanceParameters

```
{
  "scanPoolId": "string",
  "url": "http://example.com",
  "apiKey": "string",
  "timeout": "14:15:22Z",
  "scanInstanceType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
scanPoolId	string	true	none	none
url	string(uri)	true	none	none
apiKey	string¦null	false	none	none
timeout	string(time)¦null	false	none	none
scanInstanceType	ScanInstanceType	true	none	- 0: MdCore - 1: MdCloud

UpdateScanInstanceCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

UpdateScanInstanceParameters


```
{
  "id": "string",
  "url": "http://example.com",
  "apiKey": "string",
  "timeout": "14:15:22Z"
}
```


Properties

Name	Type	Required	Restrictions	Description
id	string	true	none	none
url	string(uri)¦null	false	none	none
apiKey	string¦null	false	none	none
timeout	string(time)¦null	false	none	none

DeleteScanInstanceCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

TestScanInstanceCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

TestScanInstanceParameters

```
{
  "url": "string",
  "rules": [
    "string"
  ],
  "apiKey": "string",
  "userAgent": "string",
  "isCloudInstance": true
}
```

Properties

Name	Type	Required	Restrictions	Description
url	string	true	none	none
rules	[string]	true	none	none
apiKey	string¦null	false	none	none
userAgent	string¦null	false	none	none
isCloudInstance	boolean¦null	false	none	none

GetAllScanPoolsCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    }
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entries	[ScanPool]¦null	false	none	none

GetScanPoolByIdCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entry": {
    "id": "string",
    "name": "string",
    "scanPoolType": 0,
    "isDefault": true,
    "scanInstances": [
      {
        "id": "string",
        "scanPoolId": "string",
        "url": "string",
        "apiKey": "string",
        "timeout": "string",
        "scanInstanceType": 0
      }
    ]
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entry	ScanPool¦null	false	none	none

AddScanPoolCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scanPoolId": "string"
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» scanPoolId	string¦null	false	none	none

AddScanPoolParameters


```
{
  "name": "string",
  "scanPoolType": 0
}
```


Properties

Name	Type	Required	Restrictions	Description
name	string	true	none	none
scanPoolType	ScanInstanceType	true	none	- 0: MdCore - 1: MdCloud

UpdateScanPoolCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

UpdateScanPoolParameters

```
{
  "id": "string",
  "name": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	true	none	none
name	string¦null	false	none	none

DeleteScanPoolCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

GetScanWorkflowSnapshotsFromScanQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "scanWorkflowSnapshots": [
    {
      "id": "string",
      "scanId": "string",
      "workflow": {
        "id": "string",
        "scanPoolId": "string",
        "scanProcessFlow": 0,
        "isIdentityScanningEnabled": true,
        "scanIntervalDays": 0,
        "rules": [
          "string"
        ],
        "type": 0,
        "workflowSupportedTechnologies": {
          "dlp": {
            "status": 0
          },
          "deepCdr": {
            "status": 0
          },
          "sandBox": {
            "status": 0
          },
          "metascan": {
            "status": 0
          }
        },
        "name": "string",
        "userAgent": "string",
        "scanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
```

```
},
        "failOverScanPoolId": "string",
        "failOverScanPool": {
          "id": "string",
          "name": "string",
          "scanPoolType": 0,
          "isDefault": true,
          "scanInstances": [
            {
              "id": "string",
              "scanPoolId": "string",
              "url": "string",
              "apiKey": "string",
              "timeout": "string",
              "scanInstanceType": 0
            }
          ]
        },
        "filters": {
          "id": "string",
          "minimumSize": 0,
          "maximumSize": 0,
          "minimumDateTime": "2019-08-24T14:15:22Z",
          "maximumDateTime": "2019-08-24T14:15:22Z"
        },
        "fileDefinition": {
          "blockingStrategy": 0,
          "configurableBlockingStrategy": {
            "shouldTreatFilesWithSensitiveDataAsBlocked": true,
            "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
          }
        },
        "remediations": [
          {
            "id": "string",
            "workflowId": "string",
            "isEnabled": true,
            "configuration": "string",
            "type": 0,
            "handlingType": 0
          }
        ],
        "tenantId": "string"
      },
      "startTimestamp": 0,
      "stopTimestamp": 0,
      "scanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
```

```
"scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "failOverScanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "failOverScanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ],
      "storage": {
        "id": "string",
        "name": "string",
        "groupId": "string",
        "clientId": "string",
        "accountId": "string",
        "source": "string",
        "storageFunctionality": "string",
        "storageType": {
```

```
"categoryType": 0,
          "protocolType": 0,
          "vendorType": 0
        },
        "storageStatus": {
          "message": "string",
          "statusType": 0
        },
        "workflowId": "string",
        "lastUpdated": "2019-08-24T14:15:22Z",
        "tenantId": "string"
      },
      "workflowStatistics": {
        "taggingStatistics": {
          "total": 0
        },
        "keepAllowedStatistics": {
          "total": 0
        },
        "keepBlockedStatistics": {
          "total": 0
        },
        "keepSanitizedStatistics": {
          "total": 0
        },
        "movedAllowedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "movedSanitizedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
```

```
"name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "movedBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "copyAllowedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
```

```
"source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "copyBlockedStatistics": {
          "total": 0,
          "destinationStorageInformation": {
            "id": "string",
            "name": "string",
            "groupId": "string",
            "clientId": "string",
            "accountId": "string",
            "source": "string",
            "storageFunctionality": "string",
            "storageType": {
              "name": "string",
              "categoryType": 0,
              "protocolType": 0,
              "vendorType": 0
            },
            "storageStatus": {
              "message": "string",
              "statusType": 0
            },
            "workflowId": "string",
            "lastUpdated": "2019-08-24T14:15:22Z",
            "tenantId": "string"
          }
        },
        "deleteAllowedStatistics": {
          "total": 0
        },
        "deleteBlockedStatistics": {
          "total": 0
        },
        "lastCalculatedStatisticsTime": "2019-08-24T14:15:22Z"
      },
      "account": {
        "id": "string",
        "name": "string",
```

```
"credentials": "string",
        "protocolType": 0,
        "vendorType": 0,
        "autoDiscovery": true,
        "autodiscoveryDestinationGroupId": "string"
      },
      "realTimeStartDate": "2019-08-24T14:15:22Z"
    }
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» scanWorkflowSnapshots	[ScanWorkflowSnapshot]	false	none	none

UpdateNotificationsConfigResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "newNotificationsConfig": {
    "id": "string",
    "emailNotificationsConfig": {
      "userRequestNotificationsConfig": {
        "isEnabled": true
      },
      "generatedReportNotificationsConfig": {
        "isEnabled": true,
        "recipientsType": 0
      },
      "blockedFileNotificationsConfig": {
        "isEnabled": true,
        "recipientsType": 0
      }
    },
    "webhookNotificationsConfig": {
      "scanCompletedNotificationsConfig": {
        "isEnabled": true
      },
      "fileScannedNotificationConfig": {
        "isEnabled": true,
        "notificationScope": 0
      },
      "destinationAddress": "string",
      "headers": {
        "property1": "string",
        "property2": "string"
      },
      "requestType": 0
    },
    "rabbitMqNotificationsConfig": {
      "fileScannedNotificationConfig": {
        "isEnabled": true,
        "notificationScope": 0
      },
      "rabbitMqUrl": "string",
      "headers": {
        "property1": "string",
        "property2": "string"
      }
    }
  },
  "initialNotificationsConfig": {
    "id": "string",
```

```
"userRequestNotificationsConfig": {
        "isEnabled": true
      },
      "generatedReportNotificationsConfig": {
        "isEnabled": true,
        "recipientsType": 0
      },
      "blockedFileNotificationsConfig": {
        "isEnabled": true,
        "recipientsType": 0
      }
    },
    "webhookNotificationsConfig": {
      "scanCompletedNotificationsConfig": {
        "isEnabled": true
      },
      "fileScannedNotificationConfig": {
        "isEnabled": true,
        "notificationScope": 0
      },
      "destinationAddress": "string",
      "headers": {
        "property1": "string",
        "property2": "string"
      },
      "requestType": 0
    },
    "rabbitMqNotificationsConfig": {
      "fileScannedNotificationConfig": {
        "isEnabled": true,
        "notificationScope": 0
      },
      "rabbitMqUrl": "string",
      "headers": {
        "property1": "string",
        "property2": "string"
      }
    }
  },
  "typeResults": [
    {
      "type": 0,
      "status": 0,
      "errorKey": "string",
      "errorMessage": "string"
    }
  ]
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» newNotificationsConfig	NotificationsConfig¦null	false	none	none
» initialNotificationsConfig	NotificationsConfig¦null	false	none	none
» typeResults	[NotificationTypeUpdateResult]¦null	false	none	none

NotificationsConfig

```
{
  "id": "string",
  "emailNotificationsConfig": {
    "userRequestNotificationsConfig": {
      "isEnabled": true
    },
    "generatedReportNotificationsConfig": {
      "isEnabled": true,
      "recipientsType": 0
    },
    "blockedFileNotificationsConfig": {
      "isEnabled": true,
      "recipientsType": 0
    }
  },
  "webhookNotificationsConfig": {
    "scanCompletedNotificationsConfig": {
      "isEnabled": true
    },
    "fileScannedNotificationConfig": {
      "isEnabled": true,
      "notificationScope": 0
    },
    "destinationAddress": "string",
    "headers": {
      "property1": "string",
      "property2": "string"
    },
    "requestType": 0
  },
  "rabbitMqNotificationsConfig": {
    "fileScannedNotificationConfig": {
      "isEnabled": true,
      "notificationScope": 0
    },
    "rabbitMqUrl": "string",
    "headers": {
      "property1": "string",
      "property2": "string"
    }
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
emailNotificationsConfig	EmailNotificationsConfig	false	none	none
webhookNotificationsConfig	WebhookNotificationsConfig	false	none	none
rabbitMqNotificationsConfig	RabbitMqNotificationsConfig	false	none	none

EmailNotificationsConfig

```
{
  "userRequestNotificationsConfig": {
    "isEnabled": true
  },
  "generatedReportNotificationsConfig": {
    "isEnabled": true,
    "recipientsType": 0
  },
  "blockedFileNotificationsConfig": {
    "isEnabled": true,
    "recipientsType": 0
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
userRequestNotificationsConfig	UserRequestNotificationsConfig	false	none	none
generatedReportNotificationsConfig	GeneratedReportNotificationsConfig	false	none	none
blockedFileNotificationsConfig	BlockedFileNotificationsConfig	false	none	none

UserRequestNotificationsConfig

```
{
  "isEnabled": true
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	NotificationsConfigBase	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

NotificationsConfigBase

```
{
  "isEnabled": true
}
```

Properties

Name	Type	Required	Restrictions	Description
isEnabled	boolean	false	none	none

GeneratedReportNotificationsConfig

```
{
  "isEnabled": true,
  "recipientsType": 0
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	NotificationsConfigBase	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» recipientsType	GeneratedReportNotificationsRecipientsType	false	none	- 0: OwnerOnly - 1: AllRegisteredUsers

GeneratedReportNotificationsRecipientsType

```
0
```

- 0: OwnerOnly
- 1: AllRegisteredUsers

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: OwnerOnly - 1: AllRegisteredUsers

Enumerated Values

Property	Value
anonymous	0
anonymous	1

BlockedFileNotificationsConfig

```
{
  "isEnabled": true,
  "recipientsType": 0
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	NotificationsConfigBase	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» recipientsType	BlockedFileNotificationsRecipientsType	false	none	- 0: AllRegisteredUsers - 1: FileOwner - 2: FileOwnerAndAllRegisteredUsers

BlockedFileNotificationsRecipientsType

```
0
```

- 0: AllRegisteredUsers
- 1: FileOwner
- 2: FileOwnerAndAllRegisteredUsers

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: AllRegisteredUsers - 1: FileOwner - 2: FileOwnerAndAllRegisteredUsers

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

WebhookNotificationsConfig

```
{
  "scanCompletedNotificationsConfig": {
    "isEnabled": true
  },
  "fileScannedNotificationConfig": {
    "isEnabled": true,
    "notificationScope": 0
  },
  "destinationAddress": "string",
  "headers": {
    "property1": "string",
    "property2": "string"
  },
  "requestType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
scanCompletedNotificationsConfig	ScanCompletedNotificationsConfig	false	none	none
fileScannedNotificationConfig	FileScannedNotificationConfig	false	none	none
destinationAddress	string¦null	false	none	none
headers	object¦null	false	none	none
» **additionalProperties**	string	false	none	none
requestType	RequestType	false	none	- 0: Post - 1: Put

ScanCompletedNotificationsConfig

```
{
  "isEnabled": true
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	NotificationsConfigBase	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

FileScannedNotificationConfig

```
{
  "isEnabled": true,
  "notificationScope": 0
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	NotificationsConfigBase	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» notificationScope	FileNotificationScope	false	none	- 0: BlockedFilesOnly - 1: AllFiles

FileNotificationScope

```
0
```

- 0: BlockedFilesOnly
- 1: AllFiles

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: BlockedFilesOnly - 1: AllFiles

Enumerated Values

Property	Value
anonymous	0
anonymous	1

RequestType

```
0
```

- 0: Post
- 1: Put

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Post - 1: Put

Enumerated Values

Property	Value
anonymous	0
anonymous	1

RabbitMqNotificationsConfig

```
{
  "fileScannedNotificationConfig": {
    "isEnabled": true,
    "notificationScope": 0
  },
  "rabbitMqUrl": "string",
  "headers": {
    "property1": "string",
    "property2": "string"
  }
}
```


Properties

Name	Type	Required	Restrictions	Description
fileScannedNotificationConfig	FileScannedNotificationConfig	false	none	none
rabbitMqUrl	string¦null	false	none	none
headers	object¦null	false	none	none
» **additionalProperties**	string	false	none	none

NotificationTypeUpdateResult


```
{
  "type": 0,
  "status": 0,
  "errorKey": "string",
  "errorMessage": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
type	NotificationType	false	none	- 0: Email - 1: Webhook - 2: RabbitMq
status	UpdateStatus	false	none	- 0: Success - 1: Failed - 2: Skipped
errorKey	string¦null	false	none	none
errorMessage	string¦null	false	none	none

NotificationType

```
0
```

- 0: Email
- 1: Webhook
- 2: RabbitMq

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Email - 1: Webhook - 2: RabbitMq

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

UpdateStatus

```
0
```

- 0: Success
- 1: Failed
- 2: Skipped

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Success - 1: Failed - 2: Skipped

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

NotificationsConfigParameters

```
{
  "emailNotificationsConfig": {
    "generatedReportNotificationsConfig": {
      "isEnabled": true,
      "recipientsType": 0
    },
    "userRequestNotificationsConfig": {
      "isEnabled": true
    },
    "blockedFileNotificationsConfig": {
      "isEnabled": true,
      "recipientsType": 0
    }
  },
  "webhookNotificationsConfig": {
    "scanCompletedNotificationsConfig": {
      "isEnabled": true
    },
    "fileScannedNotificationConfig": {
      "isEnabled": true,
      "notificationScope": 0
    },
    "destinationAddress": "string",
    "headers": {
      "property1": "string",
      "property2": "string"
    },
    "requestType": 0
  },
  "rabbitMqNotificationsConfig": {
    "fileScannedNotificationConfig": {
      "isEnabled": true,
      "notificationScope": 0
    },
    "rabbitMqUrl": "string",
    "headers": {
      "property1": "string",
      "property2": "string"
    }
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
emailNotificationsConfig	EmailNotificationsConfig2¦null	false	none	none
webhookNotificationsConfig	WebhookNotificationsConfig2¦null	false	none	none
rabbitMqNotificationsConfig	RabbitMqNotificationsConfig2¦null	false	none	none

EmailNotificationsConfig2

```
{
  "generatedReportNotificationsConfig": {
    "isEnabled": true,
    "recipientsType": 0
  },
  "userRequestNotificationsConfig": {
    "isEnabled": true
  },
  "blockedFileNotificationsConfig": {
    "isEnabled": true,
    "recipientsType": 0
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
generatedReportNotificationsConfig	GeneratedReportNotificationsConfig2¦null	false	none	none
userRequestNotificationsConfig	UserRequestNotificationsConfig2¦null	false	none	none
blockedFileNotificationsConfig	BlockedFileNotificationsConfig2¦null	false	none	none

GeneratedReportNotificationsConfig2

```
{
  "isEnabled": true,
  "recipientsType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
isEnabled	boolean	false	none	none
recipientsType	GeneratedReportNotificationsRecipientsType2	false	none	- 0: OwnerOnly - 1: AllRegisteredUsers

GeneratedReportNotificationsRecipientsType2

```
0
```

- 0: OwnerOnly
- 1: AllRegisteredUsers

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: OwnerOnly - 1: AllRegisteredUsers

Enumerated Values

Property	Value
anonymous	0
anonymous	1

UserRequestNotificationsConfig2

```
{
  "isEnabled": true
}
```

Properties

Name	Type	Required	Restrictions	Description
isEnabled	boolean	false	none	none

BlockedFileNotificationsConfig2

```
{
  "isEnabled": true,
  "recipientsType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
isEnabled	boolean	false	none	none
recipientsType	BlockedFileNotificationsRecipientsType2	false	none	- 0: AllRegisteredUsers - 1: FileOwner - 2: FileOwnerAndAllRegisteredUsers

BlockedFileNotificationsRecipientsType2

```
0
```

- 0: AllRegisteredUsers
- 1: FileOwner
- 2: FileOwnerAndAllRegisteredUsers

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: AllRegisteredUsers - 1: FileOwner - 2: FileOwnerAndAllRegisteredUsers

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

WebhookNotificationsConfig2

```
{
  "scanCompletedNotificationsConfig": {
    "isEnabled": true
  },
  "fileScannedNotificationConfig": {
    "isEnabled": true,
    "notificationScope": 0
  },
  "destinationAddress": "string",
  "headers": {
    "property1": "string",
    "property2": "string"
  },
  "requestType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
scanCompletedNotificationsConfig	ScanCompletedNotificationsConfig2¦null	false	none	none
fileScannedNotificationConfig	FileScannedNotificationConfig2¦null	false	none	none
destinationAddress	string¦null	false	none	none
headers	object¦null	false	none	none
» **additionalProperties**	string	false	none	none
requestType	RequestType¦null	false	none	- 0: Post - 1: Put

ScanCompletedNotificationsConfig2

```
{
  "isEnabled": true
}
```

Properties

Name	Type	Required	Restrictions	Description
isEnabled	boolean	true	none	none

FileScannedNotificationConfig2

```
{
  "isEnabled": true,
  "notificationScope": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
isEnabled	boolean	true	none	none
notificationScope	FileNotificationScope	true	none	- 0: BlockedFilesOnly - 1: AllFiles

RabbitMqNotificationsConfig2

```
{
  "fileScannedNotificationConfig": {
    "isEnabled": true,
    "notificationScope": 0
  },
  "rabbitMqUrl": "string",
  "headers": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
fileScannedNotificationConfig	FileScannedNotificationConfig2¦null	false	none	none
rabbitMqUrl	string¦null	false	none	none
headers	object¦null	false	none	none
» **additionalProperties**	string	false	none	none

UpdateSmtpConfigCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "newSmtpConfig": {
    "host": "string",
    "baseUrl": "string",
    "port": 0,
    "senderAddress": "string",
    "senderName": "string",
    "domain": "string",
    "username": "string",
    "secureSocketOption": 0,
    "ignoreCertWarnings": true,
    "isSmtpEnabled": true
  },
  "oldSmtpConfig": {
    "host": "string",
    "baseUrl": "string",
    "port": 0,
    "senderAddress": "string",
    "senderName": "string",
    "domain": "string",
    "username": "string",
    "secureSocketOption": 0,
    "ignoreCertWarnings": true,
    "isSmtpEnabled": true
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» newSmtpConfig	SmtpConfig¦null	false	none	none
» oldSmtpConfig	SmtpConfig¦null	false	none	none

SmtpConfig

```
{
  "host": "string",
  "baseUrl": "string",
  "port": 0,
  "senderAddress": "string",
  "senderName": "string",
  "domain": "string",
  "username": "string",
  "secureSocketOption": 0,
  "ignoreCertWarnings": true,
  "isSmtpEnabled": true
}
```

Properties

Name	Type	Required	Restrictions	Description
host	string	false	none	none
baseUrl	string	false	none	none
port	integer(int32)	false	none	none
senderAddress	string	false	none	none
senderName	string	false	none	none
domain	string	false	none	none
username	string	false	none	none
secureSocketOption	SecureSocketOption	false	none	- 0: None - 1: Auto - 2: SslOnConnect - 3: StartTls - 4: StartTlsWhenAvailable
ignoreCertWarnings	boolean	false	none	none
isSmtpEnabled	boolean	false	none	none

SecureSocketOption

```
0
```

- 0: None
- 1: Auto
- 2: SslOnConnect
- 3: StartTls
- 4: StartTlsWhenAvailable

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: None - 1: Auto - 2: SslOnConnect - 3: StartTls - 4: StartTlsWhenAvailable

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4

UpdateSmtpConfigParameters

```
{
  "host": "string",
  "port": 0,
  "baseUrl": "http://example.com",
  "senderAddress": "string",
  "senderName": "string",
  "domain": "string",
  "username": "string",
  "password": "string",
  "secureSocketOption": 0,
  "ignoreCertWarnings": true,
  "isSmtpEnabled": true
}
```

Properties

Name	Type	Required	Restrictions	Description
host	string¦null	false	none	none
port	integer(int32)¦null	false	none	none
baseUrl	string(uri)¦null	false	none	none
senderAddress	string¦null	false	none	none
senderName	string¦null	false	none	none
domain	string¦null	false	none	none
username	string¦null	false	none	none
password	string¦null	false	none	none
secureSocketOption	SecureSocketOption¦null	false	none	- 0: None - 1: Auto - 2: SslOnConnect - 3: StartTls - 4: StartTlsWhenAvailable
ignoreCertWarnings	boolean¦null	false	none	none
isSmtpEnabled	boolean¦null	false	none	none

GetSmtpConfigResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "host": "string",
  "port": 0,
  "baseUrl": "string",
  "senderAddress": "string",
  "senderName": "string",
  "domain": "string",
  "username": "string",
  "secureSocketOption": 0,
  "ignoreCertWarnings": true,
  "isSmtpEnabled": true
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» host	string¦null	false	none	none
» port	integer(int32)	false	none	none
» baseUrl	string¦null	false	none	none
» senderAddress	string¦null	false	none	none
» senderName	string¦null	false	none	none
» domain	string¦null	false	none	none
» username	string¦null	false	none	none
» secureSocketOption	SecureSocketOption	false	none	- 0: None - 1: Auto - 2: SslOnConnect - 3: StartTls - 4: StartTlsWhenAvailable
» ignoreCertWarnings	boolean	false	none	none
» isSmtpEnabled	boolean	false	none	none

GetNotificationsConfigResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "emailNotificationsConfig": {
    "generatedReportNotificationsConfig": {
      "isEnabled": true,
      "recipientsType": 0
    },
    "userRequestNotificationsConfig": {
      "isEnabled": true
    },
    "blockedFileNotificationsConfig": {
      "isEnabled": true,
      "recipientsType": 0
    }
  },
  "webhookNotificationsConfig": {
    "scanCompletedNotificationsConfig": {
      "isEnabled": true
    },
    "fileScannedNotificationConfig": {
      "isEnabled": true,
      "notificationScope": 0
    },
    "destinationAddress": "string",
    "headers": {
      "property1": "string",
      "property2": "string"
    },
    "requestType": 0
  },
  "rabbitMqNotificationsConfig": {
    "fileScannedNotificationConfig": {
      "isEnabled": true,
      "notificationScope": 0
    },
    "rabbitMqUrl": "string",
    "headers": {
      "property1": "string",
      "property2": "string"
    }
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» emailNotificationsConfig	EmailNotificationsConfig2¦null	false	none	none
» webhookNotificationsConfig	WebhookNotificationsConfig2¦null	false	none	none
» rabbitMqNotificationsConfig	RabbitMqNotificationsConfig2¦null	false	none	none

OnlineActivationParameters

```
{
  "key": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
key	string	true	none	none

Deactivation

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

RetentionConfigParameters


```
{
  "retentionAuditDays": 1,
  "reportRetention": {
    "isEnabled": true,
    "retentionDays": 1
  },
  "rtpHistoryRetention": {
    "isEnabled": true,
    "retentionDays": 1
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
retentionAuditDays	integer(int32)	false	none	none
reportRetention	ReportRetentionParameters	true	none	none
rtpHistoryRetention	ReportRetentionParameters	true	none	none

ReportRetentionParameters

```
{
  "isEnabled": true,
  "retentionDays": 1
}
```

Properties

Name	Type	Required	Restrictions	Description
isEnabled	boolean	false	none	none
retentionDays	integer(int32)	false	none	none

GetRetentionConfig

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "retentionAuditDays": 0,
  "reportRetentionConfiguration": {
    "isEnabled": true,
    "retentionDays": 0
  },
  "rtpHistoryRetentionConfiguration": {
    "isEnabled": true,
    "retentionDays": 0
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» retentionAuditDays	integer(int32)	false	none	none
» reportRetentionConfiguration	ReportRetention¦null	false	none	none
» rtpHistoryRetentionConfiguration	ReportRetention¦null	false	none	none

ReportRetention


```
{
  "isEnabled": true,
  "retentionDays": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
isEnabled	boolean	false	none	none
retentionDays	integer(int32)	false	none	none

GenerateEncryptionKeyCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

GetEncryptionKeyCreationDateQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "creationDate": "2019-08-24T14:15:22Z"
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» creationDate	string(date-time)¦null	false	none	none

UpdateSingleUserSessionCommandResponse


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

PostSingleUserSessionParameters

```
{
  "isSingleUserSession": true
}
```

Properties

Name	Type	Required	Restrictions	Description
isSingleUserSession	boolean	true	none	none

GetTelemetryConfigQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "telemetryConfig": {
    "isEnabled": true,
    "sendTelemetryToOpswat": true,
    "autoInstrumentation": true,
    "optelCollectors": [
      {
        "metricEndpoint": {
          "endpointUrl": "string",
          "endpointApiKey": "string",
          "protocol": 0
        },
        "traceEndpoint": {
          "endpointUrl": "string",
          "endpointApiKey": "string",
          "protocol": 0
        },
        "isOpswatCollector": true
      }
    ]
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» telemetryConfig	TelemetryConfig	false	none	none

TelemetryConfig

```
{
  "isEnabled": true,
  "sendTelemetryToOpswat": true,
  "autoInstrumentation": true,
  "optelCollectors": [
    {
      "metricEndpoint": {
        "endpointUrl": "string",
        "endpointApiKey": "string",
        "protocol": 0
      },
      "traceEndpoint": {
        "endpointUrl": "string",
        "endpointApiKey": "string",
        "protocol": 0
      },
      "isOpswatCollector": true
    }
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
isEnabled	boolean	false	none	none
sendTelemetryToOpswat	boolean¦null	false	none	none
autoInstrumentation	boolean¦null	false	none	none
optelCollectors	[OptelCollector]¦null	false	none	none

OptelCollector

```
{
  "metricEndpoint": {
    "endpointUrl": "string",
    "endpointApiKey": "string",
    "protocol": 0
  },
  "traceEndpoint": {
    "endpointUrl": "string",
    "endpointApiKey": "string",
    "protocol": 0
  },
  "isOpswatCollector": true
}
```

Properties

Name	Type	Required	Restrictions	Description
metricEndpoint	OptelCollectorEndpoint	false	none	none
traceEndpoint	OptelCollectorEndpoint	false	none	none
isOpswatCollector	boolean¦null	false	none	none

OptelCollectorEndpoint

```
{
  "endpointUrl": "string",
  "endpointApiKey": "string",
  "protocol": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
endpointUrl	string	false	none	none
endpointApiKey	string¦null	false	none	none
protocol	OptelCollectorEndpointProtocol	false	none	- 0: GRPC - 1: HTTP

OptelCollectorEndpointProtocol

```
0
```

- 0: GRPC
- 1: HTTP

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: GRPC - 1: HTTP

Enumerated Values

Property	Value
anonymous	0
anonymous	1

UpdateTelemetryConfigCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

UpdateTelemetryConfigCommandRequest

```
{
  "telemetryConfig": {
    "isEnabled": true,
    "sendTelemetryToOpswat": true,
    "autoInstrumentation": true,
    "optelCollectors": [
      {
        "metricEndpoint": {
          "endpointUrl": "string",
          "endpointApiKey": "string",
          "protocol": 0
        },
        "traceEndpoint": {
          "endpointUrl": "string",
          "endpointApiKey": "string",
          "protocol": 0
        },
        "isOpswatCollector": true
      }
    ]
  },
  "userInfo": {
    "userId": "string",
    "userName": "string"
  }
}
```

Properties

Name	Type	Required	Restrictions	Description
telemetryConfig	TelemetryConfig	false	none	none
userInfo	UserInfo¦null	false	none	none

UserInfo

```
{
  "userId": "string",
  "userName": "string"
}
```


Properties

Name	Type	Required	Restrictions	Description
userId	string	false	none	none
userName	string	false	none	none

SsoConfig


```
{
  "isEnabled": true,
  "ssoProvider": 0,
  "ssoType": 0,
  "baseUrl": "string",
  "clientId": "string",
  "clientSecret": "string",
  "authority": "string",
  "identityProviderCertificate": "string",
  "identityProviderIssuer": "string",
  "loginUrl": "string",
  "logoutUrl": "string",
  "administratorRoleName": "string",
  "readonlyRoleName": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
isEnabled	boolean	false	none	none
ssoProvider	SingleSignOnProvider	false	none	- 0: Azure - 1: Okta - 2: Auth0 - 3: Adfs - 4: PingFederate - 5: Opswat
ssoType	SingleSignOnType	false	none	- 0: OpenIdConnect - 1: Saml2
baseUrl	string	false	none	none
clientId	string	false	none	none
clientSecret	string	false	none	none
authority	string	false	none	none
identityProviderCertificate	string	false	none	none
identityProviderIssuer	string	false	none	none
loginUrl	string	false	none	none
logoutUrl	string	false	none	none
administratorRoleName	string	false	none	none
readonlyRoleName	string	false	none	none

SingleSignOnProvider

```
0
```

- 0: Azure
- 1: Okta
- 2: Auth0
- 3: Adfs
- 4: PingFederate
- 5: Opswat

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Azure - 1: Okta - 2: Auth0 - 3: Adfs - 4: PingFederate - 5: Opswat

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4
anonymous	5

SingleSignOnType

```
0
```

- 0: OpenIdConnect
- 1: Saml2

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: OpenIdConnect - 1: Saml2

Enumerated Values

Property	Value
anonymous	0
anonymous	1

SsoConfigResponse


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "clientId": "string",
  "authority": "string",
  "baseUrl": "string",
  "isEnabled": true,
  "ssoType": 0,
  "ssoProvider": 0,
  "identityProviderIssuer": "string",
  "identityProviderLoginUrl": "string",
  "identityProviderLogoutUrl": "string",
  "administratorRoleName": "string",
  "readonlyRoleName": "string",
  "ssoConfig": {
    "isEnabled": true,
    "ssoProvider": 0,
    "ssoType": 0,
    "baseUrl": "string",
    "clientId": "string",
    "clientSecret": "string",
    "authority": "string",
    "identityProviderCertificate": "string",
    "identityProviderIssuer": "string",
    "loginUrl": "string",
    "logoutUrl": "string",
    "administratorRoleName": "string",
    "readonlyRoleName": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» clientId	string¦null	false	none	none
» authority	string¦null	false	none	none
» baseUrl	string¦null	false	none	none
» isEnabled	boolean	false	none	none
» ssoType	SingleSignOnType2	false	none	- 0: OpenIdConnect
» ssoProvider	SingleSignOnProvider2	false	none	- 0: Azure - 1: Okta - 2: Auth0 - 3: Adfs - 4: PingFederate
» identityProviderIssuer	string¦null	false	none	none
» identityProviderLoginUrl	string¦null	false	none	none
» identityProviderLogoutUrl	string¦null	false	none	none
» administratorRoleName	string¦null	false	none	none
» readonlyRoleName	string¦null	false	none	none
» ssoConfig	SsoConfig¦null	false	none	none

SingleSignOnType2

```
0
```

- 0: OpenIdConnect

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: OpenIdConnect

Enumerated Values

Property	Value
anonymous	0

SingleSignOnProvider2

```
0
```

- 0: Azure
- 1: Okta
- 2: Auth0
- 3: Adfs
- 4: PingFederate

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Azure - 1: Okta - 2: Auth0 - 3: Adfs - 4: PingFederate

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3

anonymous	4

UpdateSsoConfigParameters

```
{
  "clientId": "string",
  "clientSecret": "string",
  "authority": "string",
  "baseUrl": "string",
  "identityProviderIssuer": "string",
  "identityProviderLoginUrl": "string",
  "identityProviderLogoutUrl": "string",
  "identityProviderCertificate": {
    "content": "string"
  },
  "isEnabled": true,
  "ssoType": 0,
  "ssoProvider": 0,
  "administratorRoleName": "string",
  "readonlyRoleName": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
clientId	string	false	none	none
clientSecret	string	false	none	none
authority	string	false	none	none
baseUrl	string	false	none	none
identityProviderIssuer	string	false	none	none
identityProviderLoginUrl	string	false	none	none
identityProviderLogoutUrl	string	false	none	none
identityProviderCertificate	IdentityProviderCertificate	false	none	none
isEnabled	boolean	true	none	none
ssoType	integer(int32)	false	none	none
ssoProvider	integer(int32)	false	none	none
administratorRoleName	string	false	none	none
readonlyRoleName	string	false	none	none

IdentityProviderCertificate

```
{
  "content": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
content	string¦null	false	none	none

EnumerateStorages

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "groupId": "string",
      "accountId": "string",
      "name": "string",
      "storageClientId": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "scansInProgress": true,
      "isRtpInProgress": true,
      "isScheduledScanInProgress": true,
      "storageFunctionality": "string",
      "$type": "string"
    }
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entries	[Storage]¦null	false	none	none

GetStorage


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "storage": {
    "id": "string",
    "groupId": "string",
    "accountId": "string",
    "name": "string",
    "storageClientId": "string",
    "storageType": {
      "name": "string",
      "categoryType": 0,
      "protocolType": 0,
      "vendorType": 0
    },
    "storageStatus": {
      "message": "string",
      "statusType": 0
    },
    "scansInProgress": true,
    "isRtpInProgress": true,
    "isScheduledScanInProgress": true,
    "storageFunctionality": "string",
    "$type": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» storage	Storage¦null	false	none	none

EnumerateStorageUsersQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "fullName": "string"
    }
  ]
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entries	[StorageUser]¦null	false	none	none

StorageUser

```
{
  "id": "string",
  "fullName": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string¦null	false	none	none
fullName	string¦null	false	none	none

AddNewStorageCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "storageId": "string"
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» storageId	string¦null	false	none	none

AddBatchStorageParameters

```
{
  "addStorageParametersList": [
    {
      "name": "string",
      "storageVendorType": 0,
      "storageProtocolType": 0,
      "credentialsFile": "string",
      "credentials": "string",
      "source": "string",
      "groupId": "string",
      "accountId": "string",
      "workflowId": "string",
      "transferMetadataAndChecksum": true
    }
  ]
}
```

Properties

Name	Type	Required	Restrictions	Description
addStorageParametersList	[AddStorageParameters]	false	none	none

AddStorageParameters

```
{
  "name": "string",
  "storageVendorType": 0,
  "storageProtocolType": 0,
  "credentialsFile": "string",
  "credentials": "string",
  "source": "string",
  "groupId": "string",
  "accountId": "string",
  "workflowId": "string",
  "transferMetadataAndChecksum": true
}
```

Properties

Name	Type	Required	Restrictions	Description
name	string	true	none	none
storageVendorType	StorageVendorType2	true	none	- 0: AmazonS3 - 1: OneDrive - 2: Box - 3: DellIsilon - 4: AzureFiles - 5: SmbCompatible - 6: S3Compatible - 7: AzureBlob - 8: AlibabaCloud - 9: GoogleCloud - 10: Sftp - 11: SharePoint - 12: Nfs - 13: Wasabi - 14: OracleCloud - 15: NetAppOntap - 16: Cubbit - 17: Mft - 18: SharepointOnPrem - 19: Ftp - 20: Oracle - 21: NfsSisl - 22: ScalityRing
storageProtocolType	StorageProtocolType2	true	none	- 0: Smb - 1: GraphSdk - 2: BoxSdk - 3: AwsSdk - 4: AzureBlobSdk - 5: AlibabaCloudSdk - 6: GoogleCloudSdk - 7: AzureFilesSdk - 8: Sftp - 9: Nfs - 10: Mft - 11: SharepointOnPrem - 12: Ftp - 13: OracleSdk
credentialsFile	string(binary)¦null	false	none	none
credentials	string¦null	false	none	none
source	string	true	none	none
groupId	string¦null	false	none	none

accountId	string¦null	false	none	none
workflowId	string¦null	false	none	none
transferMetadataAndChecksum	boolean¦null	false	none	none

GetBatchResultsResponse


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "addStorageResultInfos": [
    {
      "result": 0,
      "responseKey": "string",
      "responseMessage": "string",
      "responseMessageParams": {
        "property1": "string",
        "property2": "string"
      },
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "source": "string",
      "name": "string",
      "vendorType": 0,
      "protocolType": 0
    }
  ],
  "isCompleted": true
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» addStorageResultInfos	[AddStorageResultInfo]¦null	false	none	none
» isCompleted	boolean	false	none	none

AddStorageResultInfo

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "storageStatus": {
    "message": "string",
    "statusType": 0
  },
  "source": "string",
  "name": "string",
  "vendorType": 0,
  "protocolType": 0
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» storageStatus	StorageStatus	false	none	none
» source	string	false	none	none
» name	string	false	none	none
» vendorType	StorageVendorType2	false	none	- 0: AmazonS3 - 1: OneDrive - 2: Box - 3: Dellsilon - 4: AzureFiles - 5: SmbCompatible - 6: S3Compatible - 7: AzureBlob - 8: AlibabaCloud - 9: GoogleCloud - 10: Sftp - 11: SharePoint - 12: Nfs - 13: Wasabi - 14: OracleCloud - 15: NetAppOntap - 16: Cubbit - 17: Mft - 18: SharepointOnPrem - 19: Ftp - 20: Oracle - 21: NfsSisl - 22: ScalityRing
» protocolType	StorageProtocolType2	false	none	- 0: Smb - 1: GraphSdk - 2: BoxSdk - 3: AwsSdk - 4: AzureBlobSdk - 5: AlibabaCloudSdk - 6: GoogleCloudSdk - 7: AzureFilesSdk - 8: Sftp - 9: Nfs - 10: Mft - 11: SharepointOnPrem - 12: Ftp - 13: OracleSdk

UpdateNewStorageCommandResponse


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "updatedStorage": {
    "id": "string",
    "name": "string",
    "groupId": "string",
    "clientId": "string",
    "accountId": "string",
    "source": "string",
    "storageFunctionality": "string",
    "storageType": {
      "name": "string",
      "categoryType": 0,
      "protocolType": 0,
      "vendorType": 0
    },
    "storageStatus": {
      "message": "string",
      "statusType": 0
    },
    "workflowId": "string",
    "lastUpdated": "2019-08-24T14:15:22Z",
    "tenantId": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» updatedStorage	NewStorage¦null	false	none	none

EnumerateStorages2


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "groupId": "string",
      "accountId": "string",
      "name": "string",
      "storageClientId": "string",
      "storageType": {
        "name": "string",
        "categoryType": 0,
        "protocolType": 0,
        "vendorType": 0
      },
      "storageStatus": {
        "message": "string",
        "statusType": 0
      },
      "scansInProgress": true,
      "isRtpInProgress": true,
      "isScheduledScanInProgress": true,
      "storageFunctionality": "string",
      "$type": "string"
    }
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entries	[Storage]¦null	false	none	none

AddScanScheduleParameters

```
{
  "name": "string",
  "workflowId": "string",
  "recurrencePattern": {
    "type": "string",
    "minutelyRecurrence": {
      "everyNumberOfMinutes": 5
    },
    "hourlyRecurrence": {
      "everyNumberOfHours": 1
    },
    "dailyRecurrence": {
      "everyNumberOfDays": 1
    },
    "weeklyRecurrence": {
      "daysOfWeek": [
        0
      ],
      "everyNumberOfWeeks": 0
    },
    "monthlyRecurrence": {
      "everyNumberOfMonths": 1,
      "dayOfMonth": 1
    },
    "startDate": "2019-08-24T14:15:22Z",
    "timeToStart": "2019-08-24T14:15:22Z",
    "avoidConcurrentScheduleRuns": true
  },
  "scanPartition": "string",
  "priority": 0,
  "$type": "string"
}
```

Properties

allOf - discriminator: ScanScheduleParameters.$type

Name	Type	Required	Restrictions	Description
anonymous	ScanScheduleParameters	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

ScanScheduleParameters

```
{
  "name": "string",
  "workflowId": "string",
  "recurrencePattern": {
    "type": "string",
    "minutelyRecurrence": {
      "everyNumberOfMinutes": 5
    },
    "hourlyRecurrence": {
      "everyNumberOfHours": 1
    },
    "dailyRecurrence": {
      "everyNumberOfDays": 1
    },
    "weeklyRecurrence": {
      "daysOfWeek": [
        0
      ],
      "everyNumberOfWeeks": 0
    },
    "monthlyRecurrence": {
      "everyNumberOfMonths": 1,
      "dayOfMonth": 1
    },
    "startDate": "2019-08-24T14:15:22Z",
    "timeToStart": "2019-08-24T14:15:22Z",
    "avoidConcurrentScheduleRuns": true
  },
  "scanPartition": "string",
  "priority": 0,
  "$type": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
name	string	true	none	none
workflowId	string	true	none	none
recurrencePattern	RecurrencePatternParameters	true	none	none
scanPartition	string¦null	false	none	none
priority	integer(int32)	false	none	none
$type	string	true	none	none

RecurrencePatternParameters

```
{
  "type": "string",
  "minutelyRecurrence": {
    "everyNumberOfMinutes": 5
  },
  "hourlyRecurrence": {
    "everyNumberOfHours": 1
  },
  "dailyRecurrence": {
    "everyNumberOfDays": 1
  },
  "weeklyRecurrence": {
    "daysOfWeek": [
      0
    ],
    "everyNumberOfWeeks": 0
  },
  "monthlyRecurrence": {
    "everyNumberOfMonths": 1,
    "dayOfMonth": 1
  },
  "startDate": "2019-08-24T14:15:22Z",
  "timeToStart": "2019-08-24T14:15:22Z",
  "avoidConcurrentScheduleRuns": true
}
```

Properties

Name	Type	Required	Restrictions	Description
type	string	false	none	none
minutelyRecurrence	MinutelyRecurrenceParameters¦null	false	none	none
hourlyRecurrence	HourlyRecurrenceParameters¦null	false	none	none
dailyRecurrence	DailyRecurrenceParameters¦null	false	none	none
weeklyRecurrence	WeeklyRecurrenceParameters¦null	false	none	none
monthlyRecurrence	MonthlyRecurrenceParameters¦null	false	none	none
startDate	string(date-time)¦null	false	none	none
timeToStart	string(date-time)¦null	false	none	none
avoidConcurrentScheduleRuns	boolean	false	none	none

MinutelyRecurrenceParameters

```
{
  "everyNumberOfMinutes": 5
}
```

Properties

Name	Type	Required	Restrictions	Description
everyNumberOfMinutes	integer(int32)	false	none	none

HourlyRecurrenceParameters

```
{
  "everyNumberOfHours": 1
}
```

Properties

Name	Type	Required	Restrictions	Description
everyNumberOfHours	integer(int32)	false	none	none

DailyRecurrenceParameters

```
{
  "everyNumberOfDays": 1
}
```

Properties

Name	Type	Required	Restrictions	Description
everyNumberOfDays	integer(int32)	false	none	none

WeeklyRecurrenceParameters

```
{
  "daysOfWeek": [
    0
  ],
  "everyNumberOfWeeks": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
daysOfWeek	[DayOfWeek]	false	none	[- 0: Sunday - 1: Monday - 2: Tuesday - 3: Wednesday - 4: Thursday - 5: Friday - 6: Saturday]
everyNumberOfWeeks	integer(int32)	false	none	none

DayOfWeek

```
0
```

- 0: Sunday
- 1: Monday
- 2: Tuesday
- 3: Wednesday
- 4: Thursday
- 5: Friday
- 6: Saturday

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Sunday - 1: Monday - 2: Tuesday - 3: Wednesday - 4: Thursday - 5: Friday - 6: Saturday

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4
anonymous	5
anonymous	6

MonthlyRecurrenceParameters


```
{
  "everyNumberOfMonths": 1,
  "dayOfMonth": 1
}
```


Properties

Name	Type	Required	Restrictions	Description
everyNumberOfMonths	integer(int32)	false	none	none
dayOfMonth	integer(int32)	false	none	none

UpdateScanScheduleParameters

```
{
  "name": "string",
  "workflowId": "string",
  "recurrencePattern": {
    "type": "string",
    "minutelyRecurrence": {
      "everyNumberOfMinutes": 5
    },
    "hourlyRecurrence": {
      "everyNumberOfHours": 1
    },
    "dailyRecurrence": {
      "everyNumberOfDays": 1
    },
    "weeklyRecurrence": {
      "daysOfWeek": [
        0
      ],
      "everyNumberOfWeeks": 0
    },
    "monthlyRecurrence": {
      "everyNumberOfMonths": 1,
      "dayOfMonth": 1
    },
    "startDate": "2019-08-24T14:15:22Z",
    "timeToStart": "2019-08-24T14:15:22Z",
    "avoidConcurrentScheduleRuns": true
  },
  "scanPartition": "string",
  "priority": 0,
  "$type": "string",
  "id": "string"
}
```


Properties

allOf - discriminator: ScanScheduleParameters.$type

Name	Type	Required	Restrictions	Description
anonymous	ScanScheduleParameters	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» id	string¦null	false	none	none

UpdateScanScheduleCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

EnumerateScanSchedules

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "name": "string",
      "workflowId": "string",
      "scanConfigurationId": "string",
      "type": "string",
      "scanPartition": "string",
      "createdAt": "2019-08-24T14:15:22Z",
      "lastScanTime": "2019-08-24T14:15:22Z",
      "nextScanTime": "2019-08-24T14:15:22Z",
      "recurrencePatternDto": {
        "type": 0,
        "minutelyRecurrence": {
          "everyNumberOfMinutes": 0
        },
        "hourlyRecurrence": {
          "everyNumberOfHours": 0
        },
        "dailyRecurrence": {
          "everyNumberOfDays": 0
        },
        "weeklyRecurrence": {
          "daysOfWeek": [
            0
          ],
          "everyNumberOfWeeks": 0
        },
        "monthlyRecurrence": {
          "everyNumberOfMonths": 0,
          "dayOfMonth": 0
        },
        "startDate": "2019-08-24T14:15:22Z",
        "timeToStart": "2019-08-24T14:15:22Z",
        "avoidConcurrentScheduleRuns": true
      },
      "createdBy": "string",
      "priority": 0,
      "storageId": "string"
    }
  ],
  "totalCount": 0
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entries	[ScheduledScanResponse]¦null	false	none	none
» totalCount	integer(int64)	false	none	none

ScheduledScanResponse

```
{
  "id": "string",
  "name": "string",
  "workflowId": "string",
  "scanConfigurationId": "string",
  "type": "string",
  "scanPartition": "string",
  "createdAt": "2019-08-24T14:15:22Z",
  "lastScanTime": "2019-08-24T14:15:22Z",
  "nextScanTime": "2019-08-24T14:15:22Z",
  "recurrencePatternDto": {
    "type": 0,
    "minutelyRecurrence": {
      "everyNumberOfMinutes": 0
    },
    "hourlyRecurrence": {
      "everyNumberOfHours": 0
    },
    "dailyRecurrence": {
      "everyNumberOfDays": 0
    },
    "weeklyRecurrence": {
      "daysOfWeek": [
        0
      ],
      "everyNumberOfWeeks": 0
    },
    "monthlyRecurrence": {
      "everyNumberOfMonths": 0,
      "dayOfMonth": 0
    },
    "startDate": "2019-08-24T14:15:22Z",
    "timeToStart": "2019-08-24T14:15:22Z",
    "avoidConcurrentScheduleRuns": true
  },
  "createdBy": "string",
  "priority": 0,
  "storageId": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string¦null	false	none	none
name	string¦null	false	none	none
workflowId	string	true	none	WorkflowId required if ScanConfigurationId is null
scanConfigurationId	string¦null	false	none	This property is obsolete and will be removed in the future. Use WorkflowId instead.
type	string¦null	false	none	none
scanPartition	string¦null	false	none	none
createdAt	string(date-time)	false	none	none
lastScanTime	string(date-time)¦null	false	none	none
nextScanTime	string(date-time)	false	none	none
recurrencePatternDto	RecurrencePattern¦null	false	none	none
createdBy	string¦null	false	none	none
priority	MessagePriority	false	none	- 0: Low - 1: Medium - 2: High
storageId	string	false	none	none

RecurrencePattern

```
{
  "type": 0,
  "minutelyRecurrence": {
    "everyNumberOfMinutes": 0
  },
  "hourlyRecurrence": {
    "everyNumberOfHours": 0
  },
  "dailyRecurrence": {
    "everyNumberOfDays": 0
  },
  "weeklyRecurrence": {
    "daysOfWeek": [
      0
    ],
    "everyNumberOfWeeks": 0
  },
  "monthlyRecurrence": {
    "everyNumberOfMonths": 0,
    "dayOfMonth": 0
  },
  "startDate": "2019-08-24T14:15:22Z",
  "timeToStart": "2019-08-24T14:15:22Z",
  "avoidConcurrentScheduleRuns": true
}
```

Properties

Name	Type	Required	Restrictions	Description
type	RecurrencePatternType	false	none	- 0: Hourly - 1: Daily - 2: Weekly - 3: Monthly - 4: Minutely
minutelyRecurrence	MinutelyRecurrence¦null	false	none	none
hourlyRecurrence	HourlyRecurrence¦null	false	none	none
dailyRecurrence	DailyRecurrence¦null	false	none	none
weeklyRecurrence	WeeklyRecurrence¦null	false	none	none
monthlyRecurrence	MonthlyRecurrence¦null	false	none	none
startDate	string(date-time)¦null	false	none	none
timeToStart	string(date-time)¦null	false	none	none
avoidConcurrentScheduleRuns	boolean	false	none	none

RecurrencePatternType

```
0
```

- 0: Hourly
- 1: Daily
- 2: Weekly
- 3: Monthly
- 4: Minutely

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Hourly - 1: Daily - 2: Weekly - 3: Monthly - 4: Minutely

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4

MinutelyRecurrence

```
{
  "everyNumberOfMinutes": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
everyNumberOfMinutes	integer(int32)	false	none	none

HourlyRecurrence

```
{
  "everyNumberOfHours": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
everyNumberOfHours	integer(int32)	false	none	none

DailyRecurrence

```
{
  "everyNumberOfDays": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
everyNumberOfDays	integer(int32)	false	none	none

WeeklyRecurrence

```
{
  "daysOfWeek": [
    0
  ],
  "everyNumberOfWeeks": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
daysOfWeek	[DayOfWeek]	false	none	[- 0: Sunday - 1: Monday - 2: Tuesday - 3: Wednesday - 4: Thursday - 5: Friday - 6: Saturday]
everyNumberOfWeeks	integer(int32)	false	none	none

MonthlyRecurrence

```
{
  "everyNumberOfMonths": 0,
  "dayOfMonth": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
everyNumberOfMonths	integer(int32)	false	none	none
dayOfMonth	integer(int32)	false	none	none

SortDirectionFilter

```
0
```

- 0: Ascending
- 1: Descending

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Ascending - 1: Descending

Enumerated Values

Property	Value
anonymous	0
anonymous	1

DeleteScanSchedulesCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "areAllDeleted": true
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» result	ResultType6	false	none	- 0: Success - 1: FailedInvalidListOfIds - 2: StorageForbidden
» areAllDeleted	boolean	false	none	none

ResultType6


```
0
```


- 0: Success
- 1: FailedInvalidListOfIds
- 2: StorageForbidden

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Success - 1: FailedInvalidListOfIds - 2: StorageForbidden

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

RetrieveUserNameFromStorageQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "userName": "string"
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» userName	string¦null	false	none	none

EnumerateActiveUsers


```
{
  "entries": [
    {
      "id": "string",
      "userName": "string",
      "fullName": "string",
      "email": "string",
      "roles": [
        {
          "bindingId": "string",
          "roleId": "string",
          "roleName": "string",
          "expiresAt": "2019-08-24T14:15:22Z"
        }
      ],
      "directPermissions": [
        {
          "id": "string",
          "permissionName": "string",
          "expiresAt": "2019-08-24T14:15:22Z"
        }
      ]
    }
  ],
  "totalCount": 0
}
```


Properties

Name	Type	Required	Restrictions	Description
entries	[ActiveUser]¦null	false	none	none
totalCount	integer(int64)	false	none	none

ActiveUser

```
{
  "id": "string",
  "userName": "string",
  "fullName": "string",
  "email": "string",
  "roles": [
    {
      "bindingId": "string",
      "roleId": "string",
      "roleName": "string",
      "expiresAt": "2019-08-24T14:15:22Z"
    }
  ],
  "directPermissions": [
    {
      "id": "string",
      "permissionName": "string",
      "expiresAt": "2019-08-24T14:15:22Z"
    }
  ]
}
```


Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
userName	string	false	none	none
fullName	string	false	none	none
email	string	false	none	none
roles	[UserRoleAssignment]¦null	false	none	none
directPermissions	[UserPermission]¦null	false	none	none

UserRoleAssignment


```
{
  "bindingId": "string",
  "roleId": "string",
  "roleName": "string",
  "expiresAt": "2019-08-24T14:15:22Z"
}
```

Properties

Name	Type	Required	Restrictions	Description
bindingId	string(guid)	false	none	none
roleId	string(guid)	false	none	none
roleName	string	false	none	none
expiresAt	string(date-time)¦null	false	none	none

UserPermission

```
{
  "id": "string",
  "permissionName": "string",
  "expiresAt": "2019-08-24T14:15:22Z"
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string(guid)	false	none	none
permissionName	string	false	none	none
expiresAt	string(date-time)¦null	false	none	none

OrderType

```
0
```

- 0: Descending
- 1: Ascending

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Descending - 1: Ascending

Enumerated Values

Property	Value
anonymous	0
anonymous	1

OrderBy

```
0
```

- 0: Id
- 1: FullName
- 2: Username
- 3: Email
- 4: Role

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: Id - 1: FullName - 2: Username - 3: Email - 4: Role

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	3
anonymous	4

RegisterUserParameters

```
{
  "fullName": "string",
  "userName": "string",
  "password": "string",
  "email": "string",
  "ssoUserId": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
fullName	string	true	none	none
userName	string	true	none	none
password	string¦null	false	none	none
email	string	true	none	none
ssoUserId	string¦null	false	none	none

CreateUserParameters

```
{
  "fullName": "string",
  "userName": "string",
  "password": "string",
  "email": "string",
  "ssoUserId": "string",
  "role": 0
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RegisterUserParameters	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» role	UserRole	true	none	- 0: None - 1: ReadOnlyAdministrator - 2: Administrator - 4: SsoAdministrator - 5: SsoReadOnlyAdministrator - 6: TenantAdministrator - 999: Invalid

UserRole

```
0
```

- 0: None
- 1: ReadOnlyAdministrator
- 2: Administrator
- 4: SsoAdministrator
- 5: SsoReadOnlyAdministrator
- 6: TenantAdministrator
- 999: Invalid

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: None - 1: ReadOnlyAdministrator - 2: Administrator - 4: SsoAdministrator - 5: SsoReadOnlyAdministrator - 6: TenantAdministrator - 999: Invalid

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2
anonymous	4
anonymous	5
anonymous	6
anonymous	999

AuthenticateUserCommandResponseWithoutTokens

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "fullName": "string",
  "email": "string",
  "userName": "string",
  "role": 0,
  "user": {
    "id": "string",
    "userName": "string",
    "fullName": "string",
    "email": "string",
    "role": 0,
    "roles": [
      "string"
    ],
    "tenantId": "string",
    "ssoUserId": "string",
    "passwordHash": "string",
    "passwordSalt": "string",
    "passwordHashAlgorithm": 0
  },
  "userId": "string"
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» fullName	string¦null	false	none	none
» email	string¦null	false	none	none
» userName	string¦null	false	none	none
» role	UserRole¦null	false	none	- 0: None - 1: ReadOnlyAdministrator - 2: Administrator - 4: SsoAdministrator - 5: SsoReadOnlyAdministrator - 6: TenantAdministrator - 999: Invalid
» user	UserDto¦null	false	none	none
» userId	string¦null	false	none	none

UserDto

```
{
  "id": "string",
  "userName": "string",
  "fullName": "string",
  "email": "string",
  "role": 0,
  "roles": [
    "string"
  ],
  "tenantId": "string",
  "ssoUserId": "string",
  "passwordHash": "string",
  "passwordSalt": "string",
  "passwordHashAlgorithm": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	false	none	none
userName	string	false	none	none
fullName	string	false	none	none
email	string	false	none	none
role	UserRole	false	none	- 0: None - 1: ReadOnlyAdministrator - 2: Administrator - 4: SsoAdministrator - 5: SsoReadOnlyAdministrator - 6: TenantAdministrator - 999: Invalid
roles	[string]	false	none	none
tenantId	string¦null	false	none	none
ssoUserId	string¦null	false	none	none
passwordHash	string(byte)¦null	false	none	none
passwordSalt	string(byte)¦null	false	none	none
passwordHashAlgorithm	integer(int32)¦null	false	none	none

AuthenticateUserCommandRequest

```
{
  "userName": "string",
  "password": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
userName	string	true	none	login name of the user
password	string	true	none	password

CurrentUserInfo


```
{
  "username": "string",
  "email": "string",
  "authProvider": "string",
  "role": "string",
  "id": "string",
  "roles": [
    {
      "bindingId": "string",
      "roleId": "string",
      "roleName": "string",
      "expiresAt": "2019-08-24T14:15:22Z"
    }
  ],
  "directPermissions": [
    {
      "id": "string",
      "permissionName": "string",
      "expiresAt": "2019-08-24T14:15:22Z"
    }
  ]
}
```


Properties

Name	Type	Required	Restrictions	Description
username	string	false	none	none
email	string	false	none	none
authProvider	string	false	none	none
role	string	false	none	none
id	string	false	none	none
roles	[UserRoleAssignment]¦null	false	none	none
directPermissions	[UserPermission]¦null	false	none	none

UpdateUserCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "user": {
    "id": "string",
    "userName": "string",
    "fullName": "string",
    "email": "string",
    "role": 0,
    "roles": [
      "string"
    ],
    "tenantId": "string",
    "ssoUserId": "string",
    "passwordHash": "string",
    "passwordSalt": "string",
    "passwordHashAlgorithm": 0
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» user	UserDto¦null	false	none	none

UpdateUserRoleParameters

```
{
  "userId": "string",
  "role": 0
}
```


Properties

Name	Type	Required	Restrictions	Description
userId	string	true	none	none
role	UserRole	true	none	- 0: None - 1: ReadOnlyAdministrator - 2: Administrator - 4: SsoAdministrator - 5: SsoReadOnlyAdministrator - 6: TenantAdministrator - 999: Invalid

EnumerateUsers


```
{
  "entries": [
    {
      "id": "string",
      "userName": "string",
      "fullName": "string",
      "email": "string",
      "role": 0,
      "roles": [
        "string"
      ],
      "tenantId": "string",
      "ssoUserId": "string",
      "passwordHash": "string",
      "passwordSalt": "string",
      "passwordHashAlgorithm": 0
    }
  ],
  "totalCount": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
entries	[UserDto]¦null	false	none	none
totalCount	integer(int64)	false	none	none

UserCleanupCommandResponse


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "user": {
    "id": "string",
    "userName": "string",
    "fullName": "string",
    "email": "string",
    "role": 0,
    "roles": [
      "string"
    ],
    "tenantId": "string",
    "ssoUserId": "string",
    "passwordHash": "string",
    "passwordSalt": "string",
    "passwordHashAlgorithm": 0
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» user	UserDto¦null	false	none	none

UpdateUserParameters

```
{
  "userId": "string",
  "role": 0,
  "userName": "string",
  "fullName": "string",
  "currentPassword": "string",
  "newPassword": "string",
  "email": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
userId	string¦null	false	none	none
role	UserRole	false	none	- 0: None - 1: ReadOnlyAdministrator - 2: Administrator - 4: SsoAdministrator - 5: SsoReadOnlyAdministrator - 6: TenantAdministrator - 999: Invalid
userName	string¦null	false	none	none
fullName	string¦null	false	none	none
currentPassword	string¦null	false	none	none
newPassword	string¦null	false	none	none
email	string¦null	false	none	none

RequestResetPasswordCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

RequestResetPasswordCommandRequest

```
{
  "userInput": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
userInput	string	true	none	none

UpdatePasswordCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "user": {
    "id": "string",
    "userName": "string",
    "fullName": "string",
    "email": "string",
    "role": 0,
    "roles": [
      "string"
    ],
    "tenantId": "string",
    "ssoUserId": "string",
    "passwordHash": "string",
    "passwordSalt": "string",
    "passwordHashAlgorithm": 0
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» user	UserDto¦null	false	none	none

UpdatePasswordCommandRequest

```
{
  "newPassword": "stringstring",
  "newPasswordConfirmation": "stringstring",
  "secureToken": "string"
}
```


Properties

Name	Type	Required	Restrictions	Description
newPassword	string	true	none	none
newPasswordConfirmation	string	true	none	none
secureToken	string	false	none	none

GetUserPermissionsQueryResponse


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "roles": [
    {
      "bindingId": "string",
      "roleId": "string",
      "roleName": "string",
      "expiresAt": "2019-08-24T14:15:22Z"
    }
  ],
  "directPermissions": [
    {
      "id": "string",
      "permissionName": "string",
      "expiresAt": "2019-08-24T14:15:22Z"
    }
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» roles	[UserRoleAssignment]¦null	false	none	none
» directPermissions	[UserPermission]¦null	false	none	none

GetUserTourQueryResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "userId": "string",
  "userTours": [
    {
      "tourType": 0,
      "finalizedDate": "2019-08-24T14:15:22Z"
    }
  ]
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» userId	string¦null	false	none	none
» userTours	[UserTour]¦null	false	none	none

UserTour

```
{
  "tourType": 0,
  "finalizedDate": "2019-08-24T14:15:22Z"
}
```

Properties

Name	Type	Required	Restrictions	Description
tourType	UserTourType	false	none	- 0: OnboardingUi - 1: WorkflowUi - 2: StorageUnits
finalizedDate	string(date-time)	false	none	none

UserTourType

```
0
```

- 0: OnboardingUi
- 1: WorkflowUi
- 2: StorageUnits

Properties

Name	Type	Required	Restrictions	Description
anonymous	integer	false	none	- 0: OnboardingUi - 1: WorkflowUi - 2: StorageUnits

Enumerated Values

Property	Value
anonymous	0
anonymous	1
anonymous	2

FinalizeUserTour

```
{
  "userId": "string",
  "userTourType": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
userId	string	true	none	none
userTourType	UserTourType	true	none	- 0: OnboardingUi - 1: WorkflowUi - 2: StorageUnits

WebhookEventBasedRtpCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "correlationId": "string"
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» correlationId	string¦null	false	none	none

WebhookEventBasedRtpParameters

```
{
  "metadata": "string",
  "storageClientId": "string",
  "scanId": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
metadata	string	true	none	none
storageClientId	string	true	none	none
scanId	string¦null	false	none	none

GetAllWorkflowsCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entries": [
    {
      "id": "string",
      "scanPoolId": "string",
      "scanProcessFlow": 0,
      "isIdentityScanningEnabled": true,
      "scanIntervalDays": 0,
      "rules": [
        "string"
      ],
      "type": 0,
      "workflowSupportedTechnologies": {
        "dlp": {
          "status": 0
        },
        "deepCdr": {
          "status": 0
        },
        "sandBox": {
          "status": 0
        },
        "metascan": {
          "status": 0
        }
      },
      "name": "string",
      "userAgent": "string",
      "scanPool": {
        "id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "failOverScanPoolId": "string",
```

```
"id": "string",
        "name": "string",
        "scanPoolType": 0,
        "isDefault": true,
        "scanInstances": [
          {
            "id": "string",
            "scanPoolId": "string",
            "url": "string",
            "apiKey": "string",
            "timeout": "string",
            "scanInstanceType": 0
          }
        ]
      },
      "filters": {
        "id": "string",
        "minimumSize": 0,
        "maximumSize": 0,
        "minimumDateTime": "2019-08-24T14:15:22Z",
        "maximumDateTime": "2019-08-24T14:15:22Z"
      },
      "fileDefinition": {
        "blockingStrategy": 0,
        "configurableBlockingStrategy": {
          "shouldTreatFilesWithSensitiveDataAsBlocked": true,
          "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
        }
      },
      "remediations": [
        {
          "id": "string",
          "workflowId": "string",
          "isEnabled": true,
          "configuration": "string",
          "type": 0,
          "handlingType": 0
        }
      ],
      "tenantId": "string"
    }
  ]
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entries	[Workflow]¦null	false	none	none

GetWorkflowByIdCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "entry": {
    "id": "string",
    "scanPoolId": "string",
    "scanProcessFlow": 0,
    "isIdentityScanningEnabled": true,
    "scanIntervalDays": 0,
    "rules": [
      "string"
    ],
    "type": 0,
    "workflowSupportedTechnologies": {
      "dlp": {
        "status": 0
      },
      "deepCdr": {
        "status": 0
      },
      "sandBox": {
        "status": 0
      },
      "metascan": {
        "status": 0
      }
    },
    "name": "string",
    "userAgent": "string",
    "scanPool": {
      "id": "string",
      "name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    },
    "failOverScanPoolId": "string",
    "failOverScanPool": {
```

```
"name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    },
    "filters": {
      "id": "string",
      "minimumSize": 0,
      "maximumSize": 0,
      "minimumDateTime": "2019-08-24T14:15:22Z",
      "maximumDateTime": "2019-08-24T14:15:22Z"
    },
    "fileDefinition": {
      "blockingStrategy": 0,
      "configurableBlockingStrategy": {
        "shouldTreatFilesWithSensitiveDataAsBlocked": true,
        "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
      }
    },
    "remediations": [
      {
        "id": "string",
        "workflowId": "string",
        "isEnabled": true,
        "configuration": "string",
        "type": 0,
        "handlingType": 0
      }
    ],
    "tenantId": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» entry	Workflow¦null	false	none	none

AddWorkflowCommandResponse


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "workflowId": "string"
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» workflowId	string¦null	false	none	none

AddWorkflowParameters

```
{
  "scanPoolId": "string",
  "rules": [
    "string"
  ],
  "name": "string",
  "userAgent": "string",
  "failOverScanPoolId": "string",
  "filters": {
    "minimumSize": 0,
    "maximumSize": 0,
    "minimumDateTime": "2019-08-24T14:15:22Z",
    "maximumDateTime": "2019-08-24T14:15:22Z"
  },
  "scanProcessFlow": 0,
  "isIdentityScanningEnabled": true,
  "scanIntervalDays": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
scanPoolId	string¦null	false	none	none
rules	[string]¦null	false	none	none
name	string	true	none	none
userAgent	string¦null	false	none	none
failOverScanPoolId	string¦null	false	none	none
filters	AddFilters¦null	false	none	none
scanProcessFlow	ScanProcessFlow¦null	false	none	- 0: WithScanning - 1: WithoutScanning
isIdentityScanningEnabled	boolean¦null	false	none	none
scanIntervalDays	integer¦null	false	none	none

AddFilters

```
{
  "minimumSize": 0,
  "maximumSize": 0,
  "minimumDateTime": "2019-08-24T14:15:22Z",
  "maximumDateTime": "2019-08-24T14:15:22Z"
}
```


Properties

Name	Type	Required	Restrictions	Description
minimumSize	integer(int64)	false	none	none
maximumSize	integer(int64)	false	none	none
minimumDateTime	string(date-time)	false	none	none
maximumDateTime	string(date-time)	false	none	none

UpdateWorkflowCommandResponse


```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

UpdateWorkflowParameters


```
{
  "id": "string",
  "scanPoolId": "string",
  "scanProcessFlow": 0,
  "rules": [
    "string"
  ],
  "name": "string",
  "userAgent": "string",
  "failOverScanPoolId": "string",
  "failOverDisabled": true,
  "filters": {
    "minimumSize": 0,
    "maximumSize": 0,
    "minimumDateTime": "2019-08-24T14:15:22Z",
    "maximumDateTime": "2019-08-24T14:15:22Z"
  },
  "fileDefinition": {
    "blockingStrategy": 0,
    "configurableBlockingStrategy": {
      "shouldTreatFilesWithSensitiveDataAsBlocked": true,
      "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
    }
  },
  "isIdentityScanningEnabled": true,
  "scanIntervalDays": 0
}
```

Properties

Name	Type	Required	Restrictions	Description
id	string	true	none	none
scanPoolId	string¦null	false	none	none
scanProcessFlow	ScanProcessFlow¦null	false	none	- 0: WithScanning - 1: WithoutScanning
rules	[string]¦null	false	none	none
name	string¦null	false	none	none
userAgent	string¦null	false	none	none
failOverScanPoolId	string¦null	false	none	none
failOverDisabled	boolean¦null	false	none	none
filters	UpdateFilters¦null	false	none	none
fileDefinition	FileDefinition¦null	false	none	none
isIdentityScanningEnabled	boolean¦null	false	none	none
scanIntervalDays	integer¦null	false	none	none

UpdateFilters

```
{
  "minimumSize": 0,
  "maximumSize": 0,
  "minimumDateTime": "2019-08-24T14:15:22Z",
  "maximumDateTime": "2019-08-24T14:15:22Z"
}
```

Properties

Name	Type	Required	Restrictions	Description
minimumSize	integer(int64)	false	none	none
maximumSize	integer(int64)	false	none	none
minimumDateTime	string(date-time)	false	none	none
maximumDateTime	string(date-time)	false	none	none

DeleteWorkflowCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```

Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

SetDefaultWorkflowCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none

RefreshSupportedTechnologiesCommandResponse

```
{
  "result": 0,
  "responseKey": "string",
  "responseMessage": "string",
  "responseMessageParams": {
    "property1": "string",
    "property2": "string"
  },
  "workflowDto": {
    "id": "string",
    "scanPoolId": "string",
    "scanProcessFlow": 0,
    "isIdentityScanningEnabled": true,
    "scanIntervalDays": 0,
    "rules": [
      "string"
    ],
    "type": 0,
    "workflowSupportedTechnologies": {
      "dlp": {
        "status": 0
      },
      "deepCdr": {
        "status": 0
      },
      "sandBox": {
        "status": 0
      },
      "metascan": {
        "status": 0
      }
    },
    "name": "string",
    "userAgent": "string",
    "scanPool": {
      "id": "string",
      "name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    },
    "failOverScanPoolId": "string",
    "failOverScanPool": {
```

```
"name": "string",
      "scanPoolType": 0,
      "isDefault": true,
      "scanInstances": [
        {
          "id": "string",
          "scanPoolId": "string",
          "url": "string",
          "apiKey": "string",
          "timeout": "string",
          "scanInstanceType": 0
        }
      ]
    },
    "filters": {
      "id": "string",
      "minimumSize": 0,
      "maximumSize": 0,
      "minimumDateTime": "2019-08-24T14:15:22Z",
      "maximumDateTime": "2019-08-24T14:15:22Z"
    },
    "fileDefinition": {
      "blockingStrategy": 0,
      "configurableBlockingStrategy": {
        "shouldTreatFilesWithSensitiveDataAsBlocked": true,
        "shouldTreatFilesWithVulnerabilitiesAsBlocked": true
      }
    },
    "remediations": [
      {
        "id": "string",
        "workflowId": "string",
        "isEnabled": true,
        "configuration": "string",
        "type": 0,
        "handlingType": 0
      }
    ],
    "tenantId": "string"
  }
}
```


Properties

allOf

Name	Type	Required	Restrictions	Description
anonymous	RestResponse	false	none	none

and

Name	Type	Required	Restrictions	Description
anonymous	object	false	none	none
» workflowDto	Workflow|null	false	none	none

RefreshSupportedTechnologiesCommandRequest

```
{
  "workflowId": "string"
}
```

Properties

Name	Type	Required	Restrictions	Description
workflowId	string	false	none	none

OPSWAT.

