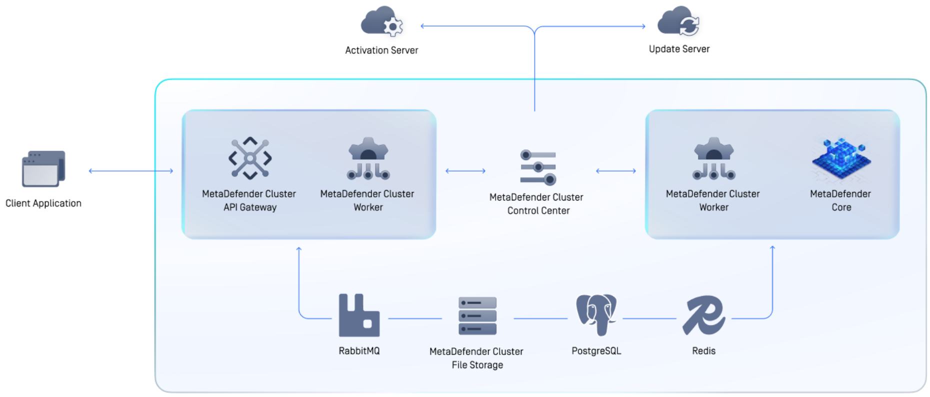
MetaDefender Cluster
v2.5.2

Table of Contents

Operating

Overview


Activation Server
Update Server
Client Application
MetaDefender Cluster API Gateway
MetaDefender Cluster Worker
MetaDefender Cluster Control Center
MetaDefender Cluster Worker
MetaDefender Core
RabbitMQ
MetaDefender Cluster File Storage
PostgreSQL
Redis

The MetaDefender Cluster (MD Cluster) is an approach to serve very large deployments while offering improved auto-scaling, high availability and fault tolerant capabilities for MetaDefender Core.

The MetaDefender Cluster consists of several components:

Component	Functionalities
MD Cluster Control Center	Assist administrators with user management, system health monitoring, and deploying or upgrading MetaDefender Core or MD Cluster API Gateway without any downtime.
MD Cluster Identity Service	Assist MD Cluster Control Center and MD Cluster API Gateway in client authentication, managing user activity sessions and authorization.
MD Cluster File Storage	Securely store and share files asynchronously across components in the cluster. The component manages the duration and duplication of files.
MD Cluster Worker	Deploy and monitor activities of MetaDefender Core and MD Cluster API Gateway.
MD Cluster API Gateway	Accept file scans, fetch scan statuses, and process download requests from clients.
MetaDefender Core	Scan the accepted files.
RabbitMQ - Message broker	Receives tasks from MD Cluster API Gateway and forwards them to MetaDefender Core instances for processing.
Redis - Caching server	Store in-progress results in memory for rapid retrieval.
PostgreSQL - Database sever	Permanently store scan results, configuration and executive reports.

The MD Cluster offers users two distinct interfaces. The first is a RESTful interface provided by MD Cluster **API Gateway** for applications to upload files for scanning, retrieve scan status, download processed files, or abort file scanning. The other is a Web UI provided by MD Cluster **Control Center** for the system administrator to manage licenses and users, modify workflow configurations, monitor the overall system, and remotely deploy or upgrade MD Cluster **API Gateway** or **MetaDefender Core**.

When a file is submitted to the MD Cluster **API Gateway** for scanning, its body content is securely transmitted to MD Cluster **File Storage** for subsequent use. **API Gateway** submits a scan task in RabbitMQ queue, and responds to the application with `data_id`. The task is delivered to healthy **MetaDefender Core** instances and one of them will accept the task. The file corresponding to the task is transmitted from MD Cluster **File Storage** to the instance's local storage, and the processing of the file takes place. Scan results produced by the processing are continuously recorded in Redis for fast retrieval and are finally stored in the PostgreSQL database for long-term storage. If created, the sanitized or watermarked file is securely transmitted to MD Cluster **File Storage** for future download by MD Cluster **API Gateway**.

In certain rate situations, if one of the **MetaDefender Core** instances unexpectedly ceases operation, its 'broken' files are delivered to other **MetaDefender Core** instances for continued processing without the need for applications to resubmit the files. By leveraging MD Cluster **File Storage** and RabbitMQ, **MetaDefender Core** instances within MetaDefender Cluster can collaborate in distributing the workload of archive extraction, greatly decreasing the overall time required to process archive files while utilizing the resources much more efficiently.

Using the Web Console from by MD Cluster **Control Center**, the system administrator is able to adjust workflow settings centrally and, after which the updates are automatically synced across all **MetaDefender Core** instances. The administrator can scale out the number of MD Cluster **API Gateway** or **MetaDefender Core** instances if additional power is required. He or she can also upgrade the instances seamlessly while the file processing is occurring. All statistical data and health information for components, along with executive reports, can be accessed easily through the Web UI of MD Cluster **Control Center**.

System requirements

Windows

Component	Minimum version	Dependencies	Recommended System Specs
PostgreSQL Database Server	16.9		Vendor recommendation
RabbitMQ Messaging Broker	3.13.0	64 bit Erlang/OTP from version 26.0 to 26.2.x.	Vendor recommendation
MetaDefender Cluster File Storage	2.5.2	Microsoft Visual C++ Redistributable 2019 version 14.29.30139.0 or above.	Minimum of 8 CPU cores and 8 GB of RAM required.
MetaDefender Cluster Control Center	2.5.2	Microsoft Visual C++ Redistributable 2019 version 14.29.30139.0 or above.	Minimum of 4 CPU cores and 4 GB of RAM required.
MetaDefender Cluster Identity Service	2.5.2	Microsoft Visual C++ Redistributable 2019 version 14.29.30139.0 or above.	Minimum of 4 CPU cores and 4 GB of RAM required.
MetaDefender Cluster Worker for **MetaDefender Cluster API Gateway**	2.5.2	Microsoft Visual C++ Redistributable 2019 version 14.29.30139.0 or above.	Minimum of 4 CPU cores and 8 GB of RAM required.
MetaDefender Cluster Worker for **MetaDefender Core**	2.5.2	Microsoft Visual C++ Redistributable 2019 version 14.29.30139.0 or above.	System Configuration

Info

`WMIC`, by default, is disabled since Windows 11. To enable it, please run the following command as Administrator in Command Prompt:

```
DISM /Online /Add-Capability /CapabilityName:WMIC
```

Debian/Ubuntu or Red Hat/Rocky

Component	Minimum version	Dependencies	Recommended System Specs
PostgreSQL Database Server	16.9		Vendor recommendation
Redis Caching Server	7.0.5		Vendor recommendation
RabbitMQ Messaging Broker	3.13.0	64 bit Erlang/OTP 25.0 or above.	Vendor recommendation
MetaDefender Cluster File Storage	2.5.2	`uuid` package. `tar` tool. `lsb_release` tool.	Minimum of 8 CPU cores and 8 GB of RAM required.
MetaDefender Cluster Control Center	2.5.2	`uuid` package. `tar` tool. `lsb_release` tool.	Minimum of 4 CPU cores and 4 GB of RAM required.
MetaDefender Cluster Identity Service	2.5.2	`uuid` package. `tar` tool. `lsb_release` tool.	Minimum of 4 CPU cores and 4 GB of RAM required.
MetaDefender Cluster Worker for **MetaDefender Cluster API Gateway**	2.5.2	`uuid` package. `tar` tool. `lsb_release` tool.	Minimum of 4 CPU cores and 8 GB of RAM required.

Component	Minimum version	Dependencies	Recommended System Specs
MetaDefender Cluster Worker for **MetaDefender Core**	2.5.2	`uuid` package. `tar` tool. `lsb_release` tool.	System configuration

ⓘ **Info**

`tar`, by default, is not included in some Linux distributions. Please run the following command in Terminal to install `tar`:

- Debian/Ubuntu: `sudo apt install tar`
- Red Hat/Rocky: `sudo dnf install tar`

ⓘ **Info**

`lsb_release`, by default, is not included in **Rocky**. Please run the following command in Terminal to install `lsb_release`

```
sudo dnf install -y yum-utils
```

```
sudo dnf config-manager --set-enabled devel
```

```
sudo dnf update -y
```

```
sudo dnf install -y redhat-lsb-core
```

Installation

This section includes guidance for installing and setting up the MetaDefender Cluster on physical machines, virtual machines, or in containers.

Physical or Virtual Machine-Based Setup

Prerequisite

Before executing the setup, please ensure System requirements are met and that any necessary dependencies are installed.

Installation order

MetaDefender Cluster consists of the following components along with their corresponding default ports.

Step	Component	How to install (short name)	Default port
1	Redis Caching Server	Redis	6379
2	RabbitMQ Message Broker	RabbitMQ	5672
3	PostgreSQL Database Server	PostgreSQL	5432
4	MetaDefender Cluster **File Storage**	MD Cluster File Storage	8890
5	MetaDefender Cluster **Identity Service**	MD Cluster Identity Service	8891
6	MetaDefender Cluster **Control Center**	MD Cluster Control Center	8892
7	MetaDefender Cluster **Worker**	MD Cluster Worker	8893

The system administrator should adhere to the following service installation sequence to prevent conflicts:

1. Install Redis, RabbitMQ, Postgres, MD Cluster **File Storage**, MD Cluster **Identity Service**.
2. Install MD Cluster **Control Center**.
3. Install MD Cluster **Worker** on the targeted machines (for deploying MD Cluster **API Gateway** or **MetaDefender Core**).

> **Warning**
>
> An exception rule for the firewall needs to be created to permit both incoming (inbound) and outgoing (outbound) connections to every component.

> **Info**
>
> While many components can be set up on a single machine, they should be installed individually on different machines according to their features. Kindly consult Best practices for further information.

Installation

Install Redis Caching Server

> **Info**
>
> Redis version 7.0 or higher is required.
>
> Only Redis on Linux is officially recommended.

1. Follow steps to install Redis Caching server.
2. Access Redis configuration file `/etc/redis/redis.conf` for editing.
3. Comment out the `bind` setting and set `protected-mode` option to **no**.

redis.conf none

```
...
# The following line should be commented
# bind 127.0.0.1
...
# The following line should be uncommented and set to no
protected-mode no
...
```

4. Restart the service.

bash

```
# Red Hat/Rocky
 $ sudo systemctl enable redis
 $ sudo systemctl restart redis

# Debian/Ubuntu
$ sudo systemctl enable redis-server
$ sudo systemctl restart redis-server
```

Install RabbitMQ Message Broker

Info

RabbitMQ version 3.13.0 or higher is required.

Warning

RabbitMQ functions effectively only with specific supported versions of Erlang. Please refer to the link for the Erlang-RabbitMQ compatibility matrix.

Windows

1. Download Erlang and follow the instructions to install Erlang.
2. Download RabbitMQ for Windows.
3. Run the executable file as administrator, follow instructions to complete the RabbitMQ installation.
4. In Command Prompt, change working directory to `<RabbitMQ installation folder>/rabbitmq_server-<version>/sbin` and run the following command:

None bash

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

Linux

1. Download Erlang and follow the instructions to install Erlang and its dependencies.

2. Download RabbitMQ for Red Hat/Rocky or Debian/Ubuntu .
3. In Terminal, run the following command:

bash

```
# Red Hat/Rocky
$ sudo rpm -Uvh --nodeps rabbitmq-server-<rabbitmq
version>.el8.noarch.rpm
$ sudo systemctl enable rabbitmq-server
$ sudo systemctl start rabbitmq-server

# Debian/Ubuntu
$ sudo dpkg -i rabbitmq-server_<rabbitmq version>_all.deb
```

4. In Terminal, run the following command:

None bash

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

Install PostgreSQL Database Server

> **Info**
>
> PostgreSQL version 16.9 or higher is required.
>
> `pg_trgm` extension is required for PostgreSQL running on Linux.

1. Download PostgreSQL Database Server.
2. Follow steps to setup Postgres Database Server to allow connections from external applications.
3. Restart Postgres Database Server.

Install MD Cluster File Storage

1. Build Ignition file for MD Cluster **File Storage** service.
2. Start **Command Prompt as Administrator on Windows** or **Terminal on Linux** and run the following command:

bash

```
# Windows
> msiexec.exe /i <md_cluster_file_storage_package> /qn

# Debian or Ubuntu
$ sudo apt -y install uuid
$ sudo dpkg -i <md_cluster_file_storage_package> || sudo apt
install -f

# Red Hat or Rocky
$ sudo dnf -y install uuid
$ sudo yum install <md_cluster_file_storage_package> -y
```

3. Check the service status.

Install MD Cluster Identity Service

1. Build Ignition file for MD Cluster **Identity Service**.
2. Start **Command Prompt as Administrator on Windows** or **Terminal on Linux** and run the following command:

bash

```
# Windows
> msiexec.exe /i <md_cluster_identity_service_package> /qn

# Ubuntu or Debian
$ sudo apt -y install uuid
$ sudo dpkg -i <md_cluster_identity_service_package> || sudo
apt install -f

# Red Hat or Rocky
$ sudo dnf -y install uuid
$ sudo yum install <md_cluster_identity_service_package> -y
```

3. Check the service status.

Install MD Cluster Control Center

1. Build Ignition file for MD Cluster **Control Center** service.
2. Start **Command Prompt as Administrator on Windows** or **Terminal on Linux** and run the following command:

bash

```
# Windows
> msiexec.exe /i <md_cluster_control_center_package> /qn

# Ubuntu or Debian
$ sudo apt -y install uuid
$ sudo dpkg -i <md_cluster_control_center_package> || sudo apt
install -f

# Red Hat or Rocky
$ sudo dnf -y install uuid
$ sudo yum install <md_cluster_control_center_package> -y
```

3. Check the service status.

Setup Data Lake and Data Warehouse

1. Go to `C:\Program Files\OPSWAT\MetaDefender Cluster Control Center` directory in **Windows Command Prompt** or `/usr/sbin` directory in **Linux Terminal**.
2. Run the following command:

bash

```
# Windows
> md-cluster-dbready.exe --host=<postgres-host> --port=
<postgres-port> --user=<postgres-user> --password=<postgres-
password> --target=lake,warehouse

# Linux (Ubuntu, Debian, Red Hat or Rocky)
$ md-cluster-dbready --host=<postgres-host> --port=<postgres-
port> --user=<postgres-user> --password=<postgres-password> --
target=lake,warehouse
```

ℹ **Info**

Make sure the `postgres-user` possesses **superuser** rights to successfully create the database.

Install MD Cluster Worker

Info

You need to prepare at least **two** workers: one for **MetaDefender Core** and the other for MD Cluster **API Gateway**.

1. Build Ignition file for MD Cluster **Worker** service.
2. Start **Command Prompt as Administrator on Windows** or **Terminal on Linux** and run the following command:

bash

```
# Windows
> msiexec.exe /i <md_cluster_worker_package> /qn

# Ubuntu or Debian
$ sudo apt -y install uuid
$ sudo dpkg -i <md_cluster_worker_package> || sudo apt install
-f

# Red Hat or Rocky
$ sudo dnf -y install uuid
$ sudo yum install <md_cluster_worker_package> -y
```

3. Check the service status.
4. Repeat the above steps for other MD Cluster **Workers**.

Configurations

When all MD Cluster **Worker** instances are installed successfully, it marks a completed installation. Now, heading to essential configuration steps.

Connect essential services

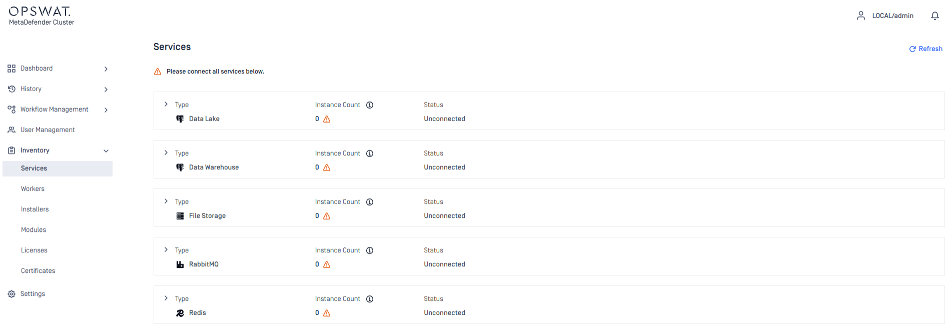
Info

Essential services for the Cluster includes Redis, Postgres, RabbitMQ, and MD Cluster **File Storage**.

The system is operational only when MD Cluster **Control Center** can effectively connect to all essential services.

1. Sign in to MD Cluster **Control Center** web console with the initial administrator user account that you created in Install MD Cluster **Identity Service**.

2. Go to `Inventory` > `Services`, open the relevant service category, and click on `Add service`.

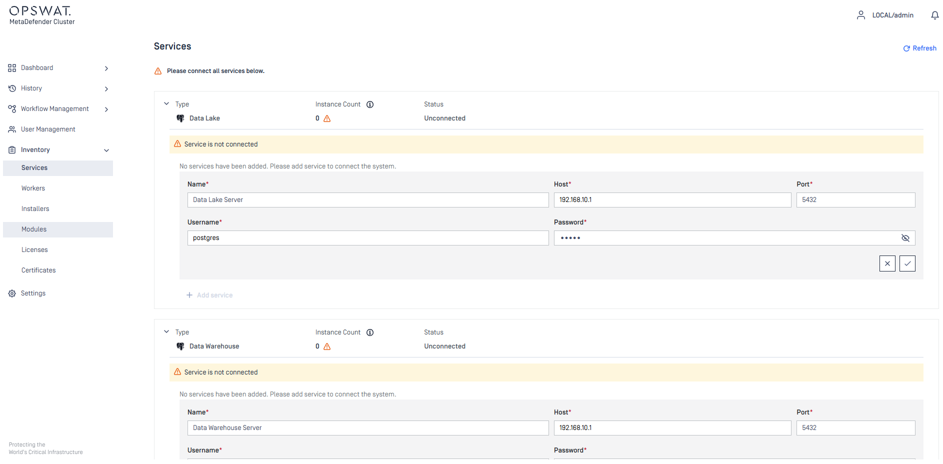


Services

Please connect all services below.

Type	Instance Count	Status
Data Lake	0 ⚠	Unconnected
Data Warehouse	0 ⚠	Unconnected
File Storage	0 ⚠	Unconnected
RabbitMQ	0 ⚠	Unconnected
Redis	0 ⚠	Unconnected

3. Complete all necessary fields as specified by the selected services.



Services

Please connect all services below.

Type	Instance Count	Status
Data Lake	0 ⚠	Unconnected

Service is not connected

No services have been added. Please add service to connect the system.

Name*: Data Lake Server
Host*: 192.168.10.1
Port*: 5432
Username*: postgres
Password*: •••••

Type	Instance Count	Status
Data Warehouse	0 ⚠	Unconnected

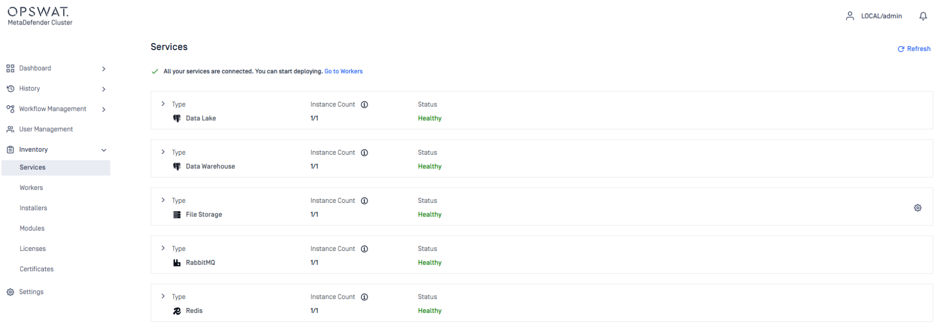
Service is not connected

No services have been added. Please add service to connect the system.

Name*: Data Warehouse Server
Host*: 192.168.10.1
Port*: 5432
Username*
Password*

4. Save result.

5. Check status of the service connections.



Services

✓ All your services are connected. You can start deploying. Go to Workers

Type	Instance Count	Status
Data Lake	1/1	Healthy
Data Warehouse	1/1	Healthy
File Storage	1/1	Healthy
RabbitMQ	1/1	Healthy
Redis	1/1	Healthy

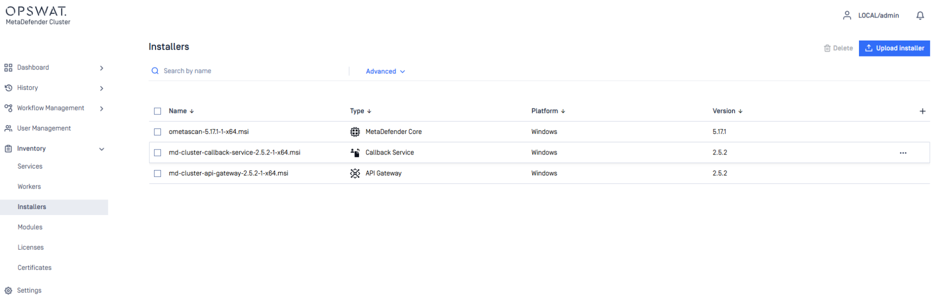
Submit MetaDefender Core and MD Cluster API Gateway installers

Info

The installers of **MetaDefender Core,** MD Cluster **API Gateway** and MD Cluster **Callback Service** will be subsequently deployed on MD Cluster **Worker** remotely by MD Cluster **Control Center**.

Various versions of installation files may be submitted to MD Cluster **Control Center**. The correct version to install will be chosen during Deployment phase.

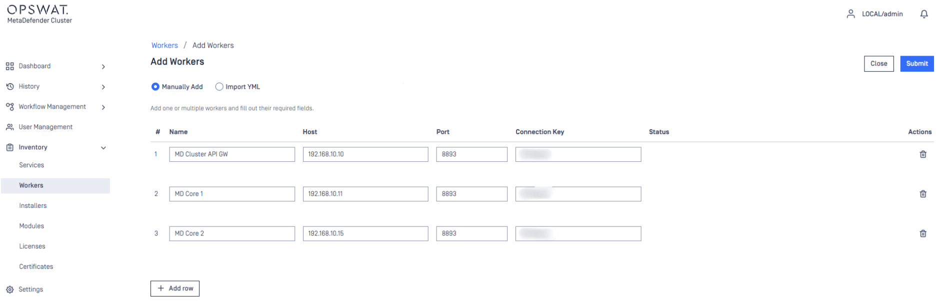
1. Sign in to MD Cluster **Control Center** web console with the initial administrator user account that you created in Install MD Cluster **Identity Service**.
2. Go to `Inventory` > Installers and select `Upload installer` .
3. Select **MetaDefender Core,** MD Cluster **API Gateway** or MD Cluster **Callback Service** installation files.
4. Click `Upload` .



☐	Name ↓	Type ↓	Platform ↓	Version ↓	+
☐	ometascan-5.17.1-1-x64.msi	MetaDefender Core	Windows	5.17.1	
☐	md-cluster-callback-service-2.5.2-1-x64.msi	Callback Service	Windows	2.5.2	···
☐	md-cluster-api-gateway-2.5.2-1-x64.msi	API Gateway	Windows	2.5.2	

Connect to MD Cluster Workers

1. Sign in to the MD Cluster **Control Center** web console with the initial administrator account that you created during the installation of MD Cluster **Identity Service**.
2. Go to `Inventory` > `Workers` and select `Add workers`.
3. Complete the required fields to add new workers and click `Submit`.



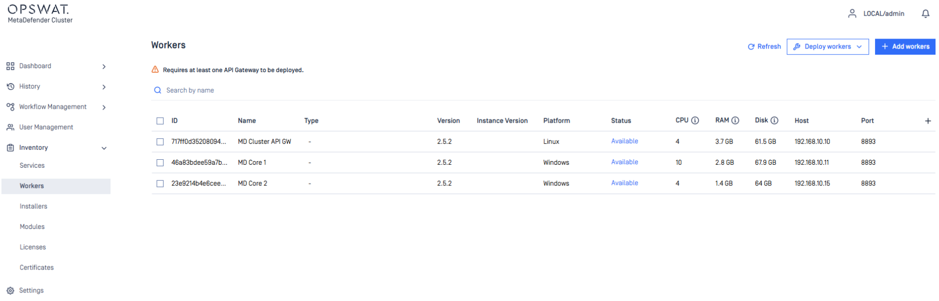
Add Workers

● Manually Add ○ Import YML

Add one or multiple workers and fill out their required fields.

#	Name	Host	Port	Connection Key	Status	Actions
1	MD Cluster API GW	192.168.10.10	8893	[redacted]		🗑
2	MD Core 1	192.168.10.11	8893	[redacted]		🗑
3	MD Core 2	192.168.10.15	8893	[redacted]		🗑

4. Check the status of MD Cluster **Worker** connections.



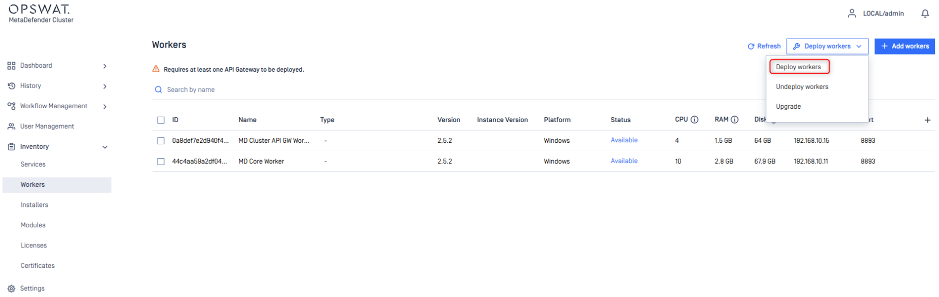
Workers

Requires at least one API Gateway to be deployed.

□	ID	Name	Type	Version	Instance Version	Platform	Status	CPU	RAM	Disk	Host	Port
□	717ff0d35208094...	MD Cluster API GW	-	2.5.2		Linux	Available	4	3.7 GB	61.5 GB	192.168.10.10	8893
□	46a83bdee59a7b...	MD Core 1	-	2.5.2		Windows	Available	10	2.8 GB	67.9 GB	192.168.10.11	8893
□	23e9214b4e6cee...	MD Core 2	-	2.5.2		Windows	Available	4	1.4 GB	64 GB	192.168.10.15	8893

Deploy MetaDefender Core and MD Cluster API Gateway instances

1. Sign in to the MD Cluster **Control Center** web console with the initial administrator account that you created during the installation of MD Cluster **Identity Service**.
2. Go to `Inventory` > `Workers` and select `Deploy workers`.

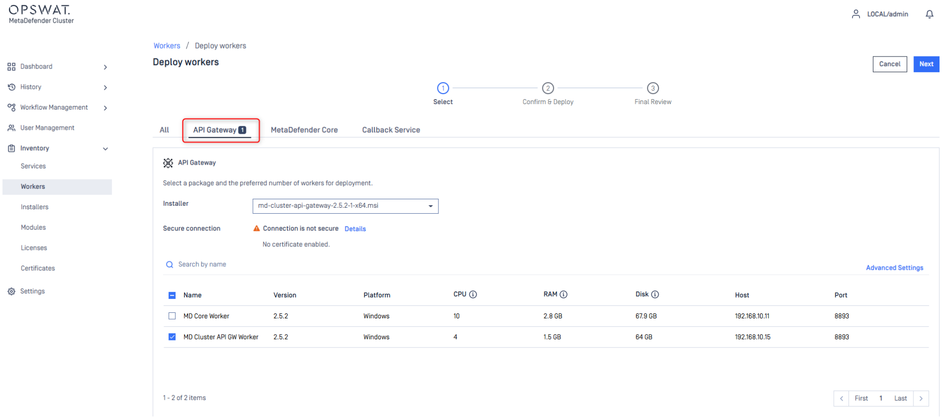


Workers

Requires at least one API Gateway to be deployed.

□	ID	Name	Type	Version	Instance Version	Platform	Status	CPU	RAM	Disk		rt
□	0a8def7e2d940f4...	MD Cluster API GW Wor...	-	2.5.2		Windows	Available	4	1.5 GB	64 GB	192.168.10.15	8893
□	44c4aa59a2df04...	MD Core Worker	-	2.5.2		Windows	Available	10	2.8 GB	67.9 GB	192.168.10.11	8893

3. In API Gateway tab, select workers for MD Cluster API Gateway deployment and decide which installer version will be used to deploy on the selected workers.



Deploy workers

API Gateway

Select a package and the preferred number of workers for deployment.

Installer: md-cluster-api-gateway-2.5.2-1-x64.msi

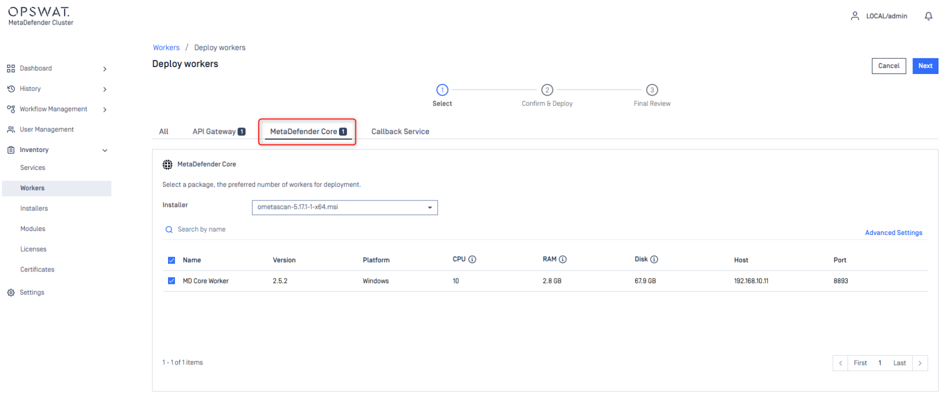
Secure connection: Connection is not secure Details

No certificate enabled.

☐	Name	Version	Platform	CPU	RAM	Disk	Host	Port
□	MD Core Worker	2.5.2	Windows	10	2.8 GB	67.9 GB	192.168.10.11	8893
☑	MD Cluster API GW Worker	2.5.2	Windows	4	1.5 GB	64 GB	192.168.10.15	8893

1 - 2 of 2 items

4. Repeat the step for selection of workers for MetaDefender Core deployment.



Select a package, the preferred number of workers for deployment.

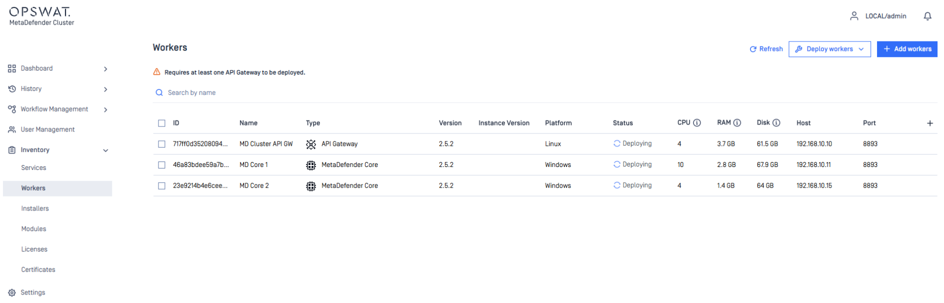
Installer: ometascan-5.17.1-1-x64.msi

☑	Name	Version	Platform	CPU	RAM	Disk	Host	Port
☑	MD Core Worker	2.5.2	Windows	10	2.8 GB	67.9 GB	192.168.10.11	8893

1 - 1 of 1 items

> **Info**
>
> MetaDefender Cluster is only ready to use when both MD Cluster API Gateway and MetaDefender Core are deployed.

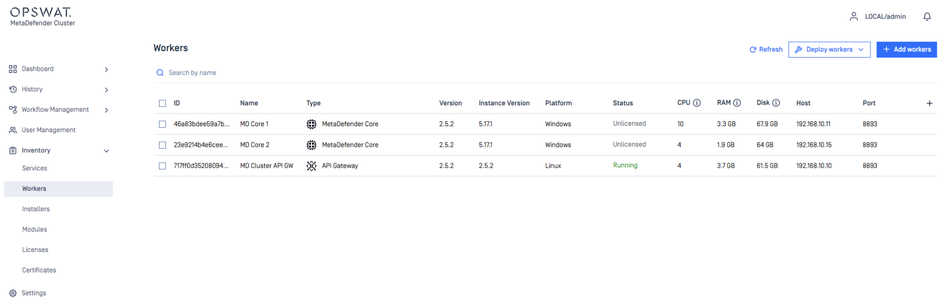
5. Click `Next`.
6. Confirm the deployment details, then click `Deploy` and `Finish`.
7. Hold off until the deployment is completed successfully.



Requires at least one API Gateway to be deployed.

□	ID	Name	Type	Version	Instance Version	Platform	Status	CPU	RAM	Disk	Host	Port
□	717ff0d35208094...	MD Cluster API GW	API Gateway	2.5.2		Linux	Deploying	4	3.7 GB	61.5 GB	192.168.10.10	8893
□	46a83bdee59a7b...	MD Core 1	MetaDefender Core	2.5.2		Windows	Deploying	10	2.8 GB	67.9 GB	192.168.10.11	8893
□	23e9214b4e6cee...	MD Core 2	MetaDefender Core	2.5.2		Windows	Deploying	4	1.4 GB	64 GB	192.168.10.15	8893

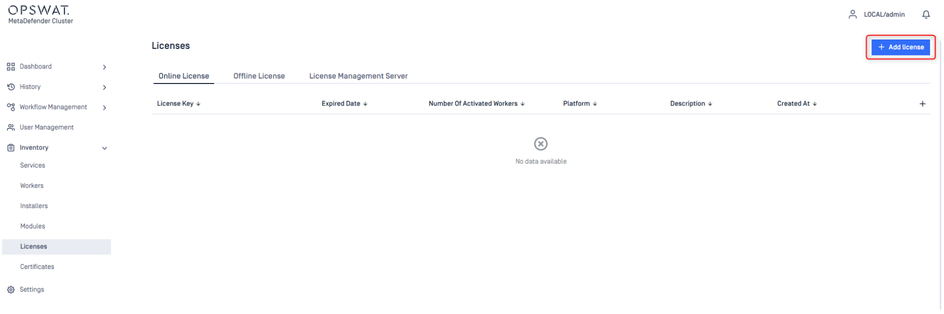
8. Although MetaDefender Core instances are deployed successfully on workers, a valid license is required for them to function.



□	ID	Name	Type	Version	Instance Version	Platform	Status	CPU	RAM	Disk	Host	Port
□	46a83bdee59a7b...	MD Core 1	MetaDefender Core	2.5.2	5.17.1	Windows	Unlicensed	10	3.3 GB	67.9 GB	192.168.10.11	8893
□	23e9214b4e6cee...	MD Core 2	MetaDefender Core	2.5.2	5.17.1	Windows	Unlicensed	4	1.9 GB	64 GB	192.168.10.15	8893
□	717ff0d35208094...	MD Cluster API GW	API Gateway	2.5.2	2.5.2	Linux	Running	4	3.7 GB	61.5 GB	192.168.10.10	8893

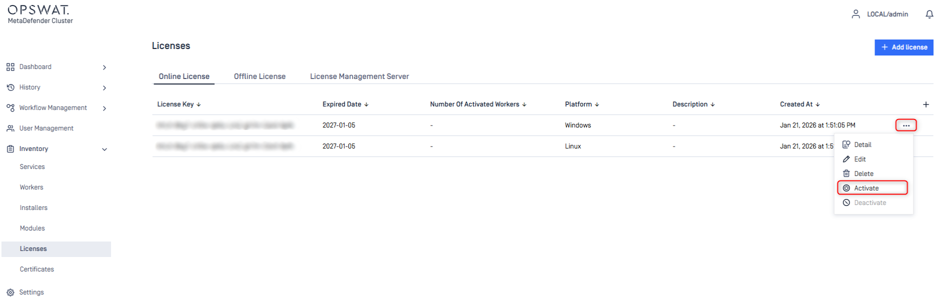
Activate online license

1. Sign in to the MD Cluster **Control Center** web console with the initial administrator account that you created during the installation of MD Cluster **Identity Service**.
2. Go to `Inventory` > `Licenses` and select `Online License`.
3. Click `Add license` and input your license key.



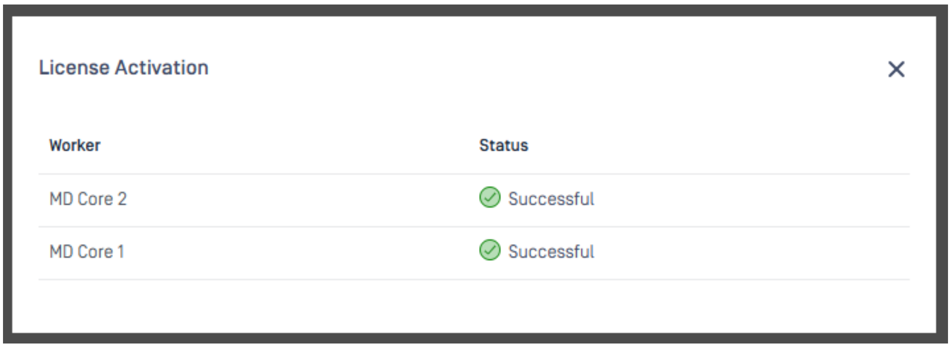
License Key ↓	Expired Date ↓	Number Of Activated Workers ↓	Platform ↓	Description ↓	Created At ↓

4. Click the three-dot icon beside your selected license key and choose `Activate`.



License Key ↓	Expired Date ↓	Number Of Activated Workers ↓	Platform ↓	Description ↓	Created At ↓
[redacted]	2027-01-05	-	Windows	-	Jan 21, 2026 at 1:51:05 PM
[redacted]	2027-01-05	-	Linux	-	Jan 21, 2026 at 1:5

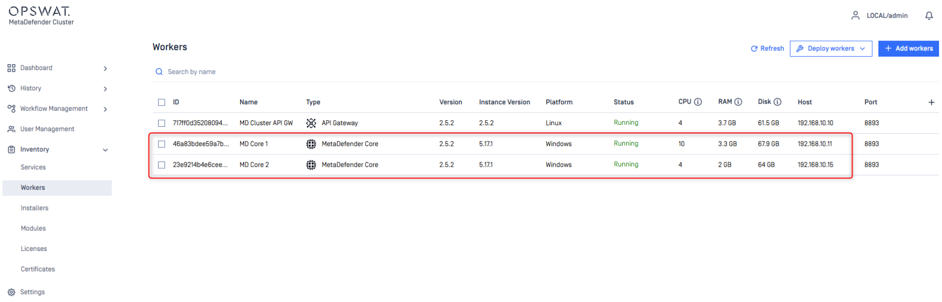
5. Wait until the license activation finishes on the deployed MetaDefender Core instances.



License Activation

Worker	Status
MD Core 2	✓ Successful
MD Core 1	✓ Successful

6. Navigate to `Inventory` > `Workers` to ensure the running status of the workers hosting MetaDefender Core.



□	ID	Name	Type	Version	Instance Version	Platform	Status	CPU ⓘ	RAM ⓘ	Disk ⓘ	Host	Port
□	717ff0d35208094...	MD Cluster API GW	API Gateway	2.5.2	2.5.2	Linux	Running	4	3.7 GB	61.5 GB	192.168.10.10	8893
□	46a83bdee59a7b...	MD Core 1	MetaDefender Core	2.5.2	5.17.1	Windows	Running	10	3.3 GB	67.9 GB	192.168.10.11	8893
□	23e9214b4e6cee...	MD Core 2	MetaDefender Core	2.5.2	5.17.1	Windows	Running	4	2 GB	64 GB	192.168.10.15	8893

7. MD Cluster **API Gateway** can now efficiently accept scan requests.

MetaDefender Cluster File Storage

Ignition file

Info

The ignition file is required only for a clean installation.

The following fields are essential for the ignition file:

- `secure.connection_key`
- `secure.private_key`
- `secure.certificate`

To install MetaDefender Cluster (MD Cluster) **File Storage** server, ignition file in YML format is required at the following location:

- Windows: `C:\opswat\md_cluster_file_storage.yml`
- Linux: `/etc/opswat/md_cluster_file_storage.yml`

The ignition file includes fields:

Key path	Value type	Accepted values	Required	Description
`secure.connection_key`	String	A string from 4 to 64 character long containing digits from 0 to 9 and characters from a/A to z/Z	**Required**	An arbitrary string that enables clients to connect to the server. Use this value as input when adding MD Cluster **File Storage** in the UI of MD Cluster **Control Center**.
`secure.private_key`	String		**Required**	Content of private key in X509 format.
`secure.certificate`	String		**Required**	Content of certificate in X509 format.
`storage.path`	String		Optional	Path to an **existing** directory where the MD Cluster File Storage server stores its files. The server requires full permissions to access the path in Linux.
`rest.host`	String		Optional	IP address (V4/V6) or host where the server resides on. Default value is `'*'` **Notes:** value `'*'` allows the service to accept connections from all network interfaces. To bind the service to a specific interface, specify its IP address or domain name. For example, to listen on all IPv4 interfaces, set the host to `0.0.0.0`

Key path	Value type	Accepted values	Required	Description
`rest.port`	Number		Optional	The port where the server resides on. Default value is `8890`
`log.streams[@].log_type`	String	- `file` - `syslog`	Optional	Type of log device.
`log.streams[@].log_level`	String	- `dump` - `debug` - `info` - `warning` - `error`	Optional	Level of log message.

Key path	Value type	Accepted values	Required	Description
`log.streams[@].log_path`	String	If `log.streams[@].log_type` is `"file"` then `log.streams[@].log_path` is the path to a file on file system where logs are written. If `log.streams[@].log_type` is `"syslog"` then - `log.streams[@].log_path` can be `[tcp/udp]://host:port` where host:port is the host and port to a remote syslog server that supports TCP or UDP protocol. - `log.streams[@].log_path` can be `"local"` to write log to local syslog server (Linux only).	Optional	Location where logs are written.

Info

If `storage.path` is not defined in the Ignition file, MD Cluster File Storage will save the submitted files in the default storage directory according to the platform:

- On Windows, `<install-directory>/data/storage`
- On Linux, `/var/lib/md-cluster-file-storage/storage`

Warning

The default storage directory will be deleted when MD Cluster **File Storage** is uninstalled.

Configuration file

After successfully installing, MD Cluster File Storage generates a configuration file with changeable settings at the following location:

- Windows: `C:\Program Files\OPSWAT\MetaDefender Cluster File Storage\md_cluster_file_storage.yml`
- Linux: `/etc/md-cluster-file-storage/md_cluster_file_storage.yml`

> **Info**
>
> The service must be restarted to take the new configurations into effect.

Sample

> **Info**
>
> OpenSSL or a similar tool (e.g., ssh-keygen) can create a pair of public and private keys in X.509 format.

yaml

```
secure:
  connection_key: "1234abcd" # [0-9a-zA-Z]{4,64}
  private_key: |
        -----BEGIN PRIVATE KEY-----

MIIJQwIBADANBgkqhkiG9w0BAQEFAASCCS0wggkpAgEAAoICAQCjYtuWaICCY0
tJ

PubxpIgIL+WWmz/fmK8IQr11Wtee6/IUyUlo5I602mq1qcLhT/kmpoR8Di3DAm
HK

nSWdPWtn1BtXLErLlUiHgZDrZWInmEBjKM1DZf+CvNGZ+EzPgBv5nTekLWcfI5
ZZ

toGuIP1Dl/IkNDw8zFz4cpiMe/BFGemyxdHhLrKHSm8Eo+nT734tItnHKT/m6D
SU

0xlZ13d6ehLRm7/+Nx47M3XMTRH5qKP/7TTE2s0U6+M0tsGI2zpRi+m6jzhNyM
BT

J1u58qAe3ZW5/+YAiuZYAB6n5bhUp4oFuB5wYbcBywVR8ujInpF8buWQUjy5N8
pS

Np7szdYsnLJpvAd0sibrNPjC0FQCNrpNjgJmIK3+mKk4kXX7ZTwefoAzTK4l2p
HN

uC53QVc/EF++GBLAxmvCDq9ZpMIYi7OmzkkAKKC9Ue6Ef217LFQCFIBKIzv9cg
i9

fwPMLhrKleoVRNsecBsCP569WgJXhUnwf2lon4fEZr3+vRuc9shfqnV0nPN1IM
Sn

zXCast7I2fiuRXdIz96KjlGQpP4XfNVA+RGL7aMnWOFIaVrKWLzAtgzoGMTvP/
Au

ehKXncBJhYtW0ltTioVx+5yTYSAZWl+IssmXjefxJqYi2/7QWmv1QC9psNcjTM
aB

QLN03T1Qelbs7Y27sxdEnNUth4kI+wIDAQABAoICAFWe8MQZb37k2gdAV3Y6aq
8f

qokKQqbCNLd3giGFwYkezHXoJfg6Di7oZxNcKyw35LFEghkgtQqErQqo35VPIo
H+

vXUpWOjnCmM4muFA9/cX6mYMc8TmJsg0ewLdBCOZVw+wPABlaqz+0UOiSMMftp
k9

fz9JwGd8ERyBsT+tk3Qi6D0vPZVsC1KqxxL/cwIFd3Hf2ZBtJXe0KBn1pktWht
5A
```

```
Kqx9mld2Ovl7NjgiC1Fx9r+fZw/iOabFFwQA4dr+R8mEMK/7bd4VXfQ1o/QGGb
MT

G+ulFrsiDyP+rBIAaGC0i7gDjLAIBQeDhP409ZhswIEc/GBtODU372a2CQK/u4
Q/

HBQvuBtKFNkGUooLgCCbFxzgNUGc83GB/6IwbEM7R5uXqsFiE71LpmroDyjKTl
Q8

YZkpIcLNVLw0usoGYHFm2rvCyEVlfsE3Ub8cFyTFk50SeOcF2QL2xzKmmbZEpX
gl

xBHR0hjgon0IKJDGfor4bHO7Nt+1Ece8u2oTEKvpz5aIn44OeC5mApRGy83/0b
vs

esnWjDE/bGpoT8qFuy+0urDEPNId44XcJm1IRIlG56ErxC3l0s11wrIpTmXXck
qw

zFR9s2z7f0zjeyxqZg4NTPI7wkM3M8BXlvp2GTBIeoxrWB4V3YArwu8QF80QBg
Vz

mgHl24nTg00UH1OjZsABAoIBAQDOxftSDbSqGytcWqPYP3SZHAWDA0O4ACEM+e
Cw

au9ASutl0IDlNDMJ8nC2ph25BMe5hHDWp2cGQJog7pZ/3qQogQho2gUniKDifN
77

40QdyllTzTVROqmP8+efreIvqlzHmuqaGfGs5oTkZaWj5su+B+bT+9rIwZcwf
s5

YRINhQRx17qa++xh5mfE25c+M9fiIBTiNSo4lTxWMBShnK8xrGaMEmN7W0qTMb
FH

PgQz5FcxRjCCqwHilwNBeLDTp/ZECEB7y34khVh531mBE2mNzSVIQcGZP1I/Dv
Xj

W7UUNdgFwii/GW+6M0uUDy23UVQpbFzcV8o1C2nZc4Fb4zwBAoIBAQDKSJkFww
uR

naVJS6WxOKjX8MCu9/cKPnwBv2mmI2jgGxHTw5sr3ahmF5eTb8Zo19BowytN+t
r6

2ZFoIBA9Ubc9esEAU8l3fggdfM82cuR9sGcfQVoCh8tMg6BP8IBLOmbSUhN3PG
2m

39I802u0fFNVQCJKhx1m1MFFLOu7lVcDS9JN+oYVPb6MDfBLm5jOiPuYkFZ4gH
79

J7gXI0/YKhaJ7yXthYVkdrSF6Eooer4RZgma62Dd1VNzSq3JBo6rYjF7Lvd+Rw
```

```
DC

R1thHrmf/IXplxpNVkoMVxtzbrrbgnC25QmvRYc0rlS/kvM4yQhMH3eA7IycDZ
Mp

Y+0xm7I7jTT7AoIBAGKzKIMDXdCxBWKhNYJ8z7hiItNl1IZZMW2TPUiY0rl6ya
Ch

BVXjM9W0r07QPnHZsUiByqb743adkbTUjmxdJzjaVtxN7ZXwZvOVrY7I7fPWYn
CE

fXCr4+IVpZI/ZHZWpGX6CGSgT6EOjCZ5IUufIvEpqVSmtF8MqfXO9o9uIYLokr
WQ

x1dBl5UnuTLDqw8bChq7O5y6yfuWaOWvL7nxI8NvSsfj4y635gIa/0dFeBYZEf
HI

UlGdNVomwXwYEzgE/c19ruIowX7HU/NgxMWTMZhpazlxgesXybel+YNcfDQ4e3
RM

OMz3ZFiaMaJsGGNf4++d9TmMgk4Ns6oDs6Tb9AECggEBAJYzd+SOYo26iBu3nw
3L

65uEeh6xou8pXH0Tu4gQrPQTRZZ/nT3iNgOwqu1gRuxcq7TOjt41UdqIKO8vN7
/A

aJavCpaKoIMowy/aGCbvAvjNPpU3unU8jdl/t08EXs79S5IKPcgAx87sTTi7KD
N5

SYt4tr2uPEe53NTXuSatilG5QCyExIELOuzWAMKzg7CAiIlNS9foWeLyVkBgCQ
6S

me/L8ta+mUDy37K6vC34jh9vK9yrwF6X44ItRoOJafCaVfGI+175q/eWcqTX4q
+I

G4tKls4sL4mgOJLq+ra50aYMxbcuommctPMXU6CrrYyQpPTHMNVDQy2ttFdsq9
iK

TncCggEBAMmt/8yvPflS+xv3kg/ZBvR9JB1In2n3rUCYYD47ReKFqJ03Vmq5C9
nY

56s9w7OUO8perBXlJYmKZQhO4293lvxZD2Iq4NcZbVSCMoHAUzhzY3brdgtSIx
a2

gGveGAezZ38qKIU26dkz7deECY4vrsRkwhpTW0LGVCpjcQoaKvymAoCmAs8V2o
Mr

Ziw1YQ9uOUoWwOqm1wZqmVcOXvPIS2gWAs3fQlWjH9hkcQTMsUaXQDOD0aqkSY
3E
```

```
NqOvbCV1/oUpRi3076khCoAXI1bKSn/AvR3KDP14B5toHI/F5OTSEiGhhHesgR
rs
        fBrpEY1IATtPq1taBZZogRqI3rOkkPk=
        -----END PRIVATE KEY-----
  certificate: |
        -----BEGIN CERTIFICATE-----

MIIF5jCCA86gAwIBAgIJANq50IuwPFKgMA0GCSqGSIb3DQEBCwUAMIGGMQswCQ
YD

VQQGEwJHQjEQMA4GA1UECAwHRXJld2hvbjETMBEGA1UEBwwKQWxsIGFyb3VuZD
Eb

MBkGA1UECgwSbGlid2Vic29ja2V0cy10ZXN0MRIwEAYDVQQDDAlsb2NhbGhvc3
Qx

HzAdBgkqhkiG9w0BCQEWEG5vbmVAaW52YWxpZC5vcmcwIBcNMTgwMzIwMDQxNj
A3

WhgPMjExODAyMjQwNDE2MDdaMIGGMQswCQYDVQQGEwJHQjEQMA4GA1UECAwHRX
Jl

d2hvbjETMBEGA1UEBwwKQWxsIGFyb3VuZDEbMBkGA1UECgwSbGlid2Vic29ja2
V0

cy10ZXN0MRIwEAYDVQQDDAlsb2NhbGhvc3QxHzAdBgkqhkiG9w0BCQEWEG5vbm
VA

aW52YWxpZC5vcmcwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCjYt
uW

aICCY0tJPubxpIgIL+WWmz/fmK8IQr11Wtee6/IUyUlo5I602mq1qcLhT/kmpo
R8

Di3DAmHKnSWdPWtn1BtXLErLlUiHgZDrZWInmEBjKM1DZf+CvNGZ+EzPgBv5nT
ek

LWcfI5ZZtoGuIP1Dl/IkNDw8zFz4cpiMe/BFGemyxdHhLrKHSm8Eo+nT734tIt
nH

KT/m6DSU0xlZ13d6ehLRm7/+Nx47M3XMTRH5qKP/7TTE2s0U6+M0tsGI2zpRi+
m6

jzhNyMBTJ1u58qAe3ZW5/+YAiuZYAB6n5bhUp4oFuB5wYbcBywVR8ujInpF8bu
WQ

Ujy5N8pSNp7szdYsnLJpvAd0sibrNPjC0FQCNrpNjgJmIK3+mKk4kXX7ZTwefo
Az

TK4l2pHNuC53QVc/EF++GBLAxmvCDq9ZpMIYi7OmzkkAKKC9Ue6Ef217LFQCFI
```

```
BK

Izv9cgi9fwPMLhrKleoVRNsecBsCP569WgJXhUnwf2lon4fEZr3+vRuc9shfqn
V0

nPN1IMSnzXCast7I2fiuRXdIz96KjlGQpP4XfNVA+RGL7aMnWOFIaVrKWLzAtg
zo

GMTvP/AuehKXncBJhYtW0ltTioVx+5yTYSAZWl+IssmXjefxJqYi2/7QWmv1QC
9p

sNcjTMaBQLN03T1Qelbs7Y27sxdEnNUth4kI+wIDAQABo1MwUTAdBgNVHQ4EFg
QU

9mYU23tW2zsomkKTAXarjr2vjuswHwYDVR0jBBgwFoAU9mYU23tW2zsomkKTAX
ar

jr2vjuswDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAgEANjIBMr
ow

YNCbhAJdP7dhlhT2RUFRdeRUJD0IxrH/hkvb6myHHnK8nOYezFPjUlmRKUgNED
uA

xbnXZzPdCRNV9V2mShbXvCyiDY7WCQE2Bn44z26O0uWVk+7DNNLH9BnkwUtOnM
9P

wtmD9phWexm4q2GnTsiL6Ul6cy0QlTJWKVLEUQQ6yda582e23J1AXqtqFcpfoE
34

H3afEiGy882b+ZBiwkeV+oq6XVF8sFyr9zYrv9CvWTYlkpTQfLTZSsgPdEHYVc
jv

xQ2D+XyDR0aRLRlvxUa9dHGFHLICG34Juq5Ai6lM1EsoD8HSsJpMcmrH7MWw2c
Kk

ujC3rMdFTtte83wF1uuF4FjUC72+SmcQN7A386BC/nk2TTsJawTDzqwOu/VdZv
2g

1WpTHlumlClZeP+G/jkSyDwqNnTu1aodDmUa4xZodfhP1HWPwUKFcq8oQr148Q
YA

AOlbUOJQU7QwRWd1VbnwhDtQWXC92A2w1n/xkZSR1BM/NUSDhkBSUU1WjMbWg6
Gg

mnIZLRerQCu1Oozr87rOQqQakPkyt8BUSNK3K42j2qcfhAONdRl8Hq8Qs5pupy
+s

8sdCGDlwR3JNCMv6u48OK87F4mcIxhkSefFJUFII25pCGN5WtE4p5l+9cnO1Gr
IX
```

```
e2Hl/7M0c/lbZ4FvXgARlex2rkgS0Ka06HE=
-----END CERTIFICATE-----
```

MetaDefender Identity Service

Ignition file

ⓘ Info

The ignition file is required only for a clean installation.

The following fields are essential for the ignition file:

- `secure.connection_key`
- `secure.private_key`
- `secure.certificate`
- `database.host`
- `database.port`
- `database.user`
- `database.password`

To install MetaDefender Cluster (MD Cluster) Identity Service server, ignition file in YML format is required at the following location:

- Windows: `C:\opswat\md_cluster_identity_service.yml`
- Linux: `/etc/opswat/md_cluster_identity_service.yml`

The ignition file includes fields:

Key path	Value type	Accepted values	Required	Description
`secure.connection_key`	String	A string from 4 to 64 character long containing digits from 0 to 9 and characters from a/A to z/Z	**Required**	An arbitrary string that enables clients to connect to the server. Use this value for the key `identity.connection_key` in configuration file of MD Cluster Control Center.
`secure.private_key`	String		**Required**	Content of private key in X509 format.
`secure.certificate`	String		**Required**	Content of certificate in X509 format.
`database.host`	String		**Required**	IP address / domain name of the server where PostgreSQL server locates.
`database.port`	Number		**Required**	Port of PostgreSQL server is listening for connections from clients.
`database.user`	String		**Required**	PostgreSQL server's user. SUPERUSER privilege is required to setup the server's database and extensions for the first time.
`database.password`	String		**Required**	PostgreSQL server's user credentials.

Key path	Value type	Accepted values	Required	Description
`rest.host`	String		Optional	IP address (V4/V6) or host where the server resides on. Default value is `'*'` **Notes:** value `'*'` allows the service to accept connections from all network interfaces. To bind the service to a specific interface, specify its IP address or domain name. For example, to listen on all IPv4 interfaces, set the host to `0.0.0.0`
`rest.port`	Number		Optional	The port where the server resides on. Default value is `8891`
`log.streams[@].log_type`	String	• `file` • `syslog`	Optional	Type of log device.
`log.streams[@].log_level`	String	• `dump` • `debug` • `info` • `warning` • `error`	Optional	Level of log message.

Key path	Value type	Accepted values	Required	Description
`log.streams[@].log_path`	String	If `log.streams[@].log_type` is `"file"` then `log.streams[@].log_path` is the path to a file on file system where logs are written. If `log.streams[@].log_type` is `"syslog"` then - `log.streams[@].log_path` can be `[tcp/udp]://host:port` where host:port is the host and port to a remote syslog server that supports TCP or UDP protocol. - `log.streams[@].log_path` can be `"local"` to write log to local syslog server (Linux only).	Optional	Location where logs are written.
`user.name`	String		Optional	User name for the initial administrator user account.
`user.password`	String		Optional	Password for the initial administrator user account.
`user.email`	String	Basic email format, a string starts with non `whitespace`/non `@` characters, contains one `@` symbol, and ends with non `whitespace`/non `@` characters.	Optional	E-mail address for the initial administrator user account.

Key path	Value type	Accepted values	Required	Description
`user.apikey`	String	string of exactly 36 characters composed of uppercase and lowercase letters (A-Z, a-z) and digits (0-9)	Optional	API key for the initial administrator user account.

Configuration file

After successfully installing, MD Cluster Identity Service generates a configuration file with changeable settings at the following location

- Windows: `C:\Program Files\OPSWAT\MetaDefender Cluster Identity Service\md_cluster_identity_service.yml`
- Linux: `/etc/md-cluster-identity-service/md_cluster_identity_service.yml`

Info

The service must be restarted to take the new configurations into effect.

Sample

Warning

`database.host`, `database.port`, `database.user`, and `database.password` should be updated with the appropriate values of your Postgres host/IP, port, username, and password.

Info

OpenSSL or a similar tool (e.g., ssh-keygen) can create a pair of public and private keys in X.509 format.

yaml

```
database:
  host: "your_postgres_host"
  port: 5432
  user: "your_postgres_username"
  password: "your_postgres_admin_password"
secure:
  connection_key: "1234abcd"    # [0-9a-zA-Z]{4,64}
  certificate: |
    -----BEGIN CERTIFICATE-----

MIIF5jCCA86gAwIBAgIJANq50IuwPFKgMA0GCSqGSIb3DQEBCwUAMIGGMQswCQ
YD

VQQGEwJHQjEQMA4GA1UECAwHRXJld2hvbjETMBEGA1UEBwwKQWxsIGFyb3VuZD
Eb

MBkGA1UECgwSbGlid2Vic29ja2V0cy10ZXN0MRIwEAYDVQQDDAlsb2NhbGhvc3
Qx

HzAdBgkqhkiG9w0BCQEWEG5vbmVAaW52YWxpZC5vcmcwIBcNMTgwMzIwMDQxNj
A3

WhgPMjExODAyMjQwNDE2MDdaMIGGMQswCQYDVQQGEwJHQjEQMA4GA1UECAwHRX
Jl

d2hvbjETMBEGA1UEBwwKQWxsIGFyb3VuZDEbMBkGA1UECgwSbGlid2Vic29ja2
V0

cy10ZXN0MRIwEAYDVQQDDAlsb2NhbGhvc3QxHzAdBgkqhkiG9w0BCQEWEG5vbm
VA

aW52YWxpZC5vcmcwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCjYt
uW

aICCY0tJPubxpIgIL+WWmz/fmK8IQr11Wtee6/IUyUlo5I602mq1qcLhT/kmpo
R8

Di3DAmHKnSWdPWtn1BtXLErLlUiHgZDrZWInmEBjKM1DZf+CvNGZ+EzPgBv5nT
ek

LWcfI5ZZtoGuIP1Dl/IkNDw8zFz4cpiMe/BFGemyxdHhLrKHSm8Eo+nT734tIt
nH

KT/m6DSU0xlZ13d6ehLRm7/+Nx47M3XMTRH5qKP/7TTE2s0U6+M0tsGI2zpRi+
m6

jzhNyMBTJ1u58qAe3ZW5/+YAiuZYAB6n5bhUp4oFuB5wYbcBywVR8ujInpF8bu
WQ
```

```
Ujy5N8pSNp7szdYsnLJpvAd0sibrNPjC0FQCNrpNjgJmIK3+mKk4kXX7ZTwefo
Az

TK4l2pHNuC53QVc/EF++GBLAxmvCDq9ZpMIYi7OmzkkAKKC9Ue6Ef217LFQCFI
BK

Izv9cgi9fwPMLhrKleoVRNsecBsCP569WgJXhUnwf2lon4fEZr3+vRuc9shfqn
V0

nPN1IMSnzXCast7I2fiuRXdIz96KjlGQpP4XfNVA+RGL7aMnWOFIaVrKWLzAtg
zo

GMTvP/AuehKXncBJhYtW0ltTioVx+5yTYSAZWl+IssmXjefxJqYi2/7QWmv1QC
9p

sNcjTMaBQLN03T1Qelbs7Y27sxdEnNUth4kI+wIDAQABo1MwUTAdBgNVHQ4EFg
QU

9mYU23tW2zsomkKTAXarjr2vjuswHwYDVR0jBBgwFoAU9mYU23tW2zsomkKTAX
ar

jr2vjuswDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAgEANjIBMr
ow

YNCbhAJdP7dhlhT2RUFRdeRUJD0IxrH/hkvb6myHHnK8nOYezFPjUlmRKUgNED
uA

xbnXZzPdCRNV9V2mShbXvCyiDY7WCQE2Bn44z26O0uWVk+7DNNLH9BnkwUtOnM
9P

wtmD9phWexm4q2GnTsiL6Ul6cy0QlTJWKVLEUQQ6yda582e23J1AXqtqFcpfoE
34

H3afEiGy882b+ZBiwkeV+oq6XVF8sFyr9zYrv9CvWTYlkpTQfLTZSsgPdEHYVc
jv

xQ2D+XyDR0aRLRlvxUa9dHGFHLICG34Juq5Ai6lM1EsoD8HSsJpMcmrH7MWw2c
Kk

ujC3rMdFTtte83wF1uuF4FjUC72+SmcQN7A386BC/nk2TTsJawTDzqwOu/VdZv
2g

1WpTHlumlClZeP+G/jkSyDwqNnTu1aodDmUa4xZodfhP1HWPwUKFcq8oQr148Q
YA

AOlbUOJQU7QwRWd1VbnwhDtQWXC92A2w1n/xkZSR1BM/NUSDhkBSUU1WjMbWg6
Gg

mnIZLRerQCu1Oozr87rOQqQakPkyt8BUSNK3K42j2qcfhAONdRl8Hq8Qs5pupy
+s
```

```
8sdCGDlwR3JNCMv6u48OK87F4mcIxhkSefFJUFII25pCGN5WtE4p5l+9cnO1Gr
IX
    e2Hl/7M0c/lbZ4FvXgARlex2rkgS0Ka06HE=
    -----END CERTIFICATE-----

  private_key: |
    -----BEGIN PRIVATE KEY-----

MIIJQwIBADANBgkqhkiG9w0BAQEFAASCCS0wggkpAgEAAoICAQCjYtuWaICCY0
tJ

PubxpIgIL+WWmz/fmK8IQr11Wtee6/IUyUlo5I602mq1qcLhT/kmpoR8Di3DAm
HK

nSWdPWtn1BtXLErLlUiHgZDrZWInmEBjKM1DZf+CvNGZ+EzPgBv5nTekLWcfI5
ZZ

toGuIP1Dl/IkNDw8zFz4cpiMe/BFGemyxdHhLrKHSm8Eo+nT734tItnHKT/m6D
SU

0xlZ13d6ehLRm7/+Nx47M3XMTRH5qKP/7TTE2s0U6+M0tsGI2zpRi+m6jzhNyM
BT

J1u58qAe3ZW5/+YAiuZYAB6n5bhUp4oFuB5wYbcBywVR8ujInpF8buWQUjy5N8
pS

Np7szdYsnLJpvAd0sibrNPjC0FQCNrpNjgJmIK3+mKk4kXX7ZTwefoAzTK4l2p
HN

uC53QVc/EF++GBLAxmvCDq9ZpMIYi7OmzkkAKKC9Ue6Ef217LFQCFIBKIzv9cg
i9

fwPMLhrKleoVRNsecBsCP569WgJXhUnwf2lon4fEZr3+vRuc9shfqnV0nPN1IM
Sn

zXCast7I2fiuRXdIz96KjlGQpP4XfNVA+RGL7aMnWOFIaVrKWLzAtgzoGMTvP/
Au

ehKXncBJhYtW0ltTioVx+5yTYSAZWl+IssmXjefxJqYi2/7QWmv1QC9psNcjTM
aB

QLN03T1Qelbs7Y27sxdEnNUth4kI+wIDAQABAoICAFWe8MQZb37k2gdAV3Y6aq
8f

qokKQqbCNLd3giGFwYkezHXoJfg6Di7oZxNcKyw35LFEghkgtQqErQqo35VPIo
H+

vXUpWOjnCmM4muFA9/cX6mYMc8TmJsg0ewLdBCOZVw+wPABlaqz+0UOiSMMftp
k9
```

```
fz9JwGd8ERyBsT+tk3Qi6D0vPZVsC1KqxxL/cwIFd3Hf2ZBtJXe0KBn1pktWht
5A

Kqx9mld2Ovl7NjgiC1Fx9r+fZw/iOabFFwQA4dr+R8mEMK/7bd4VXfQ1o/QGGb
MT

G+ulFrsiDyP+rBIAaGC0i7gDjLAIBQeDhP409ZhswIEc/GBtODU372a2CQK/u4
Q/

HBQvuBtKFNkGUooLgCCbFxzgNUGc83GB/6IwbEM7R5uXqsFiE71LpmroDyjKTl
Q8

YZkpIcLNVLw0usoGYHFm2rvCyEVlfsE3Ub8cFyTFk50SeOcF2QL2xzKmmbZEpX
gl

xBHR0hjgon0IKJDGfor4bHO7Nt+1Ece8u2oTEKvpz5aIn44OeC5mApRGy83/0b
vs

esnWjDE/bGpoT8qFuy+0urDEPNId44XcJm1IRIlG56ErxC3l0s11wrIpTmXXck
qw

zFR9s2z7f0zjeyxqZg4NTPI7wkM3M8BXlvp2GTBIeoxrWB4V3YArwu8QF80QBg
Vz

mgHl24nTg00UH1OjZsABAoIBAQDOxftSDbSqGytcWqPYP3SZHAWDA0O4ACEM+e
Cw

au9ASutl0IDlNDMJ8nC2ph25BMe5hHDWp2cGQJog7pZ/3qQogQho2gUniKDifN
77

40QdykllTzTVROqmP8+efreIvqlzHmuqaGfGs5oTkZaWj5su+B+bT+9rIwZcwf
s5

YRINhQRx17qa++xh5mfE25c+M9fiIBTiNSo4lTxWMBShnK8xrGaMEmN7W0qTMb
FH

PgQz5FcxRjCCqwHilwNBeLDTp/ZECEB7y34khVh531mBE2mNzSVIQcGZP1I/Dv
Xj

W7UUNdgFwii/GW+6M0uUDy23UVQpbFzcV8o1C2nZc4Fb4zwBAoIBAQDKSJkFww
uR

naVJS6WxOKjX8MCu9/cKPnwBv2mmI2jgGxHTw5sr3ahmF5eTb8Zo19BowytN+t
r6

2ZFoIBA9Ubc9esEAU8l3fggdfM82cuR9sGcfQVoCh8tMg6BP8IBLOmbSUhN3PG
2m

39I802u0fFNVQCJKhx1m1MFFLOu7lVcDS9JN+oYVPb6MDfBLm5jOiPuYkFZ4gH
```

```
79

J7gXI0/YKhaJ7yXthYVkdrSF6Eooer4RZgma62Dd1VNzSq3JBo6rYjF7Lvd+RwDC

R1thHrmf/IXplxpNVkoMVxtzbrrbgnC25QmvRYc0rlS/kvM4yQhMH3eA7IycDZMp

Y+0xm7I7jTT7AoIBAGKzKIMDXdCxBWKhNYJ8z7hiItNl1IZZMW2TPUiY0rl6yaCh

BVXjM9W0r07QPnHZsUiByqb743adkbTUjmxdJzjaVtxN7ZXwZvOVrY7I7fPWYnCE

fXCr4+IVpZI/ZHZWpGX6CGSgT6EOjCZ5IUufIvEpqVSmtF8MqfXO9o9uIYLokrWQ

x1dBl5UnuTLDqw8bChq7O5y6yfuWaOWvL7nxI8NvSsfj4y635gIa/0dFeBYZEfHI

UlGdNVomwXwYEzgE/c19ruIowX7HU/NgxMWTMZhpazlxgesXybel+YNcfDQ4e3RM

OMz3ZFiaMaJsGGNf4++d9TmMgk4Ns6oDs6Tb9AECggEBAJYzd+SOYo26iBu3nw3L

65uEeh6xou8pXH0Tu4gQrPQTRZZ/nT3iNgOwqu1gRuxcq7TOjt41UdqIKO8vN7/A

aJavCpaKoIMowy/aGCbvAvjNPpU3unU8jdl/t08EXs79S5IKPcgAx87sTTi7KDN5

SYt4tr2uPEe53NTXuSatilG5QCyExIELOuzWAMKzg7CAiIlNS9foWeLyVkBgCQ6S

me/L8ta+mUDy37K6vC34jh9vK9yrwF6X44ItRoOJafCaVfGI+175q/eWcqTX4q+I

G4tKls4sL4mgOJLq+ra50aYMxbcuommctPMXU6CrrYyQpPTHMNVDQy2ttFdsq9iK

TncCggEBAMmt/8yvPflS+xv3kg/ZBvR9JB1In2n3rUCYYD47ReKFqJ03Vmq5C9nY

56s9w7OUO8perBXlJYmKZQhO4293lvxZD2Iq4NcZbVSCMoHAUzhzY3brdgtSIxa2

gGveGAezZ38qKIU26dkz7deECY4vrsRkwhpTW0LGVCpjcQoaKvymAoCmAs8V2oMr
```

```
Ziw1YQ9uOUoWwOqm1wZqmVcOXvPIS2gWAs3fQlWjH9hkcQTMsUaXQDOD0aqkSY
3E

NqOvbCV1/oUpRi3076khCoAXI1bKSn/AvR3KDP14B5toHI/F5OTSEiGhhHesgR
rs
    fBrpEY1IATtPq1taBZZogRqI3rOkkPk=
    -----END PRIVATE KEY-----
```

MetaDefender Cluster Worker

Ignition file

Info

The ignition file is required only for a clean installation.

The following fields are essential for the ignition file:

- `secure.connection_key`
- `secure.private_key`
- `secure.certificate`

To install MetaDefender Cluster (MD Cluster) Worker server, ignition file in YML format is required at the following location:

- Windows: `C:\opswat\md_cluster_worker.yml`
- Linux: `/etc/opswat/md_cluster_worker.yml`

The ignition file includes fields:

Key path	Value type	Accepted values	Required	Description
`secure.connection_key`	String	A string from 4 to 64 character long containing digits from 0 to 9 and characters from a/A to z/Z	**Required**	An arbitrary string that enables clients to connect to the server. Use this value as input when adding a MD Cluster Worker in the UI of the MD Cluster Control Center.
`secure.private_key`	String		**Required**	Content of private key in X509 format.
`secure.certificate`	String		**Required**	Content of certificate in X509 format.
`rest.host`	String		Optional	IP address (V4/V6) or host where the server resides on. Default value is `'*'` **Notes:** value `'*'` allows the service to accept connections from all network interfaces. To bind the service to a specific interface, specify its IP address or domain name. For example, to listen on all IPv4 interfaces, set the host to `0.0.0.0`

Key path	Value type	Accepted values	Required	Description
rest.port	Number	A string from 4 to 64 character long containing digits from 0 to 9 and characters from a/A to z/Z	Optional	The port where the server resides on. Default value is 8893
log.streams[@].log_type	String	- file - syslog	Optional	Type of log device.
log.streams[@].log_level	String	- dump - debug - info - warning - error	Optional	Level of log message.
log.streams[@].log_path	String	If log.streams[@].log_type is "file" then log.streams[@].log_path is the path to a file on file system where logs are written. If log.streams[@].log_type is "syslog" then - log.streams[@].log_path can be [tcp/udp]://host:port where host:port is the host and port to a remote syslog server that supports TCP or UDP protocol. - log.streams[@].log_path can be "local" to write log to local syslog server (Linux only).	Optional	Location where logs are written.

Configuration file

After successfully installing, MD Cluster Worker generates a configuration file with changeable settings at the following location:

- Windows: `C:\Program Files\OPSWAT\MetaDefender Cluster Worker\md_cluster_worker.yml`
- Linux: `/etc/md-cluster-worker/md_cluster_worker.yml`

> **Info**
>
> The service must be restarted to take the new configurations into effect.

Sample

> **Info**
>
> OpenSSL or a similar tool (e.g., ssh-keygen) can create a pair of public and private keys in X509 format.

yaml

```
secure:
  connection_key: 1234abcd
  private_key: |
      -----BEGIN PRIVATE KEY-----

MIIJQwIBADANBgkqhkiG9w0BAQEFAASCCS0wggkpAgEAAoICAQCjYtuWaICCY0
tJ

PubxpIgIL+WWmz/fmK8IQr11Wtee6/IUyUlo5I602mq1qcLhT/kmpoR8Di3DAm
HK

nSWdPWtn1BtXLErLlUiHgZDrZWInmEBjKM1DZf+CvNGZ+EzPgBv5nTekLWcfI5
ZZ

toGuIP1Dl/IkNDw8zFz4cpiMe/BFGemyxdHhLrKHSm8Eo+nT734tItnHKT/m6D
SU

0xlZ13d6ehLRm7/+Nx47M3XMTRH5qKP/7TTE2s0U6+M0tsGI2zpRi+m6jzhNyM
BT

J1u58qAe3ZW5/+YAiuZYAB6n5bhUp4oFuB5wYbcBywVR8ujInpF8buWQUjy5N8
pS

Np7szdYsnLJpvAd0sibrNPjC0FQCNrpNjgJmIK3+mKk4kXX7ZTwefoAzTK4l2p
HN

uC53QVc/EF++GBLAxmvCDq9ZpMIYi7OmzkkAKKC9Ue6Ef217LFQCFIBKIzv9cg
i9

fwPMLhrKleoVRNsecBsCP569WgJXhUnwf2lon4fEZr3+vRuc9shfqnV0nPN1IM
Sn

zXCast7I2fiuRXdIz96KjlGQpP4XfNVA+RGL7aMnWOFIaVrKWLzAtgzoGMTvP/
Au

ehKXncBJhYtW0ltTioVx+5yTYSAZWl+IssmXjefxJqYi2/7QWmv1QC9psNcjTM
aB

QLN03T1Qelbs7Y27sxdEnNUth4kI+wIDAQABAoICAFWe8MQZb37k2gdAV3Y6aq
8f

qokKQqbCNLd3giGFwYkezHXoJfg6Di7oZxNcKyw35LFEghkgtQqErQqo35VPIo
H+

vXUpWOjnCmM4muFA9/cX6mYMc8TmJsg0ewLdBCOZVw+wPABlaqz+0UOiSMMftp
k9

fz9JwGd8ERyBsT+tk3Qi6D0vPZVsC1KqxxL/cwIFd3Hf2ZBtJXe0KBn1pktWht
5A
```

```
Kqx9mld2Ovl7NjgiC1Fx9r+fZw/iOabFFwQA4dr+R8mEMK/7bd4VXfQ1o/QGGb
MT

G+ulFrsiDyP+rBIAaGC0i7gDjLAIBQeDhP409ZhswIEc/GBtODU372a2CQK/u4
Q/

HBQvuBtKFNkGUooLgCCbFxzgNUGc83GB/6IwbEM7R5uXqsFiE71LpmroDyjKTl
Q8

YZkpIcLNVLw0usoGYHFm2rvCyEVlfsE3Ub8cFyTFk50SeOcF2QL2xzKmmbZEpX
gl

xBHR0hjgon0IKJDGfor4bHO7Nt+1Ece8u2oTEKvpz5aIn44OeC5mApRGy83/0b
vs

esnWjDE/bGpoT8qFuy+0urDEPNId44XcJm1IRIlG56ErxC3l0s11wrIpTmXXck
qw

zFR9s2z7f0zjeyxqZg4NTPI7wkM3M8BXlvp2GTBIeoxrWB4V3YArwu8QF80QBg
Vz

mgHl24nTg00UH1OjZsABAoIBAQDOxftSDbSqGytcWqPYP3SZHAWDA0O4ACEM+e
Cw

au9ASutl0IDlNDMJ8nC2ph25BMe5hHDWp2cGQJog7pZ/3qQogQho2gUniKDifN
77

40QdyklITzTVROqmP8+efreIvqlzHmuqaGfGs5oTkZaWj5su+B+bT+9rIwZcwf
s5

YRINhQRx17qa++xh5mfE25c+M9fiIBTiNSo4lTxWMBShnK8xrGaMEmN7W0qTMb
FH

PgQz5FcxRjCCqwHilwNBeLDTp/ZECEB7y34khVh531mBE2mNzSVIQcGZP1I/Dv
Xj

W7UUNdgFwii/GW+6M0uUDy23UVQpbFzcV8o1C2nZc4Fb4zwBAoIBAQDKSJkFww
uR

naVJS6WxOKjX8MCu9/cKPnwBv2mmI2jgGxHTw5sr3ahmF5eTb8Zo19BowytN+t
r6

2ZFoIBA9Ubc9esEAU8l3fggdfM82cuR9sGcfQVoCh8tMg6BP8IBLOmbSUhN3PG
2m

39I802u0fFNVQCJKhx1m1MFFLOu7lVcDS9JN+oYVPb6MDfBLm5jOiPuYkFZ4gH
79

J7gXI0/YKhaJ7yXthYVkdrSF6Eooer4RZgma62Dd1VNzSq3JBo6rYjF7Lvd+Rw
```

```
DC

R1thHrmf/IXplxpNVkoMVxtzbrrbgnC25QmvRYc0rlS/kvM4yQhMH3eA7IycDZ
Mp

Y+0xm7I7jTT7AoIBAGKzKIMDXdCxBWKhNYJ8z7hiItNl1IZZMW2TPUiY0rl6ya
Ch

BVXjM9W0r07QPnHZsUiByqb743adkbTUjmxdJzjaVtxN7ZXwZvOVrY7I7fPWYn
CE

fXCr4+IVpZI/ZHZWpGX6CGSgT6EOjCZ5IUufIvEpqVSmtF8MqfXO9o9uIYLokr
WQ

x1dBl5UnuTLDqw8bChq7O5y6yfuWaOWvL7nxI8NvSsfj4y635gIa/0dFeBYZEf
HI

UlGdNVomwXwYEzgE/c19ruIowX7HU/NgxMWTMZhpazlxgesXybel+YNcfDQ4e3
RM

OMz3ZFiaMaJsGGNf4++d9TmMgk4Ns6oDs6Tb9AECggEBAJYzd+SOYo26iBu3nw
3L

65uEeh6xou8pXH0Tu4gQrPQTRZZ/nT3iNgOwqu1gRuxcq7TOjt41UdqIKO8vN7
/A

aJavCpaKoIMowy/aGCbvAvjNPpU3unU8jdl/t08EXs79S5IKPcgAx87sTTi7KD
N5

SYt4tr2uPEe53NTXuSatilG5QCyExIELOuzWAMKzg7CAiIlNS9foWeLyVkBgCQ
6S

me/L8ta+mUDy37K6vC34jh9vK9yrwF6X44ItRoOJafCaVfGI+175q/eWcqTX4q
+I

G4tKls4sL4mgOJLq+ra50aYMxbcuommctPMXU6CrrYyQpPTHMNVDQy2ttFdsq9
iK

TncCggEBAMmt/8yvPflS+xv3kg/ZBvR9JB1In2n3rUCYYD47ReKFqJ03Vmq5C9
nY

56s9w7OUO8perBXlJYmKZQhO4293lvxZD2Iq4NcZbVSCMoHAUzhzY3brdgtSIx
a2

gGveGAezZ38qKIU26dkz7deECY4vrsRkwhpTW0LGVCpjcQoaKvymAoCmAs8V2o
Mr

Ziw1YQ9uOUoWwOqm1wZqmVcOXvPIS2gWAs3fQlWjH9hkcQTMsUaXQDOD0aqkSY
3E
```

```
NqOvbCV1/oUpRi3076khCoAXI1bKSn/AvR3KDP14B5toHI/F5OTSEiGhhHesgR
rs
      fBrpEY1IATtPq1taBZZogRqI3rOkkPk=
      -----END PRIVATE KEY-----
  certificate: |
      -----BEGIN CERTIFICATE-----

MIIF5jCCA86gAwIBAgIJANq50IuwPFKgMA0GCSqGSIb3DQEBCwUAMIGGMQswCQ
YD

VQQGEwJHQjEQMA4GA1UECAwHRXJld2hvbjETMBEGA1UEBwwKQWxsIGFyb3VuZD
Eb

MBkGA1UECgwSbGlid2Vic29ja2V0cy10ZXN0MRIwEAYDVQQDDAlsb2NhbGhvc3
Qx

HzAdBgkqhkiG9w0BCQEWEG5vbmVAaW52YWxpZC5vcmcwIBcNMTgwMzIwMDQxNj
A3

WhgPMjExODAyMjQwNDE2MDdaMIGGMQswCQYDVQQGEwJHQjEQMA4GA1UECAwHRX
Jl

d2hvbjETMBEGA1UEBwwKQWxsIGFyb3VuZDEbMBkGA1UECgwSbGlid2Vic29ja2
V0

cy10ZXN0MRIwEAYDVQQDDAlsb2NhbGhvc3QxHzAdBgkqhkiG9w0BCQEWEG5vbm
VA

aW52YWxpZC5vcmcwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCjYt
uW

aICCY0tJPubxpIgIL+WWmz/fmK8IQr11Wtee6/IUyUlo5I602mq1qcLhT/kmpo
R8

Di3DAmHKnSWdPWtn1BtXLErLlUiHgZDrZWInmEBjKM1DZf+CvNGZ+EzPgBv5nT
ek

LWcfI5ZZtoGuIP1Dl/IkNDw8zFz4cpiMe/BFGemyxdHhLrKHSm8Eo+nT734tIt
nH

KT/m6DSU0xlZ13d6ehLRm7/+Nx47M3XMTRH5qKP/7TTE2s0U6+M0tsGI2zpRi+
m6

jzhNyMBTJ1u58qAe3ZW5/+YAiuZYAB6n5bhUp4oFuB5wYbcBywVR8ujInpF8bu
WQ

Ujy5N8pSNp7szdYsnLJpvAd0sibrNPjC0FQCNrpNjgJmIK3+mKk4kXX7ZTwefo
Az

TK4l2pHNuC53QVc/EF++GBLAxmvCDq9ZpMIYi7OmzkkAKKC9Ue6Ef217LFQCFI
```

```
BK

Izv9cgi9fwPMLhrKleoVRNsecBsCP569WgJXhUnwf2lon4fEZr3+vRuc9shfqn
V0

nPN1IMSnzXCast7I2fiuRXdIz96KjlGQpP4XfNVA+RGL7aMnWOFIaVrKWLzAtg
zo

GMTvP/AuehKXncBJhYtW0ltTioVx+5yTYSAZWl+IssmXjefxJqYi2/7QWmv1QC
9p

sNcjTMaBQLN03T1Qelbs7Y27sxdEnNUth4kI+wIDAQABo1MwUTAdBgNVHQ4EFg
QU

9mYU23tW2zsomkKTAXarjr2vjuswHwYDVR0jBBgwFoAU9mYU23tW2zsomkKTAX
ar

jr2vjuswDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAgEANjIBMr
ow

YNCbhAJdP7dhlhT2RUFRdeRUJD0IxrH/hkvb6myHHnK8nOYezFPjUlmRKUgNED
uA

xbnXZzPdCRNV9V2mShbXvCyiDY7WCQE2Bn44z26O0uWVk+7DNNLH9BnkwUtOnM
9P

wtmD9phWexm4q2GnTsiL6Ul6cy0QlTJWKVLEUQQ6yda582e23J1AXqtqFcpfoE
34

H3afEiGy882b+ZBiwkeV+oq6XVF8sFyr9zYrv9CvWTYlkpTQfLTZSsgPdEHYVc
jv

xQ2D+XyDR0aRLRlvxUa9dHGFHLICG34Juq5Ai6lM1EsoD8HSsJpMcmrH7MWw2c
Kk

ujC3rMdFTtte83wF1uuF4FjUC72+SmcQN7A386BC/nk2TTsJawTDzqwOu/VdZv
2g

1WpTHlumlClZeP+G/jkSyDwqNnTu1aodDmUa4xZodfhP1HWPwUKFcq8oQr148Q
YA

AOlbUOJQU7QwRWd1VbnwhDtQWXC92A2w1n/xkZSR1BM/NUSDhkBSUU1WjMbWg6
Gg

mnIZLRerQCu1Oozr87rOQqQakPkyt8BUSNK3K42j2qcfhAONdRl8Hq8Qs5pupy
+s

8sdCGDlwR3JNCMv6u48OK87F4mcIxhkSefFJUFII25pCGN5WtE4p5l+9cnO1Gr
IX
```

```
e2Hl/7M0c/lbZ4FvXgARlex2rkgS0Ka06HE=
-----END CERTIFICATE-----
```

MetaDefender Cluster Control Center

Ignition file

ⓘ **Info**

The ignition file is required only for a clean installation.

The following fields are essential for the ignition file:

- `identity.host`
- `identity.port`
- `identity.connection_key`
- `database.host`
- `database.port`
- `database.user`
- `database.password`
- `secure.encryption_key`

To install MetaDefender Cluster Control (MD Cluster) Center server, ignition file in YML format is required at the following location:

- Windows: `C:\opswat\md_cluster_control_center.yml`
- Linux: `/etc/opswat/md_cluster_control_center.yml`

The ignition file includes fields:

Key path	Value type	Accepted values	Required	Description
`identity.host`	String		**Required**	IP address of the server where MD Cluster Identity Service server locates.
`identity.port`	String		**Required**	Port of MD Cluster Identity Service server is listening for connections from clients.
`identity.connection_key`	String	A string from 4 to 64 character long containing digits from 0 to 9 and characters from a/A to z/Z	**Required**	The access key required to connect to the MD Cluster Identity Service server, ensuring it matches the value used by the server.
`database.host`	String		**Required**	IP address / domain name of the server where PostgreSQL server locates.
`database.port`	Number		**Required**	Port of PostgreSQL server is listening for connections from clients.
`database.user`	String		**Required**	PostgreSQL server's user. SUPERUSER privilege is required to setup the server's database and extensions for the first time.
`database.password`	String		**Required**	PostgreSQL server's user credentials.
`secure.encryption_key`	String	A 32-character plain text composed of characters 'a'-'z' and digits '0'-'9'.	**Required**	The encryption key is used to encrypt the sensitive data in the database.
`rest.port`	Number		Optional	The port where the server resides on. Default value is `8892`
`rest.log_path`	String		Optional	Location where logs are written.

Key path	Value type	Accepted values	Required	Description
`rest.log_level`	String	- `dump` - `debug` - `info` - `warning` - `error`	Optional	Level of log message.
`log.streams[@].log_type`	String	- `file` - `syslog`	Optional	Type of log device.
`log.streams[@].log_level`	String	- `dump` - `debug` - `info` - `warning` - `error`	Optional	Level of log message.

Key path	Value type	Accepted values	Required	Description
`log.streams[@].log_path`	String	If `log.streams[@].log_type` is `"file"` then `log.streams[@].log_path` is the path to a file on file system where logs are written. If `log.streams[@].log_type` is `"syslog"` then - `log.streams[@].log_path` can be `[tcp/udp]://host:port` where host:port is the host and port to a remote syslog server that supports TCP or UDP protocol. - `log.streams[@].log_path` can be `"local"` to write log to local syslog server (Linux only).	Optional	Location where logs are written.

Warning

Avoid using the loopback IP address (such as `localhost` or `127.0.0.1`) for key `identity.host` .

It may prevent MD Cluster **API Gateway** from successfully establishing a connection to MD Cluster **Identity Service**.

Configuration file

After successfully installing, MD Cluster Control Center generates a configuration file with changeable settings at the following location:

- Windows: `C:\Program Files\OPSWAT\MetaDefender Cluster Control Center\md_cluster_control_center.yml`
- Linux: `/etc/md-cluster-control-center/md_cluster_control_center.yml`

Info

The service must be restarted to take the new configurations into effect.

Sample

Warning

`database.host`, `database.port`, `database.user`, and `database.password` should be updated with the appropriate values of your Postgres host/IP, port, username, and password.

`identity.host` should be updated with the appropriate host or IP of your MD Cluster Identity Service.

yaml

```yaml
database:
  host: "your_postgres_host_ip"
  port: 5432
  user: "your_postgres_username"
  password: "your_postgres_admin_password"
identity:
  host: "your_md_cluster_identity_service_host_ip"
  port: 8891
  connection_key: "1234abcd"
secure:
  encryption_key: "12345678123456781234567812345678" # [a-z0-9]{32}
```

Container-Based Setup

ℹ **Prerequisite**

Before running the setup, please check [System Requirements] to install all required dependencies of MetaDefender Cluster (MD Cluster).

Setup order requirement

Please follow the installation order to complete the system setup properly.

Order	Service	Notes
1	Redis, RabbitMQ, PostgreSQL and MD Cluster Identity Service	- Could be setup in parallel in any order among them. - Make sure they are all fully functional and accessible before proceeding to the next setup order #2.
2	MD Cluster Control Center	- Ensure it's able to connect to those services in #1 - Make sure it is fully functional and accessible.
3	MD Cluster File Storage	- Ensure they're able to connect to MD Cluster Control Center - Make sure it is fully functional and accessible.
4	MD Cluster Worker for MD Cluster API Gateway and MD Cluster Worker for MetaDefender Core	- Could be setup in parallel in any order among them. - Ensure they're able to connect to MD Cluster Control Center - Make sure they are all fully functional and accessible.

Image name and version

All the images can be found at OPSWAT Docker Hub with the following information:

Info

version is the currently release version.

MD Cluster Identity Service

Docker image bash

```
opswat/metadefender-cluster:identity-service-<version>-debian-
12
```

MD Cluster File Storage

Docker image bash

```
opswat/metadefender-cluster:file-storage-<version>-debian-12
```

MD Cluster Control Center

Docker image bash

```
opswat/metadefender-cluster:control-center-<version>-debian-12
```

MD Cluster Worker for MD Cluster API Gateway

Docker image bash

```
opswat/metadefender-cluster:worker-api-gateway-<version>-
debian-12
```

MD Cluster Worker for MetaDefender Core

Docker image bash

```
opswat/metadefender-cluster:worker-core-<version>-debian-12
```

Environment variables

1. MD Cluster Identity Service

Environment Variable	Necessity	Description
`MDCLS_IDENTITY_SERVICE_DB_HOST`	Required	Provide the database host for MD Cluster Identity Service
`MDCLS_IDENTITY_SERVICE_DB_PORT`	Optional	Provide the database port for MD Cluster Identity Service Default: 5432
`MDCLS_IDENTITY_SERVICE_DB_USER`	Required	Provide the database user for MD Cluster Identity Service
`MDCLS_IDENTITY_SERVICE_DB_PASSWORD`	Required	Provide the database password for MD Cluster Identity Service

Environment Variable	Necessity	Description
`MDCLS_USER`	Required	Define the information to initiate the administrator account. This account is to automatically do the following tasks: • Add Redis to MD Cluster Control Center if specified. • Add RabbitMQ to MD Cluster Control Center if specified. • Add Data Lake to MD Cluster Control Center if specified. • Add Data Warehouse to MD Cluster Control Center if specified. • Add MD Cluster File Storage to MD Cluster Control Center if specified. • Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. • Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.

Environment Variable	Necessity	Description
`MDCLS_PASSWORD`	Required	Define the information to initiate the administrator account. This account is to automatically do the following tasks: • Add Redis to MD Cluster Control Center if specified. • Add RabbitMQ to MD Cluster Control Center if specified. • Add Data Lake to MD Cluster Control Center if specified. • Add Data Warehouse to MD Cluster Control Center if specified. • Add MD Cluster File Storage to MD Cluster Control Center if specified. • Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. • Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.

Environment Variable	Necessity	Description
`MDCLS_EMAIL`	Required	Define the information to initiate the administrator account. This account is to automatically do the following tasks: • Add Redis to MD Cluster Control Center if specified. • Add RabbitMQ to MD Cluster Control Center if specified. • Add Data Lake to MD Cluster Control Center if specified. • Add Data Warehouse to MD Cluster Control Center if specified. • Add MD Cluster File Storage to MD Cluster Control Center if specified. • Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. • Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.

Environment Variable	Necessity	Description
`MDCLS_APIKEY`	Optional	Define the information to initiate the administrator account. This account is to automatically do the following tasks: - Add Redis to MD Cluster Control Center if specified. - Add RabbitMQ to MD Cluster Control Center if specified. - Add Data Lake to MD Cluster Control Center if specified. - Add Data Warehouse to MD Cluster Control Center if specified. - Add MD Cluster File Storage to MD Cluster Control Center if specified. - Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. - Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.

Environment Variable	Necessity	Description
MDCLS_IDENTITY_SERVICE_CONNECTION_KEY	Required	Define the connection key in order to register to Control Center. Must be 4 to 64 characters long, using only letters and digits (0–9, a–z, A–Z).
MDCLS_IDENTITY_SERVICE_PORT	Optional	Define the expose port for MD Cluster Identity Service Default: 8891
LOG_LEVEL	Optional	Define the log level. Default value: info Accepted values: info / debug / error / warning

Start MD Cluster Identity Service container with docker run:

bash

```
docker run -d --name md-cluster-identity-service \
        -e MDCLS_IDENTITY_SERVICE_DB_HOST=<your_postgres_host>
\
  -e MDCLS_IDENTITY_SERVICE_DB_USER=<your_postgres_user> \
  -e MDCLS_IDENTITY_SERVICE_DB_PASSWORD=
<your_postgres_password> \
  -e MDCLS_IDENTITY_SERVICE_CONNECTION_KEY=
<your_connection_key> \
  -e MDCLS_USER=<your_admin_user> \
  -e MDCLS_PASSWORD=<your_admin_password> \
  -e MDCLS_EMAIL=<your_admin_email> \
  -p 8891:8891 opswat/metadefender-cluster:identity-service-
<version>-debian-12
```

2. MD Cluster File Storage

Environment Variable	Necessity	Description
`MDCLS_FILE_STORAGE_CONNECTION_KEY`	Required	Define the connection key in order to register to MD Cluster Control Center. Must be 4 to 64 characters long, using only letters and digits [0-9, a-z, A-Z].
`MDCLS_FILE_STORAGE_PORT`	Optional	Define the expose port for MD Cluster File Storage. Default is 8890.
`MDCLS_FILE_STORAGE_HOST`	Optional	Define the MD Cluster File Storage's host address. If it's not specified, it will get the container's internal IP address.
`LOG_LEVEL`	Optional	Define the log level. Default value: `info`. Accepted values: `info` / `debug` / `error` / `warning`.
`MDCLS_CONTROL_CENTER_HOST`	Required	Provide the MD Cluster Control Center's host address.
`MDCLS_CONTROL_CENTER_PORT`	Optional	Provide the MD Cluster Control Center's port. Default is 8892.

Environment Variable	Necessity	Description
`MDCLS_USER`	Required	Define the information to initiate the administrator account. This account is to automatically do the following tasks: - Add Redis to MD Cluster Control Center if specified. - Add RabbitMQ to MD Cluster Control Center if specified. - Add Data Lake to MD Cluster Control Center if specified. - Add Data Warehouse to MD Cluster Control Center if specified. - Add MD Cluster File Storage to MD Cluster Control Center if specified. - Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. - Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.

Environment Variable	Necessity	Description
`MDCLS_PASSWORD`	Required	Define the information to initiate the administrator account. This account is to automatically do the following tasks: - Add Redis to MD Cluster Control Center if specified. - Add RabbitMQ to MD Cluster Control Center if specified. - Add Data Lake to MD Cluster Control Center if specified. - Add Data Warehouse to MD Cluster Control Center if specified. - Add MD Cluster File Storage to MD Cluster Control Center if specified. - Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. - Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.

Environment Variable	Necessity	Description
`MDCLS_APIKEY`	Optional	Define the information to initiate the administrator account. This account is to automatically do the following tasks: - Add Redis to MD Cluster Control Center if specified. - Add RabbitMQ to MD Cluster Control Center if specified. - Add Data Lake to MD Cluster Control Center if specified. - Add Data Warehouse to MD Cluster Control Center if specified. - Add MD Cluster File Storage to MD Cluster Control Center if specified. - Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. - Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.

Info

Persistent storage is located at `/opt/opswat/md-cluster-file-storage`. If end-users require data to be retained across container lifecycles, they must mount a volume to this path with `777` permissions to ensure full read/write access for all processes.

Start MD Cluster File Storage container with docker run.

bash

```bash
docker run -d --name md-cluster-file-storage \
        -e MDCLS_FILE_STORAGE_CONNECTION_KEY=
<your_connection_key> \
  -e MDCLS_FILE_STORAGE_PORT=8890 \
  -e MDCLS_CONTROL_CENTER_HOST=<control-center_host_address> \
  -e MDCLS_USER=<your_admin_user> \
  -e MDCLS_PASSWORD=<your_admin_password> \
  -p 8890:8890 opswat/metadefender-cluster:file-storage-
<version>-debian-12
```

3. MD Cluster Control Center

Environment Variable	Necessity	Description
`MDCLS_LAKE_DB_HOST`	Optional	Provide the database host for Data Lake. In case that the end-user does not have the Data Lake, it's required to provide this variable to automate the database preparation.
`MDCLS_LAKE_DB_PORT`	Optional	Provide the database port for Data Lake. Default is 5432.
`MDCLS_LAKE_DB_USER`	Optional	Provide the database user for Data Lake. In case that the end-user does not have the Data Lake, it's required to provide this variable to automate the database preparation.
`MDCLS_LAKE_DB_PASSWORD`	Optional	Provide the database password for Data Lake. In case that the end-user does not have the Data Lake, it's required to provide this variable to automate the database preparation.
`MDCLS_WAREHOUSE_DB_HOST`	Optional	Provide the database host for Data Warehouse. In case that the end-user does not have the Data Warehouse, it's required to provide this variable to automate the database preparation.
`MDCLS_WAREHOUSE_DB_PORT`	Optional	Provide the database port for Data Warehouse. Default is 5432.
`MDCLS_WAREHOUSE_DB_USER`	Optional	Provide the database user for Data Warehouse. In case that the end-user does not have the Data Warehouse, it's required to provide this variable to automate the database preparation.

Environment Variable	Necessity	Description
`MDCLS_WAREHOUSE_DB_PASSWORD`	Optional	Provide the database password for Data Warehouse. In case that the end-user does not have the Data Warehouse, it's required to provide this variable to automate the database preparation.
`MDCLS_CACHE_HOST`	Optional	Provide the caching host (Redis).
`MDCLS_CACHE_PORT`	Optional	Provide the caching port (Redis).
`MDCLS_CACHE_USER`	Optional	Provide the caching username (Redis). If the end-user does not provide it, Redis will be added without authentication.
`MDCLS_CACHE_PASSWORD`	Optional	Provide the caching password (Redis). If the end-user does not provide it, Redis will be added without authentication. Do not support double quotes (`"`) and backslash (`\`) in the password.
`MDCLS_BROKER_HOST`	Optional	Provide the broker host (RabbitMQ).
`MDCLS_BROKER_PORT`	Optional	Provide the broker port (RabbitMQ).
`MDCLS_BROKER_USER`	Optional	Provide the broker username (RabbitMQ).
`MDCLS_BROKER_PASSWORD`	Optional	Provide the broker password (RabbitMQ).
`MDCLS_CONTROL_CENTER_DB_HOST`	Required	Provide the database host for MD Cluster Control Center.
`MDCLS_CONTROL_CENTER_DB_PORT`	Optional	Provide the database port for MD Cluster Control Center. Default is 5432.

Environment Variable	Necessity	Description
`MDCLS_CONTROL_CENTER_DB_USER`	Required	Provide the database username for MD Cluster Control Center.
`MDCLS_CONTROL_CENTER_DB_PASSWORD`	Required	Provide the database password for MD Cluster Control Center.

Environment Variable	Necessity	Description
`MDCLS_USER`	Required	Provide the administrator account that is defined in MD Cluster Identity Service. This account is to automatically do the following tasks: • Add Redis to MD Cluster Control Center if specified. • Add RabbitMQ to MD Cluster Control Center if specified. • Add Data Lake to MD Cluster Control Center if specified. • Add Data Warehouse to MD Cluster Control Center if specified. • Add MD Cluster File Storage to MD Cluster Control Center if specified. • Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. • Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.

Environment Variable	Necessity	Description
`MDCLS_PASSWORD`	Required	Provide the administrator account that is defined in MD Cluster Identity Service. This account is to automatically do the following tasks: • Add Redis to MD Cluster Control Center if specified. • Add RabbitMQ to MD Cluster Control Center if specified. • Add Data Lake to MD Cluster Control Center if specified. • Add Data Warehouse to MD Cluster Control Center if specified. • Add MD Cluster File Storage to MD Cluster Control Center if specified. • Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. • Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.

Environment Variable	Necessity	Description
`MDCLS_APIKEY`	Optional	Provide the administrator account that is defined in MD Cluster Identity Service. This account is to automatically do the following tasks: • Add Redis to MD Cluster Control Center if specified. • Add RabbitMQ to MD Cluster Control Center if specified. • Add Data Lake to MD Cluster Control Center if specified. • Add Data Warehouse to MD Cluster Control Center if specified. • Add MD Cluster File Storage to MD Cluster Control Center if specified. • Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. • Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.
`MDCLS_IDENTITY_SERVICE_HOST`	Required	Provide the MD Cluster Identity Service host in order to add it to MD Cluster Control Center.

Environment Variable	Necessity	Description
`MDCLS_IDENTITY_SERVICE_PORT`	Optional	Provide the IMD Cluster dentity Service port in order to add it to MD Cluster Control Center. Default is 8891.
`MDCLS_IDENTITY_SERVICE_CONNECTION_KEY`	Required	Provide the MD Cluster Identity Service connection key in order to add it to MD Cluster Control Center. Must be 4 to 64 characters long, using only letters and digits (0–9, a–z, A–Z).
`MDCLS_CONTROL_CENTER_ENCRYPTION_KEY`	Required	Define the encryption key for communication between MD Cluster Control Center and the services. Must be 32 characters long and contain only lowercase letters (a–z) and digits (0–9).
`MDCLS_CERT_PATH`	Optional	Provide the directory path that contains the certificate and private key in order to enable https Note: when provide this variable, it's supposed to mount this path to `/certs/` as volume For example: `--volume /your-path:/certs` Note: In cases where SSL fails to enable due to the File Storage service not being ready, the end-user can either restart the MD Cluster Control Center or manually activate SSL as a workaround.
`MDCLS_OLMS_HOST`	Required	Provide the OLMS host URL, including the `http://` or `https://` prefix.
`MDCLS_OLMS_PORT`	Required	Specify the OLMS REST API port
`MDCLS_OLMS_TOKEN`	Required	Provide the authentication token for accessing OLMS.

Environment Variable	Necessity	Description
`MDCLS_OLMS_RULE`	Required	Enter the rule configuration.
`MDCLS_OLMS_SOCKET_PORT`	Required	Specify the socket port.
`MDCLS_OLMS_DESCRIPTION`	Optional	Provide a brief description.
`LOG_LEVEL`	Optional	Define the log level. Default value: `info`. Accepted values: `info` / `debug` / `error` / `warning` .

Start MD Cluster Control Center container with Docker run.

bash

```bash
docker run -d --name md-cluster-control-center \
        -e MDCLS_CONTROL_CENTER_DB_HOST=<your_postgre_host> \
  -e MDCLS_CONTROL_CENTER_DB_USER=<your_postgre_user> \
  -e MDCLS_CONTROL_CENTER_DB_PASSWORD=<your_postgre_password>
\
  -e MDCLS_IDENTITY_SERVICE_HOST=
<your_identity_service_host_address> \
  -e MDCLS_USER=<your_admin_user> \
  -e MDCLS_PASSWORD=<your_admin_password> \
  -e MDCLS_IDENTITY_SERVICE_CONNECTION_KEY=
<your_connection_key> \
  -e MDCLS_CONTROL_CENTER_ENCRYPTION_KEY=<your_encryption_key>
\
  -e MDCLS_CERT_PATH=/certs \
  -v /new-certificates:/certs \
  -p 8892:8892 opswat/metadefender-distributed-
cluster:control-center-<version>-debian-12
```

4. MD Cluster Worker for API Gateway

Environment Variable	Necessity	Description
`MDCLS_WORKER_CONNECTION_KEY`	Required	Define the connection key in order to register to MD Cluster Control Center. Must be 4 to 64 characters long, using only letters and digits (0–9, a–z, A–Z).
`MDCLS_WORKER_PORT`	Optional	Define the expose worker's port. Default is 8893.
`MDCLS_WORKER_HOST`	Optional	Define the worker's host address. If it's not specified, it will get the container's internal IP address.
`MDCLS_CONTROL_CENTER_HOST`	Required	Provide the MD Cluster Control Center's host address.
`MDCLS_CONTROL_CENTER_PORT`	Optional	Provide the MD Cluster Control Center's port Default is 8892.

Environment Variable	Necessity	Description
`MDCLS_USER`	Required	Provide the administrator account that is defined in MD Cluster Identity Service. It can be optional if the end-user provides the MDDC_APIKEY. This account is to automatically do the following tasks: • Add Redis to MD Cluster Control Center if specified. • Add RabbitMQ to MD Cluster Control Center if specified. • Add Data Lake to MD Cluster Control Center if specified. • Add Data Warehouse to MD Cluster Control Center if specified. • Add MD Cluster File Storage to MD Cluster Control Center if specified. • Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. • Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.

Environment Variable	Necessity	Description
`MDCLS_PASSWORD`	Required	Provide the administrator account that is defined in MD Cluster Identity Service. It can be optional if the end-user provides the MDDC_APIKEY. This account is to automatically do the following tasks: • Add Redis to MD Cluster Control Center if specified. • Add RabbitMQ to MD Cluster Control Center if specified. • Add Data Lake to MD Cluster Control Center if specified. • Add Data Warehouse to MD Cluster Control Center if specified. • Add MD Cluster File Storage to MD Cluster Control Center if specified. • Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. • Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.

Environment Variable	Necessity	Description
`MDCLS_APIKEY`	Optional	Provide the administrator account that is defined in MD Cluster Identity Service. This account is to automatically do the following tasks: • Add Redis to MD Cluster Control Center if specified. • Add RabbitMQ to MD Cluster Control Center if specified. • Add Data Lake to MD Cluster Control Center if specified. • Add Data Warehouse to MD Cluster Control Center if specified. • Add MD Cluster File Storage to MD Cluster Control Center if specified. • Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. • Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.
`MDCLS_API_GATEWAY_PORT`	Optional	Define the expose port to scan files via MD Cluster API Gateway. Default is 8899.
`LOG_LEVEL`	Optional	Define the log level. Default value: `info`. Accepted values: `info` / `debug` / `error` / `warning`.

Info

If multiple MD Cluster API Gateway containers are deployed on the same host, make sure their ports are configured to avoid conflicts.

Start MD Cluster Worker for MD Cluster API Gateway container with Docker run.

bash

```
docker run -d --name md-cluster-worker-api-gateway \
        -e MDCLS_WORKER_CONNECTION_KEY=<your_connection_key> \
  -e MDCLS_WORKER_HOST=<your_worker_host_address> \
  -e MDCLS_CONTROL_CENTER_HOST=
<your_control_center_host_address> \
  -e MDCLS_USER=<your_admin_user> \
  -e MDCLS_PASSWORD=<your_admin_password> \
  -e MDCLS_API_GATEWAY_PORT=8899 \
  -p 8893:8893 -p 8899:8899 opswat/metadefender-
cluster:worker-api-gateway-<version>-debian-12
```

5. MD Cluster Worker for Core

Environment Variable	Necessity	Description
`MDCLS_WORKER_CONNECTION_KEY`	Required	Define the connection key in order to register to MD Cluster Control Center. Must be 4 to 64 characters long, using only letters and digits (0–9, a–z, A–Z).
`MDCLS_WORKER_PORT`	Optional	Define the expose worker's port. Default is 8893.
`MDCLS_WORKER_HOST`	Optional	Define the worker's host address. If it's not specified, it will get the container's internal IP address.
`MDCLS_CONTROL_CENTER_HOST`	Required	Provide the MD Cluster Control Center's host address.
`MDCLS_CONTROL_CENTER_PORT`	Optional	Provide the MD Cluster Control Center's port. Default is 8892.

Environment Variable	Necessity	Description
`MDCLS_USER`	Required	Provide the administrator account that is defined in MD Cluster Identity Service. It can be optional if the end-user provides the MDDC_APIKEY. This account is to automatically do the following tasks: • Add Redis to MD Cluster Control Center if specified. • Add RabbitMQ to MD Cluster Control Center if specified. • Add Data Lake to MD Cluster Control Center if specified. • Add Data Warehouse to MD Cluster Control Center if specified. • Add MD Cluster File Storage to MD Cluster Control Center if specified. • Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. • Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.

Environment Variable	Necessity	Description
`MDCLS_PASSWORD`	Required	Provide the administrator account that is defined in MD Cluster Identity Service. It can be optional if the end-user provides the MDDC_APIKEY. This account is to automatically do the following tasks: • Add Redis to MD Cluster Control Center if specified. • Add RabbitMQ to MD Cluster Control Center if specified. • Add Data Lake to MD Cluster Control Center if specified. • Add Data Warehouse to MD Cluster Control Center if specified. • Add MD Cluster File Storage to MD Cluster Control Center if specified. • Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. • Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.

Environment Variable	Necessity	Description
`MDCLS_APIKEY`	Optional	Provide the administrator account that is defined in MD Cluster Identity Service. This account is to automatically do the following tasks: - Add Redis to MD Cluster Control Center if specified. - Add RabbitMQ to MD Cluster Control Center if specified. - Add Data Lake to MD Cluster Control Center if specified. - Add Data Warehouse to MD Cluster Control Center if specified. - Add MD Cluster File Storage to MD Cluster Control Center if specified. - Add MD Cluster Worker to MD Cluster Control Center, upload MD Cluster API Gateway installer to MD Cluster Control Center, and deploy MD Cluster API Gateway to MD Cluster Worker. - Add MD Cluster Worker to MD Cluster Control Center, upload MetaDefender Core installer to MD Cluster Control Center, and deploy MetaDefender Core to MD Cluster Worker.
`LOG_LEVEL`	Optional	Define the log level. Default value: `info`. Accepted values: `info` / `debug` / `error` / `warning`.
`MDCLS_LICENSE_KEY`	Optional	Provide the license key to activate MetaDefender Core.
`MDCLS_LICENSE_DESCRIPTION`	Optional	Define a description of the license key.

Info

If multiple MetaDefender Core containers are deployed on the same host, make sure their ports and hosts are configured to avoid conflicts.

Start MD Cluster Worker for MetaDefender Core container with Docker run.

bash

```
docker run -d --name md-cluster-worker-core \
        -e MDCLS_WORKER_CONNECTION_KEY=<your_connection_key> \
  -e MDCLS_WORKER_HOST=<your_core_host_address> \
  -e MDCLS_CONTROL_CENTER_HOST=<your_control_center_host_address> \
  -e MDCLS_USER=<your_admin_user> \
  -e MDCLS_PASSWORD=>your_admin_password> \
  -p 8893:8893 opswat/metadefender-cluster:worker-core-<version>-debian-12
```

Start MetaDefender Cluster with Docker Compose

1. Create a local file named `docker-compose.yaml` and copy the following content to this file:

yaml yaml

```
services:
  redis:
    image: redis:7.0.5
    container_name: redis
    ports:
      - "6379:6379"
    networks:
      - mddc
  rabbitmq:
    image: rabbitmq:3.13.0
    container_name: rabbitmq
    restart: always
    healthcheck:
      test: ["CMD", "rabbitmq-diagnostics", "-q","ping"]
      interval: 10s
      timeout: 10s
      retries: 30
      start_period: 20s
    environment:
      - RABBITMQ_DEFAULT_USER=admin
      - RABBITMQ_DEFAULT_PASS=admin
    ports:
      - "5672:5672"
      - "15672:15672"
    networks:
      - mddc
  postgres:
    image: postgres:14.17
    container_name: postgres
    ports:
      - 5432:5432
    networks:
      - mddc
    environment:
      - POSTGRES_USER=admin
      - POSTGRES_PASSWORD=admin
    healthcheck:
      test: [ "CMD", "pg_isready", "-U", "admin", "-d",
"postgres" ]
      interval: 10s
      timeout: 10s
      retries: 30
  identity-service:
    env_file:
    - .env.example
    image: opswat/metadefender-cluster:identity-service-2.0.0-
debian-12
    container_name: identity-service
    ports:
      - 8891:8891
    networks:
      - mddc
    deploy:
      restart_policy:
        condition: on-failure
    depends_on:
      postgres:
        condition: service_healthy
        restart: true
  file-storage:
    env_file:
    - .env.example
    image: opswat/metadefender-cluster:file-storage-2.0.0-debian-12
    container_name: file-storage
    ports:
      - 8890:8890
    networks:
      - mddc
    deploy:
      restart_policy:
        condition: on-failure
    depends_on:
      postgres:
        condition: service_healthy
        restart: true
  control-center:
    env_file:
    - .env.example
    image: opswat/metadefender-cluster:control-center-2.0.0-debian-12
    container_name: control-center
    ports:
      - 8892:8892
    networks:
      - mddc
    deploy:
      restart_policy:
        condition: on-failure
    healthcheck:
      test: ["CMD", "true"]
      interval: 60s
      start_period: 90s
      start_interval: 60s
    depends_on:
      - identity-service
      - redis
      - rabbitmq
      - file-storage
  worker-api-gateway:
    env_file:
    - .env.example
    image: opswat/metadefender-cluster:worker-api-gateway-
2.0.0-debian-12
    container_name: worker-api-gateway
    ports:
      - "8893"
      - 7777:7777
    networks:
      - mddc
    deploy:
      restart_policy:
        condition: on-failure
    healthcheck:
      test: ["CMD", "true"]
      interval: 5s
      timeout: 2s
      start_period: 30s
    depends_on:
      control-center:
        condition: service_healthy
  worker-core:
    env_file:
    - .env.example
    image: opswat/metadefender-cluster:worker-core-2.0.0-
debian-12
    container_name: worker-core
    ports:
      - "8008"
    networks:
      - mddc
    deploy:
      restart_policy:
        condition: on-failure
    depends_on:
      control-center:
        condition: service_healthy
      worker-api-gateway:
        condition: service_healthy

networks:
  mddc:
    driver: bridge
    ipam:
      config:
        - subnet: 10.0.0.0/24
          gateway: 10.0.0.1
```

```
##Ensure to replace with your specific image tag
```

2. Prepare an environment variable file named `.env.example` and provide with your own values
3. Run the application with the command:

yaml

```
docker compose up -d
```

Known limitation

- When the host experiences resource limitations or degraded performance, some containers may fail to start properly. In such cases, restarting the container is recommended to restore normal operation.

Recommended Setup

Although it is possible to install Redis Caching Server, RabbitMQ Message Broker and Postgres Database Server and MetaDefender Cluster (MD Cluster) File Storage on the same machine, they should be installed separately on various machines to optimize their performance.

Redis Caching Server

The caching server consumes a large amount of memory while operating; hence, a machine with ample and high-speed memory is best suited to this component.

RabbitMQ Message Broker

The broker is one of keys that powers MetaDefender Cluster architecture, it ensures tasks are delivered to MetaDefender Core instances in an equitable manner, delivering a "broken" task to a healthy MetaDefender Core instance and spreading tasks to new instances if more MetaDefender Core instances are added to system. For that reason, the broker should be hosted on a separate machine.

Postgres Database Server

MetaDefender Cluster database is split into three main clusters.

Data Lake stores scan results and other details related to requests such `data_id`, hashes, etc.

Since Data Lake is shared among MetaDefender **Core** and MD Cluster **API Gateway** instances, it should be hosted on a large-volume and high-speed disk. The network is also essential to Data Lake; a high-speed network is necessary.

Data Warehouse, which prepares materials for building executive reports, uses a single connection to Data Lake and collects data periodically.

Since executive reports may be stored for a long period of time for MD Cluster Control Center to access, Data warehouse should be hosted on a large-volume machine.

MetaDefender Cluster File Storage

MD Cluster **File Storage** is shared among **MetaDefender Core** and MD Cluster **API Gateway** instances. The server consumes a large amount of disk to store the submitted files from instances. Since all file-related traffic goes through MD Cluster **File Storage**, a high speed network is essential. Rocky 9.0 is recommended to host MD Cluster **File Storage**.

MetaDefender Cluster API Gateway

Due to differences in Operating System and Nginx support on Windows and Linux, MD Cluster **API Gateway** should be hosted on a Linux machine running Rocky 9.0 for high throughput of file scan submissions.

MetaDefender Core

One of strong aspects of MetaDefender Cluster is that it can support a hybrid architecture in which MD Cluster **API Gateway** and MD Cluster **File Storage** instances may be hosted on Linux while **MetaDefender Core** instances can be on Windows. Therefore, based on customer requirements, **MetaDefender Core** instances can be hosted on Windows or Linux machines.

> **Warning**
>
> It is recommended to setup all MetaDefender Core instances on Windows or on Linux.
>
> Mixed OS run is unsupported.

License activation

MetaDefender Cluster supports two types of license activations:

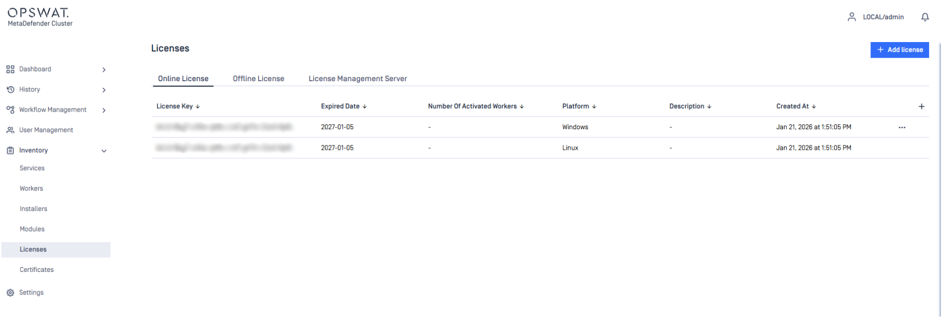
- Online Activation
- Offline Activation
- License Management Server Activation

Online Activation

MetaDefender Cluster (MD Cluster) supports seamless license activation for every deployed **MetaDefender Core** instance. The license key must be provided to MD Cluster **Control Center** and will be manually activated on each individual **MetaDefender Core** instance. If necessary, multiple license keys may also be supplied.

Adding License

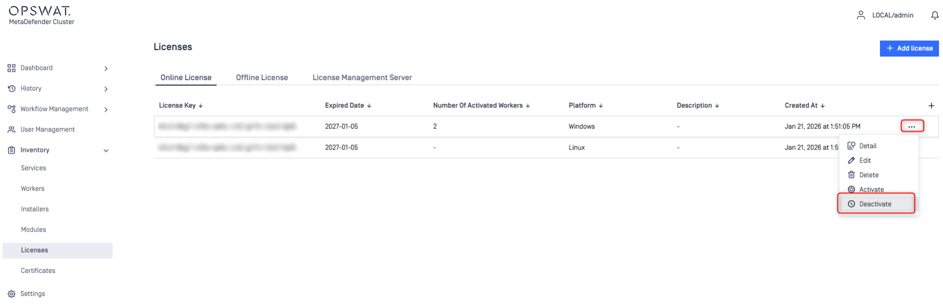
1. Sign in to MD Cluster **Control Center** console.
2. Go to `Inventory` > `Licenses` and select `Add license`.
3. Input your license key and click `Add`.



License Key ↓	Expired Date ↓	Number Of Activated Workers ↓	Platform ↓	Description ↓	Created At ↓
[redacted]	2027-01-05	-	Windows	-	Jan 21, 2026 at 1:51:05 PM
[redacted]	2027-01-05	-	Linux	-	Jan 21, 2026 at 1:51:05 PM

License Activation

1. Sign in to MD Cluster **Control Center** console.
2. From the left side bar, go to `Inventory` > `Licenses`.
3. From the list of available licenses, choose the key you wish to use for activation.
4. Click `Activate` to apply the license key to the appropriate instance(s).



License Key ↓	Expired Date ↓	Number Of Activated Workers ↓	Platform ↓	Description ↓	Created At ↓
[redacted]	2027-01-05	2	Windows	-	Jan 21, 2026 at 1:51:05 PM
[redacted]	2027-01-05	-	Linux	-	Jan 21, 2026 at 1:5

Info

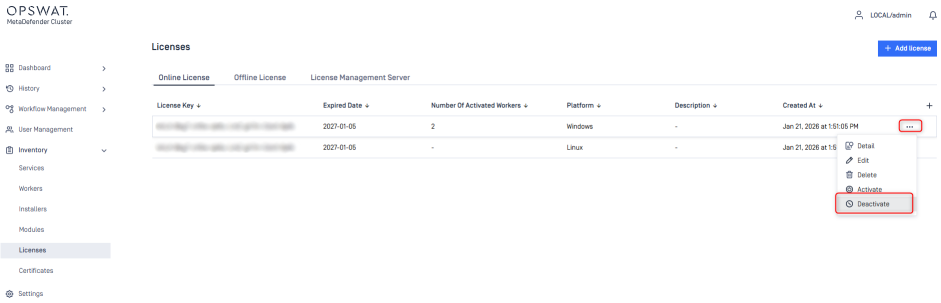
Once `Activate` is clicked, the license key will be applied to all unlicensed **MetaDefender Core** instances. Ensure that your license quota is sufficient to cover all unlicensed instances.

You can view the number of activated instances and available slots by selecting **Details** on the license key. At the moment, **Details** can only be viewed after activation is successful.

License Deactivation

Follow these steps to deactivate your license:

1. Sign in to MD Cluster **Control Center** console.
2. From the left side bar, select `Inventory` > `Licenses`.
3. From the list of available licenses, choose the key you wish to use for deactivation.
4. Click Deactivate to remove the license from all **MetaDefender Core** instances currently activated with the license key.



License Key ↓	Expired Date ↓	Number Of Activated Workers ↓	Platform ↓	Description ↓	Created At ↓
[redacted]	2027-01-05	2	Windows	-	Jan 21, 2026 at 1:51:05 PM
[redacted]	2027-01-05	-	Linux	-	Jan 21, 2026 at 1:5

Info

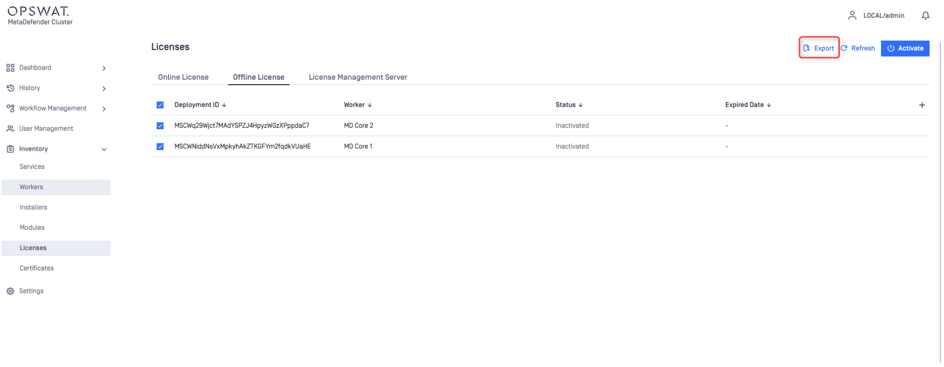
Once deactivated, the license slots will become available and can be reassigned when necessary.

When **MetaDefender Core** instances are undeployed through MD Cluster **Control Center**, their associated licenses are automatically deactivated.

Offline Activation

Collect Deployment IDs

1. Sign in to MetaDefender Cluster (MD Cluster) **Control Center** console.
2. Go to `Inventory` > `Licenses` and select `Offline License` tab.
3. Select Deployment IDs of MetaDefender Core instances you prefer to activate.
4. Press `Export` at the top right corner and save the exported file to your location of choice.



☑	Deployment ID ↓	Worker ↓	Status ↓	Expired Date ↓	+
☑	MSCWq29Wjct7MAdYSPZJ4HpyzWGzXPppdaC7	MD Core 2	Inactivated	-	
☑	MSCWNiddNsVxMpkyhAkZTKGFYm2fqdkVUaHE	MD Core 1	Inactivated	-	

Info

MetaDefender Cluster **Control Center** only displays the Deployment IDs of MetaDefender Core instances that have not been activated thus far.

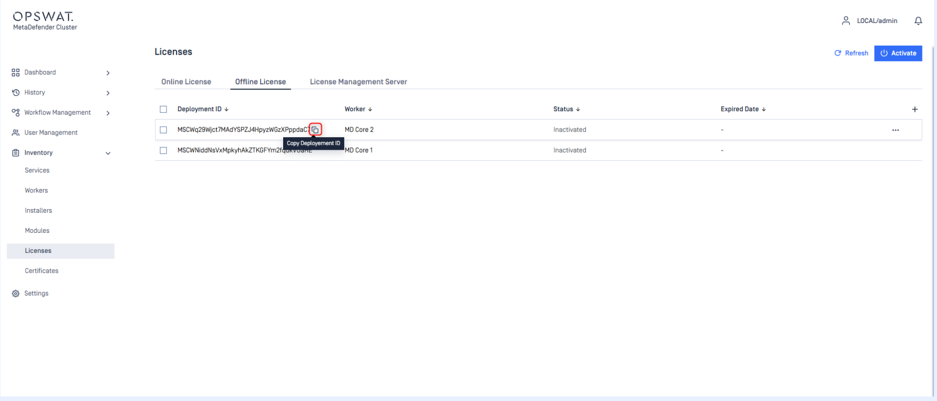
Info

The exported file includes a list of chosen Deployment IDs that will be used for activation in the subsequent stage.

Info

To collect the Deployment ID of a **single** MetaDefender Core instance, please:

1. Hover your mouse over the preferred Deployment ID to display the copy button.
2. Press the copy button.
3. Retain the copied Deployment ID and proceed to the next stage.

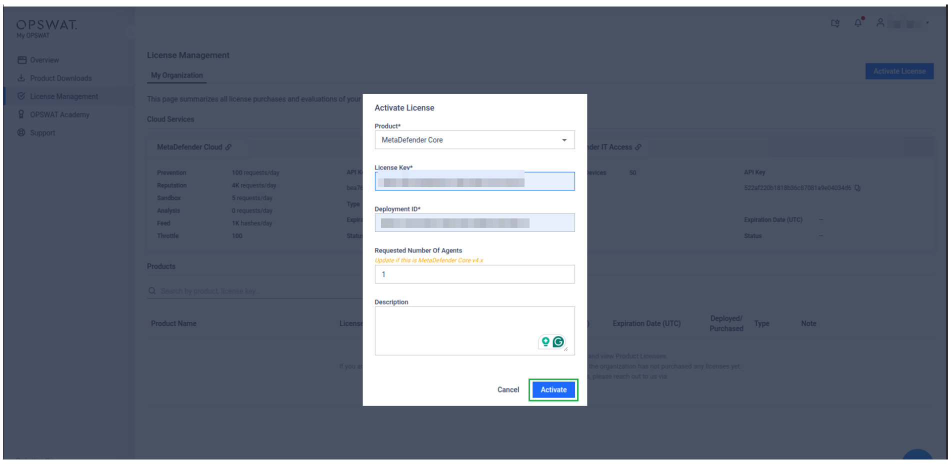


Licenses

☐	Deployment ID ↓	Worker ↓	Status ↓	Expired Date ↓
☐	MSCWq29Wjct7MAdYSPZJ4HpyzWGzXPppdaC7	MD Core 2	Inactivated	-
☐	MSCWNiddNsVxMpkyhAkZTKGFYm2fqukVUaHE	MD Core 1	Inactivated	-

Activate license with Deployment ID

1. Sign in to MyOPSWAT with your account.
2. Navigate to `License Management` on the left side panel.
3. Click `Activate License`.
4. Fill out all necessary information, including your Activation Key, Deployment ID and selection of the Package you require.



Activate License

Product*: MetaDefender Core

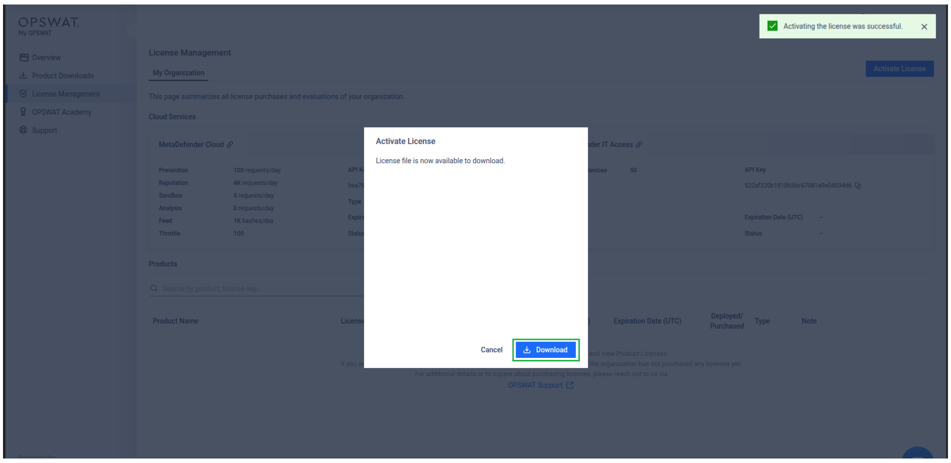
License Key*: [redacted]

Deployment ID*: [redacted]

Requested Number Of Agents
Update if this is MetaDefender Core v4.x
1

Description

5. Click `Activate`.
6. Click `Download` and store the license file to your secure location.



Activating the license was successful.

License Management

This page summarizes all license purchases and evaluations of your organization.

Cloud Services

MetaDefender Cloud	
Prevention	100 requests/day
Reputation	4K requests/day
Sandbox	5 requests/day
Analysis	0 requests/day
Feed	1K hashes/day
Throttle	100

API Key: 522af220b1818b36c87081a9e04034d6

Expiration Date (UTC): --

Status: --

Products

Search by product, license key...

Product Name | Expiration Date (UTC) | Deployed/ Purchased | Type | Note

For additional details or to inquire about purchasing licenses, please reach out to us via OPSWAT Support

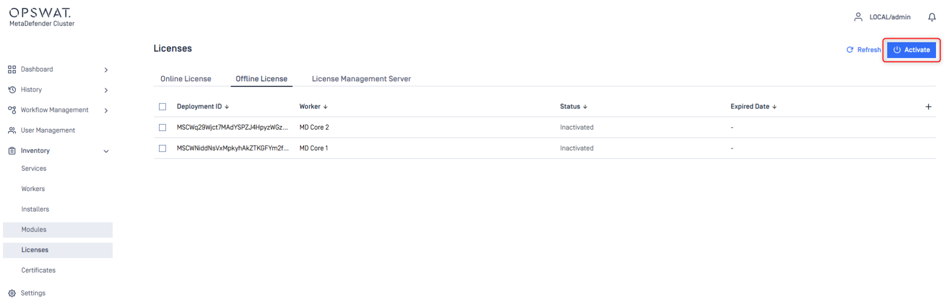
Activate License

License file is now available to download.

> **Info**
>
> The license file is associated with one unique Deployment ID. The users must carry out steps 3 to 6 for every deployment ID on their list.

Activate MetaDefender Core instances with license files

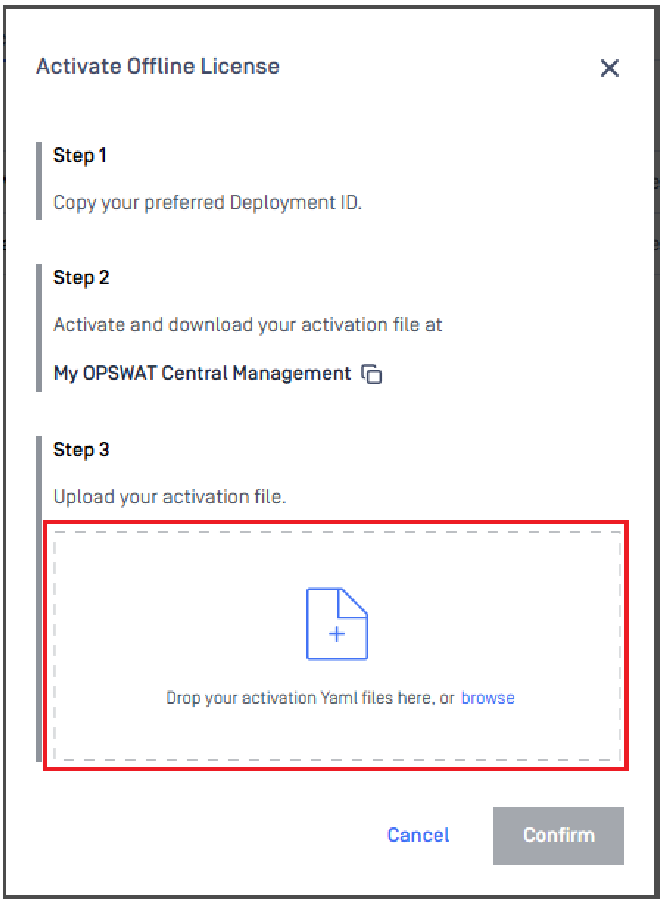
1. Sign in to MD Cluster **Control Center** console.
2. Go to `Inventory` > `Licenses` and select `Offline License` tab.
3. Click `Activate`.



Licenses

Deployment ID	Worker	Status	Expired Date
MSCWq29Wjct7MAdYSPZJ4HpyzWGz...	MD Core 2	Inactivated	-
MSCWNiddNsVxMpkyhAkZTKGFYm2f...	MD Core 1	Inactivated	-

4. Drop the license files into the dash area for submission.



Activate Offline License

Step 1

Copy your preferred Deployment ID.

Step 2

Activate and download your activation file at

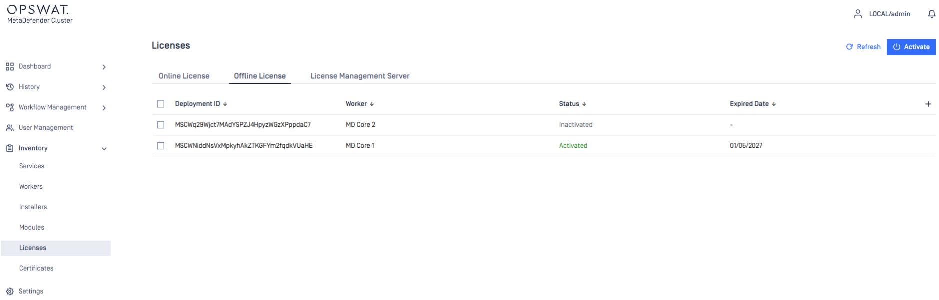
My OPSWAT Central Management

Step 3

Upload your activation file.

Drop your activation Yaml files here, or browse

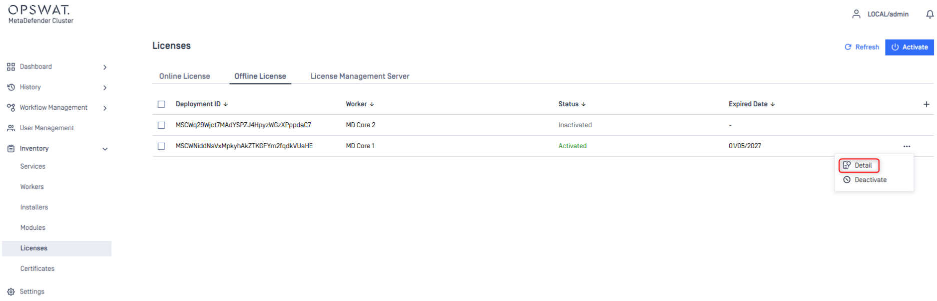
5. Click `Confirm` to complete.
6. MD Cluster **Control Center** activates MetaDefender Core instances associated with the provided license files and displays their activation status.



Licenses

□	Deployment ID ↓	Worker ↓	Status ↓	Expired Date ↓	+
□	MSCWq29Wjct7MAdYSPZJ4HpyzWGzXPppdaC7	MD Core 2	Inactivated	-	
□	MSCWNiddNsVxMpkyhAkZTKGFYm2fqdkVUaHE	MD Core 1	Activated	01/05/2027	

7. Select an activated MetaDefender Core instance and press `Details` to view the license details



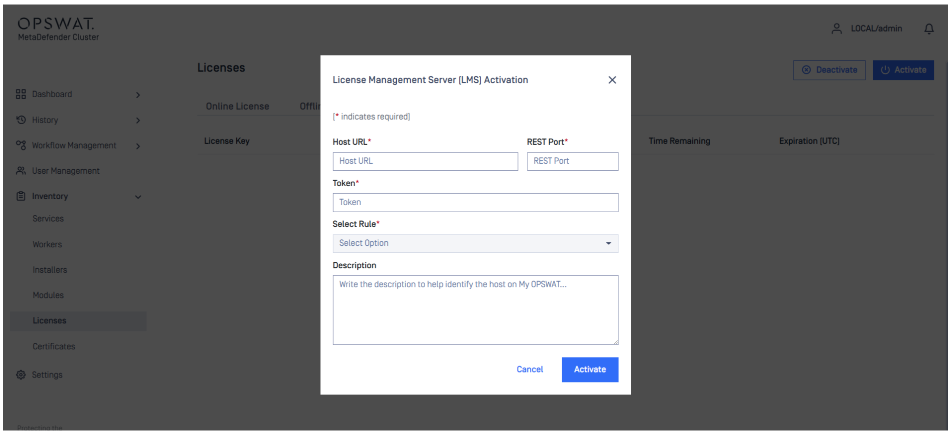
Licenses

□	Deployment ID ↓	Worker ↓	Status ↓	Expired Date ↓
□	MSCWq29Wjct7MAdYSPZJ4HpyzWGzXPppdaC7	MD Core 2	Inactivated	-
□	MSCWNiddNsVxMpkyhAkZTKGFYm2fqdkVUaHE	MD Core 1	Activated	01/05/2027

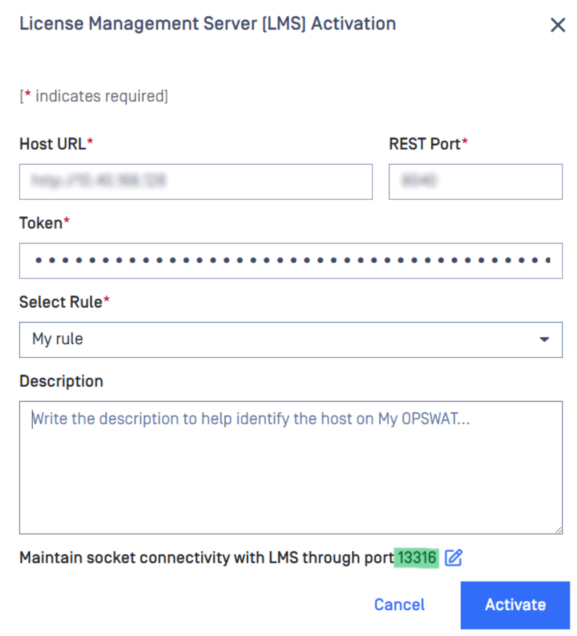
License Management Server Activation

Connect to License Management Server (LMS)

1. Sign in to MetaDefender Cluster **Control Center** console.
2. Go to `Inventory` > `Licenses` and select License Management Server tab.
3. Select `Activate` and provide the necessary information in the required fields:
    a. **Host URL**: The URL of the License Management Server to connect to.
    b. **REST Port**: The port of the License Management Server.
    c. **Token**: Access token obtained from the License Management Server.



4. After input all required fields, the connection to LMS will be established and available rules can be selected under `Select Rules`.



License Management Server (LMS) Activation

[* indicates required]

Host URL*: [redacted]

REST Port*: [redacted]

Token*: ••

Select Rule*: My rule

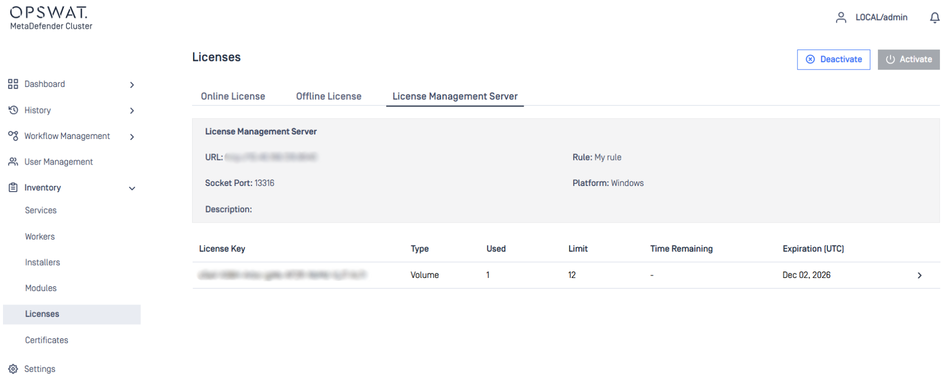
Description: Write the description to help identify the host on My OPSWAT...

Maintain socket connectivity with LMS through port 13316

Info

The port number defined in "Maintain socket connectivity with LMS through port `<port_number>`" is required to sustain connectivity between MD Cluster Control Center and License Management Server. In order for successfully activation, **confirm that the port is properly configured and allowed in firewall setting**.

5. Select the appropriate rule and choose Activate. Upon successful completion, the license details will be shown.



Licenses

License Management Server

URL: [redacted]

Rule: My rule

Socket Port: 13316

Platform: Windows

Description:

License Key	Type	Used	Limit	Time Remaining	Expiration (UTC)
[redacted]	Volume	1	12	-	Dec 02, 2026

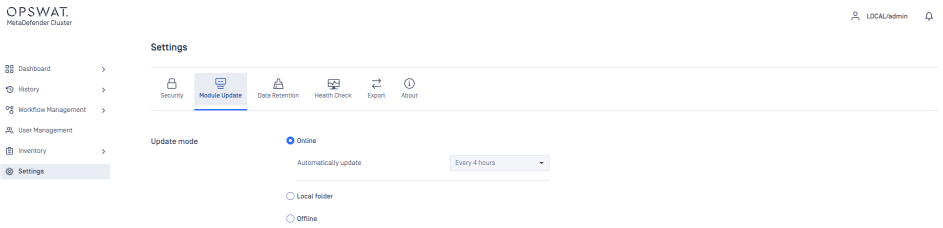
ⓘ **Info**

Once activation is successful, the License Management Server will manage the license status of all MetaDefender Core instances. Please ensure sufficient quota is available.

6. Check instance status under `Inventory` > `Workers` and confirm all MetaDefender Core instances is activated successfully.

Module update

MetaDefender Cluster (MD Cluster) introduces three modes of Module Update. To switch between the modes, please sign in to MD Cluster **Control Center** console, navigate to `Settings` and select `Module Update` tab.



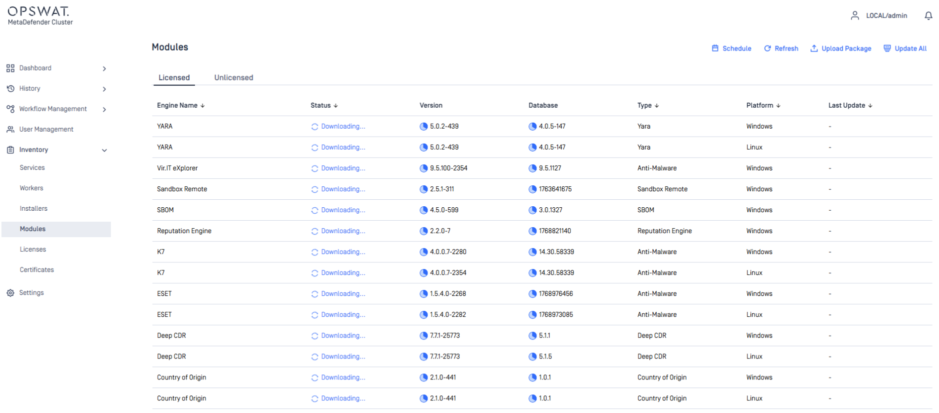
Update mode
● Online
Automatically update: Every 4 hours
○ Local folder
○ Offline

Info

Online update mode is enabled by default.

Online module update

In online update mode, MD Cluster **Control Center** will base its checks on the activated licenses to find and download the latest engine packages from OPSWAT online update infrastructure, repeating this process every four hours.



Engine Name ↓	Status ↓	Version	Database	Type ↓	Platform ↓	Last Update ↓
YARA	Downloading...	5.0.2-439	4.0.5-147	Yara	Windows	-
YARA	Downloading...	5.0.2-439	4.0.5-147	Yara	Linux	-
Vir.IT eXplorer	Downloading...	9.5.100-2354	9.5.1127	Anti-Malware	Windows	-
Sandbox Remote	Downloading...	2.5.1-311	1763641675	Sandbox Remote	Windows	-
SBOM	Downloading...	4.5.0-599	3.0.1327	SBOM	Windows	-
Reputation Engine	Downloading...	2.2.0-7	1768821140	Reputation Engine	Windows	-
K7	Downloading...	4.0.0.7-2280	14.30.58339	Anti-Malware	Windows	-
K7	Downloading...	4.0.0.7-2354	14.30.58339	Anti-Malware	Linux	-
ESET	Downloading...	1.5.4.0-2268	1768976456	Anti-Malware	Windows	-
ESET	Downloading...	1.5.4.0-2282	1768973085	Anti-Malware	Linux	-
Deep CDR	Downloading...	7.7.1-25773	5.1.1	Deep CDR	Windows	-
Deep CDR	Downloading...	7.7.1-25773	5.1.5	Deep CDR	Linux	-
Country of Origin	Downloading...	2.1.0-441	1.0.1	Country of Origin	Windows	-
Country of Origin	Downloading...	2.1.0-441	1.0.1	Country of Origin	Linux	-

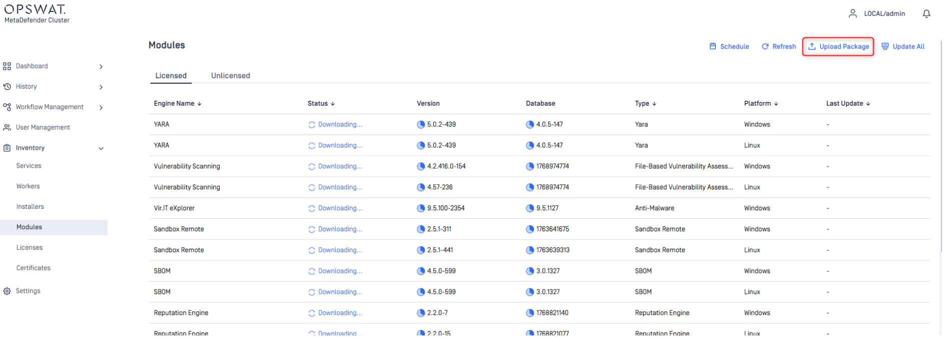
All downloaded engine packages are verified and stored in MD Cluster **File Storage** for licensed MetaDefender Core instances to pull, install, or upgrade on their end. Through this mechanism, the instances cease to independently pull the engine packages from the update infrastructure, conserving network bandwidth while enhancing their readiness.

Offline module update

In offline update mode, administrators must download the licensed engine packages from **MetaDefender Update Downloader** and upload them manually to MD Cluster **Control Center**.

> **Info**
>
> Please reference here for more details about downloading engine packages from MetaDefender Update Downloader.

1. Sign in to MD Cluster **Control Center** console.
2. Go to `Inventory` > `Modules`.
3. Press `Upload Package` at the top right corner.

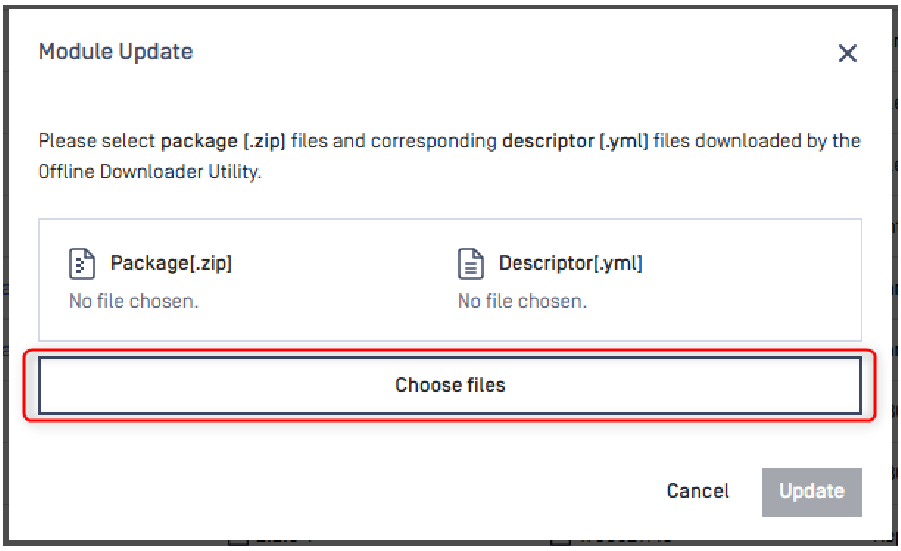


Modules

Engine Name ↓	Status ↓	Version	Database	Type ↓	Platform ↓	Last Update ↓
YARA	Downloading...	5.0.2-439	4.0.5-147	Yara	Windows	-
YARA	Downloading...	5.0.2-439	4.0.5-147	Yara	Linux	-
Vulnerability Scanning	Downloading...	4.2.416.0-154	1768974774	File-Based Vulnerability Assess...	Windows	-
Vulnerability Scanning	Downloading...	4.57-236	1768974774	File-Based Vulnerability Assess...	Linux	-
Vir.IT eXplorer	Downloading...	9.5.100-2354	9.5.1127	Anti-Malware	Windows	-
Sandbox Remote	Downloading...	2.5.1-311	1763641675	Sandbox Remote	Windows	-
Sandbox Remote	Downloading...	2.5.1-441	1763639313	Sandbox Remote	Linux	-
SBOM	Downloading...	4.5.0-599	3.0.1327	SBOM	Windows	-
SBOM	Downloading...	4.5.0-599	3.0.1327	SBOM	Linux	-
Reputation Engine	Downloading...	2.2.0-7	1768821140	Reputation Engine	Windows	-
Reputation Engine	Downloading...	2.2.0-15	1768821077	Reputation Engine	Linux	-

> **Info**
>
> `Update All` is always disabled if Offline Update mode is selected in `Settings` > `Module Update`.

4. Choose your engine package files.



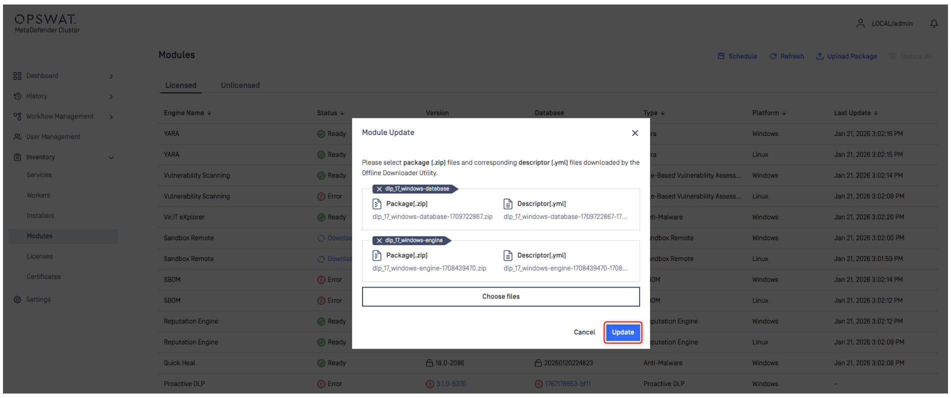
Module Update

Please select **package (.zip)** files and corresponding **descriptor (.yml)** files downloaded by the Offline Downloader Utility.

Package[.zip]
No file chosen.

Descriptor[.yml]
No file chosen.

5. Click `Update` to submit the package files.



Engine Name ↓	Status ↓	Version	Database	Type ↓	Platform ↓	Last Update ↓
YARA	Ready			ra	Windows	Jan 21, 2026 3:02:16 PM
YARA	Ready			ra	Linux	Jan 21, 2026 3:02:15 PM
Vulnerability Scanning	Ready			e-Based Vulnerability Assess...	Windows	Jan 21, 2026 3:02:14 PM
Vulnerability Scanning	Error			e-Based Vulnerability Assess...	Linux	Jan 21, 2026 3:02:09 PM
Vir.IT eXplorer	Ready			ti-Malware	Windows	Jan 21, 2026 3:02:20 PM
Sandbox Remote	Downloa			ndbox Remote	Windows	Jan 21, 2026 3:02:00 PM
Sandbox Remote	Downloa			ndbox Remote	Linux	Jan 21, 2026 3:01:59 PM
SBOM	Error			OM	Windows	Jan 21, 2026 3:02:14 PM
SBOM	Error			OM	Linux	Jan 21, 2026 3:02:12 PM
Reputation Engine	Ready			putation Engine	Windows	Jan 21, 2026 3:02:12 PM
Reputation Engine	Ready			putation Engine	Linux	Jan 21, 2026 3:02:09 PM
Quick Heal	Ready	18.0-2086	20260120224823	Anti-Malware	Windows	Jan 21, 2026 3:02:08 PM
Proactive DLP	Error	3.1.0-5370	1767178853-bf11	Proactive DLP	Windows	-

Module Update

Please select **package (.zip)** files and corresponding **descriptor (.yml)** files downloaded by the Offline Downloader Utility.

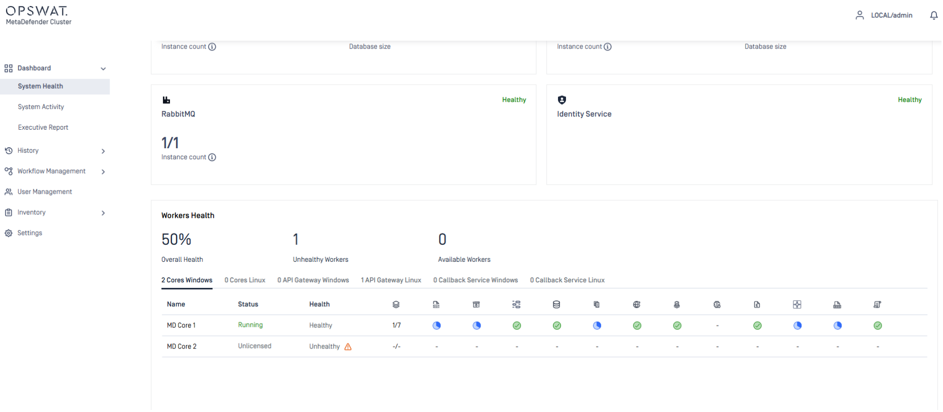
dlp_17_windows-database
Package[.zip]
dlp_17_windows-database-1709722867.zip
Descriptor[.yml]
dlp_17_windows-database-1709722867-17...

dlp_17_windows-engine
Package[.zip]
dlp_17_windows-engine-1708439470.zip
Descriptor[.yml]
dlp_17_windows-engine-1708439470-1708...

Info

Package files from various engines can be selected simultaneously.

6. Wait until engine packages are ready.
7. Engine update statuses on MetaDefender Core instances can be monitored in `Dashboard` > `System Health` > `Worker Health`.



Instance count | Database size | Instance count | Database size

RabbitMQ — Healthy
1/1 Instance count

Identity Service — Healthy

Workers Health

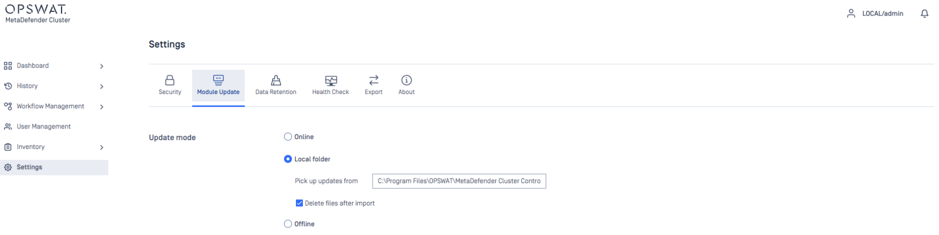
50% Overall Health
1 Unhealthy Workers
0 Available Workers

2 Cores Windows | 0 Cores Linux | 0 API Gateway Windows | 1 API Gateway Linux | 0 Callback Service Windows | 0 Callback Service Linux

Name	Status	Health														
MD Core 1	Running	Healthy	1/7	◔	◔	✓	✓	◔	✓	✓	-	✓	◔	◔	✓
MD Core 2	Unlicensed	Unhealthy ⚠	-/-	-	-	-	-	-	-	-	-	-	-	-	-

Local folder update

In local folder update, the administrator can specify the path to a folder in `Pick updates from` option. MD Cluster **Control Center** autonomously gathers packages that are added to the folder, then verifies the packages and stores them in MD Cluster **File Storage** for licensed MetaDefender Core instances to retrieve, install, or update on their end.



Settings

Update mode
○ Online
● Local folder
Pick up updates from: C:\Program Files\OPSWAT\MetaDefender Cluster Contro
☑ Delete files after import
○ Offline

The administrator has the option to select `Delete files after import` so that MD Cluster **Control Center** wipes all packages upon success.

Info

Network-mapped drives and UNC paths are permitted.

- On Windows, ensure that MD Cluster Control Center has permission to access to the folder.
- On Linux, ensure that the folder has read and execute permissions set (e.g., `chmod 755`).

MetaDefender Cluster Documentation

The users can consult this web page or, alternatively, they can download the manual in pdf format from the link below:

MetaDefender Cluster manual

- Download link
- SHA256:

High Availability

Overview

In MetaDefender Cluster (MD Cluster), critical components for its continuous operation include RabbitMQ, Redis, Postgres, and MD Cluster File Storage. Any disruption of these components will lead to an interruption in the scanning processes and result in a failed verdict for the processed files. To prevent the interruption, high-availability solutions must be implemented on the components.

A strategy for achieving high availability is the replication and redundancy of essential components. The key concept is that if a single component fails, the redundant system takes over seamlessly, avoiding any interruption in service. Following are guidelines to set up the high availability solution on individual components and apply them in MetaDefender Cluster.

- High availability support for MD Cluster File Storage.
- High availability support for RabbitMQ.
- High availability support for Redis.
- High availability support for Data lake.

High Availability support for File Storage

Key concept

The High Availability solution for MetaDefender Cluster (MD Cluster) **File Storage** is implemented in this manner:

- A file is stored across multiple MD Cluster **File Storages** by MD Cluster **API Gateway** or **MetaDefender Core**.
- MD Cluster **API Gateway** or **MetaDefender Core** must request all MD Cluster **File Storage** instances for a file existence or a file download.

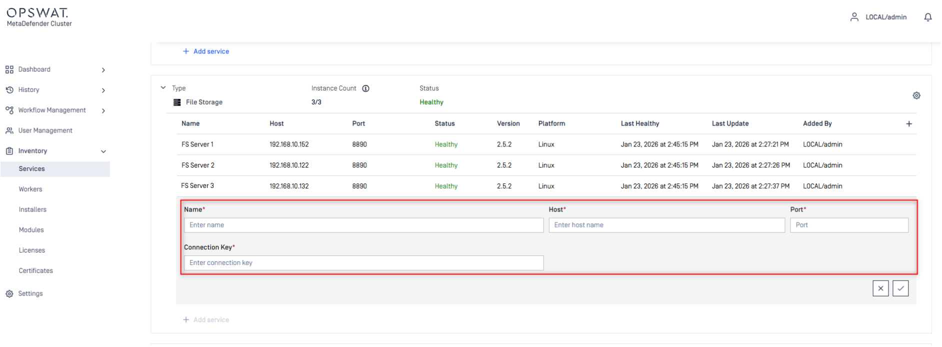
Info

A minimum of **three** MD Cluster File Storage instances must be installed on separate hosts for High Availability solution to function properly.

All MD Cluster File Storage instances must be of an identical version.

Setup Instructions

1. Setup MD Cluster **File Storage** instances on individual servers.
2. Sign to MD Cluster **Control Center** console with your Administrator account.
3. Navigate to `Inventory` > `Services`.
4. Expand the `File Storage Service` group.
5. Click `Add service`.
6. Enter the values for `Name`, `Host`, `Port` and `Connection Key` fields of individual MD Cluster File Storage instances set up in Step 1.
7. Click the `Check` icon in the bottom right to complete.

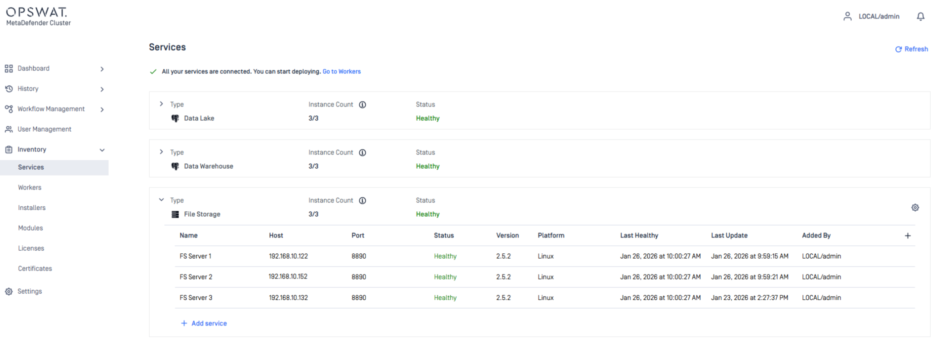


Type: File Storage
Instance Count: 3/3
Status: Healthy

Name	Host	Port	Status	Version	Platform	Last Healthy	Last Update	Added By
FS Server 1	192.168.10.152	8890	Healthy	2.5.2	Linux	Jan 23, 2026 at 2:45:15 PM	Jan 23, 2026 at 2:27:21 PM	LOCAL/admin
FS Server 2	192.168.10.122	8890	Healthy	2.5.2	Linux	Jan 23, 2026 at 2:45:15 PM	Jan 23, 2026 at 2:27:26 PM	LOCAL/admin
FS Server 3	192.168.10.132	8890	Healthy	2.5.2	Linux	Jan 23, 2026 at 2:45:15 PM	Jan 23, 2026 at 2:27:37 PM	LOCAL/admin

Name*: Enter name
Host*: Enter host name
Port*: Port
Connection Key*: Enter connection key

8. Ensure all MD Cluster `File Storage` instances are healthy and reachable by MD Cluster `Control Center`.



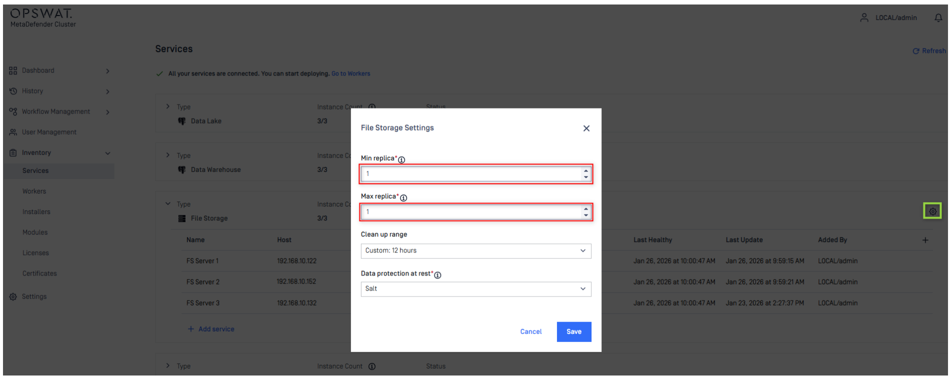
Services

All your services are connected. You can start deploying. Go to Workers

Type	Instance Count	Status
Data Lake	3/3	Healthy
Data Warehouse	3/3	Healthy
File Storage	3/3	Healthy

Name	Host	Port	Status	Version	Platform	Last Healthy	Last Update	Added By
FS Server 1	192.168.10.122	8890	Healthy	2.5.2	Linux	Jan 26, 2026 at 10:00:27 AM	Jan 26, 2026 at 9:59:15 AM	LOCAL/admin
FS Server 2	192.168.10.152	8890	Healthy	2.5.2	Linux	Jan 26, 2026 at 10:00:27 AM	Jan 26, 2026 at 9:59:21 AM	LOCAL/admin
FS Server 3	192.168.10.132	8890	Healthy	2.5.2	Linux	Jan 26, 2026 at 10:00:27 AM	Jan 23, 2026 at 2:27:37 PM	LOCAL/admin

9. Click on the gear icon in the top left of `File Storage Service` group to configure the minimum and maximum replicas.



File Storage Settings

Min replica*: 1

Max replica*: 1

Clean up range: Custom: 12 hours

Data protection at rest*: Salt

- Minimum replica: The minimum number of data copies that must be written for the operation to succeed.
- Maximum replica: The maximum of data copies stored across the system.

> **Info**
>
> To balance performance and high availability efficiency, the minimum and maximum replicas should be set to the following values:
>
> - `Min replica` = 2
> - `Max replica` = 3

10. Click `Save` to complete.

High Availability support for RabbitMQ

Info

A minimum of **three** RabbitMQ nodes must be installed on separate hosts for High Availability solution to function properly.

An **odd number** of RabbitMQ nodes is required.

All RabbitMQ nodes must be of an identical version.

RabbitMQ cluster

1. Install RabbitMQ nodes on servers.
2. Ensure each node can resolve its own hostname and those of the others.

- Start **Command Prompt on Windows** or **Terminal on Linux**, and run the following command to get hostname.

bash

```bash
# Windows
> hostname

# Debian/Ubuntu or Red Hat/Rocky
$ hostname
```

- In **Command Prompt on Windows** or **Terminal on Linux** of any RabbitMQ node, and run the following command to ping to the other using its hostname.

bash

```bash
# Windows
> ping <other_node_hostname>

# Debian/Ubuntu or Red Hat/Rocky
$ ping <other_node_hostname>
```

3. On each RabbitMQ nodes, open the following ports.

Default port	Process
5672	Used by MD Cluster **Control Center**, MD Cluster **API Gateway** and **MetaDefender Core**.
4369	Used by discovery daemon on each RabbitMQ nodes and `rabbitmqctl` tool.
25672	Used by each RabbitMQ nodes and `rabbitmqctl` tool to communicate to the other nodes.
15672	Used by `rabbitmq-management` plugin.

4. Verify that the Erlang cookies of all RabbitMQ nodes are identical.

Info

RabbitMQ nodes and `rabbitmqctl` tool use a **cookie** to determine whether they are allowed to communicate with each other. For two nodes to be able to communicate they must have the same shared secret called the Erlang cookie. The cookie is a string of alphanumeric characters up to 255 characters in size.

- In **Windows**, access the specified locations to check the cookie contents.

Type	Location
Server cookie	`C:\Windows\system32\config\systemprofile.erlang.cookie`
Command line cookie	`C:\Users%USERNAME%.erlang.cookie`

- In **Linux**, access the specified locations to check the cookie contents.

Type	Location
Server cookie	`/var/lib/rabbitmq/.erlang.cookie`
Command line cookie	`$HOME/.erlang.cookie`

5. Select one node to be the leader of RabbitMQ cluster.

6. In **Command Prompt on Windows** or **Terminal on Linux** of the server hosting the leader, run the following command to obtain its node name.

bash

```
# Windows
> rabbitmqctl status

# Debian/Ubuntu or Red Hat/Rocky
$ rabbitmqctl status
```

7. In **Command Prompt on Windows** or **Terminal on Linux** of each member node server, run the following command to join the node to the same cluster as the leader.

bash

```
# Windows
> rabbitmqctl stop_app
> rabbitmqctl reset
> rabbitmqctl join_cluster <leader_node_name>
> rabbitmqctl start_app

# Debian/Ubuntu or Red Hat/Rocky
$ rabbitmqctl stop_app
$ rabbitmqctl reset
$ rabbitmqctl join_cluster <leader_node_name>
$ rabbitmqctl start_app
```

8. In **Command Prompt on Windows** or **Terminal on Linux** of all nodes, ensure they are in the same cluster.

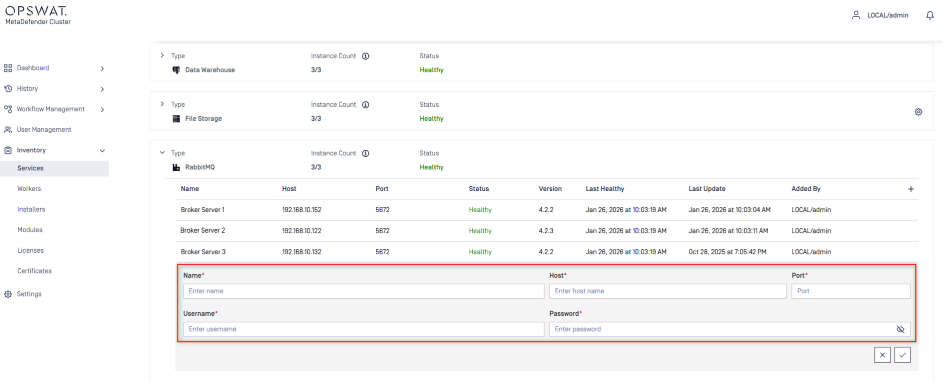
bash

```
# Windows
> rabbitmqctl cluster_status

# Debian/Ubuntu or Red Hat/Rocky
$ rabbitmqctl cluster_status
```

Setup Instructions

1. Sign to MD Cluster **Control Center** console with your Administrator account.
2. Navigate to `Inventory` > `Services`.

3. Expand the `RabbitMQ` group.
4. Click `Add service`.
5. Enter the values for `Name`, `Host`, `Port`, `Username` and `Password` fields of individual RabbitMQ nodes set up in Build RabbitMQ cluster.

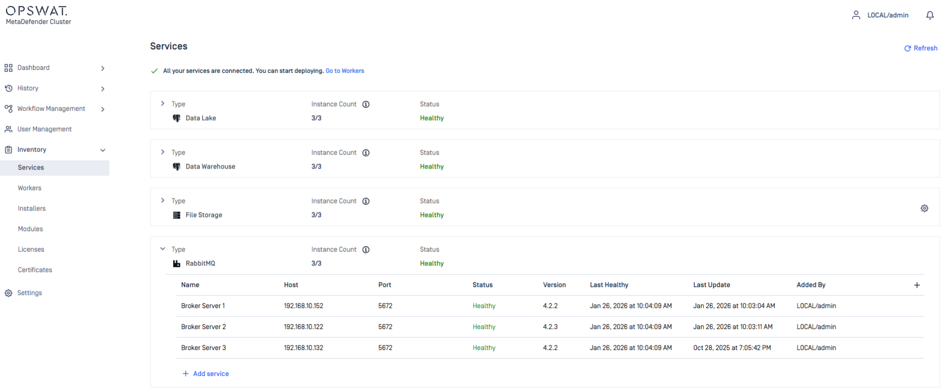


Type	Instance Count	Status
Data Warehouse	3/3	Healthy
File Storage	3/3	Healthy
RabbitMQ	3/3	Healthy

Name	Host	Port	Status	Version	Last Healthy	Last Update	Added By
Broker Server 1	192.168.10.152	5672	Healthy	4.2.2	Jan 26, 2026 at 10:03:19 AM	Jan 26, 2026 at 10:03:04 AM	LOCAL/admin
Broker Server 2	192.168.10.122	5672	Healthy	4.2.3	Jan 26, 2026 at 10:03:19 AM	Jan 26, 2026 at 10:03:11 AM	LOCAL/admin
Broker Server 3	192.168.10.132	5672	Healthy	4.2.2	Jan 26, 2026 at 10:03:19 AM	Oct 28, 2025 at 7:05:42 PM	LOCAL/admin

Name* Enter name
Host* Enter host name
Port* Port
Username* Enter username
Password* Enter password

6. Click the Check icon in the bottom right to complete.
7. Ensure all RabbitMQ nodes are reachable by the MD Cluster **Control Center**.



Services

✓ All your services are connected. You can start deploying. Go to Workers

Type	Instance Count	Status
Data Lake	3/3	Healthy
Data Warehouse	3/3	Healthy
File Storage	3/3	Healthy
RabbitMQ	3/3	Healthy

Name	Host	Port	Status	Version	Last Healthy	Last Update	Added By
Broker Server 1	192.168.10.152	5672	Healthy	4.2.2	Jan 26, 2026 at 10:04:09 AM	Jan 26, 2026 at 10:03:04 AM	LOCAL/admin
Broker Server 2	192.168.10.122	5672	Healthy	4.2.3	Jan 26, 2026 at 10:04:09 AM	Jan 26, 2026 at 10:03:11 AM	LOCAL/admin
Broker Server 3	192.168.10.132	5672	Healthy	4.2.2	Jan 26, 2026 at 10:04:09 AM	Oct 28, 2025 at 7:05:42 PM	LOCAL/admin

High Availability support for Redis

ℹ Info

A minimum of **two** Redis instances must be installed on separate hosts for High Availability solution to function properly.

An **odd number** of **Redis Sentinels** should be installed.

Redis Sentinel

1. Install Redis instances on servers.
2. Select one instance as primary. In **Linux Terminal** of the other instances (replicas), run the following command:

bash

```
# Debian/Ubuntu or Red Hat/Rocky
$ redis-cli replicaof <primary_host> <primary_port>
```

3. Build configuration file for **Redis Sentinel**.

bash

```
# The port on which the Sentinel should run
port <SENTINEL_PORT>

# By default Redis does not run as a daemon. Use 'yes' if you
need it.
# Note that Redis will write a pid file in /var/run/redis.pid
when daemonized.
daemonize yes

sentinel monitor myprimary <PRIMARY_IP> <PRIMARY_PORT> 2
# sentinel monitor <master-name> <ip> <port> <quorum>
# quorum is the number of Sentinels that need to agree about
the
# fact the master is not reachable, in order to really mark
the master as
# failing, and eventually start a failover procedure if
possible.


sentinel down-after-milliseconds myprimary 2000
# means sentinel will consider master down after 2 seconds

sentinel failover-timeout myprimary 4000
# means the chosen sentinel has 4 seconds to perform failover


sentinel parallel-syncs myprimary 2
# sets the number of replicas that can be reconfigured to use
the new master
# after a failover at the same time. The lower the number, the
more time it
# will take for the failover process to complete, however if
the replicas are
# configured to serve old data, you may not want all the
replicas to
# re-synchronize with the master at the same time. While the
replication process is
# mostly non blocking for a replica, there is a moment when it
stops to
# load the bulk data from the master. You may want to make
sure only one
# replica at a time is not reachable by setting this option to
the value of 1.
```

ℹ **Info**

Duplicate the configuration file and modify `SENTINEL_PORT` to the appropriate port that the Redis Sentinel instance listens on.

4. Install Redis Sentinel instances on servers with the corresponding configuration files.

bash

```
# Debian/Ubuntu or Red Hat/Rocky
$ sudo redis-server </path/to/sentinel-config-file> --sentinel
```

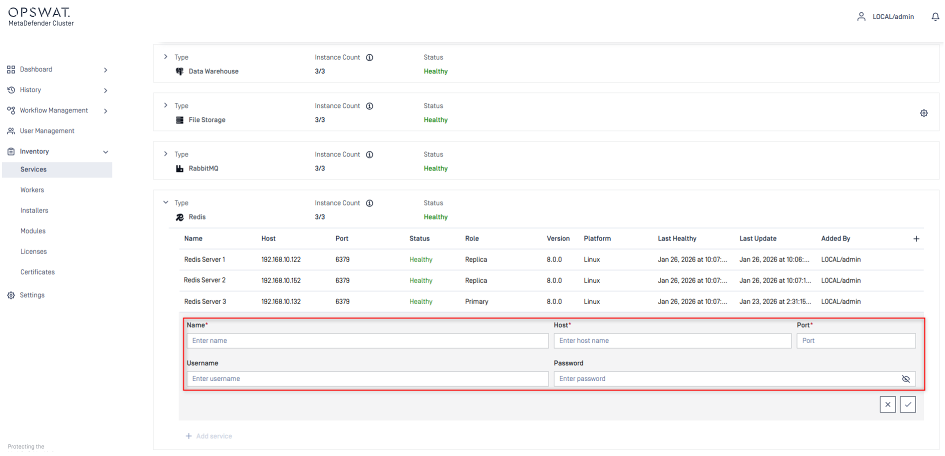
5. Verify the Redis primary and its replicas. In **Linux Terminal** of any machine, run the following command:

bash

```
# Debian/Ubuntu or Red Hat/Rocky
$ redis-cli -h <sentinel_host> -p <sentinel_port>

# Provides information about the Primary
> sentinel master myprimary

# Gives you information about the replicas connected to the
Primary
> sentinel replicas myprimary

# Provides information on the other Sentinels
> sentinel sentinels myprimary

# Provides the IP address of the current Primary
> sentinel get-master-addr-by-name myprimary
```

Setup instructions

1. Sign to MD Cluster **Control Center** console with your Administrator account.
2. Navigate to `Inventory` > `Services`.
3. Expand the `Redis` group.
4. Click `Add service`.
5. Enter the values for `Name`, `Host`, `Port`, `Username` and `Password` fields of individual Redis instance.



Type	Instance Count	Status
Data Warehouse	3/3	Healthy
File Storage	3/3	Healthy
RabbitMQ	3/3	Healthy
Redis	3/3	Healthy

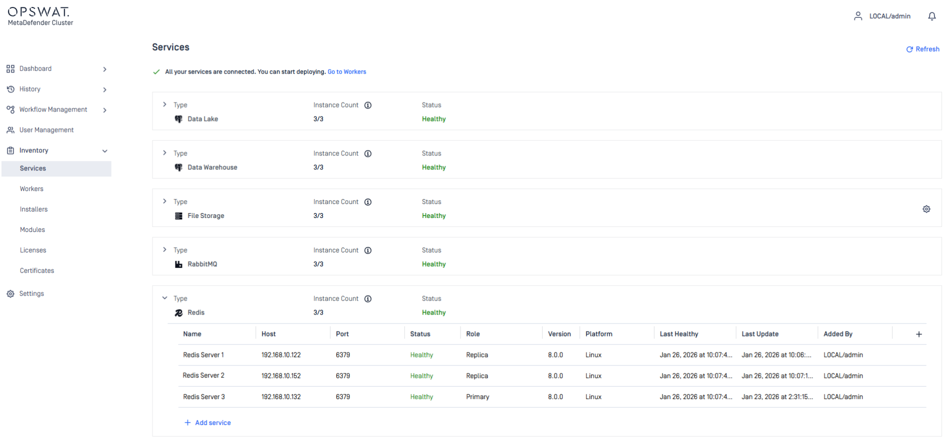
Name	Host	Port	Status	Role	Version	Platform	Last Healthy	Last Update	Added By
Redis Server 1	192.168.10.122	6379	Healthy	Replica	8.0.0	Linux	Jan 26, 2026 at 10:07:...	Jan 26, 2026 at 10:06:...	LOCAL/admin
Redis Server 2	192.168.10.152	6379	Healthy	Replica	8.0.0	Linux	Jan 26, 2026 at 10:07:...	Jan 26, 2026 at 10:07:1...	LOCAL/admin
Redis Server 3	192.168.10.132	6379	Healthy	Primary	8.0.0	Linux	Jan 26, 2026 at 10:07:...	Jan 23, 2026 at 2:31:15...	LOCAL/admin

Name* Enter name
Host* Enter host name
Port* Port
Username Enter username
Password Enter password

Warning

MD Cluster Control Center **only accepts Redis** and **not Redis Sentinel**.

6. Click the Check icon in the bottom right to complete.
7. Ensure all RabbitMQ nodes are reachable by the MD Cluster **Control Center**.



Services

✓ All your services are connected. You can start deploying. Go to Workers

Type	Instance Count	Status
Data Lake	3/3	Healthy
Data Warehouse	3/3	Healthy
File Storage	3/3	Healthy
RabbitMQ	3/3	Healthy
Redis	3/3	Healthy

Name	Host	Port	Status	Role	Version	Platform	Last Healthy	Last Update	Added By
Redis Server 1	192.168.10.122	6379	Healthy	Replica	8.0.0	Linux	Jan 26, 2026 at 10:07:4...	Jan 26, 2026 at 10:06:...	LOCAL/admin
Redis Server 2	192.168.10.152	6379	Healthy	Replica	8.0.0	Linux	Jan 26, 2026 at 10:07:4...	Jan 26, 2026 at 10:07:1...	LOCAL/admin
Redis Server 3	192.168.10.132	6379	Healthy	Primary	8.0.0	Linux	Jan 26, 2026 at 10:07:4...	Jan 23, 2026 at 2:31:15...	LOCAL/admin

High Availability support for PostgreSQL Data lake

Installation

ⓘ Info

- Replication Manager is compatible solely with Linux-based operating systems.
- The Replication Manager version in use must be compatible with the major version of the installed PostgreSQL.
- All PostgreSQL servers must be of the same version and run on the same operating system.

High availability solution for PostgreSQL data lake requires a single primary server along with a minimum of two standby servers. Both PostgreSQL and Replication Manager must be installed on every server.

ⓘ Warning

On the servers that target to run as standby:

- Do not create a PostgreSQL instance (i.e., do not execute `initdb` or any database creation scripts provided by packages).
- Ensure the destination data directory exists and is owned by the `postgres` system user.

1. Select your Linux distribution here and follow the steps to install PostgreSQL accordingly.
2. Follow the steps to install Replication Manager for PostgreSQL clusters - repmgr.

Primary configuration

1. Choose one of the installed servers to be the primary one.
2. Navigate to the folder containing `postgresql.conf` file and create a replication config file named `postgresql.replication.conf`.

Info

By default, `postgresql.conf` file is placed at

- `/var/lib/pgsql/<version>/data/` on Red Hat/Rocky.
- `/var/lib/postgresql/<version>/main` on Debian/Ubuntu.

bash

```
# Enable replication connections; set this value to at least
one more
# than the number of standbys which will connect to this
server
# (note that repmgr will execute "pg_basebackup" in WAL
streaming mode,
# which requires two free WAL senders).
#
# See: https://www.postgresql.org/docs/current/runtime-config-
replication.html#GUC-MAX-WAL-SENDERS

max_wal_senders = 10

# If using replication slots, set this value to at least one
more
# than the number of standbys which will connect to this
server.
# Note that repmgr will only make use of replication slots if
# "use_replication_slots" is set to "true" in "repmgr.conf".
# (If you are not intending to use replication slots, this
value
# can be set to "0").
#
# See: https://www.postgresql.org/docs/current/runtime-config-
replication.html#GUC-MAX-REPLICATION-SLOTS

max_replication_slots = 10

# Ensure WAL files contain enough information to enable read-
only queries
# on the standby.
#
# See: https://www.postgresql.org/docs/current/runtime-config-
wal.html#GUC-WAL-LEVEL

wal_level = 'hot_standby'

# Enable read-only queries on a standby
#
# See: https://www.postgresql.org/docs/current/runtime-config-
replication.html#GUC-HOT-STANDBY

hot_standby = on

# Enable WAL file archiving
#
# See: https://www.postgresql.org/docs/current/runtime-config-
wal.html#GUC-ARCHIVE-MODE
```

```
archive_mode = on

# Set archive command to a dummy command; this can later be
changed without
# needing to restart the PostgreSQL instance.
#
# See: https://www.postgresql.org/docs/current/runtime-config-
wal.html#GUC-ARCHIVE-COMMAND

archive_command = '/bin/true'

# This config should be added if you plan to use repmgrd for
# automatic failover
# See: https://www.repmgr.org/docs/current/repmgrd-basic-
configuration.html
shared_preload_libraries = 'repmgr'

wal_log_hints = on # for pg_rewind when rejoin
```

3. Add the replication configuration file name to the end of `postgresql.conf` file and save the modifications.

bash

```
...
include 'postgresql.replication.conf'
```

4. In Terminal, run the following commands to create `repmgr` user and database.

bash

```
$ createuser -s repmgr
$ createdb repmgr -O repmgr
```

Info

In this guideline, although the term `repmgr` is used for both user and database, any names can be used.

5. Edit `pg_hba.conf` file to configure the authentication.

bash

```
# Ensure the repmgr user has appropriate permissions in
pg_hba.conf
# and can connect in replication mode
# pg_hba.conf should contain entries similar to the following:
# Uncomment this if you want to access Postgresql database via
pgadmin with user "postgres":
#host    all             postgres        0.0.0.0/0
scram-sha-256

local   replication   repmgr
trust
host    replication   repmgr      127.0.0.1/32
trust
#or
host    replication   repmgr      0.0.0.0/0
trust

local   repmgr        repmgr
trust
host    repmgr        repmgr      127.0.0.1/32
trust
#or
host    repmgr        repmgr      0.0.0.0/0
trust
```

6. Restart PostgreSQL server.

bash

```
$ cd /path/to/pg_ctl
$ pg_ctl -D <postgresql_data_dir> restart
```

7. Create `repmgr.conf` file, fill out information in brackets and store it in a location of your choice.

> **Warning**
>
> `repmgr.conf` file should not be placed inside PostgreSQL data folder as it may be overwritten.

bash

```
node_id=<any_node_id>
node_name=<any_node_name>
# connection info of the current node
conninfo='host=<host_address_of_node> user=repmgr
dbname=repmgr connect_timeout=2'
data_directory='<postgres_data_dir>'
failover='automatic' # for repmgrd (automatic failover)
promote_command='<postgres_dir>/repmgr standby promote -f "
<your_dir>/repmgr.conf" --log-level INFO'
follow_command='<postgres_dir>/repmgr standby follow -f "
<your_dir>/repmgr.conf" -W --log-level INFO'
reconnect_attempts='5'
reconnect_interval='1'
monitor_interval_secs='1'
pg_bindir='<postgres_dir>'
# enable this so that repmgr only vote new primary
# when none of the standbys can connect to current primary
primary_visibility_consensus=true
```

Key	Red Hat/Rocky	Debian/Ubuntu
`postgres_data_dir`	`/var/lib/pgsql/<version>/data/`	`/var/lib/postgresql/<version>/main/`
`postgres_dir`	`/usr/pgsql-<version>/bin/`	`/usr/lib/postgresql/<version>/bin/`
`your_dir`	Directory to `repmgr.conf` file.	Directory to `repmgr.conf` file.

8. In Terminal, run the following commands to register the primary server.

bash

```
$ cd path/to/repmgr
$ repmgr -f <repmgr_config_file_path> primary register

INFO: connecting to primary database...
NOTICE: attempting to install extension "repmgr"
NOTICE: "repmgr" extension successfully installed
NOTICE: primary node record (id: 1) registered
```

Standby configuration

1. Create `repmgr.conf` file and modify values of `node`, `node_name`, `conninfo` accordingly.

2. Store the file in your reference location.
3. Stop PostgreSQL server.

bash

```
$ cd /path/to/pg_ctl
$ pg_ctl -D <postgresql_data_dir> stop
```

4. In Terminal, run the following commands to clone data from the primary server.

bash

```
$ cd path/to/repmgr
$ repmgr -h <primary_server_host> \
         -U repmgr -d repmgr \ # primary repmgr <user> and
<database>
         -f <standby_repmgr_config_file_path> \
         -c \ # fast checkpoint to speed up process
         standby clone \
         --dry-run # dry run to check if the primary can be
cloned

$ repmgr -h <primary_server_host> \
         -U repmgr -d repmgr \ # primary repmgr <user> and
<database>
         -f <standby_repmgr_config_file_path> \
         -c \ # fast checkpoint to speed up process
         standby clone
```

5. Start PostgreSQL server.

bash

```
$ cd /path/to/pg_ctl
$ pg_ctl -D <postgresql_data_dir> start
```

6. In Terminal, run the following commands to register the standby server.

bash

```
$ cd /path/to/repmgr
$ repmgr -f <standby_repmgr_config_file_path> \
           standby register
```

7. Check if the node was registered successfully.

bash

```
$ cd /path/to/repmgr
$ repmgr -f /etc/repmgr.conf cluster show
```

Automatic failover

In Terminal, run the following command to start Replication manager daemon on all PostgreSQL servers (including primary and standbys)

bash

```
$ cd /path/to/repmgr
$ repmgrd -f <repmgr_config_file_path>
```

Rejoin after a failure

> **Info**
>
> Replication manager daemon `repmgrd` does not automatically join a failed PostgreSQL server node to the cluster. Consequently, the cluster contains at least two primary nodes at one time, and the system administrator has to join the node to the cluster manually.

1. Ensure the failed PostgreSQL server is not running. Run the following command in Terminal to stop the server if it has, by chance, already been started by the Linux system and service manager.

bash

```
$ cd /path/to/pg_ctl
$ pg_ctl -D <postgresql_data_dir> stop
```

2. In Terminal, run the following command to rejoin the server.

bash

```
$ cd /path/to/repmgr
$ repmgr -f <repmgr_config_file> node rejoin \
         --force-rewind \ # use pg_rewind to help with diverge
timeline
         -d 'host=<current_primary>  dbname=repmgr
user=repmgr'
```

3. If the server rejoin fails, do register it as a standby. In Terminal, run the following command.

bash

```
$ cd /path/to/repmgr
$ repmgr -h <current_primary_server_host> \
         -U repmgr -d repmgr \ # primary repmgr <user> and
<database>
         -f <standby_repmgr_config_file_path> \
          -c \ # fast checkpoint to speed up process
          -F \ # this overwritten the the data folder if it
was created
          standby clone \
```

4. Start PostgreSQL server.

bash

```
$ cd /path/to/pg_ctl
$ pg_ctl -D <postgresql_data_dir> start
```

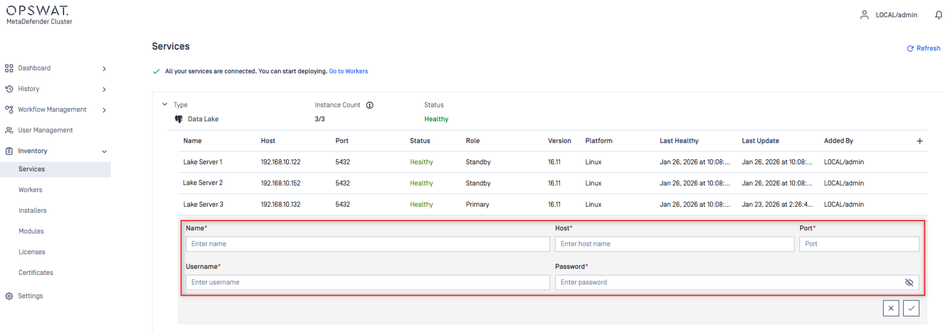
5. Force register the server as a standby.

bash

```
$ cd /path/to/repmgr
$ repmgr -f <standby_repmgr_config_file_path> \
         -F \ # forcefully overwrite an existing node record
or user --force
         standby register
```

Setup instructions

1. Sign to MD Cluster **Control Center** console with your Administrator account.
2. Navigate to `Inventory` > `Services`.
3. Expand the `Data Lake` group.
4. Click `Add service`.
5. Enter the values for `Name`, `Host`, `Port`, `Username` and `Password` fields of individual PostgreSQL instance.
6. Click the Check icon in the bottom right to complete.



Services

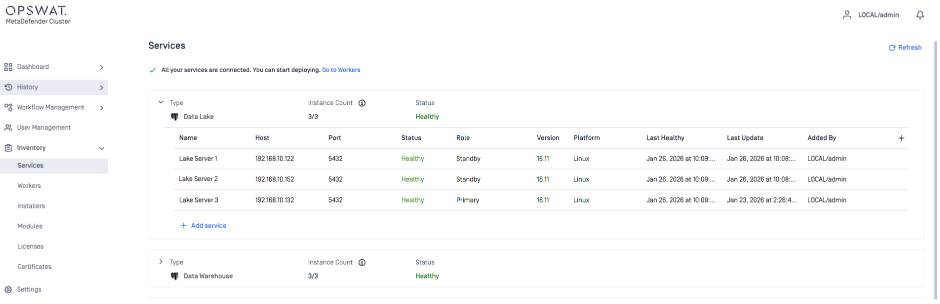
All your services are connected. You can start deploying.

Type	Instance Count	Status
Data Lake	3/3	Healthy

Name	Host	Port	Status	Role	Version	Platform	Last Healthy	Last Update	Added By
Lake Server 1	192.168.10.122	5432	Healthy	Standby	16.11	Linux	Jan 26, 2026 at 10:08:...	Jan 26, 2026 at 10:08:...	LOCAL/admin
Lake Server 2	192.168.10.152	5432	Healthy	Standby	16.11	Linux	Jan 26, 2026 at 10:08:...	Jan 26, 2026 at 10:08:...	LOCAL/admin
Lake Server 3	192.168.10.132	5432	Healthy	Primary	16.11	Linux	Jan 26, 2026 at 10:08:...	Jan 23, 2026 at 2:26:4...	LOCAL/admin

Name* Enter name
Host* Enter host name
Port* Port
Username* Enter username
Password* Enter password

7. Ensure all PostgreSQL instances are reachable by the MD Cluster **Control Center.**



Services

All your services are connected. You can start deploying.

Type	Instance Count	Status
Data Lake	3/3	Healthy

Name	Host	Port	Status	Role	Version	Platform	Last Healthy	Last Update	Added By
Lake Server 1	192.168.10.122	5432	Healthy	Standby	16.11	Linux	Jan 26, 2026 at 10:09:...	Jan 26, 2026 at 10:08:...	LOCAL/admin
Lake Server 2	192.168.10.152	5432	Healthy	Standby	16.11	Linux	Jan 26, 2026 at 10:09:...	Jan 26, 2026 at 10:08:...	LOCAL/admin
Lake Server 3	192.168.10.132	5432	Healthy	Primary	16.11	Linux	Jan 26, 2026 at 10:09:...	Jan 23, 2026 at 2:26:4...	LOCAL/admin

Type	Instance Count	Status
Data Warehouse	3/3	Healthy

System settings

This section shows MetaDefender Cluster settings.

Data Retention

You can find this feature under `Settings` > `Data Retention`.

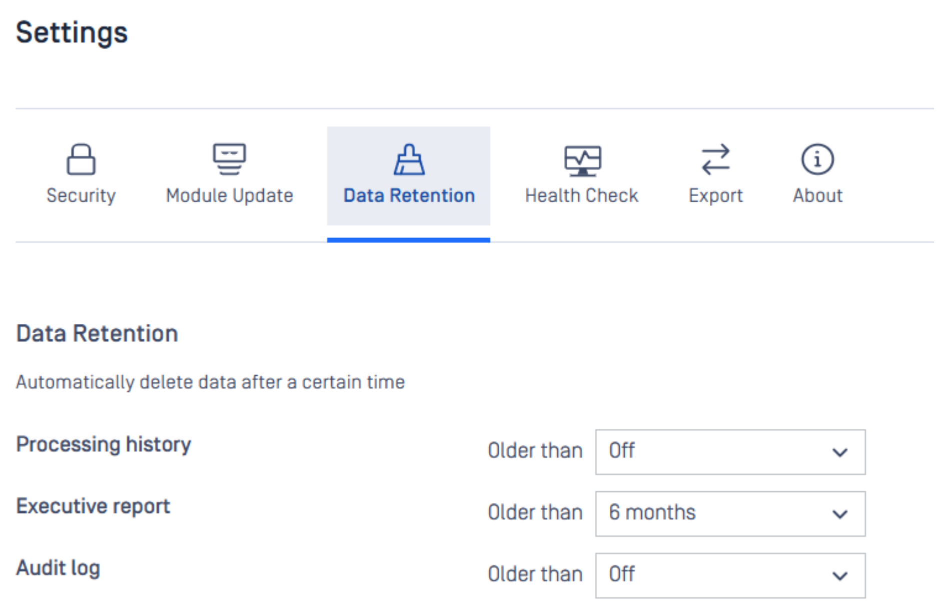
This setting enables users to define the retention period for specific data types, helping optimize system storage and maintain efficiency.

Available Data Categories

1. **Processing History:** History of scan results.
2. **Executive Report:** Statistics data.
3. **Audit Log:** Detailed logs of user actions and system events.

In case you do not want to enable automatic clean up, set the value to off. This will prevent automatic removal.

> **Warning**
>
> Disabling automatic clean-up may lead to data accumulation, which can affect system performance and increase storage costs.



Data Retention

Automatically delete data after a certain time

Processing history — Older than: Off

Executive report — Older than: 6 months

Audit log — Older than: Off

Remote Support Package Gathering

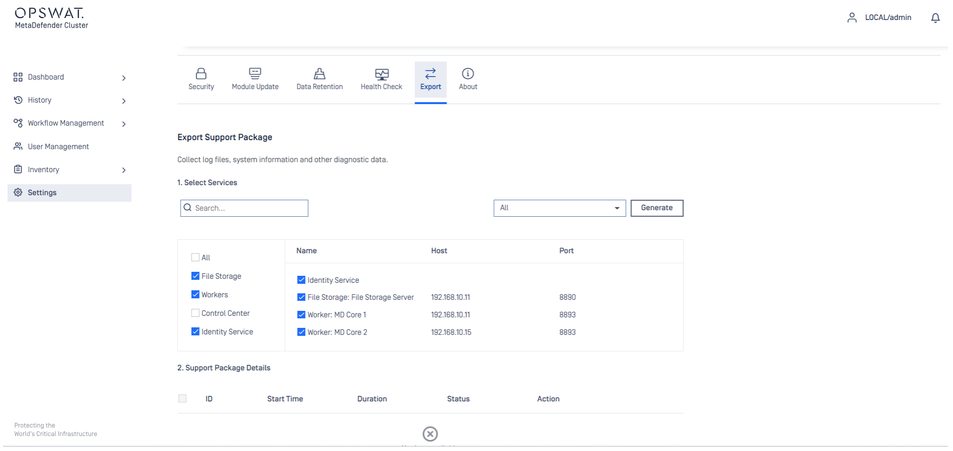
The support package contains log files and is essential for OPSWAT to troubleshoot issues. Since version **2.2.0**, it is now possible to gather a support package remotely via the web console of the MetaDefender Cluster Control Center.

Info

Ensure that all MD Cluster services are upgraded to version 2.2.0 or higher to fully support this feature.

Remote support package gathering steps:

1. Go to `Settings` > `Export` .
2. Select which MetaDefender Cluster services need to generate support package, then select `Generate`



Export Support Package

Collect log files, system information and other diagnostic data.

1. Select Services

☐ All
☑ File Storage
☑ Workers
☐ Control Center
☑ Identity Service

Name	Host	Port
☑ Identity Service		
☑ File Storage: File Storage Server	192.168.10.11	8890
☑ Worker: MD Core 1	192.168.10.11	8893
☑ Worker: MD Core 2	192.168.10.15	8893

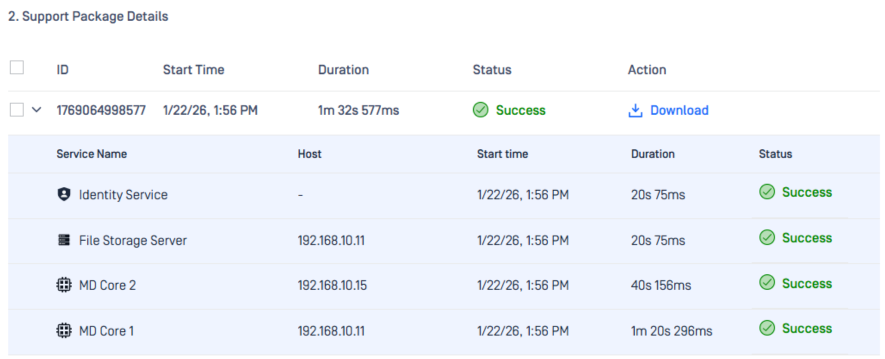
2. Support Package Details

☐	ID	Start Time	Duration	Status	Action

Info

Select the MetaDefender Cluster Worker service to generate a support package for its deployed instance as well.

3. Wait for the generation process to complete successfully. Once it is done, the download button will appear, and the support packages will be ready for download.



2. Support Package Details

☐	ID	Start Time	Duration	Status	Action
☐ ⌄	1769064998577	1/22/26, 1:56 PM	1m 32s 577ms	✓ Success	Download

Service Name	Host	Start time	Duration	Status
Identity Service	-	1/22/26, 1:56 PM	20s 75ms	✓ Success
File Storage Server	192.168.10.11	1/22/26, 1:56 PM	20s 75ms	✓ Success
MD Core 2	192.168.10.15	1/22/26, 1:56 PM	40s 156ms	✓ Success
MD Core 1	192.168.10.11	1/22/26, 1:56 PM	1m 20s 296ms	✓ Success

Info

The size of the support package may vary depending on log size and the number of days for collection. If disk space is insufficient, certain log files may be excluded from the support package.

Warning

All support package files will be downloaded to the MetaDefender Cluster Control Center. Please monitor the disk space on the host running this service when using this functionality, as the size of log files can be very large.

4. Click `Download`

Info

Some services may fail due to connection issues or insufficient disk space. In such cases, only the successfully generated support packages will be available for download. Users can view detailed error information if failures occur.

Security

Setup HTTPS

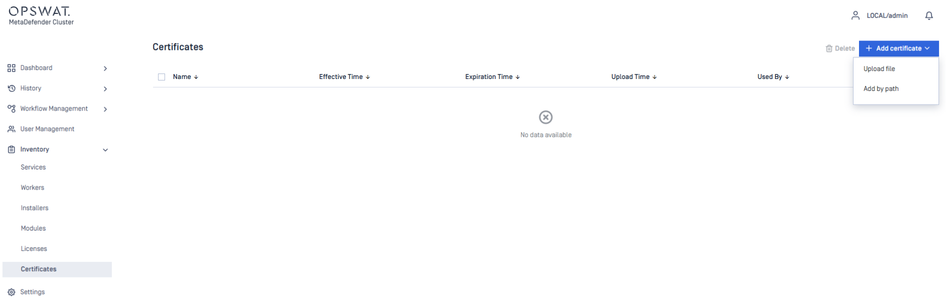
Transport Layer Security (TLS) is a cryptographic protocol that provides communications security over a computer network. Websites, like the Web Management Console, are able to use TLS to secure all communications between their servers and web browsers.

The TLS protocol aims primarily to provide confidentiality (privacy) and data integrity between two communicating computer applications.

> **Info**
>
> HTTPS is not enabled by default. As a consequence sessions between the wizard's backend and the browser may be insecure.

Steps to setup this feature:

1. Go to `Inventory` > `Certificates`
2. Click `Add certificate`
   a. To add a certificate using a file path, choose `Add by path` and enter the location of both the certificate and its corresponding private key file.
   b. To upload certificate file, select `Upload file`.



Certificate YML sample file:

yaml

```
---
private_key: |
  -----BEGIN PRIVATE KEY-----

MIIJQwIBADANBgkqhkiG9w0BAQEFAASCCS0wggkpAgEAAoICAQCjYtuWaICCY0tJ

PubxpIgIL+WWmz/fmK8IQr11Wtee6/IUyUlo5I602mq1qcLhT/kmpoR8Di3DAmHK

nSWdPWtn1BtXLErLlUiHgZDrZWInmEBjKM1DZf+CvNGZ+EzPgBv5nTekLWcfI5ZZ

toGuIP1Dl/IkNDw8zFz4cpiMe/BFGemyxdHhLrKHSm8Eo+nT734tItnHKT/m6DSU

0xlZ13d6ehLRm7/+Nx47M3XMTRH5qKP/7TTE2s0U6+M0tsGI2zpRi+m6jzhNyMBT

J1u58qAe3ZW5/+YAiuZYAB6n5bhUp4oFuB5wYbcBywVR8ujInpF8buWQUjy5N8pS

Np7szdYsnLJpvAd0sibrNPjC0FQCNrpNjgJmIK3+mKk4kXX7ZTwefoAzTK4l2pHN

uC53QVc/EF++GBLAxmvCDq9ZpMIYi7OmzkkAKKC9Ue6Ef217LFQCFIBKIzv9cgi9

fwPMLhrKleoVRNsecBsCP569WgJXhUnwf2lon4fEZr3+vRuc9shfqnV0nPN1IMSn

zXCast7I2fiuRXdIz96KjlGQpP4XfNVA+RGL7aMnWOFIaVrKWLzAtgzoGMTvP/Au

ehKXncBJhYtW0ltTioVx+5yTYSAZWl+IssmXjefxJqYi2/7QWmv1QC9psNcjTMaB

QLN03T1Qelbs7Y27sxdEnNUth4kI+wIDAQABAoICAFWe8MQZb37k2gdAV3Y6aq8f

qokKQqbCNLd3giGFwYkezHXoJfg6Di7oZxNcKyw35LFEghkgtQqErQqo35VPIoH+

vXUpWOjnCmM4muFA9/cX6mYMc8TmJsg0ewLdBCOZVw+wPABlaqz+0UOiSMMftpk9

fz9JwGd8ERyBsT+tk3Qi6D0vPZVsC1KqxxL/cwIFd3Hf2ZBtJXe0KBn1pktWht5A
```

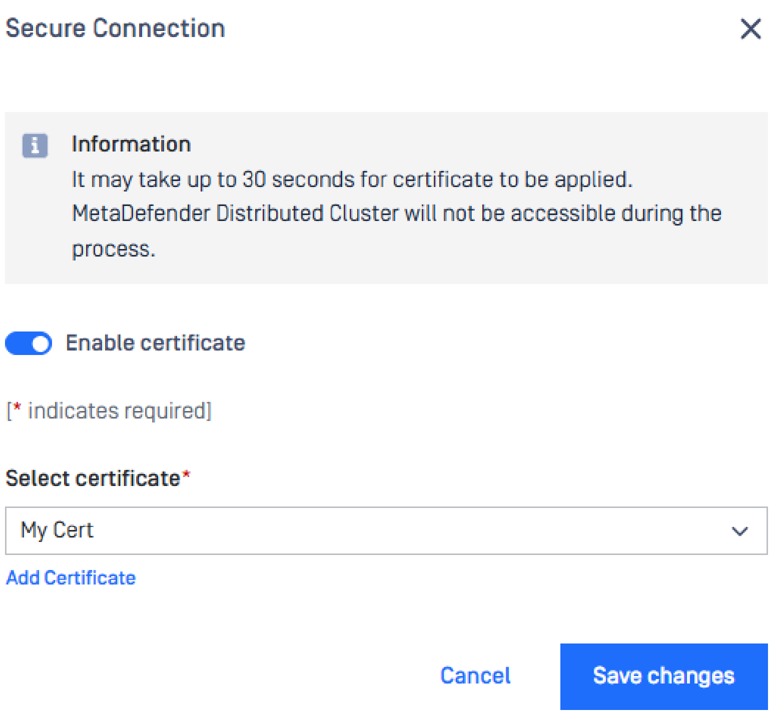
```
Kqx9mld2Ovl7NjgiC1Fx9r+fZw/iOabFFwQA4dr+R8mEMK/7bd4VXfQ1o/QGGb
MT

G+ulFrsiDyP+rBIAaGC0i7gDjLAIBQeDhP409ZhswIEc/GBtODU372a2CQK/u4
Q/

HBQvuBtKFNkGUooLgCCbFxzgNUGc83GB/6IwbEM7R5uXqsFiE71LpmroDyjKTl
Q8

YZkpIcLNVLw0usoGYHFm2rvCyEVlfsE3Ub8cFyTFk50SeOcF2QL2xzKmmbZEpX
gl

xBHR0hjgon0IKJDGfor4bHO7Nt+1Ece8u2oTEKvpz5aIn44OeC5mApRGy83/0b
vs

esnWjDE/bGpoT8qFuy+0urDEPNId44XcJm1IRIlG56ErxC3l0s11wrIpTmXXck
qw

zFR9s2z7f0zjeyxqZg4NTPI7wkM3M8BXlvp2GTBIeoxrWB4V3YArwu8QF80QBg
Vz

mgHl24nTg00UH1OjZsABAoIBAQDOxftSDbSqGytcWqPYP3SZHAWDA0O4ACEM+e
Cw

au9ASutl0IDlNDMJ8nC2ph25BMe5hHDWp2cGQJog7pZ/3qQogQho2gUniKDifN
77

40QdykllTzTVROqmP8+efreIvqlzHmuqaGfGs5oTkZaWj5su+B+bT+9rIwZcwf
s5

YRINhQRx17qa++xh5mfE25c+M9fiIBTiNSo4lTxWMBShnK8xrGaMEmN7W0qTMb
FH

PgQz5FcxRjCCqwHilwNBeLDTp/ZECEB7y34khVh531mBE2mNzSVIQcGZP1I/Dv
Xj

W7UUNdgFwii/GW+6M0uUDy23UVQpbFzcV8o1C2nZc4Fb4zwBAoIBAQDKSJkFww
uR

naVJS6WxOKjX8MCu9/cKPnwBv2mmI2jgGxHTw5sr3ahmF5eTb8Zo19BowytN+t
r6

2ZFoIBA9Ubc9esEAU8l3fggdfM82cuR9sGcfQVoCh8tMg6BP8IBLOmbSUhN3PG
2m

39I802u0fFNVQCJKhx1m1MFFLOu7lVcDS9JN+oYVPb6MDfBLm5jOiPuYkFZ4gH
79

J7gXI0/YKhaJ7yXthYVkdrSF6Eooer4RZgma62Dd1VNzSq3JBo6rYjF7Lvd+Rw
DC
```

```
R1thHrmf/IXplxpNVkoMVxtzbrrbgnC25QmvRYc0rlS/kvM4yQhMH3eA7IycDZMp
Y+0xm7I7jTT7AoIBAGKzKIMDXdCxBWKhNYJ8z7hiItNl1IZZMW2TPUiY0rl6yaCh
BVXjM9W0r07QPnHZsUiByqb743adkbTUjmxdJzjaVtxN7ZXwZvOVrY7I7fPWYnCE
fXCr4+IVpZI/ZHZWpGX6CGSgT6EOjCZ5IUufIvEpqVSmtF8MqfXO9o9uIYLokrWQ
x1dBl5UnuTLDqw8bChq7O5y6yfuWaOWvL7nxI8NvSsfj4y635gIa/0dFeBYZEfHI
UlGdNVomwXwYEzgE/c19ruIowX7HU/NgxMWTMZhpazlxgesXybel+YNcfDQ4e3RM
OMz3ZFiaMaJsGGNf4++d9TmMgk4Ns6oDs6Tb9AECggEBAJYzd+SOYo26iBu3nw3L
65uEeh6xou8pXH0Tu4gQrPQTRZZ/nT3iNgOwqu1gRuxcq7TOjt41UdqIKO8vN7/A
aJavCpaKoIMowy/aGCbvAvjNPpU3unU8jdl/t08EXs79S5IKPcgAx87sTTi7KDN5
SYt4tr2uPEe53NTXuSatilG5QCyExIELOuzWAMKzg7CAiIlNS9foWeLyVkBgCQ6S
me/L8ta+mUDy37K6vC34jh9vK9yrwF6X44ItRoOJafCaVfGI+175q/eWcqTX4q+I
G4tKls4sL4mgOJLq+ra50aYMxbcuommctPMXU6CrrYyQpPTHMNVDQy2ttFdsq9iK
TncCggEBAMmt/8yvPflS+xv3kg/ZBvR9JB1In2n3rUCYYD47ReKFqJ03Vmq5C9nY
56s9w7OUO8perBXlJYmKZQhO4293lvxZD2Iq4NcZbVSCMoHAUzhzY3brdgtSIxa2
gGveGAezZ38qKIU26dkz7deECY4vrsRkwhpTW0LGVCpjcQoaKvymAoCmAs8V2oMr
Ziw1YQ9uOUoWwOqm1wZqmVcOXvPIS2gWAs3fQlWjH9hkcQTMsUaXQDOD0aqkSY3E
NqOvbCV1/oUpRi3076khCoAXI1bKSn/AvR3KDP14B5toHI/F5OTSEiGhhHesgR
```

```
rs
  fBrpEY1IATtPq1taBZZogRqI3rOkkPk=
  -----END PRIVATE KEY-----
certificate: |
  -----BEGIN CERTIFICATE-----

MIIF5jCCA86gAwIBAgIJANq50IuwPFKgMA0GCSqGSIb3DQEBCwUAMIGGMQswCQ
YD

VQQGEwJHQjEQMA4GA1UECAwHRXJld2hvbjETMBEGA1UEBwwKQWxsIGFyb3VuZD
Eb

MBkGA1UECgwSbGlid2Vic29ja2V0cy10ZXN0MRIwEAYDVQQDDAlsb2NhbGhvc3
Qx

HzAdBgkqhkiG9w0BCQEWEG5vbmVAaW52YWxpZC5vcmcwIBcNMTgwMzIwMDQxNj
A3

WhgPMjExODAyMjQwNDE2MDdaMIGGMQswCQYDVQQGEwJHQjEQMA4GA1UECAwHRX
Jl

d2hvbjETMBEGA1UEBwwKQWxsIGFyb3VuZDEbMBkGA1UECgwSbGlid2Vic29ja2
V0

cy10ZXN0MRIwEAYDVQQDDAlsb2NhbGhvc3QxHzAdBgkqhkiG9w0BCQEWEG5vbm
VA

aW52YWxpZC5vcmcwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQCjYt
uW

aICCY0tJPubxpIgIL+WWmz/fmK8IQr11Wtee6/IUyUlo5I602mq1qcLhT/kmpo
R8

Di3DAmHKnSWdPWtn1BtXLErLlUiHgZDrZWInmEBjKM1DZf+CvNGZ+EzPgBv5nT
ek

LWcfI5ZZtoGuIP1Dl/IkNDw8zFz4cpiMe/BFGemyxdHhLrKHSm8Eo+nT734tIt
nH

KT/m6DSU0xlZ13d6ehLRm7/+Nx47M3XMTRH5qKP/7TTE2s0U6+M0tsGI2zpRi+
m6

jzhNyMBTJ1u58qAe3ZW5/+YAiuZYAB6n5bhUp4oFuB5wYbcBywVR8ujInpF8bu
WQ

Ujy5N8pSNp7szdYsnLJpvAd0sibrNPjC0FQCNrpNjgJmIK3+mKk4kXX7ZTwefo
Az

TK4l2pHNuC53QVc/EF++GBLAxmvCDq9ZpMIYi7OmzkkAKKC9Ue6Ef217LFQCFI
BK
```

```
Izv9cgi9fwPMLhrKleoVRNsecBsCP569WgJXhUnwf2lon4fEZr3+vRuc9shfqn
V0

nPN1IMSnzXCast7I2fiuRXdIz96KjlGQpP4XfNVA+RGL7aMnWOFIaVrKWLzAtg
zo

GMTvP/AuehKXncBJhYtW0ltTioVx+5yTYSAZWl+IssmXjefxJqYi2/7QWmv1QC
9p

sNcjTMaBQLN03T1Qelbs7Y27sxdEnNUth4kI+wIDAQABo1MwUTAdBgNVHQ4EFg
QU

9mYU23tW2zsomkKTAXarjr2vjuswHwYDVR0jBBgwFoAU9mYU23tW2zsomkKTAX
ar

jr2vjuswDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAgEANjIBMr
ow

YNCbhAJdP7dhlhT2RUFRdeRUJD0IxrH/hkvb6myHHnK8nOYezFPjUlmRKUgNED
uA

xbnXZzPdCRNV9V2mShbXvCyiDY7WCQE2Bn44z26O0uWVk+7DNNLH9BnkwUtOnM
9P

wtmD9phWexm4q2GnTsiL6Ul6cy0QlTJWKVLEUQQ6yda582e23J1AXqtqFcpfoE
34

H3afEiGy882b+ZBiwkeV+oq6XVF8sFyr9zYrv9CvWTYlkpTQfLTZSsgPdEHYVc
jv

xQ2D+XyDR0aRLRlvxUa9dHGFHLICG34Juq5Ai6lM1EsoD8HSsJpMcmrH7MWw2c
Kk

ujC3rMdFTtte83wF1uuF4FjUC72+SmcQN7A386BC/nk2TTsJawTDzqwOu/VdZv
2g

1WpTHlumlClZeP+G/jkSyDwqNnTu1aodDmUa4xZodfhP1HWPwUKFcq8oQr148Q
YA

AOlbUOJQU7QwRWd1VbnwhDtQWXC92A2w1n/xkZSR1BM/NUSDhkBSUU1WjMbWg6
Gg

mnIZLRerQCu1Oozr87rOQqQakPkyt8BUSNK3K42j2qcfhAONdRl8Hq8Qs5pupy
+s

8sdCGDlwR3JNCMv6u48OK87F4mcIxhkSefFJUFII25pCGN5WtE4p5l+9cnO1Gr
IX
  e2Hl/7M0c/lbZ4FvXgARlex2rkgS0Ka06HE=
  -----END CERTIFICATE-----
```

3. Go to `Settings` > `Security`
4. On the `Secure Connection` section, click `Details`
5. Select `Enable Certificate` , then select your certificate added in step 2.



Secure Connection

Information
It may take up to 30 seconds for certificate to be applied. MetaDefender Distributed Cluster will not be accessible during the process.

Enable certificate: on

[* indicates required]

Select certificate*: My Cert

Warning

Applying HTTPS settings may take some time. During this process, the MetaDefender Cluster Control Center web console will be temporarily unavailable.

Password policies

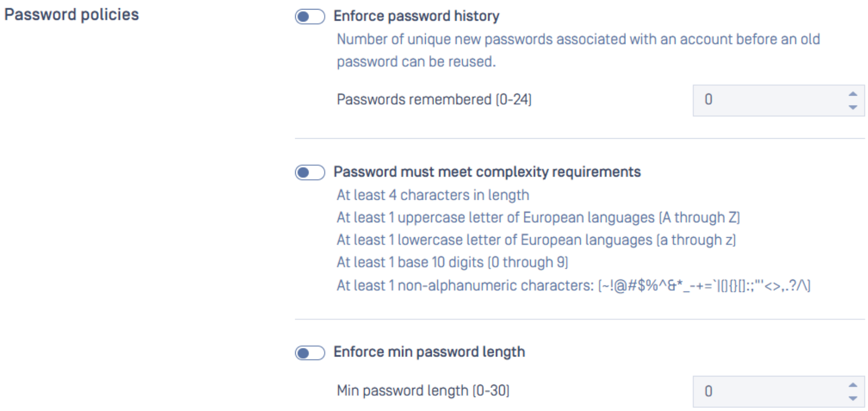
Password Policy settings are accessible under `Settings` > `Security` tab.

Info

These password policies changes only apply to new user creations and future password changes. Existing users' passwords are unaffected.

Local users' password can be enforced to meet requirements set by administrators, which includes following constraints:

- **Enforce password policy:**
    - Determines the number of unique new passwords that must be associated with a user account before an old password can be reused
    - Range: (0-24)
    - Default: 0 (to disable enforcement)
- **Minimum password length:**
    - The least number of characters that can make up a password for a user account
    - Range: (0-30)
    - Default: 0 (to disable enforcement)
- **Password must meet complexity requirements:**
    - Determines whether passwords must meet a series of guidelines that are considered important for a strong password.
    - Default: unchecked



Password policies

Enforce password history

Number of unique new passwords associated with an account before an old password can be reused.

Passwords remembered (0-24) 0

Password must meet complexity requirements

At least 4 characters in length

At least 1 uppercase letter of European languages (A through Z)

At least 1 lowercase letter of European languages (a through z)

At least 1 base 10 digits (0 through 9)

At least 1 non-alphanumeric characters: (~!@#$%^&*_-+=`|(){}[]:;"'<>,.?/\)

Enforce min password length

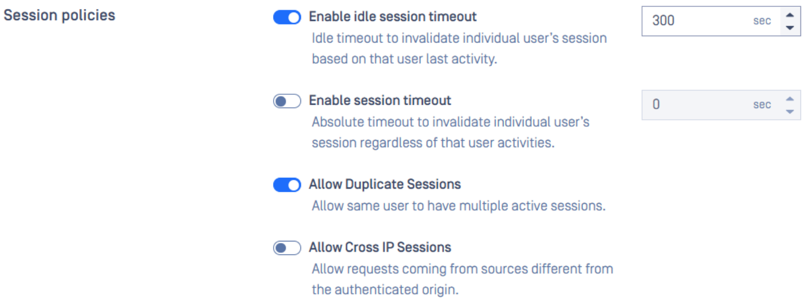
Min password length (0-30) 0

Session policies

Administrators can enforce session policies for local users to ensure compliance with organizational requirements, using the following settings:

- **Enable idle session timeout:**
    - Idle timeout automatically terminates a user's session based on how long since their last recorded activity.
    - Default: 300 seconds.
- **Enable session timeout**
    - Absolute timeout terminates an individual user's session after a fixed duration, regardless of any user activity.
    - Default: 0 (to disable enforcement)
- **Allow Duplicate Sessions**
    - Permit the same user to log in and operate multiple sessions at once.
    - Default: Enabled.

- **Allow Cross IP Sessions**
    - Permit requests from sources other than the authenticated origin.
    - Default: Disabled.



Session policies

● Enable idle session timeout — 300 sec
Idle timeout to invalidate individual user's session based on that user last activity.

○ Enable session timeout — 0 sec
Absolute timeout to invalidate individual user's session regardless of that user activities.

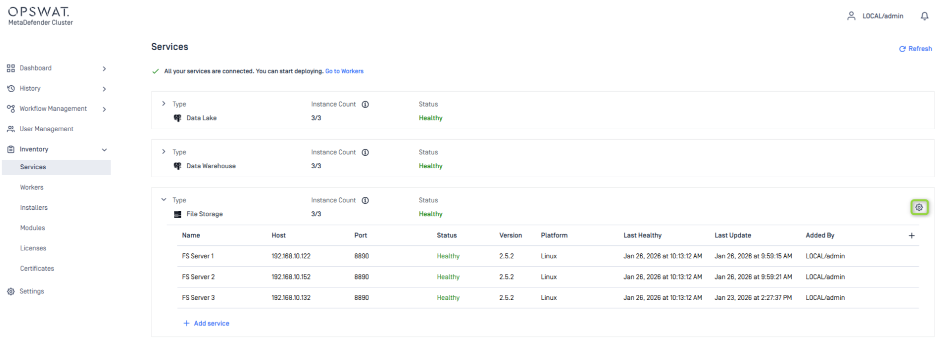
● Allow Duplicate Sessions
Allow same user to have multiple active sessions.

○ Allow Cross IP Sessions
Allow requests coming from sources different from the authenticated origin.

File Storage

MetaDefender Cluster (MD Cluster) introduces a built-in file storage server known as MD Cluster **File Storage**. The server stores and manages the live time of files and their duplications.

The administrator can set up MD Cluster to work with a single instance of MD Cluster **File Storage** or build a group of MD Cluster **File Storage** instances.



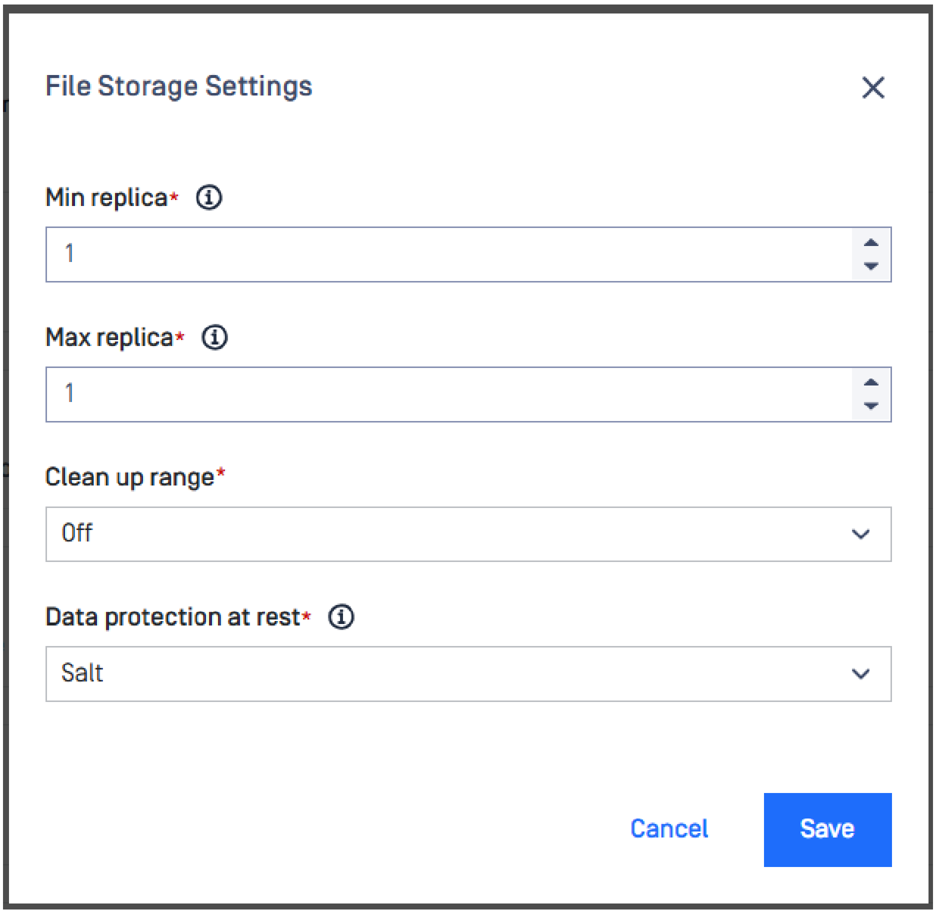
Services

All your services are connected. You can start deploying. Go to Workers

Type	Instance Count	Status
Data Lake	3/3	Healthy
Data Warehouse	3/3	Healthy
File Storage	3/3	Healthy

Name	Host	Port	Status	Version	Platform	Last Healthy	Last Update	Added By
FS Server 1	192.168.10.122	8890	Healthy	2.5.2	Linux	Jan 26, 2026 at 10:13:12 AM	Jan 26, 2026 at 9:59:15 AM	LOCAL/admin
FS Server 2	192.168.10.152	8890	Healthy	2.5.2	Linux	Jan 26, 2026 at 10:13:12 AM	Jan 26, 2026 at 9:59:21 AM	LOCAL/admin
FS Server 3	192.168.10.132	8890	Healthy	2.5.2	Linux	Jan 26, 2026 at 10:13:12 AM	Jan 23, 2026 at 2:27:37 PM	LOCAL/admin

From `Inventory` > `Services` of the MD Cluster **Control Center** web console, the administrator can click on the gear icon in the top left corner of the `File Storage` group to access MD Cluster **File Storage** settings.


File Storage Settings

Min replica* 1

Max replica* 1

Clean up range* Off

Data protection at rest* Salt

Multiple instances

When several instances of MD Cluster **File Storage** are added to the File Storage group, `Min Replica` and `Max Replica` enable the administrator to configure the operation of the storage group, as shown in the table below.

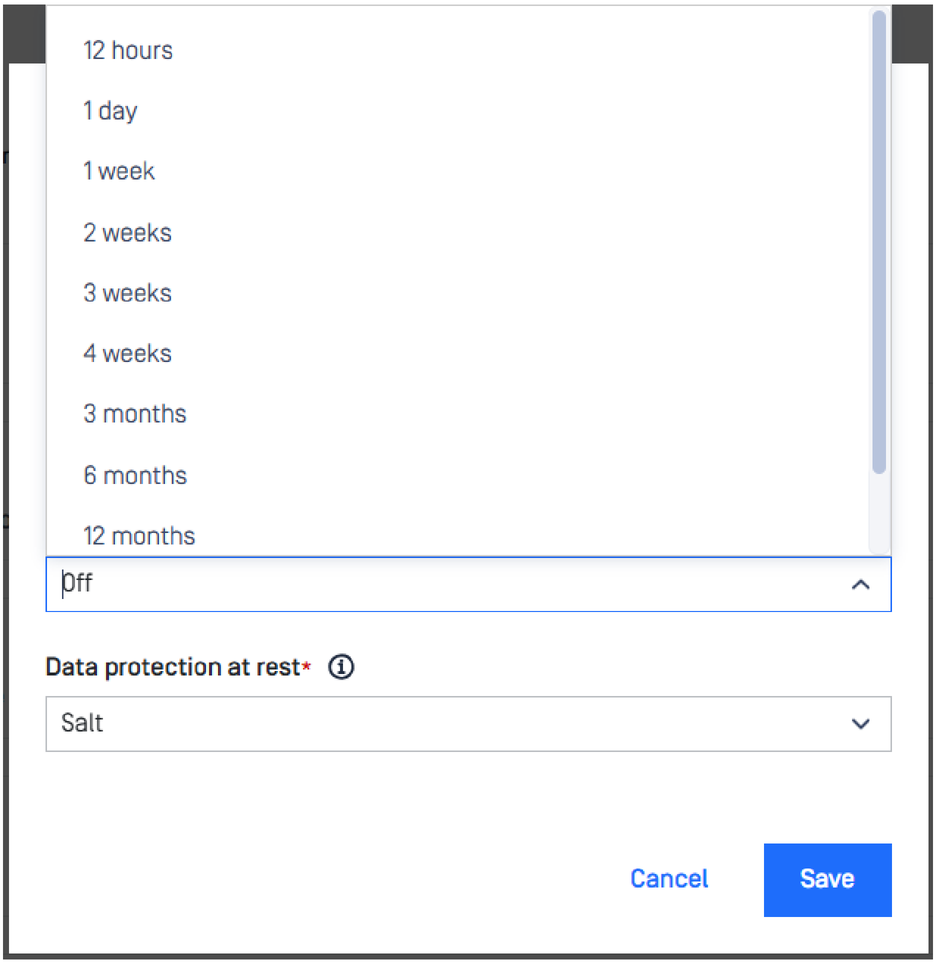
Setting	Behavior
`Min replica` = 1 `Max replica` = 1	Every file is stored without a backup across all File Storage servers. File Storage servers in the group implement a **Sharding** solution for file storage. Since there is no backup for any file, if one server in the group goes down, files managed by that server will be lost to the clients. This setup provides the best performance but also poses a high risk of data loss.
`Min replica` > 1 `Max replica` > `Min replica`	Every file is stored on at least `Min replica` number of File Storage servers and at most `Max-replica` number of servers. The setting provides **High Availability** support for File Storage. In most cases, `Min replica` and `Max replica` are configured to 2 and 3, creating a balance between performance and efficiency in High Availability.
`Min replica` > 1 `Max replica = Min replica`	Every file is fully stored on `Max replica` number of file storage servers and will not succeed if it can not be. This setting is the strictest among three options and should be considered carefully due to its impact on system performance.

Warning

Replication of a file across several MD Cluster File Storage servers significantly impacts the overall system performance. Hence, the number of replications must be evaluated thoughtfully.

Data retention

The administrator can configure data retention for files stored in the File Storage group with the `Clean up range` option. By default, this option is disabled. The administrator has the option to retain files for 12 hours, 1 day, 1 week, etc., starting from the present time.



Data protection at rest*

Salt

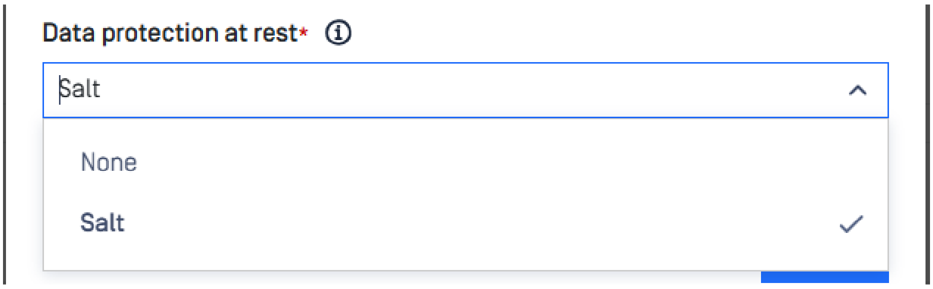
Info

Files eligible for retention include those produced by CDR, DLP, SBOM, Quarantine engines.

Package (.msi, .deb or .rpm) and module files are marked for cleanup manually and will never be affected by data retention.

Data protection

By default, all files stored on the MD Cluster File Storage server are XOR bitwise with a randomly generated binary string. This option is enabled by default to prevent the file from being executed successfully due to unexpected factors. The administrator can disable the option to optimize MD Cluster File Storage performance, though it may expose the system to security risks.



Data protection at rest* ⓘ

Salt

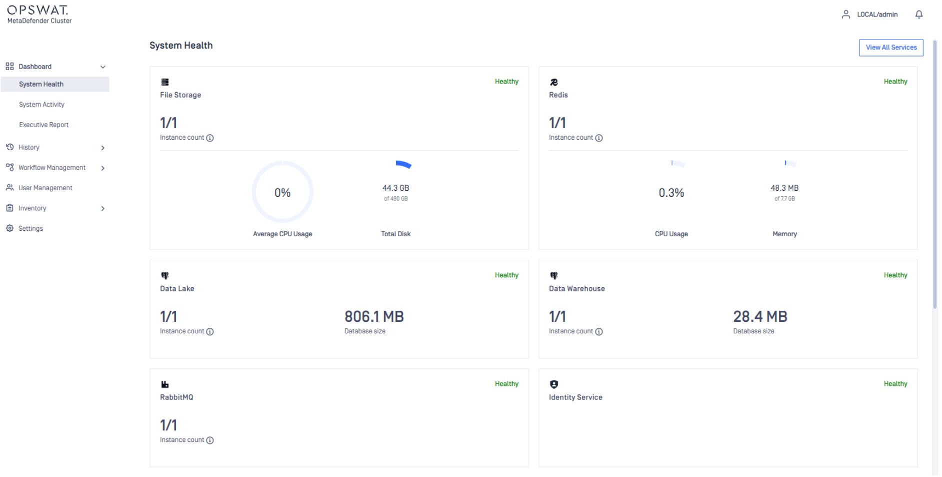
Dashboard

MetaDefender Cluster (MD Cluster) Control Center provides a Web-based user interface at default port 8892 for user to monitor the system health, system activity and executive report.

System Health

System Health show the status of all related services:

- File Storage shows the number of MD Cluster File Storage instances in File Storage cluster added to MD Cluster. The average CPU usage by the instances and the available over total disk are displayed below.
- Redis shows the number of Redis Caching services in MD Cluster. Only the CPU usage and free memory amount of the master Redis node are presented.
- RabbitMQ shows the number of RabbitMQ message brokers in the cluster.
- Data lake and warehouse shows the number of Postgres instances including the primary and its replicas. Database sizes are shown.
- Identity Service shows the health status of the MD Cluster authentication/authorization service.



System Health

File Storage — Healthy
1/1
Instance count
0%
Average CPU Usage
44.3 GB
of 490 GB
Total Disk

Redis — Healthy
1/1
Instance count
0.3%
CPU Usage
48.3 MB
of 7.7 GB
Memory

Data Lake — Healthy
1/1
Instance count
806.1 MB
Database size

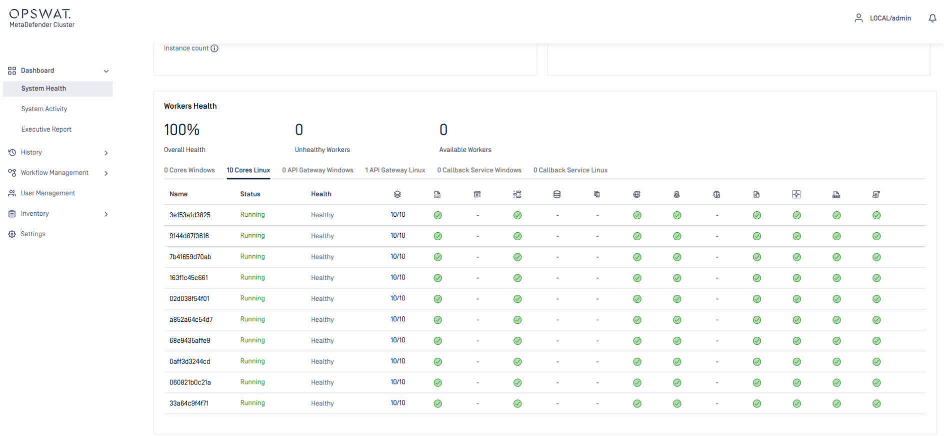
Data Warehouse — Healthy
1/1
Instance count
28.4 MB
Database size

RabbitMQ — Healthy
1/1
Instance count

Identity Service — Healthy

Worker Health

The overall health of all MD Cluster API Gateway and MetaDefender Core instances of platforms are shown here. The administrator can quickly monitor the number of AV engines installed on each MetaDefender Core instance.



Workers Health

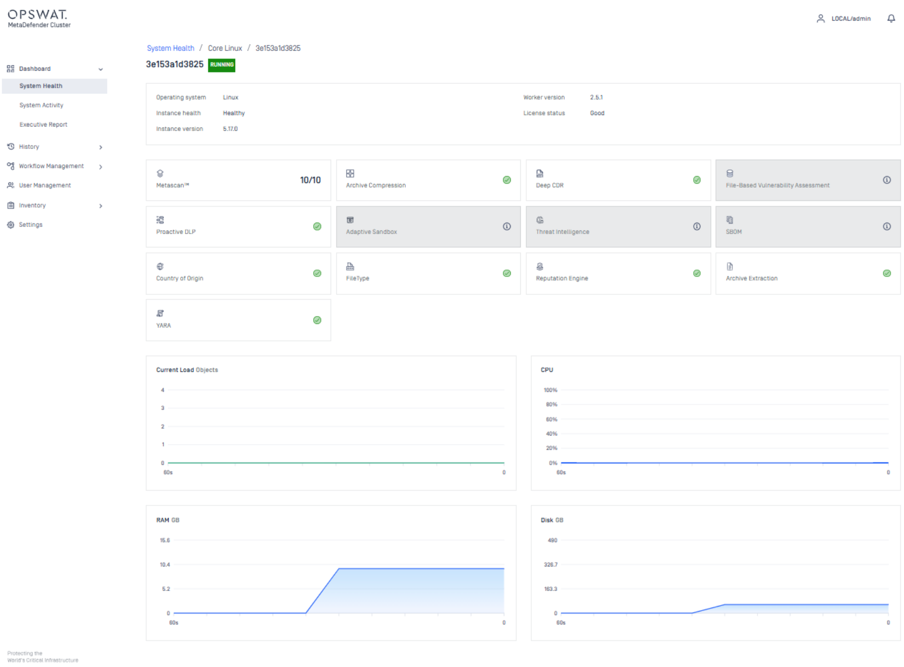
100% Overall Health

0 Unhealthy Workers

0 Available Workers

Name	Status	Health													
3e153a1d3825	Running	Healthy	10/10	✓	-	✓	-	-	✓	✓	-	✓	✓	✓	✓
9144d87f3616	Running	Healthy	10/10	✓	-	✓	-	-	✓	✓	-	✓	✓	✓	✓
7b41659d70ab	Running	Healthy	10/10	✓	-	✓	-	-	✓	✓	-	✓	✓	✓	✓
163f1c45c661	Running	Healthy	10/10	✓	-	✓	-	-	✓	✓	-	✓	✓	✓	✓
02d038f54f01	Running	Healthy	10/10	✓	-	✓	-	-	✓	✓	-	✓	✓	✓	✓
a852a64c54d7	Running	Healthy	10/10	✓	-	✓	-	-	✓	✓	-	✓	✓	✓	✓
68e9435affe9	Running	Healthy	10/10	✓	-	✓	-	-	✓	✓	-	✓	✓	✓	✓
0aff3d3244cd	Running	Healthy	10/10	✓	-	✓	-	-	✓	✓	-	✓	✓	✓	✓
060821b0c21a	Running	Healthy	10/10	✓	-	✓	-	-	✓	✓	-	✓	✓	✓	✓
33a64c9f4f71	Running	Healthy	10/10	✓	-	✓	-	-	✓	✓	-	✓	✓	✓	✓

For more details of engines installed on any MetaDefender Core instance, the administrator can hit on the row of that instance. A new page show.



3e153a1d3825 RUNNING

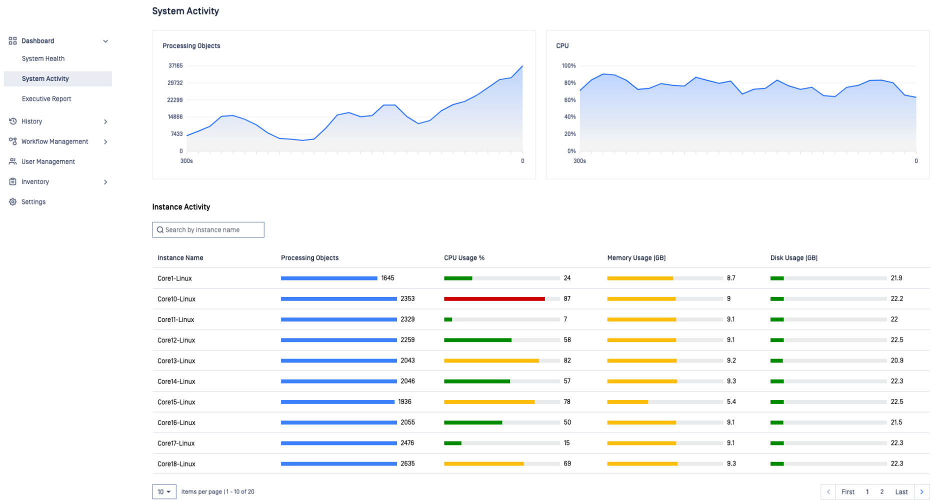
Operating system	Linux	Worker version	2.5.1
Instance health	Healthy	License status	Good
Instance version	5.17.0		

Metascan™ 10/10	Archive Compression ✓	Deep CDR ✓	File-Based Vulnerability Assessment ⓘ
Proactive DLP ✓	Adaptive Sandbox ⓘ	Threat Intelligence ⓘ	SBOM ⓘ
Country of Origin ✓	FileType ✓	Reputation Engine ✓	Archive Extraction ✓
YARA ✓			

System Activity

The overall System Activity of all MetaDefender Core instances are shown here. There are two sections of this page, System Activity and Instance Activity.

- System Activity shows the overall objects currently being processed, objects currently in-queue and average CPU usage of all MetaDefender Core instances.
- Instance Activity shows individual statistics of all MetaDefender Core instances. This includes Processing Objects, CPU Usage, Memory Usage and Disk Usage.

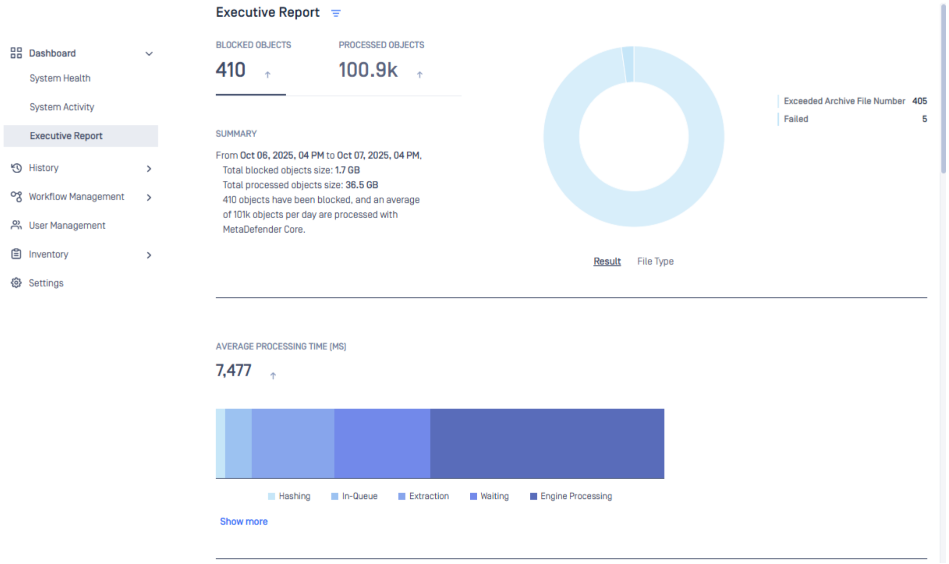


Instance Activity

Instance Name	Processing Objects	CPU Usage %	Memory Usage (GB)	Disk Usage (GB)
Core1-Linux	1645	24	8.7	21.9
Core10-Linux	2353	87	9	22.2
Core11-Linux	2329	7	9.1	22
Core12-Linux	2259	58	9.1	22.5
Core13-Linux	2043	82	9.2	20.9
Core14-Linux	2046	57	9.3	22.3
Core15-Linux	1936	78	5.4	22.5
Core16-Linux	2055	50	9.1	21.5
Core17-Linux	2476	15	9.1	22.3
Core18-Linux	2635	69	9.3	22.3

Executive Report

The Executive Report in MD Cluster provides statistical data on file scanning performance and metrics. This section will provide metrics and values to help you better understand your MetaDefender Distributed Cluster's performance.



BLOCKED OBJECTS 410

PROCESSED OBJECTS 100.9k

SUMMARY

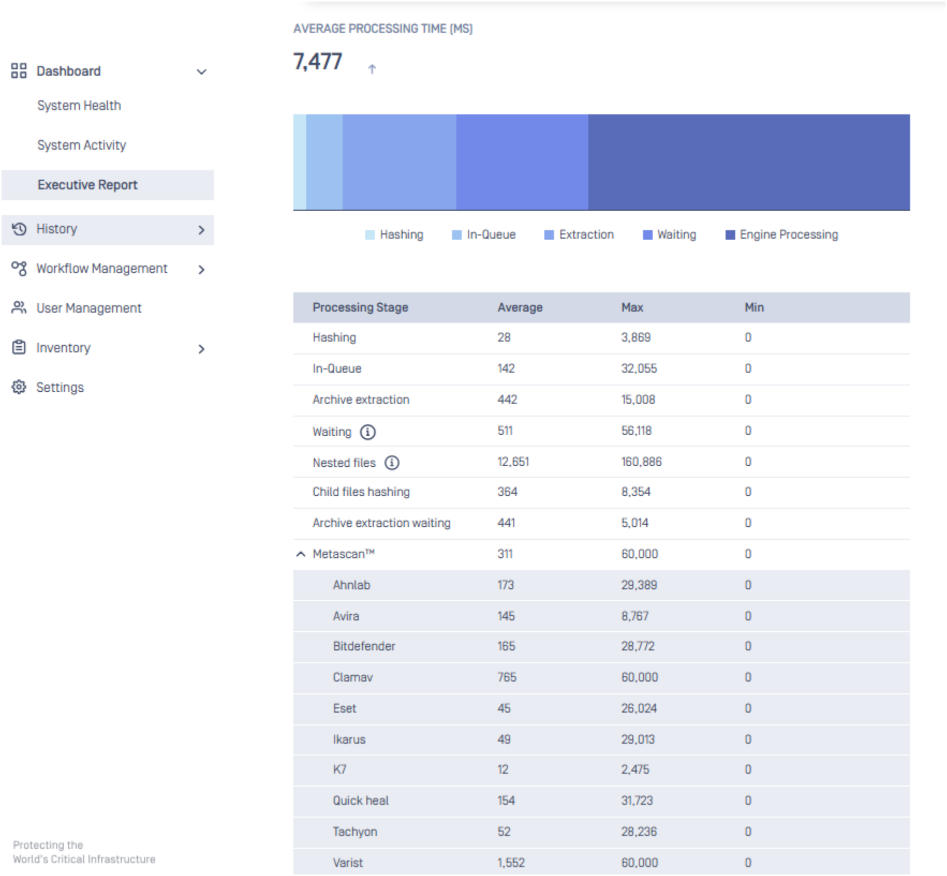
From Oct 06, 2025, 04 PM to Oct 07, 2025, 04 PM,
Total blocked objects size: 1.7 GB
Total processed objects size: 36.5 GB
410 objects have been blocked, and an average of 101k objects per day are processed with MetaDefender Core.

Exceeded Archive File Number	405
Failed	5

AVERAGE PROCESSING TIME (MS)

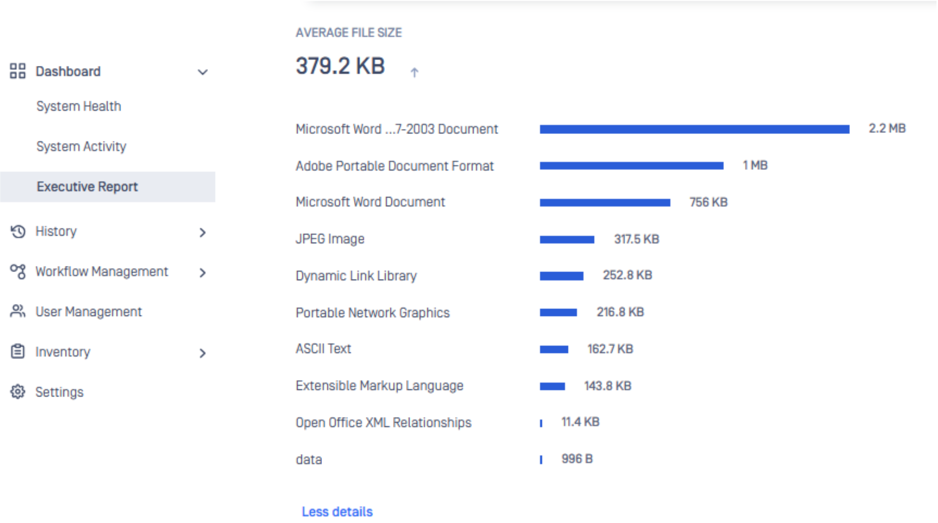
7,477

Average processing time will provide detailed information regarding the Processing Stages. The Average, Min and Max will be shown for each stage.



Processing Stage	Average	Max	Min
Hashing	28	3,869	0
In-Queue	142	32,055	0
Archive extraction	442	15,008	0
Waiting ⓘ	511	56,118	0
Nested files ⓘ	12,651	160,886	0
Child files hashing	364	8,354	0
Archive extraction waiting	441	5,014	0
Metascan™	311	60,000	0
Ahnlab	173	29,389	0
Avira	145	8,767	0
Bitdefender	165	28,772	0
Clamav	765	60,000	0
Eset	45	26,024	0
Ikarus	49	29,013	0
K7	12	2,475	0
Quick heal	154	31,723	0
Tachyon	52	28,236	0
Varist	1,552	60,000	0

Average file size will provide detailed information on the average file size of file types that were commonly scanned.



Microsoft Word ...7-2003 Document	2.2 MB
Adobe Portable Document Format	1 MB
Microsoft Word Document	756 KB
JPEG Image	317.5 KB
Dynamic Link Library	252.8 KB
Portable Network Graphics	216.8 KB
ASCII Text	162.7 KB
Extensible Markup Language	143.8 KB
Open Office XML Relationships	11.4 KB
data	996 B

History

The History section contains detailed views of processing history and audit log history.

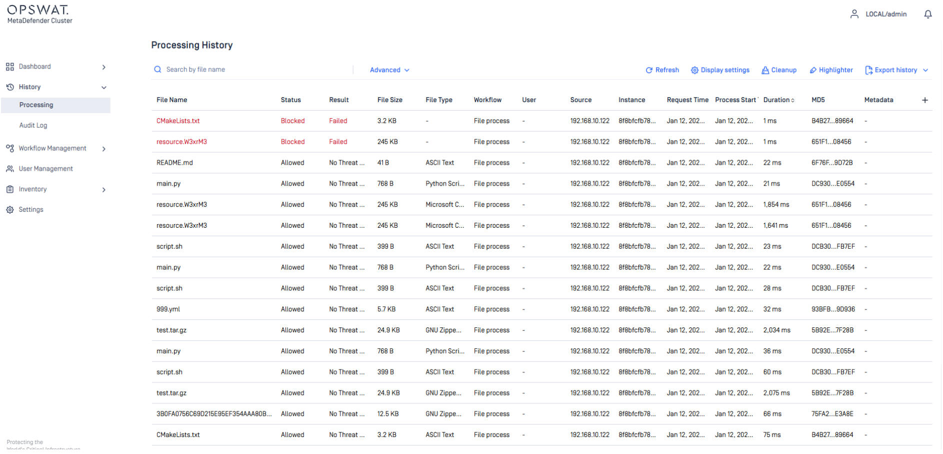
- Processing history.
- Audit log history.

Processing History

Processing history

The Processing History section shows information on all scans made on MetaDefender Cluster (MD Cluster). Search and Filter are also supported against each scan result attribute. The user can search based on the following:

- MD5, SHA1, SHA256, or SHA512 hashes.
- File name (and you can limit search result for a specific scan result, and for specific username who submitted files).
- Source (IP address).
- User name.

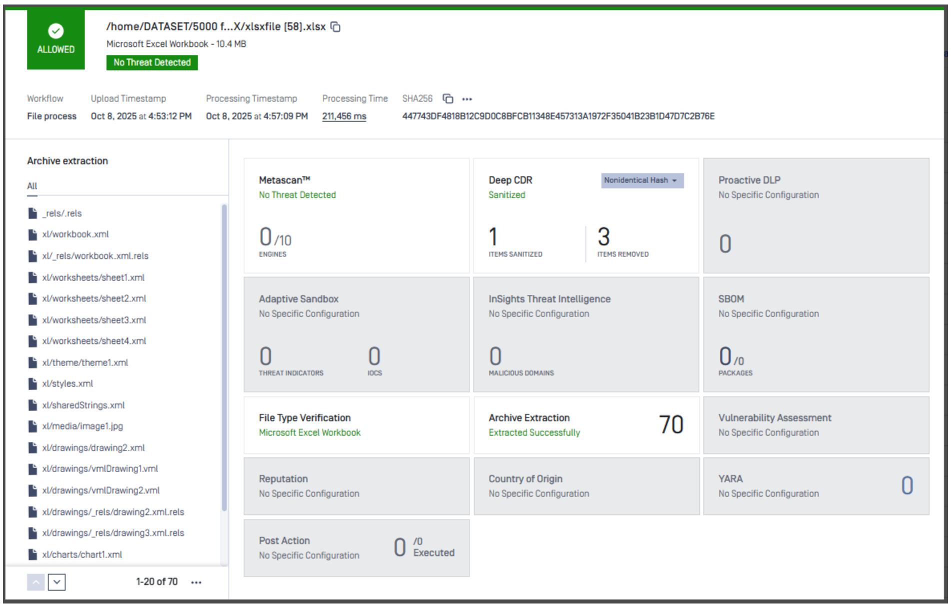


Processing History

File Name	Status	Result	File Size	File Type	Workflow	User	Source	Instance	Request Time	Process Start	Duration	MD5	Metadata
CMakeLists.txt	Blocked	Failed	3.2 KB	-	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	1 ms	B4B27...89664	-
resource.W3xrM3	Blocked	Failed	245 KB	-	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	1 ms	651F1...08456	-
README.md	Allowed	No Threat ...	41 B	ASCII Text	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	22 ms	6F76F...9D72B	-
main.py	Allowed	No Threat ...	768 B	Python Scri...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	21 ms	DC930...E0554	-
resource.W3xrM3	Allowed	No Threat ...	245 KB	Microsoft C...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	1,854 ms	651F1...08456	-
resource.W3xrM3	Allowed	No Threat ...	245 KB	Microsoft C...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	1,641 ms	651F1...08456	-
script.sh	Allowed	No Threat ...	399 B	ASCII Text	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	23 ms	DCB30...FB7EF	-
main.py	Allowed	No Threat ...	768 B	Python Scri...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	22 ms	DC930...E0554	-
script.sh	Allowed	No Threat ...	399 B	ASCII Text	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	28 ms	DCB30...FB7EF	-
999.yml	Allowed	No Threat ...	5.7 KB	ASCII Text	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	32 ms	93BFB...9D936	-
test.tar.gz	Allowed	No Threat ...	24.9 KB	GNU Zippe...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	2,034 ms	5B92E...7F28B	-
main.py	Allowed	No Threat ...	768 B	Python Scri...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	36 ms	DC930...E0554	-
script.sh	Allowed	No Threat ...	399 B	ASCII Text	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	60 ms	DCB30...FB7EF	-
test.tar.gz	Allowed	No Threat ...	24.9 KB	GNU Zippe...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	2,075 ms	5B92E...7F28B	-
3B0FA0756C69D215E95EF354AAA80B...	Allowed	No Threat ...	12.5 KB	GNU Zippe...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	66 ms	75FA2...E3A8E	-
CMakeLists.txt	Allowed	No Threat ...	3.2 KB	ASCII Text	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	75 ms	B4B27...89664	-

Scan result

Selecting a file in the Processing History will show the Scan Result of the file. Here the user will be able to see each of the Modules results.



ALLOWED

/home/DATASET/5000 f...X/xlsxfile [58].xlsx

Microsoft Excel Workbook - 10.4 MB

No Threat Detected

Workflow	Upload Timestamp	Processing Timestamp	Processing Time	SHA256
File process	Oct 8, 2025 at 4:53:12 PM	Oct 8, 2025 at 4:57:09 PM	211,456 ms	447743DF4818B12C9D0C8BFCB11348E457313A1972F35041B23B1D47D7C2B76E

Archive extraction

All

- _rels/.rels
- xl/workbook.xml
- xl/_rels/workbook.xml.rels
- xl/worksheets/sheet1.xml
- xl/worksheets/sheet2.xml
- xl/worksheets/sheet3.xml
- xl/worksheets/sheet4.xml
- xl/theme/theme1.xml
- xl/styles.xml
- xl/sharedStrings.xml
- xl/media/image1.jpg
- xl/drawings/drawing2.xml
- xl/drawings/vmlDrawing1.vml
- xl/drawings/vmlDrawing2.vml
- xl/drawings/_rels/drawing2.xml.rels
- xl/drawings/_rels/drawing3.xml.rels
- xl/charts/chart1.xml

1-20 of 70

Metascan™
No Threat Detected
0 /10 ENGINES

Deep CDR
Nonidentical Hash
Sanitized
1 ITEMS SANITIZED
3 ITEMS REMOVED

Proactive DLP
No Specific Configuration
0

Adaptive Sandbox
No Specific Configuration
0 THREAT INDICATORS
0 IOCS

InSights Threat Intelligence
No Specific Configuration
0 MALICIOUS DOMAINS

SBOM
No Specific Configuration
0 /0 PACKAGES

File Type Verification
Microsoft Excel Workbook

Archive Extraction
Extracted Successfully
70

Vulnerability Assessment
No Specific Configuration

Reputation
No Specific Configuration

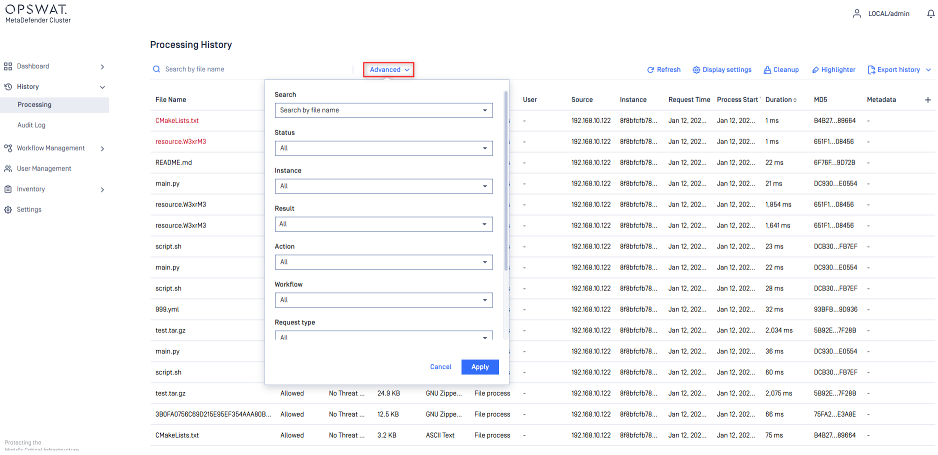
Country of Origin
No Specific Configuration

YARA
No Specific Configuration
0

Post Action
No Specific Configuration
0 /0 Executed

Filter

In Advanced Settings, there are multiple filtering options, such as the scan status, the instance that handled the file, the action taken on the file, the workflow used for processing the file, the date and time of the process, and more. Users can easily combine these to find their desired results.



Processing History

Search by file name

Search: Search by file name

Status: All

Instance: All

Result: All

Action: All

Workflow: All

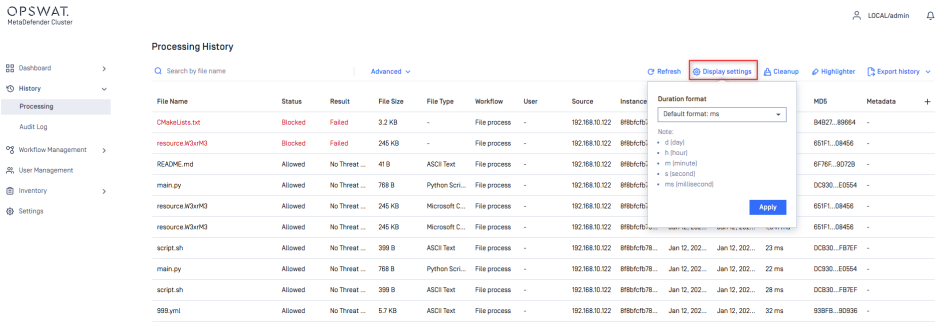
Request type: All

File Name						User	Source	Instance	Request Time	Process Start	Duration	MD5	Metadata
CMakeLists.txt						-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	1 ms	B4B27...89664	-
resource.W3xrM3						-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	1 ms	651F1...08456	-
README.md						-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	22 ms	6F76F...9D72B	-
main.py						-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	21 ms	DC930...E0554	-
resource.W3xrM3						-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	1,854 ms	651F1...08456	-
resource.W3xrM3						-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	1,641 ms	651F1...08456	-
script.sh						-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	23 ms	DCB30...FB7EF	-
main.py						-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	22 ms	DC930...E0554	-
script.sh						-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	28 ms	DCB30...FB7EF	-
999.yml						-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	32 ms	93BFB...9D936	-
test.tar.gz						-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	2,034 ms	5B92E...7F28B	-
main.py						-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	36 ms	DC930...E0554	-
script.sh						-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	60 ms	DCB30...FB7EF	-
test.tar.gz	Allowed	No Threat ...	24.9 KB	GNU Zippe...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	2,075 ms	5B92E...7F28B	-
3B0FA0756C69D215E95EF354AAA80B...	Allowed	No Threat ...	12.5 KB	GNU Zippe...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	66 ms	75FA2...E3A8E	-
CMakeLists.txt	Allowed	No Threat ...	3.2 KB	ASCII Text	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	75 ms	B4B27...89664	-

Display settings

Display Settings will allow the user to change the format of the Duration column. There are 3 options to choose from:

- Default: Shown as milliseconds or ms.
- Time Format: Shown as date h:m:s.ms.
- Short Format: Shown as date h m s ms.



Processing History

Duration format

Default format: ms

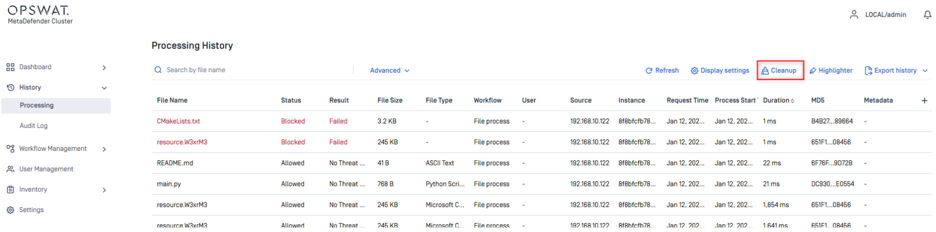
Note:

- d (day)
- h (hour)
- m (minute)
- s (second)
- ms (millisecond)

File Name	Status	Result	File Size	File Type	Workflow	User	Source	Instance				MD5	Metadata
CMakeLists.txt	Blocked	Failed	3.2 KB	-	File process	-	192.168.10.122	8f8bfcfb7				B4B27...89664	-
resource.W3xrM3	Blocked	Failed	245 KB	-	File process	-	192.168.10.122	8f8bfcfb7				651F1...08456	-
README.md	Allowed	No Threat ...	41 B	ASCII Text	File process	-	192.168.10.122	8f8bfcfb7				6F76F...9D72B	-
main.py	Allowed	No Threat ...	768 B	Python Scri...	File process	-	192.168.10.122	8f8bfcfb7				DC930...E0554	-
resource.W3xrM3	Allowed	No Threat ...	245 KB	Microsoft C...	File process	-	192.168.10.122	8f8bfcfb7				651F1...08456	-
resource.W3xrM3	Allowed	No Threat ...	245 KB	Microsoft C...	File process	-	192.168.10.122	8f8bfcfb7...				651F1...08456	-
script.sh	Allowed	No Threat ...	399 B	ASCII Text	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	23 ms	DCB30...FB7EF	-
main.py	Allowed	No Threat ...	768 B	Python Scri...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	22 ms	DC930...E0554	-
script.sh	Allowed	No Threat ...	399 B	ASCII Text	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	28 ms	DCB30...FB7EF	-
999.yml	Allowed	No Threat ...	5.7 KB	ASCII Text	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	32 ms	93BFB...9D936	-

Cleanup

Cleanup allows the user to delete the Processing History based on Time Range.

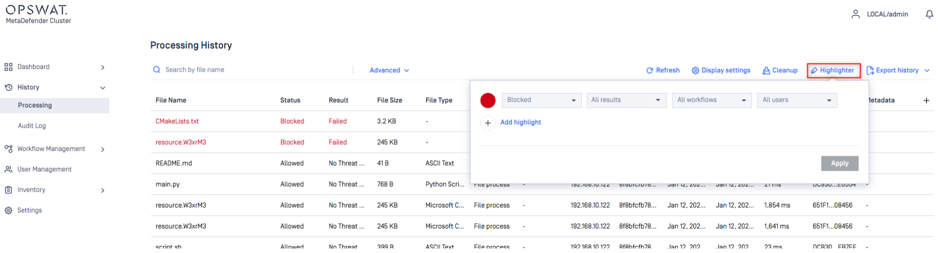


Processing History

File Name	Status	Result	File Size	File Type	Workflow	User	Source	Instance	Request Time	Process Start	Duration	MD5	Metadata
CMakeLists.txt	Blocked	Failed	3.2 KB	-	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	1 ms	B4B27...89664	-
resource.W3xrM3	Blocked	Failed	245 KB	-	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	1 ms	651F1...08456	-
README.md	Allowed	No Threat ...	41 B	ASCII Text	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	22 ms	6F76F...9D72B	-
main.py	Allowed	No Threat ...	768 B	Python Scri...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	21 ms	DC930...E0554	-
resource.W3xrM3	Allowed	No Threat ...	245 KB	Microsoft C...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	1,854 ms	651F1...08456	-
resource.W3xrM3	Allowed	No Threat ...	245 KB	Microsoft C...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	1,641 ms	651F1...08456	-

Highlight

Highlight allows the user to highlight a specific scan result based on color. This way, it will be easier to see visually for example, what is Blocked, Sanitized, Failed, etc...



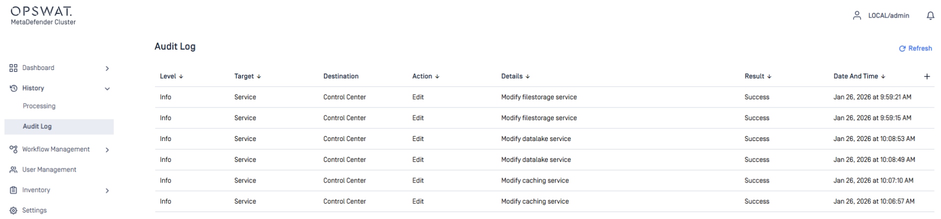
Processing History

● Blocked | All results | All workflows | All users

File Name	Status	Result	File Size	File Type									etadata
CMakeLists.txt	Blocked	Failed	3.2 KB	-									
resource.W3xrM3	Blocked	Failed	245 KB	-									
README.md	Allowed	No Threat ...	41 B	ASCII Text									
main.py	Allowed	No Threat ...	768 B	Python Scri...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	21 ms	DC930...E0554	-
resource.W3xrM3	Allowed	No Threat ...	245 KB	Microsoft C...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	1,854 ms	651F1...08456	-
resource.W3xrM3	Allowed	No Threat ...	245 KB	Microsoft C...	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	1,641 ms	651F1...08456	-
script.sh	Allowed	No Threat ...	399 B	ASCII Text	File process	-	192.168.10.122	8f8bfcfb78...	Jan 12, 202...	Jan 12, 202...	23 ms	DCB30...FB7EF	-

Audit Log

This section provides application updates such as Deploying or Undeploying an instance, license activation, configuration change etc...



Level ↓	Target ↓	Destination	Action ↓	Details ↓	Result ↓	Date And Time ↓
Info	Service	Control Center	Edit	Modify filestorage service	Success	Jan 26, 2026 at 9:59:21 AM
Info	Service	Control Center	Edit	Modify filestorage service	Success	Jan 26, 2026 at 9:59:15 AM
Info	Service	Control Center	Edit	Modify datalake service	Success	Jan 26, 2026 at 10:08:53 AM
Info	Service	Control Center	Edit	Modify datalake service	Success	Jan 26, 2026 at 10:08:49 AM
Info	Service	Control Center	Edit	Modify caching service	Success	Jan 26, 2026 at 10:07:10 AM
Info	Service	Control Center	Edit	Modify caching service	Success	Jan 26, 2026 at 10:06:57 AM

Inventory

The Inventory section contains all detailed and configurations for Services and Workers. This also incudes Module Updates, Licenses and Installers.

- Services.
- Workers.
- Modules.
- Licenses.
- Installers.

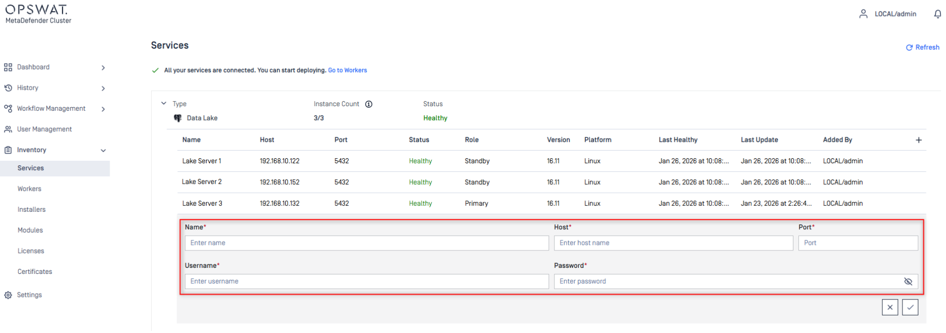
Services

This section will allow the user to add all necessary services for MetaDefender Cluster to fully function.

Data Lake and Data Warehouse

Add the Data Lake and Data Warehouse in this section. Fill out all required fields. If High Availability was configured for PostgreSQL. Add them to this section as well.

> **Info**
>
> High Availability support for Postgres is only available for Data Lake



Services

All your services are connected. You can start deploying. Go to Workers

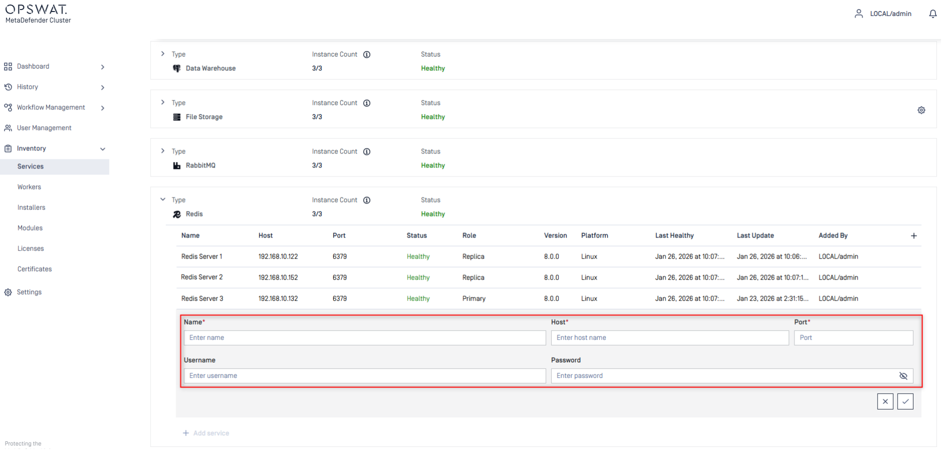
Type	Instance Count	Status
Data Lake	3/3	Healthy

Name	Host	Port	Status	Role	Version	Platform	Last Healthy	Last Update	Added By
Lake Server 1	192.168.10.122	5432	Healthy	Standby	16.11	Linux	Jan 26, 2026 at 10:08:...	Jan 26, 2026 at 10:08:...	LOCAL/admin
Lake Server 2	192.168.10.152	5432	Healthy	Standby	16.11	Linux	Jan 26, 2026 at 10:08:...	Jan 26, 2026 at 10:08:...	LOCAL/admin
Lake Server 3	192.168.10.132	5432	Healthy	Primary	16.11	Linux	Jan 26, 2026 at 10:08:...	Jan 23, 2026 at 2:26:4...	LOCAL/admin

Name* Enter name
Host* Enter host name
Port* Port
Username* Enter username
Password* Enter password

Redis

Add the Redis server in this section. Fill out all required fields. If High Availability was configured for Redis. Add them to this section as well.



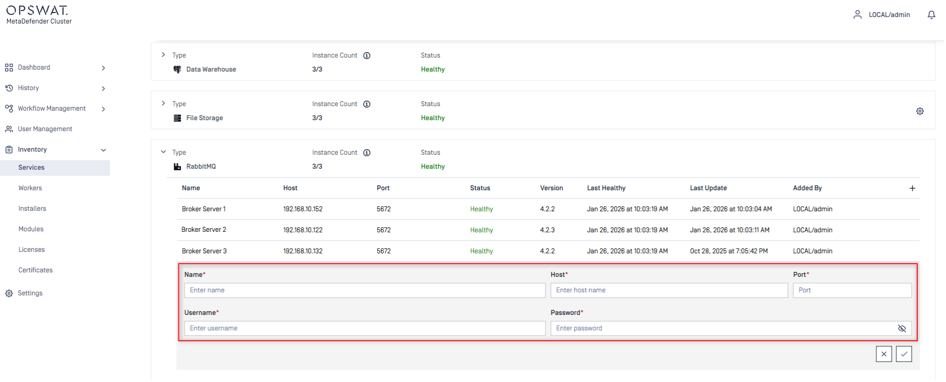
Type	Instance Count	Status
Data Warehouse	3/3	Healthy
File Storage	3/3	Healthy
RabbitMQ	3/3	Healthy
Redis	3/3	Healthy

Name	Host	Port	Status	Role	Version	Platform	Last Healthy	Last Update	Added By
Redis Server 1	192.168.10.122	6379	Healthy	Replica	8.0.0	Linux	Jan 26, 2026 at 10:07:...	Jan 26, 2026 at 10:06:...	LOCAL/admin
Redis Server 2	192.168.10.152	6379	Healthy	Replica	8.0.0	Linux	Jan 26, 2026 at 10:07:...	Jan 26, 2026 at 10:07:1...	LOCAL/admin
Redis Server 3	192.168.10.132	6379	Healthy	Primary	8.0.0	Linux	Jan 26, 2026 at 10:07:...	Jan 23, 2026 at 2:31:15...	LOCAL/admin

Name* Enter name
Host* Enter host name
Port* Port
Username Enter username
Password Enter password

RabbitMQ

Add the RabbitMQ server in this section. Fill out all required fields. If High Availability was configured for RabbitMQ. Add them to this section as well.



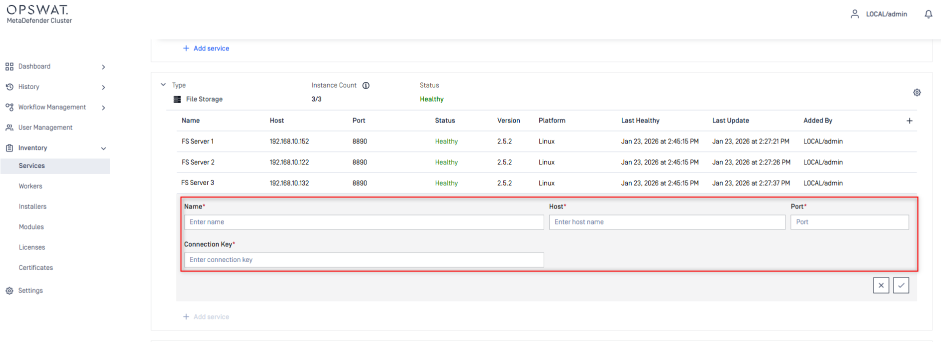
Type	Instance Count	Status
Data Warehouse	3/3	Healthy
File Storage	3/3	Healthy
RabbitMQ	3/3	Healthy

Name	Host	Port	Status	Version	Last Healthy	Last Update	Added By
Broker Server 1	192.168.10.152	5672	Healthy	4.2.2	Jan 26, 2026 at 10:03:19 AM	Jan 26, 2026 at 10:03:04 AM	LOCAL/admin
Broker Server 2	192.168.10.122	5672	Healthy	4.2.3	Jan 26, 2026 at 10:03:19 AM	Jan 26, 2026 at 10:03:11 AM	LOCAL/admin
Broker Server 3	192.168.10.132	5672	Healthy	4.2.2	Jan 26, 2026 at 10:03:19 AM	Oct 28, 2025 at 7:05:42 PM	LOCAL/admin

Name* Enter name
Host* Enter host name
Port* Port
Username* Enter username
Password* Enter password

File Storage

Add the File Storage server in this section. Fill out all required fields. If High Availability was configured for File Storage. Add them to this section as well.



Type	Instance Count	Status
File Storage	3/3	Healthy

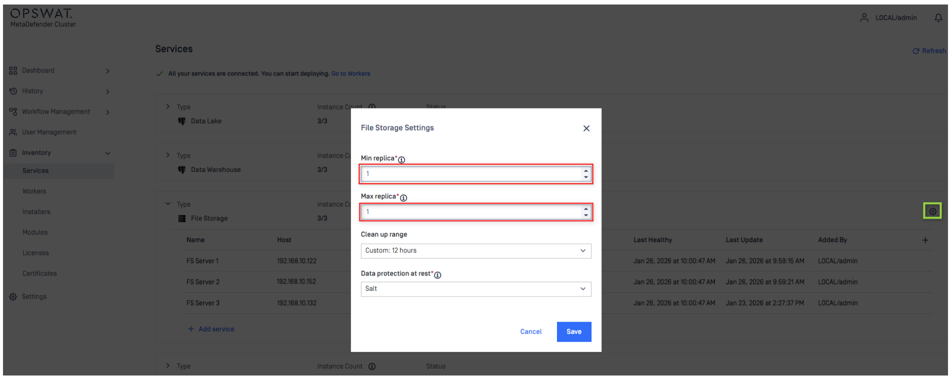
Name	Host	Port	Status	Version	Platform	Last Healthy	Last Update	Added By
FS Server 1	192.168.10.152	8890	Healthy	2.5.2	Linux	Jan 23, 2026 at 2:45:15 PM	Jan 23, 2026 at 2:27:21 PM	LOCAL/admin
FS Server 2	192.168.10.122	8890	Healthy	2.5.2	Linux	Jan 23, 2026 at 2:45:15 PM	Jan 23, 2026 at 2:27:26 PM	LOCAL/admin
FS Server 3	192.168.10.132	8890	Healthy	2.5.2	Linux	Jan 23, 2026 at 2:45:15 PM	Jan 23, 2026 at 2:27:37 PM	LOCAL/admin

Name* Enter name
Host* Enter host name
Port* Port
Connection Key* Enter connection key

File Storage Settings

Values found in the File Storage Settings can be left in its default settings. However, when configuring High Availability for File Storage. Ensure that the Minimum replica and Maximum replica has been configured correctly. For more information on High Availability for File Storage, click here.

- Min replica: The minimum number of data copies that must be written for the operation to succeed.
- Max replica: The maximum of data copies stored across the system.
- Clean up range: The number of days, weeks or months until clean up of files that are sanitized, watermarked files and files processed by Proactive DLP.
- Data protection at rest: The protection mechanism is applied to the files stored in File Storage. By default, all files are salted before being they are saved to disk.



All your services are connected. You can start deploying. Go to Workers

Type	Instance Count
Data Lake	3/3
Data Warehouse	3/3
File Storage	3/3

Name	Host	Last Healthy	Last Update	Added By
FS Server 1	192.168.10.122	Jan 26, 2026 at 10:00:47 AM	Jan 26, 2026 at 9:59:15 AM	LOCAL/admin
FS Server 2	192.168.10.152	Jan 26, 2026 at 10:00:47 AM	Jan 26, 2026 at 9:59:21 AM	LOCAL/admin
FS Server 3	192.168.10.132	Jan 26, 2026 at 10:00:47 AM	Jan 23, 2026 at 2:27:37 PM	LOCAL/admin

File Storage Settings

Min replica*: 1

Max replica*: 1

Clean up range: Custom: 12 hours

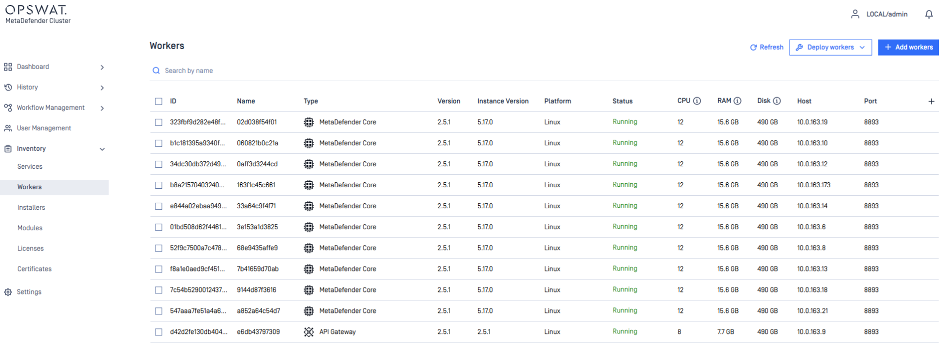
Data protection at rest*: Salt

Workers

This section will allow the system administrator to add the MetaDefender Distributed Cluster (MD Cluster) Workers which will help deploy and monitor activities of MetaDefender Core, MD Cluster API Gateway and MD Cluster Callback Service.

Info

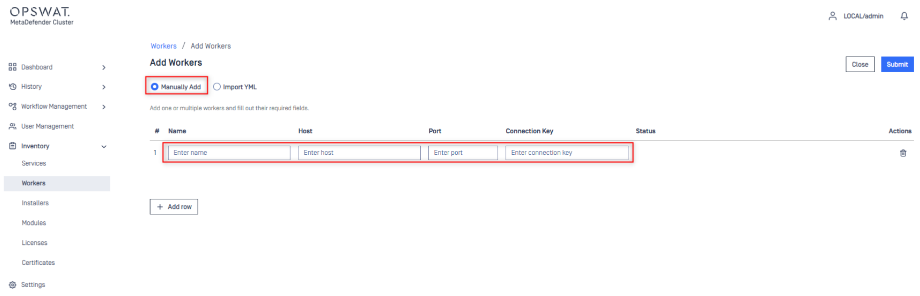
MD Cluster Callback Service is optional. However, if the scan result needs to be sent to a Webhook. This service must be installed in order to use the Callback Service feature.



□	ID	Name	Type	Version	Instance Version	Platform	Status	CPU	RAM	Disk	Host	Port
□	323fbf9d282e48f...	02d038f54f01	MetaDefender Core	2.5.1	5.17.0	Linux	Running	12	15.6 GB	490 GB	10.0.163.19	8893
□	b1c181395a9340f...	060821b0c21a	MetaDefender Core	2.5.1	5.17.0	Linux	Running	12	15.6 GB	490 GB	10.0.163.10	8893
□	34dc30db372d49...	0aff3d3244cd	MetaDefender Core	2.5.1	5.17.0	Linux	Running	12	15.6 GB	490 GB	10.0.163.12	8893
□	b8a21570403240...	163f1c45c661	MetaDefender Core	2.5.1	5.17.0	Linux	Running	12	15.6 GB	490 GB	10.0.163.173	8893
□	e844a02ebaa949...	33a64c9f4f71	MetaDefender Core	2.5.1	5.17.0	Linux	Running	12	15.6 GB	490 GB	10.0.163.14	8893
□	01bd508d62f4461...	3e153a1d3825	MetaDefender Core	2.5.1	5.17.0	Linux	Running	12	15.6 GB	490 GB	10.0.163.6	8893
□	52f9c7500a7c478...	68e9435affe9	MetaDefender Core	2.5.1	5.17.0	Linux	Running	12	15.6 GB	490 GB	10.0.163.8	8893
□	f8a1e0aed9cf451...	7b41659d70ab	MetaDefender Core	2.5.1	5.17.0	Linux	Running	12	15.6 GB	490 GB	10.0.163.13	8893
□	7c54b5290012437...	9144d87f3616	MetaDefender Core	2.5.1	5.17.0	Linux	Running	12	15.6 GB	490 GB	10.0.163.18	8893
□	547aaa7fe51a4a6...	a852a64c54d7	MetaDefender Core	2.5.1	5.17.0	Linux	Running	12	15.6 GB	490 GB	10.0.163.21	8893
□	d42d2fe130db404...	e6db43797309	API Gateway	2.5.1	2.5.1	Linux	Running	8	7.7 GB	490 GB	10.0.163.9	8893

Add new workers

When manually adding the MD Cluster Workers. Simply fill out the Name, Host, Port and Connection Key fields. The Port and Connection Key are the same values used in the configuration file of MD Cluster Worker during installation. Once added, select `Submit` and MD Cluster will validate if the Workers are added successfully or failed.



● Manually Add ○ Import YML

Add one or multiple workers and fill out their required fields.

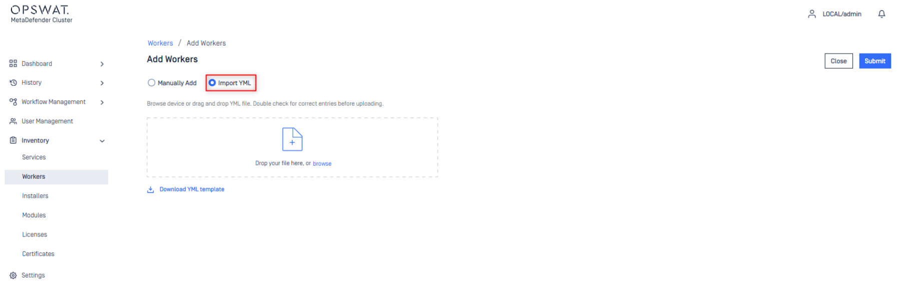
#	Name	Host	Port	Connection Key	Status	Actions
1	Enter name	Enter host	Enter port	Enter connection key		

Import workers

The system administrator has an option to import the MD Cluster Workers via YML file instead of manually adding the MD Cluster Workers via MD Cluster Control Center UI. There is a template available to be downloaded when selecting the `Import YML` option. Below you'll find an example with an MD Cluster API Gateway and three MetaDefender Cores:

yaml

```yaml
- display_name: API-Gateway
  host: 10.1.100.100
  port: 8893
  connection_key: 1234abcd
- display_name: Core-1
  host: 10.1.100.101
  port: 8893
  connection_key: 1234abcd
- display_name: Core-2
  host: 10.1.100.103
  port: 8893
  connection_key: 1234abcd
- display_name: Core-3
  host: 10.1.100.104
  port: 8893
  connection_key: 1234abcd
```



Add Workers

○ Manually Add ● Import YML

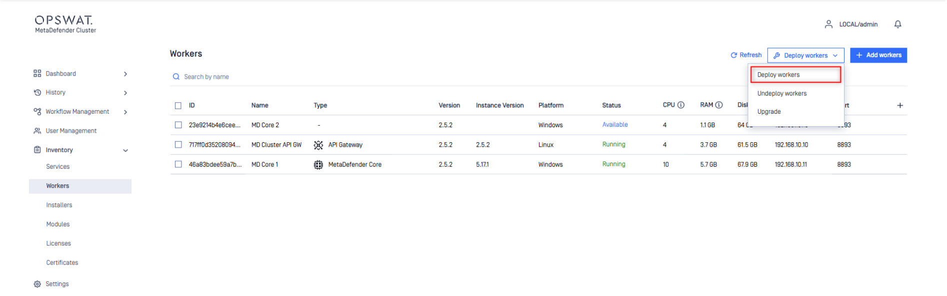
Browse device or drag and drop YML file. Double check for correct entries before uploading.

Drop your file here, or browse

Download YML template

Deploy workers

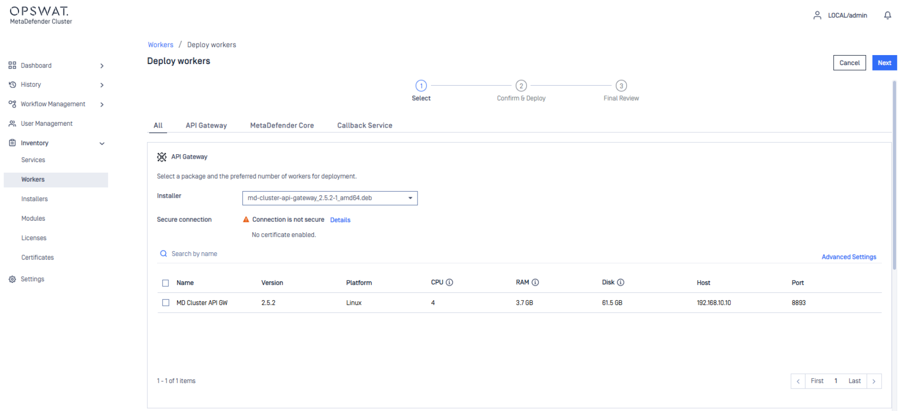
Once MD Cluster Workers have been added and their status are shown as `Available`. The system administrator can now deploy MetaDefender Core, MD Cluster API Gateway and MD Cluster Callback Service separately or all at the same time.



Workers

ID	Name	Type	Version	Instance Version	Platform	Status	CPU	RAM	Disl		rt
23e9214b4e6cee...	MD Core 2	-	2.5.2		Windows	Available	4	1.1 GB	64		93
717ff0d35208094...	MD Cluster API GW	API Gateway	2.5.2	2.5.2	Linux	Running	4	3.7 GB	61.5 GB	192.168.10.10	8893
46a83bdee59a7b...	MD Core 1	MetaDefender Core	2.5.2	5.17.1	Windows	Running	10	5.7 GB	67.9 GB	192.168.10.11	8893

To deploy the MD Cluster API Gateway. Simply choose the version of MD Cluster API Gateway and select an available MD Cluster Workers.



Deploy workers

API Gateway

Select a package and the preferred number of workers for deployment.

Installer: md-cluster-api-gateway_2.5.2-1_amd64.deb

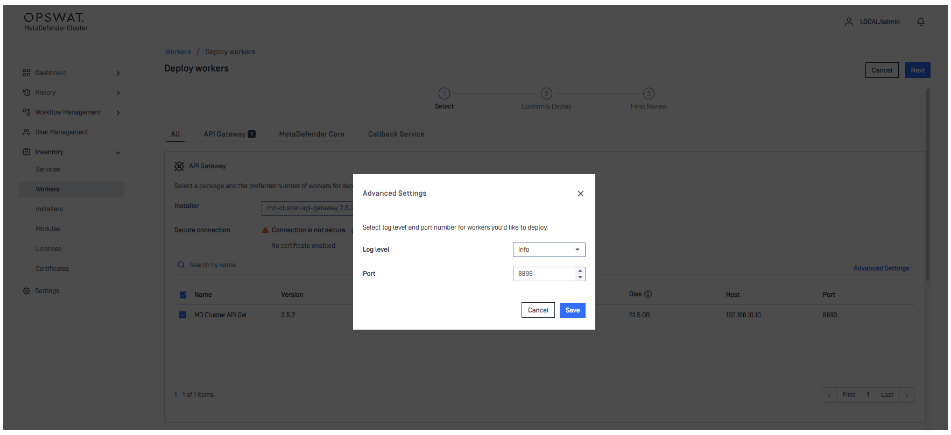
Secure connection: Connection is not secure

No certificate enabled.

Name	Version	Platform	CPU	RAM	Disk	Host	Port
MD Cluster API GW	2.5.2	Linux	4	3.7 GB	61.5 GB	192.168.10.10	8893

1 - 1 of 1 items

The Log level and Port that will be assigned to MD Cluster API Gateway can be set by clicking `Advanced Settings`.



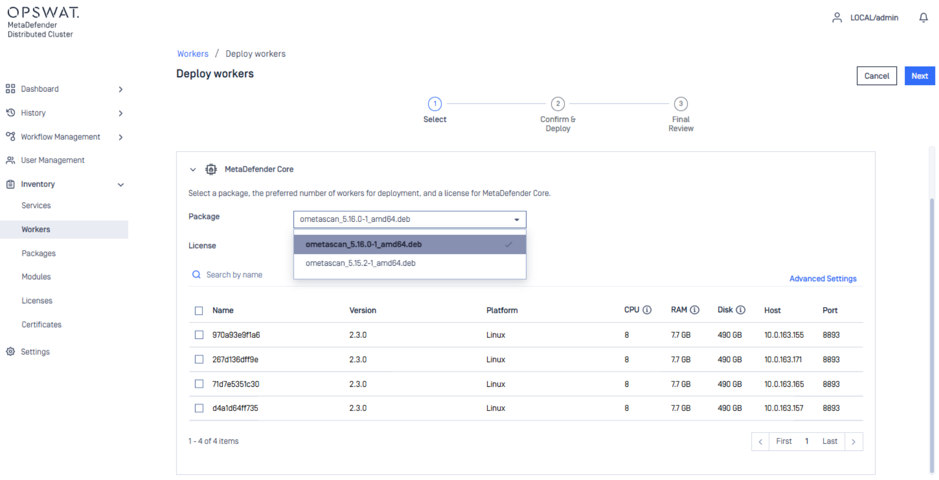
Advanced Settings

Select log level and port number for workers you'd like to deploy.

Log level: Info

Port: 8899

The system administrator can choose the version of MetaDefender Core should be deployed on available MD Cluster Workers.



Deploy workers

MetaDefender Core

Select a package, the preferred number of workers for deployment, and a license for MetaDefender Core.

Package: ometascan_5.16.0-1_amd64.deb

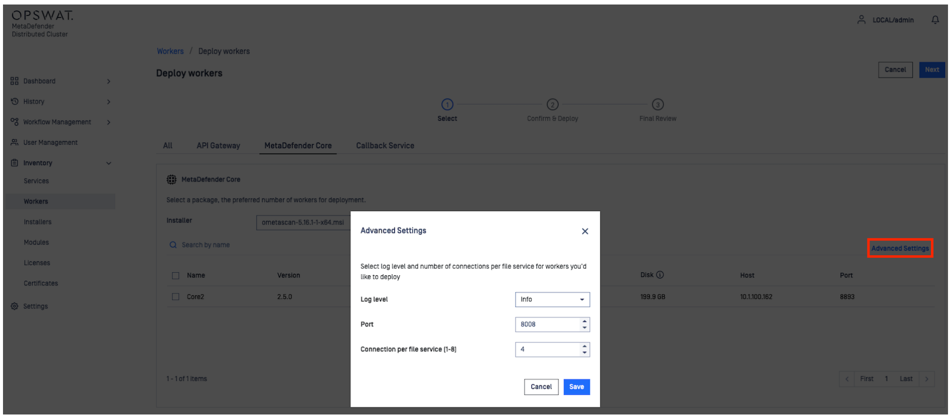
License

- ometascan_5.16.0-1_amd64.deb ✓
- ometascan_5.15.2-1_amd64.deb

□	Name	Version	Platform	CPU	RAM	Disk	Host	Port
□	970a93e9f1a6	2.3.0	Linux	8	7.7 GB	490 GB	10.0.163.155	8893
□	267d136dff9e	2.3.0	Linux	8	7.7 GB	490 GB	10.0.163.171	8893
□	71d7e5351c30	2.3.0	Linux	8	7.7 GB	490 GB	10.0.163.165	8893
□	d4a1d64ff735	2.3.0	Linux	8	7.7 GB	490 GB	10.0.163.157	8893

1 - 4 of 4 items

The system administrator can also modify the Log level, Port and Connection per file service of MetaDefender Core that will be deployed by selecting `Advanced Settings`.



Deploy workers

MetaDefender Core

Select a package, the preferred number of workers for deployment.

Installer: ometascan-5.16.1-1-x64.msi

□	Name	Version	Disk	Host	Port
□	Core2	2.5.0	199.9 GB	10.1.100.162	8893

1 - 1 of 1 items

Advanced Settings

Select log level and number of connections per file service for workers you'd like to deploy

Log level: Info

Port: 8008

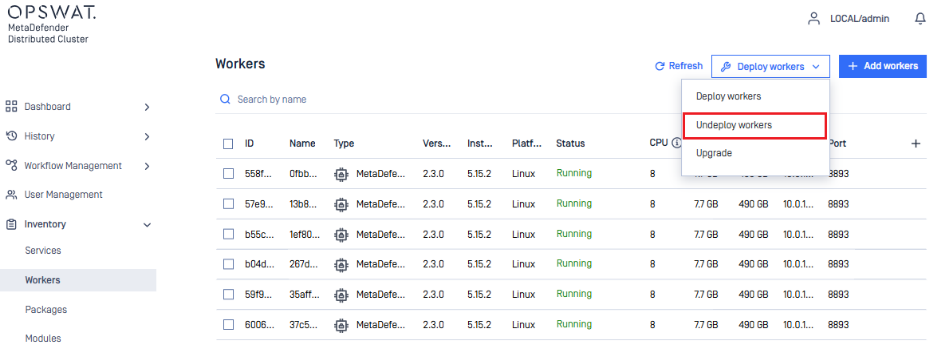
Connection per file service (1-8): 4

Cancel Save

Click `Next` to confirm the deployment and `Finish` to start deploying on selected MD Cluster Workers.

Release workers

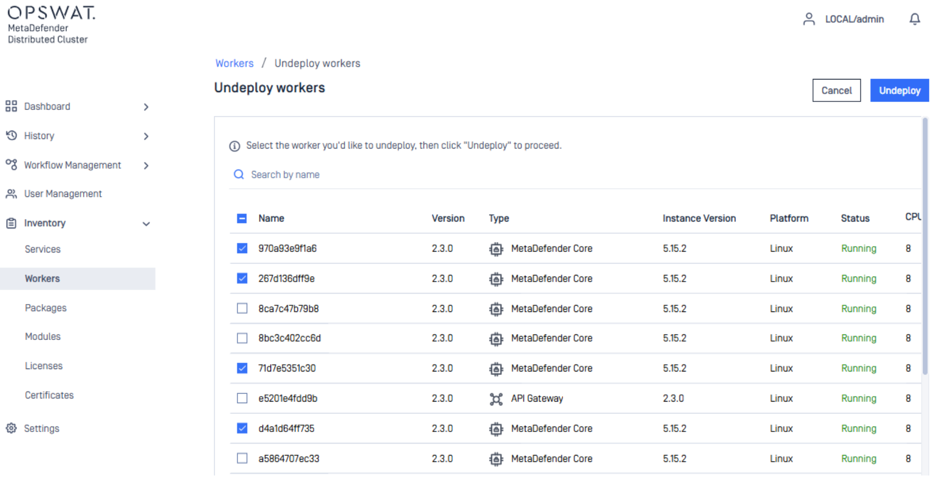
To release MD Cluster API Gateway, MD Cluster Callback Service, or MetaDefender Core on MD Cluster Workers, the system administrator can select Undeploy workers in the top right corner.



Workers

□	ID	Name	Type	Vers...	Inst...	Platf...	Status	CPU				Port
□	558f...	0fbb...	MetaDefe...	2.3.0	5.15.2	Linux	Running	8	[illegible]	[illegible]	[illegible]	8893
□	57e9...	13b8...	MetaDefe...	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.1...	8893
□	b55c...	1ef80...	MetaDefe...	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.1...	8893
□	b04d...	267d...	MetaDefe...	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.1...	8893
□	59f9...	35aff...	MetaDefe...	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.1...	8893
□	6006...	37c5...	MetaDefe...	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.1...	8893

From the list of hosting MD Cluster Worker instances, the system administrator can choose instances to release MD Cluster API Gateway, MD Cluster Callback Service or MetaDefender Core.

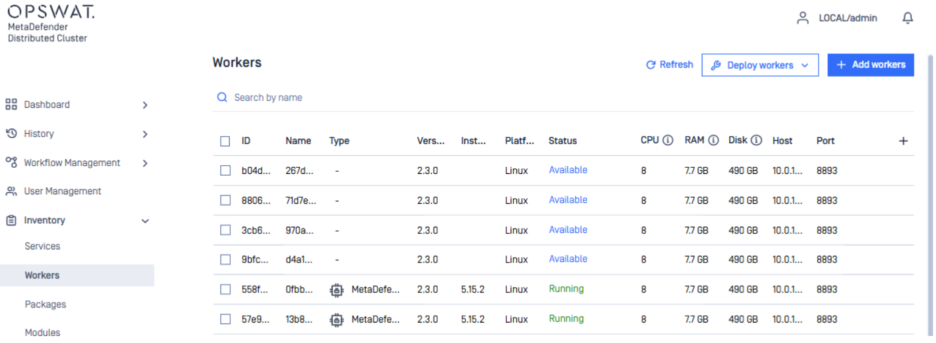


Undeploy workers

Select the worker you'd like to undeploy, then click "Undeploy" to proceed.

☐	Name	Version	Type	Instance Version	Platform	Status	CPU
☑	970a93e9f1a6	2.3.0	MetaDefender Core	5.15.2	Linux	Running	8
☑	267d136dff9e	2.3.0	MetaDefender Core	5.15.2	Linux	Running	8
☐	8ca7c47b79b8	2.3.0	MetaDefender Core	5.15.2	Linux	Running	8
☐	8bc3c402cc6d	2.3.0	MetaDefender Core	5.15.2	Linux	Running	8
☑	71d7e5351c30	2.3.0	MetaDefender Core	5.15.2	Linux	Running	8
☐	e5201e4fdd9b	2.3.0	API Gateway	2.3.0	Linux	Running	8
☑	d4a1d64ff735	2.3.0	MetaDefender Core	5.15.2	Linux	Running	8
☐	a5864707ec33	2.3.0	MetaDefender Core	5.15.2	Linux	Running	8

When the `Undeploy` button is selected. MD Cluster API Gateway, MD Cluster Callback Service, and/or MetaDefender Core on selected MD Cluster Worker instances are uninstalled. Once uninstalled, the MD Cluster Worker instances will become available to deploy new MD Cluster API Gateway, MD Cluster Callback Service or MetaDefender Core.

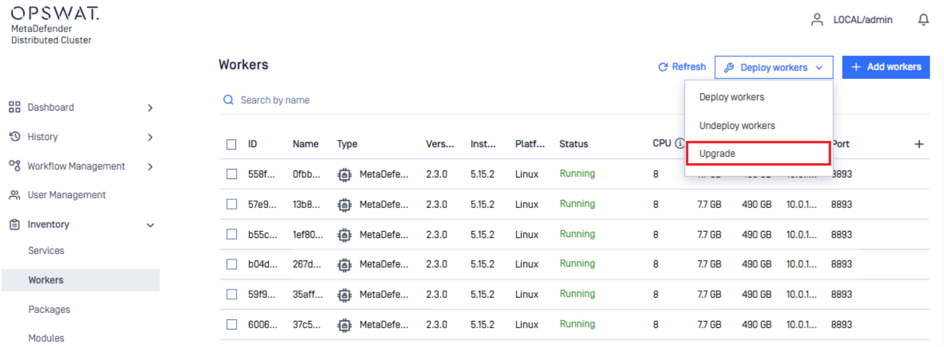


Workers

☐	ID	Name	Type	Vers...	Inst...	Platf...	Status	CPU	RAM	Disk	Host	Port
☐	b04d...	267d...	-	2.3.0		Linux	Available	8	7.7 GB	490 GB	10.0.1...	8893
☐	8806...	71d7e...	-	2.3.0		Linux	Available	8	7.7 GB	490 GB	10.0.1...	8893
☐	3cb6...	970a...	-	2.3.0		Linux	Available	8	7.7 GB	490 GB	10.0.1...	8893
☐	9bfc...	d4a1...	-	2.3.0		Linux	Available	8	7.7 GB	490 GB	10.0.1...	8893
☐	558f...	0fbb...	MetaDefe...	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.1...	8893
☐	57e9...	13b8...	MetaDefe...	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.1...	8893

Upgrade

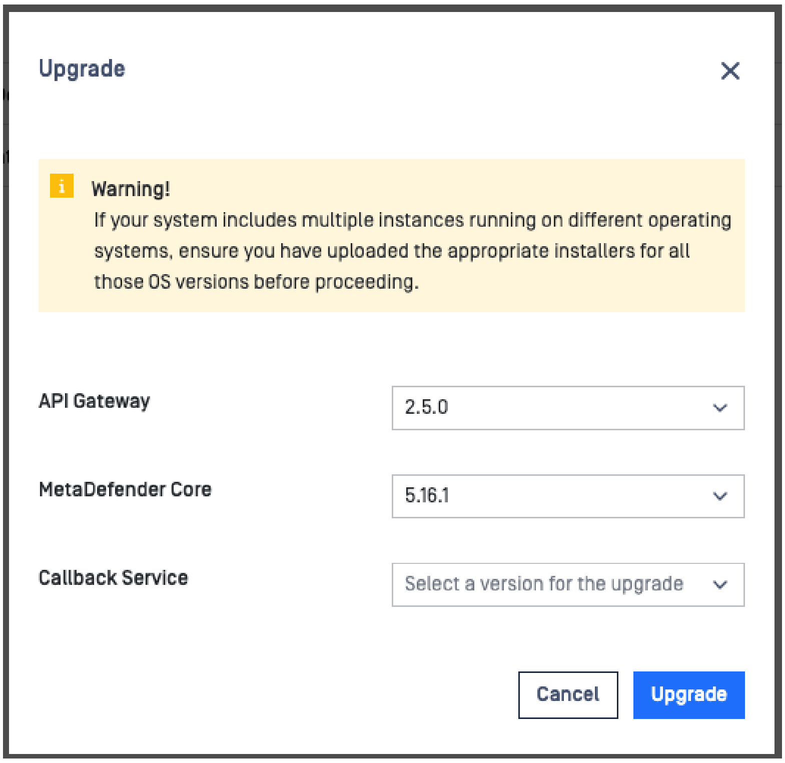
When a new version of MD Cluster API Gateway, MD Cluster Callback Service or MetaDefender Core is available. And all three installers have been uploaded to the Installers section. The system administrator can perform an upgrade for the three products by selecting the `Deploy Workers` menu and then `Upgrade`.



Workers

ID	Name	Type	Vers...	Inst...	Platf...	Status	CPU				Port
558f...	0fbb...	MetaDefe...	2.3.0	5.15.2	Linux	Running	8				8893
57e9...	13b8...	MetaDefe...	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.1...	8893
b55c...	1ef80...	MetaDefe...	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.1...	8893
b04d...	267d...	MetaDefe...	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.1...	8893
59f9...	35aff...	MetaDefe...	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.1...	8893
6006...	37c5...	MetaDefe...	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.1...	8893

The system administrator can then choose the new version of MD Cluster API Gateway, MD Cluster Callback Service and MetaDefender Core to perform the upgrade.



Upgrade

Warning!
If your system includes multiple instances running on different operating systems, ensure you have uploaded the appropriate installers for all those OS versions before proceeding.

API Gateway: 2.5.0

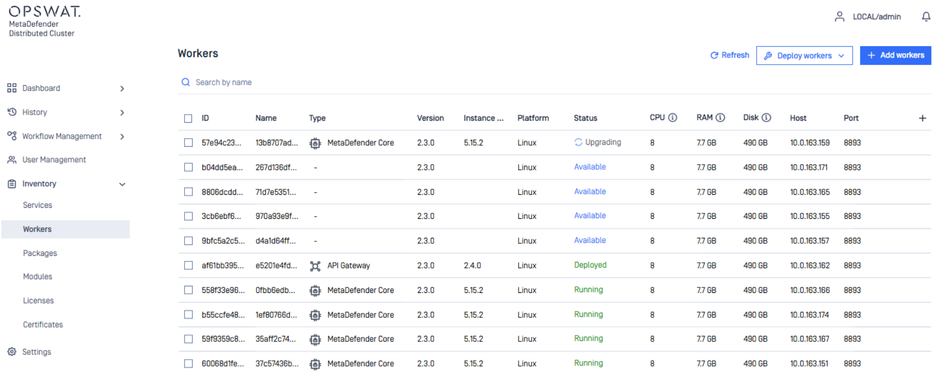
MetaDefender Core: 5.16.1

Callback Service: Select a version for the upgrade

The new versions of MD Cluster API Gateway, MD Cluster Callback Service, or MetaDefender Core are streamed to the MD Cluster Workers and remote upgrade process will takes place automatically.

Info

If there are scans currently active. MetaDefender Core will isolate itself by no longer taking scan request and finishes its current scans before upgrading. Upgrades are performed one MetaDefender Core at a time so that scans are uninterrupted.

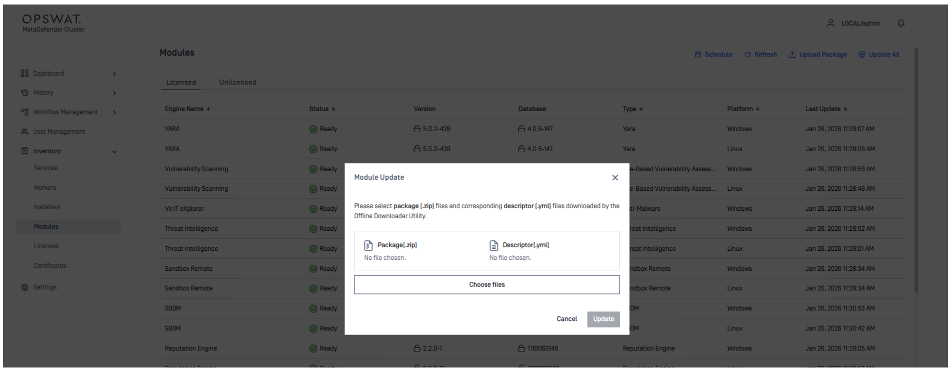


Workers

□	ID	Name	Type	Version	Instance ...	Platform	Status	CPU	RAM	Disk	Host	Port
□	57e94c23...	13b8707ad...	MetaDefender Core	2.3.0	5.15.2	Linux	Upgrading	8	7.7 GB	490 GB	10.0.163.159	8893
□	b04dd5ea...	267d136df...	-	2.3.0		Linux	Available	8	7.7 GB	490 GB	10.0.163.171	8893
□	8806dcdd...	71d7e5351...	-	2.3.0		Linux	Available	8	7.7 GB	490 GB	10.0.163.165	8893
□	3cb6ebf6...	970a93e9f...	-	2.3.0		Linux	Available	8	7.7 GB	490 GB	10.0.163.155	8893
□	9bfc5a2c5...	d4a1d64ff...	-	2.3.0		Linux	Available	8	7.7 GB	490 GB	10.0.163.157	8893
□	af61bb395...	e5201e4fd...	API Gateway	2.3.0	2.4.0	Linux	Deployed	8	7.7 GB	490 GB	10.0.163.162	8893
□	558f33e96...	0fbb6edb...	MetaDefender Core	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.163.166	8893
□	b55ccfe48...	1ef80766d...	MetaDefender Core	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.163.174	8893
□	59f9359c8...	35aff2c74...	MetaDefender Core	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.163.167	8893
□	60068d1fe...	37c57436b...	MetaDefender Core	2.3.0	5.15.2	Linux	Running	8	7.7 GB	490 GB	10.0.163.151	8893

Modules

The Modules section provides visibility to the modules that are available to MetaDefender Core to use when deployed. Modules are downloaded by MetaDefender Cluster (MD Cluster) Control Center or if deploying in an Offline Environment, Modules can be uploaded manually by the system administrator when using MetaDefender Update Downloader.



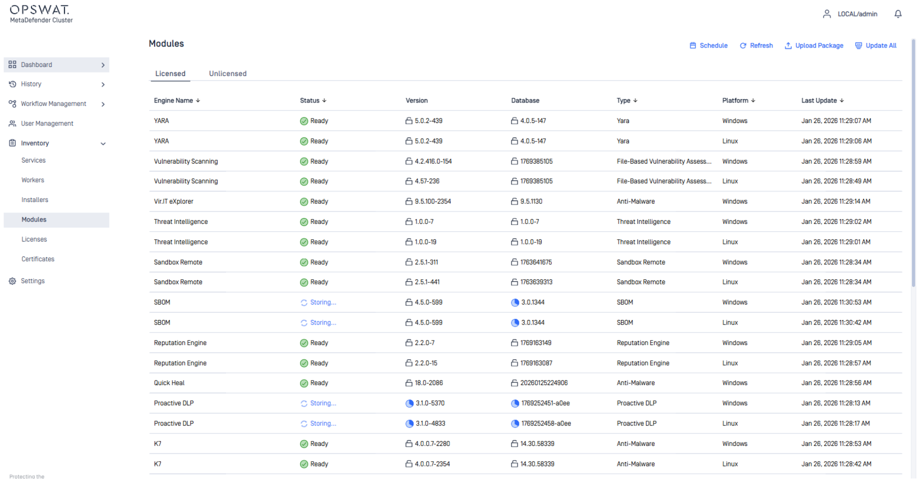
Please select **package [.zip]** files and corresponding **descriptor [.yml]** files downloaded by the Offline Downloader Utility.

No file chosen.

No file chosen.

Licensed

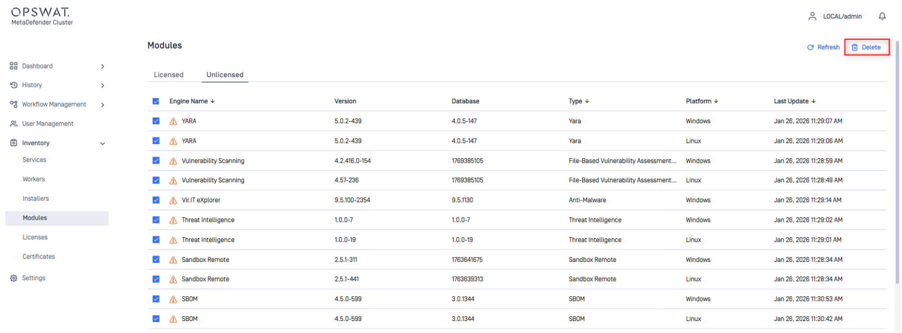
This section of the Modules will show all the modules that are available and will be used when MetaDefender Core is deployed. The modules shown in this section are based on the MetaDefender Core license. Modules are automatically downloaded if the MD Cluster Control Center has access to the internet.



Engine Name ↓	Status ↓	Version	Database	Type ↓	Platform ↓	Last Update ↓
YARA	Ready	5.0.2-439	4.0.5-147	Yara	Windows	Jan 26, 2026 11:29:07 AM
YARA	Ready	5.0.2-439	4.0.5-147	Yara	Linux	Jan 26, 2026 11:29:06 AM
Vulnerability Scanning	Ready	4.2.416.0-154	1769385105	File-Based Vulnerability Assess...	Windows	Jan 26, 2026 11:28:59 AM
Vulnerability Scanning	Ready	4.57-236	1769385105	File-Based Vulnerability Assess...	Linux	Jan 26, 2026 11:28:49 AM
Vir.IT eXplorer	Ready	9.5.100-2354	9.5.1130	Anti-Malware	Windows	Jan 26, 2026 11:29:14 AM
Threat Intelligence	Ready	1.0.0-7	1.0.0-7	Threat Intelligence	Windows	Jan 26, 2026 11:29:02 AM
Threat Intelligence	Ready	1.0.0-19	1.0.0-19	Threat Intelligence	Linux	Jan 26, 2026 11:29:01 AM
Sandbox Remote	Ready	2.5.1-311	1763641675	Sandbox Remote	Windows	Jan 26, 2026 11:28:34 AM
Sandbox Remote	Ready	2.5.1-441	1763639313	Sandbox Remote	Linux	Jan 26, 2026 11:28:34 AM
SBOM	Storing...	4.5.0-599	3.0.1344	SBOM	Windows	Jan 26, 2026 11:30:53 AM
SBOM	Storing...	4.5.0-599	3.0.1344	SBOM	Linux	Jan 26, 2026 11:30:42 AM
Reputation Engine	Ready	2.2.0-7	1769163149	Reputation Engine	Windows	Jan 26, 2026 11:29:05 AM
Reputation Engine	Ready	2.2.0-15	1769163087	Reputation Engine	Linux	Jan 26, 2026 11:28:57 AM
Quick Heal	Ready	18.0-2086	20260125224906	Anti-Malware	Windows	Jan 26, 2026 11:28:56 AM
Proactive DLP	Storing...	3.1.0-5370	1769252451-a0ee	Proactive DLP	Windows	Jan 26, 2026 11:28:13 AM
Proactive DLP	Storing...	3.1.0-4833	1769252458-a0ee	Proactive DLP	Linux	Jan 26, 2026 11:28:17 AM
K7	Ready	4.0.0.7-2280	14.30.58339	Anti-Malware	Windows	Jan 26, 2026 11:28:53 AM
K7	Ready	4.0.0.7-2354	14.30.58339	Anti-Malware	Linux	Jan 26, 2026 11:28:42 AM

Unlicensed

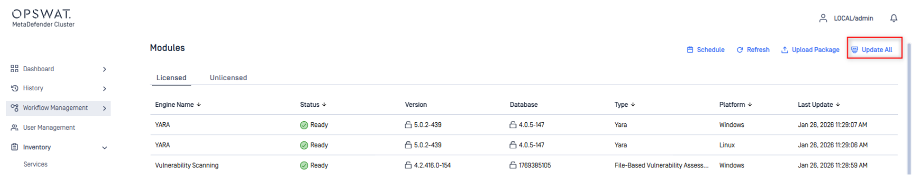
This section of Modules will show all the modules that were previously licensed. From here, since the modules are no longer licensed they can be removed by selecting all the modules and clicking `Delete`.



☑	Engine Name	Version	Database	Type	Platform	Last Update
☑	YARA	5.0.2-439	4.0.5-147	Yara	Windows	Jan 26, 2026 11:29:07 AM
☑	YARA	5.0.2-439	4.0.5-147	Yara	Linux	Jan 26, 2026 11:29:06 AM
☑	Vulnerability Scanning	4.2.416.0-154	1769385105	File-Based Vulnerability Assessment...	Windows	Jan 26, 2026 11:28:59 AM
☑	Vulnerability Scanning	4.57-236	1769385105	File-Based Vulnerability Assessment...	Linux	Jan 26, 2026 11:28:49 AM
☑	Vir.IT eXplorer	9.5.100-2354	9.5.1130	Anti-Malware	Windows	Jan 26, 2026 11:29:14 AM
☑	Threat Intelligence	1.0.0-7	1.0.0-7	Threat Intelligence	Windows	Jan 26, 2026 11:29:02 AM
☑	Threat Intelligence	1.0.0-19	1.0.0-19	Threat Intelligence	Linux	Jan 26, 2026 11:29:01 AM
☑	Sandbox Remote	2.5.1-311	1763641675	Sandbox Remote	Windows	Jan 26, 2026 11:28:34 AM
☑	Sandbox Remote	2.5.1-441	1763639313	Sandbox Remote	Linux	Jan 26, 2026 11:28:34 AM
☑	SBOM	4.5.0-599	3.0.1344	SBOM	Windows	Jan 26, 2026 11:30:53 AM
☑	SBOM	4.5.0-599	3.0.1344	SBOM	Linux	Jan 26, 2026 11:30:42 AM

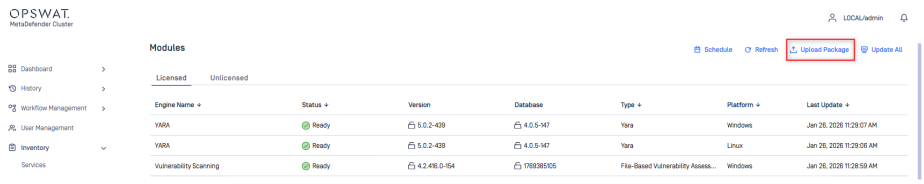
Manual Updates

The system administrator can trigger an update by simply clicking the `Update All` button.

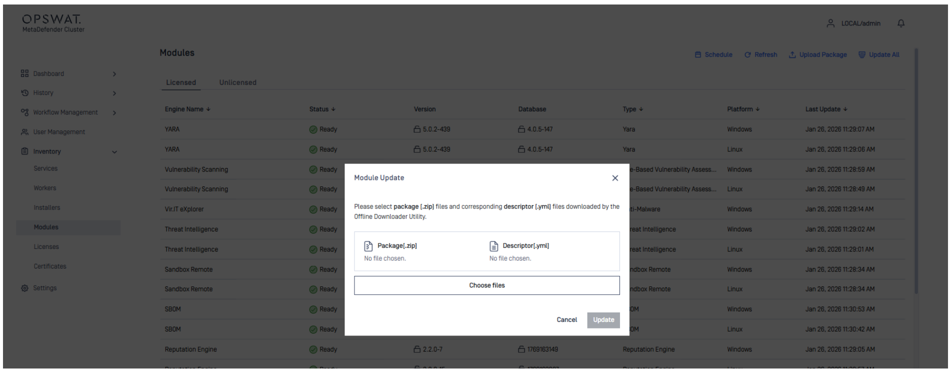


Engine Name	Status	Version	Database	Type	Platform	Last Update
YARA	Ready	5.0.2-439	4.0.5-147	Yara	Windows	Jan 26, 2026 11:29:07 AM
YARA	Ready	5.0.2-439	4.0.5-147	Yara	Linux	Jan 26, 2026 11:29:06 AM
Vulnerability Scanning	Ready	4.2.416.0-154	1769385105	File-Based Vulnerability Assess...	Windows	Jan 26, 2026 11:28:59 AM

If MD Cluster Control Center does not have access to the internet. The system administrator must use **MetaDefender Update Downloader** to download the modules. Once downloaded, the modules can be manually uploaded to MD Cluster Control Center.



Engine Name	Status	Version	Database	Type	Platform	Last Update
YARA	Ready	5.0.2-439	4.0.5-147	Yara	Windows	Jan 26, 2026 11:29:07 AM
YARA	Ready	5.0.2-439	4.0.5-147	Yara	Linux	Jan 26, 2026 11:29:06 AM
Vulnerability Scanning	Ready	4.2.416.0-154	1769385105	File-Based Vulnerability Assess...	Windows	Jan 26, 2026 11:28:59 AM



Engine Name	Status	Version	Database	Type	Platform	Last Update
YARA	Ready	5.0.2-439	4.0.5-147	Yara	Windows	Jan 26, 2026 11:29:07 AM
YARA	Ready	5.0.2-439	4.0.5-147	Yara	Linux	Jan 26, 2026 11:29:06 AM
Vulnerability Scanning	Ready			e-Based Vulnerability Assess...	Windows	Jan 26, 2026 11:28:59 AM
Vulnerability Scanning	Ready			e-Based Vulnerability Assess...	Linux	Jan 26, 2026 11:28:49 AM
Vir.IT eXplorer	Ready			i-Malware	Windows	Jan 26, 2026 11:29:14 AM
Threat Intelligence	Ready			reat Intelligence	Windows	Jan 26, 2026 11:29:02 AM
Threat Intelligence	Ready			reat Intelligence	Linux	Jan 26, 2026 11:29:01 AM
Sandbox Remote	Ready			ndbox Remote	Windows	Jan 26, 2026 11:28:34 AM
Sandbox Remote	Ready			ndbox Remote	Linux	Jan 26, 2026 11:28:34 AM
SBOM	Ready			OM	Windows	Jan 26, 2026 11:30:53 AM
SBOM	Ready			OM	Linux	Jan 26, 2026 11:30:42 AM
Reputation Engine	Ready	2.2.0-7	1769163149	Reputation Engine	Windows	Jan 26, 2026 11:29:05 AM

Module Update

Please select **package [.zip]** files and corresponding **descriptor [.yml]** files downloaded by the Offline Downloader Utility.

Package[.zip]
No file chosen.

Descriptor[.yml]
No file chosen.

Module Update Configuration

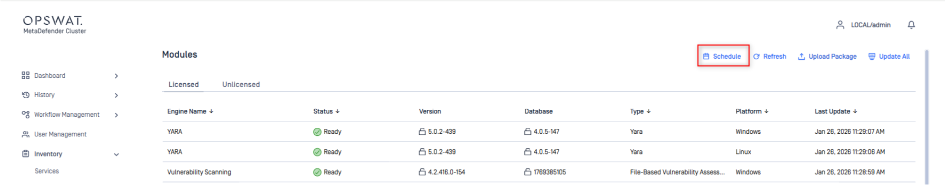
The system administrator can choose an update mechanism for the Modules.

- **Online:** Automatic Updates which will download modules from the internet.
- **Local Folder:** MD Cluster Control Center will pick up Module updates from a specific folder.
- **Offline:** Disable Automatic Updates.

Online

Choosing the Online method will allow MD Cluster Control Center to perform an automatic update by downloading the modules directly from the internet.

> **Warning**
>
> MD Cluster Control Center will need access to the following host to be able to download module updates:
>
> - https://update.dl.opswat.com Note: *IP address-based whitelisting on your firewall might fail after some time since OPSWAT uses CDN [Content Delivery Network] to faster delivery updates over the world, and IP address of edge servers might change over time.*

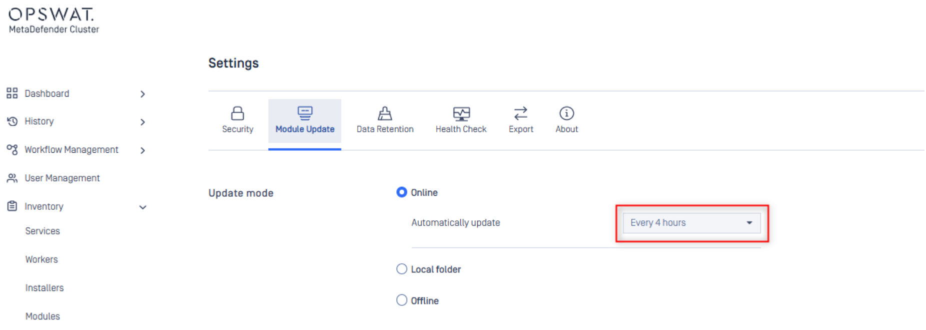


Modules

Licensed | Unlicensed

Engine Name	Status	Version	Database	Type	Platform	Last Update
YARA	Ready	5.0.2-439	4.0.5-147	Yara	Windows	Jan 26, 2026 11:29:07 AM
YARA	Ready	5.0.2-439	4.0.5-147	Yara	Linux	Jan 26, 2026 11:29:06 AM
Vulnerability Scanning	Ready	4.2.416.0-154	1769385105	File-Based Vulnerability Assess...	Windows	Jan 26, 2026 11:28:59 AM

MD Cluster Control Center will periodically check the latest version of modules every 4 hours.



Settings

Update mode: ● Online

Automatically update: Every 4 hours

○ Local folder

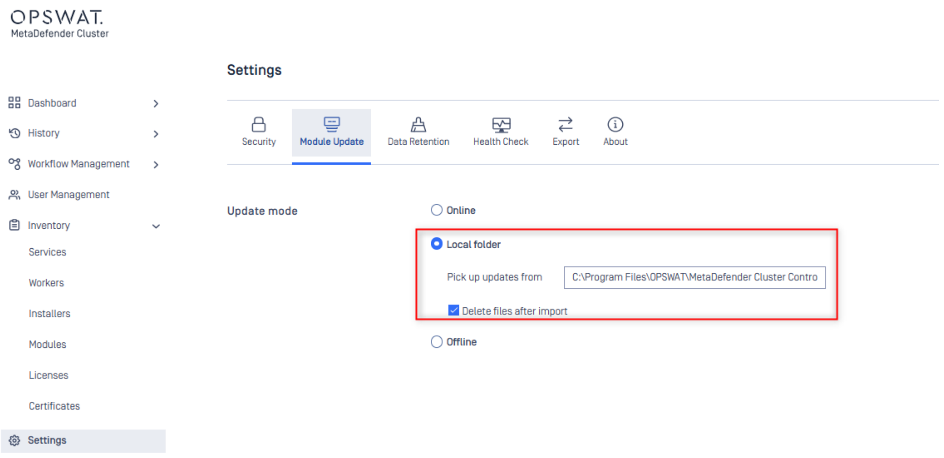
○ Offline

Local Folder

Choosing Local Folder method will allow MD Cluster Control Center to monitor any changes to the specified folder. If there are any changes, MD Cluster Control Center will apply the new module updates.

Info

MetaDefender Update Downloader must be used to download the modules.



Settings

Update mode

○ Online

● Local folder

Pick up updates from: C:\Program Files\OPSWAT\MetaDefender Cluster Contro

☑ Delete files after import

○ Offline

Offline

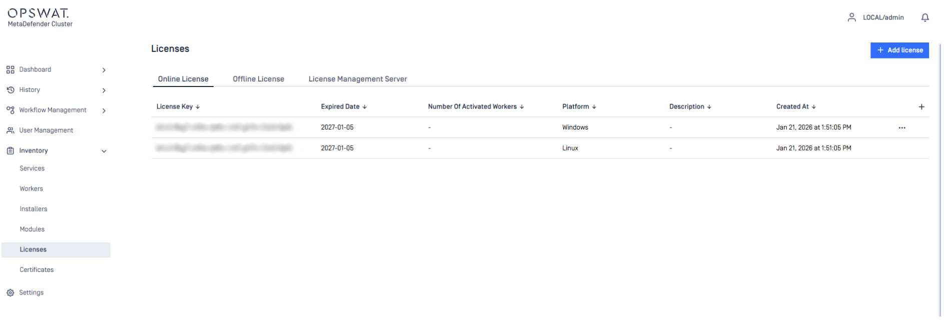
Choosing Offline method will turn off the module update mechanism.

Licenses

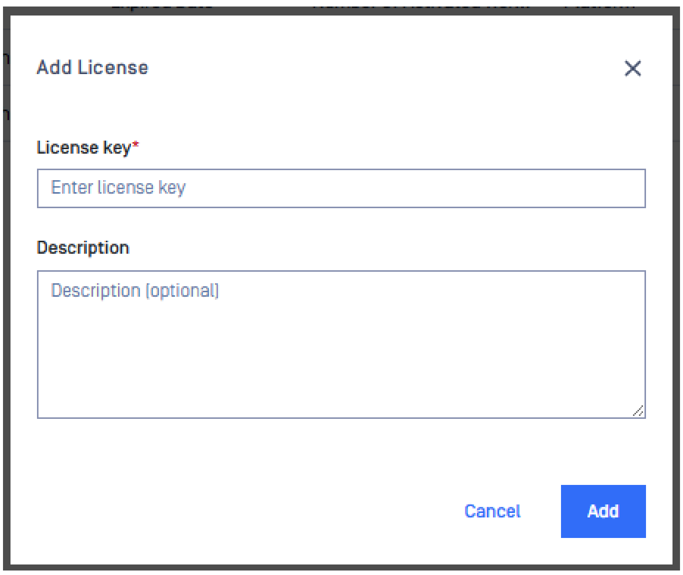
MetaDefender Core licenses can be added to this section. The system administrator can add multiple licenses but only one license can be used to activate MetaDefender Core.

Adding a License

To add your license, simply select `Add License` and enter the license key then select `Add`.



License Key ↓	Expired Date ↓	Number Of Activated Workers ↓	Platform ↓	Description ↓	Created At ↓
[redacted]	2027-01-05	-	Windows	-	Jan 21, 2026 at 1:51:05 PM
[redacted]	2027-01-05	-	Linux	-	Jan 21, 2026 at 1:51:05 PM



License key*

Enter license key

Description

Description (optional)

Online activation

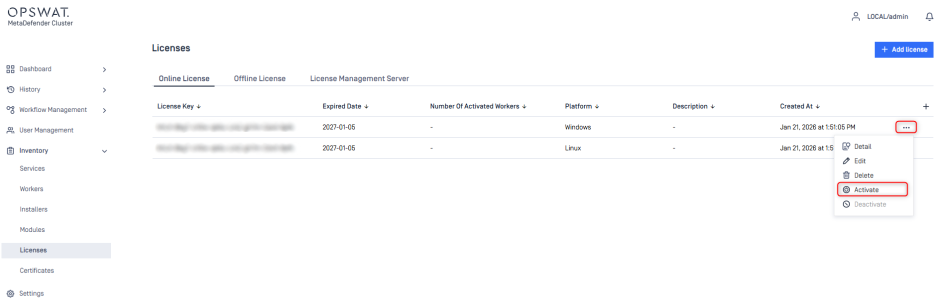
MD Cluster Control Center will connect directly to the OPSWAT licensing server if connected to the internet.

Warning

MD Cluster Control Center will need access to the following host for license activation:

- https://activation.dl.opswat.com Note: *IP address-based whitelisting on your firewall might fail after some time since OPSWAT uses CDN (Content Delivery Network) to faster delivery updates over the world, and IP address of edge servers might change over time.*

To activate the MetaDefender Core instances simply select the three dots to the right of the license and select `Activate`.



License Key ↓	Expired Date ↓	Number Of Activated Workers ↓	Platform ↓	Description ↓	Created At ↓
[redacted]	2027-01-05	-	Windows	-	Jan 21, 2026 at 1:51:05 PM
[redacted]	2027-01-05	-	Linux	-	Jan 21, 2026 at 1:5

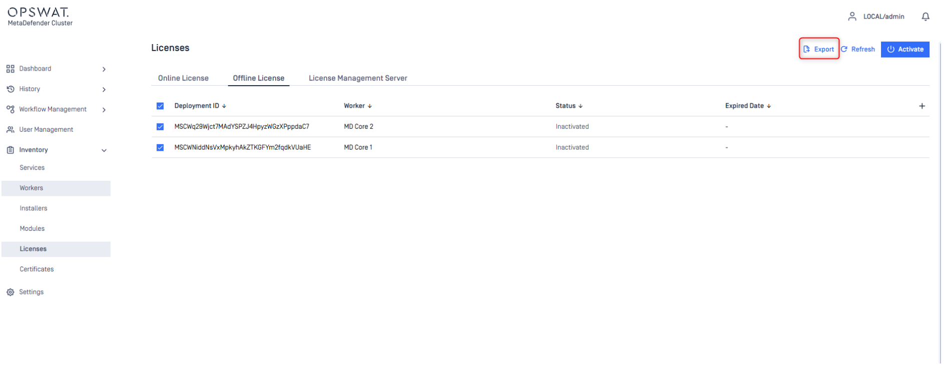
Offline activation

With no internet connection on MD Cluster Control Center, MetaDefender Core instances can be activated indirectly from a different machine that has internet connection. The system administrator must obtain all DeploymentID's of the MetaDefender Cores and then activate each DeploymentID separately to obtain their license file from My OPSWAT.

Once the system administrator obtains all the license files. Simply activate the MetaDefender Core instances on MD Cluster Control Center.

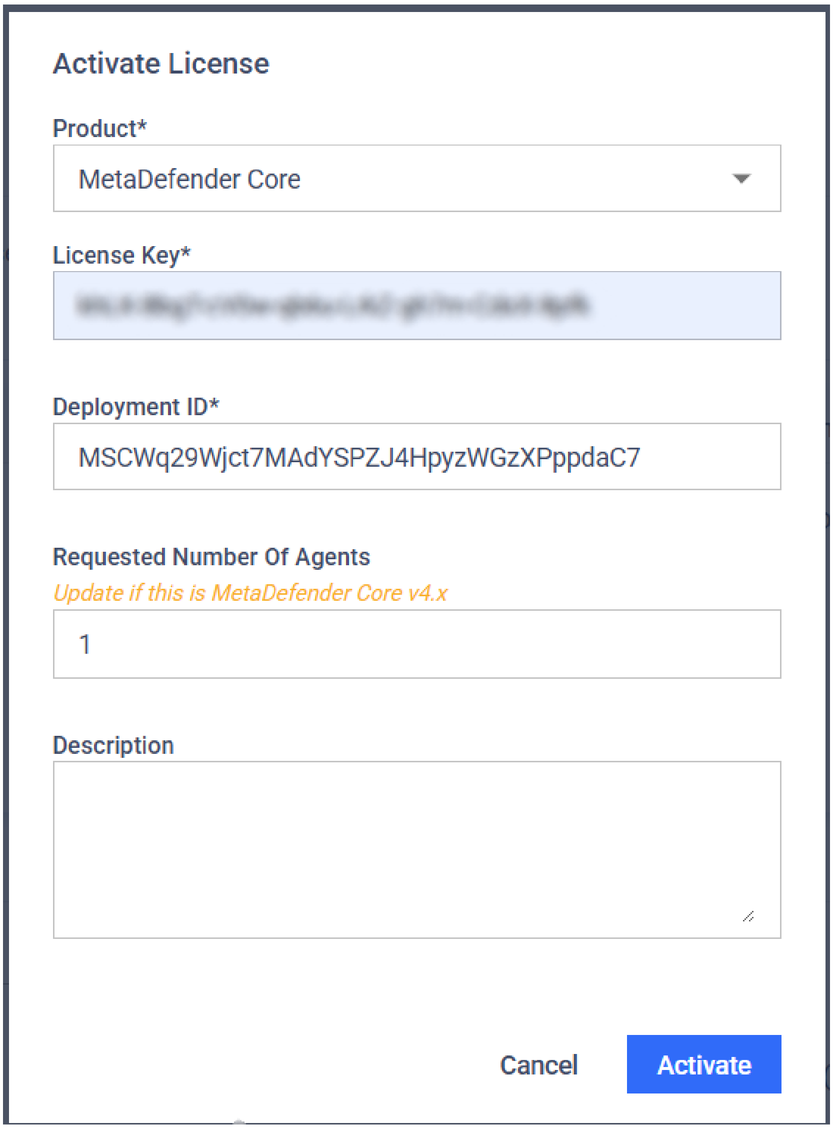
Follow the steps below to activate MetaDefender Core:

- Select a list of Deployment ID to activate then select `Export`. This will allow you to download a text file containing all the DeploymentID's that were selected.



☑	Deployment ID ↓	Worker ↓	Status ↓	Expired Date ↓
☑	MSCWq29Wjct7MAdYSPZJ4HpyzWGzXPppdaC7	MD Core 2	Inactivated	-
☑	MSCWNiddNsVxMpkyhAkZTKGFYm2fqdkVUaHE	MD Core 1	Inactivated	-

- Open the exported file to reveal all the selected DeploymentID.
- Sign in to My OPSWAT and then head to `License Management` -> `Activate License` and for each DeploymentID, click `Activate` and download the license files.



Activate License

Product*

MetaDefender Core

License Key*

[redacted]

Deployment ID*

MSCWq29Wjct7MAdYSPZJ4HpyzWGzXPppdaC7

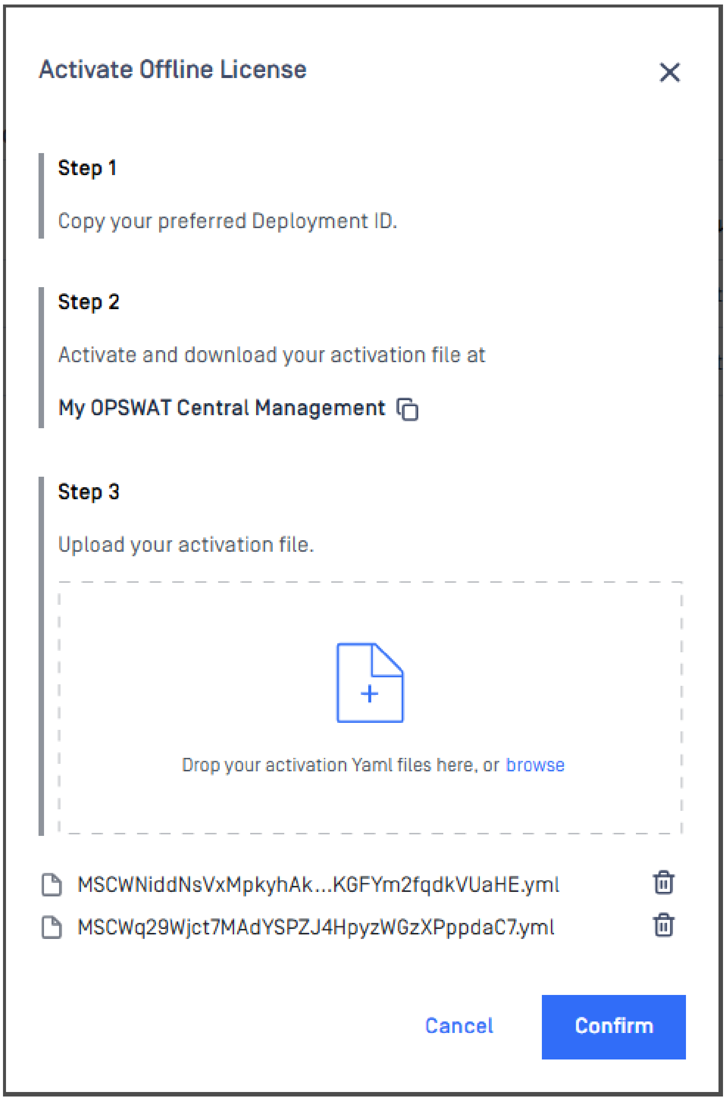
Requested Number Of Agents

Update if this is MetaDefender Core v4.x

1

Description

- Once all the license files are obtained. Head back to MD Cluster Control Center -> `Inventory` -> `Licenses` -> `Offline Licenses`. Click `Activate` to upload activation files to activate.



Activate Offline License

Step 1

Copy your preferred Deployment ID.

Step 2

Activate and download your activation file at

My OPSWAT Central Management

Step 3

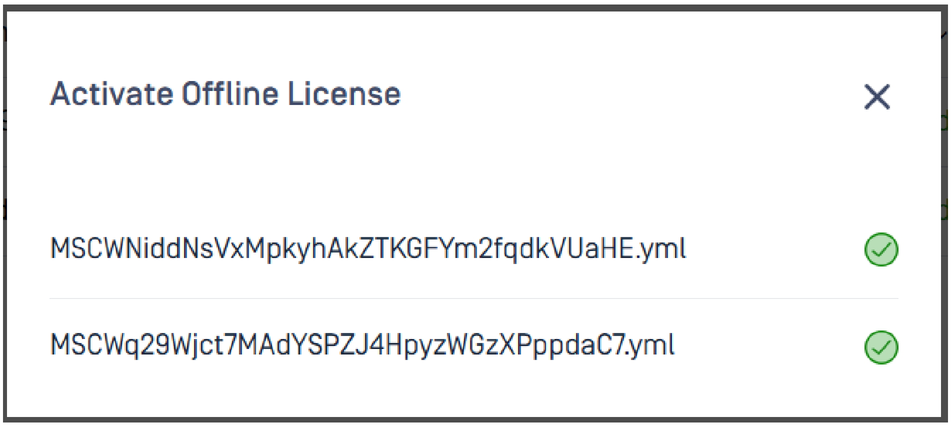
Upload your activation file.

Drop your activation Yaml files here, or browse

MSCWNiddNsVxMpkyhAk...KGFYm2fqdkVUaHE.yml

MSCWq29Wjct7MAdYSPZJ4HpyzWGzXPppdaC7.yml

- By clicking `Confirm`, MD Cluster **Control Center** will verify the validity of the activation files and start activating MetaDefender Core accordingly.



Activate Offline License

MSCWNiddNsVxMpkyhAkZTKGFYm2fqdkVUaHE.yml ✓

MSCWq29Wjct7MAdYSPZJ4HpyzWGzXPppdaC7.yml ✓

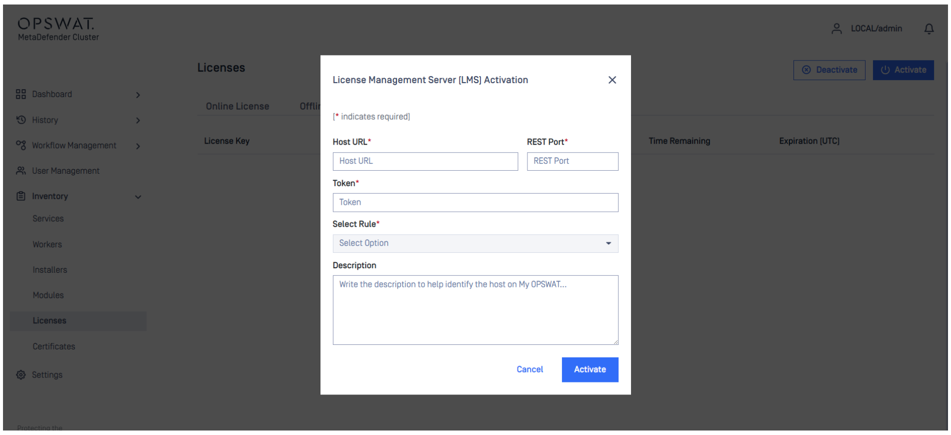
License Management Server activation

Using this licensing model, the License Management Server will act as a dedicated server that manages all license operations for MetaDefender Core. It is deisgned for organizations that prefer or required on-prem or cloud based solutions.

The License Management Server will handle the full license lifecycle - from activation to deactivation, renewal.

Follow the steps below to connect and activate with License Management Server:

- Sign in to MetaDefender Cluster **Control Center** console.
- Go to `Inventory` > `Licenses` and select License Management Server tab.
- Select `Activate` and provide the necessary information in the required fields:
  - **Host URL**: The URL of the License Management Server to connect to.
  - **REST Port**: The port of the License Management Server.
  - **Token**: Access token obtained from the License Management Server.



License Management Server (LMS) Activation

(* indicates required)

Host URL*: Host URL

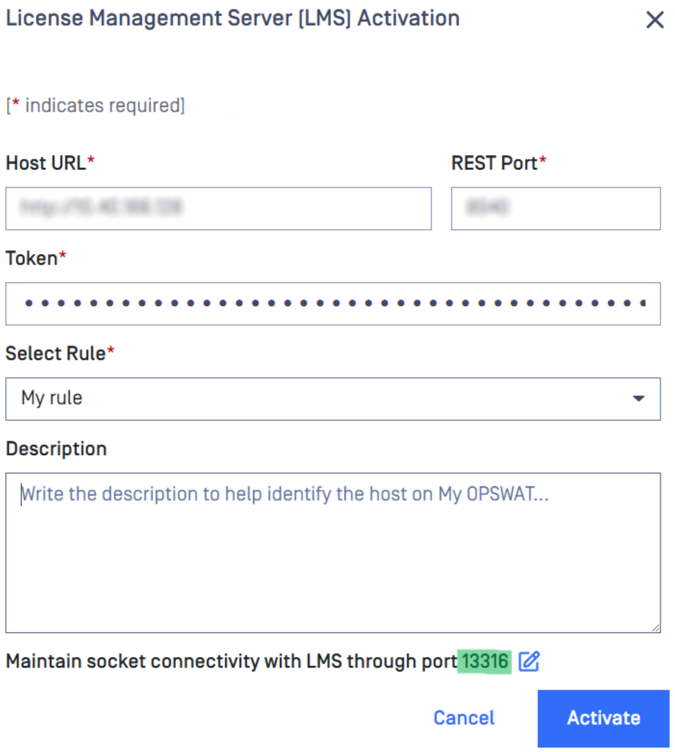
REST Port*: REST Port

Token*: Token

Select Rule*: Select Option

Description: Write the description to help identify the host on My OPSWAT...

- After input all required fields, the connection to LMS will be established and available rules can be selected under `Select Rules`.



License Management Server (LMS) Activation

[* indicates required]

Host URL*: [redacted]

REST Port*: [redacted]

Token*: ••

Select Rule*: My rule

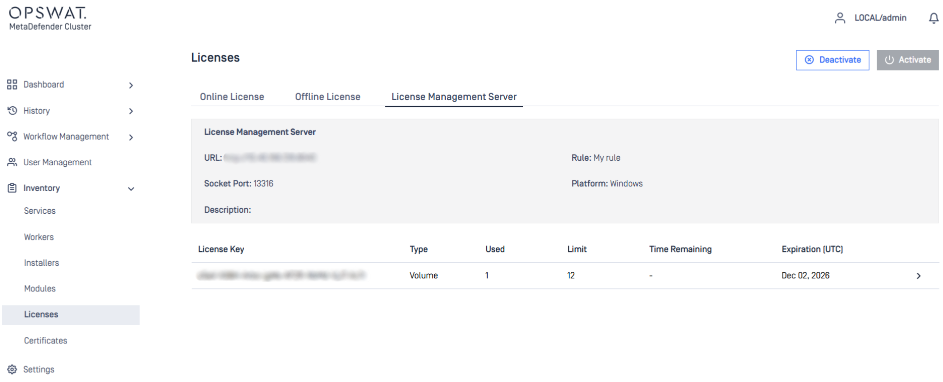
Description: Write the description to help identify the host on My OPSWAT...

Maintain socket connectivity with LMS through port 13316

Info

The port number defined in "Maintain socket connectivity with LMS through port `<port_number>`" is required to sustain connectivity between the Control Center and License Management Server. In order for successfully activation, **confirm that the port is properly configured and allowed in firewall setting**.

- Select the appropriate rule and choose Activate. Upon successful completion, the license details will be shown.



Licenses

License Management Server

URL: [redacted]

Rule: My rule

Socket Port: 13316

Platform: Windows

Description:

License Key	Type	Used	Limit	Time Remaining	Expiration (UTC)
[redacted]	Volume	1	12	-	Dec 02, 2026

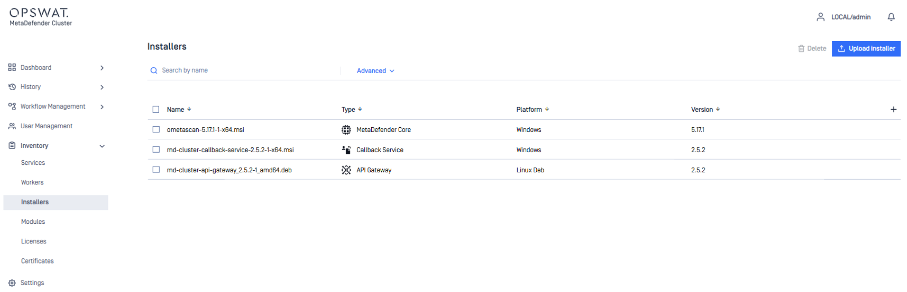
ⓘ Info

Once activation is successful, the License Management Server will manage the license status of all MetaDefender Core instances. Please ensure sufficient quota is available.

- Check instance status under `Inventory` > `Workers` and confirm all MetaDefender Core instances is activated successfully.

Installers

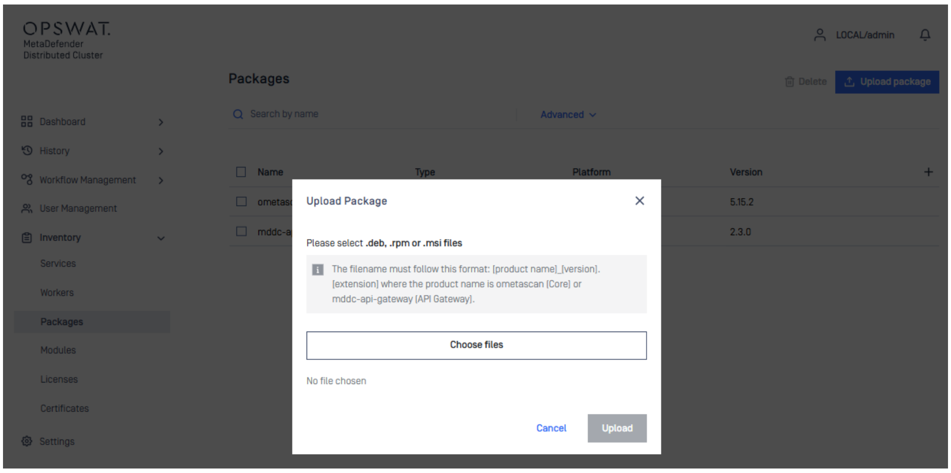
The Installers section lists all MetaDefender Cluster (MD Cluster) API Gateway, MD Cluster Callback Service and MetaDefender Core installers that have been recorded so far. These installers are used when deplying to a Worker.



Installers

□	Name	Type	Platform	Version
□	ometascan-5.17.1-1-x64.msi	MetaDefender Core	Windows	5.17.1
□	md-cluster-callback-service-2.5.2-1-x64.msi	Callback Service	Windows	2.5.2
□	md-cluster-api-gateway_2.5.2-1_amd64.deb	API Gateway	Linux Deb	2.5.2

Once OPSWAT releases a new version of the MD Cluster API Gateway, MD Cluster Callback Service and MetaDefender Core. The system administrator needs to download the installers from the OPSWAT portal and upload them to the MD Cluster Control Center by clicking `Upload Package` button in the top right corner.



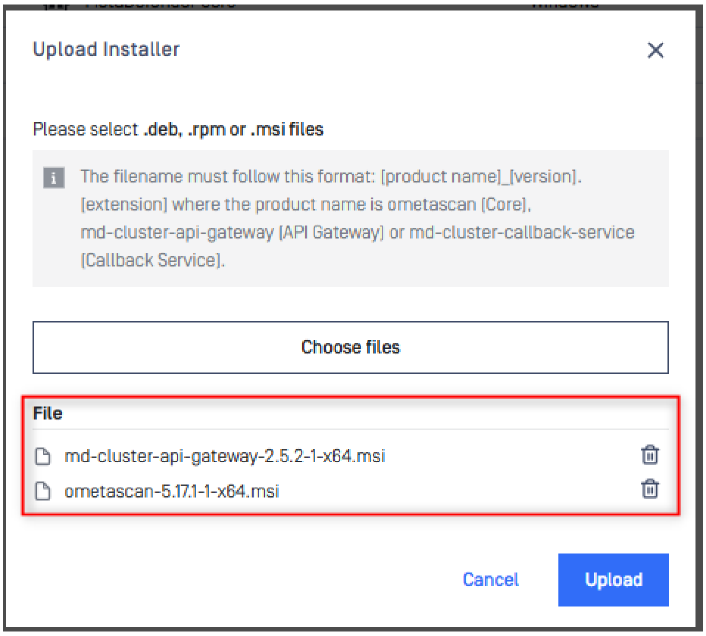
Upload Package

Please select **.deb, .rpm or .msi files**

The filename must follow this format: [product name]_[version].[extension] where the product name is ometascan (Core) or mddc-api-gateway (API Gateway).

No file chosen

The system administrator can upload several installer files at once.



Upload Installer

Please select **.deb, .rpm or .msi files**

The filename must follow this format: [product name]_[version].[extension] where the product name is ometascan (Core), md-cluster-api-gateway (API Gateway) or md-cluster-callback-service (Callback Service).

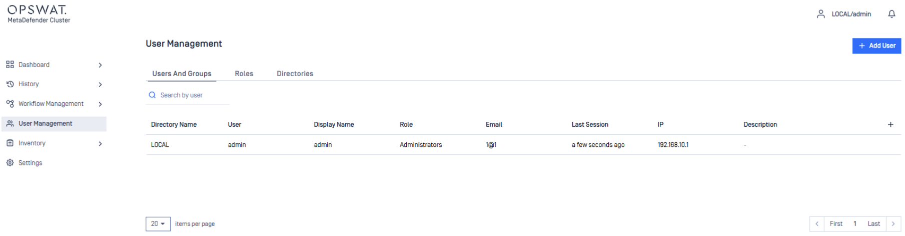
File
md-cluster-api-gateway-2.5.2-1-x64.msi
ometascan-5.17.1-1-x64.msi

Installer files are uploaded to the MD Cluster File Storage and will be ready for deployment or upgrading.

User Management

Users And Groups

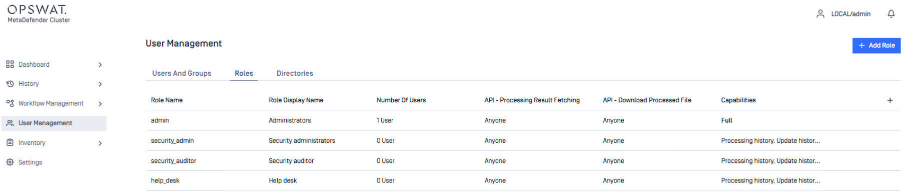
This section lists the existing users for MetaDefender Cluster. A user can be added in this section and can be assigned a specific role.



Directory Name	User	Display Name	Role	Email	Last Session	IP	Description
LOCAL	admin	admin	Administrators	1@1	a few seconds ago	192.168.10.1	-

Roles

This section list the existing roles available for MetaDefender Cluster. Each role has specific permission and can be assigned to a user.

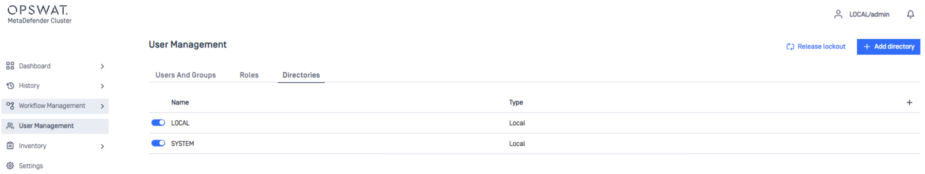


Role Name	Role Display Name	Number Of Users	API - Processing Result Fetching	API - Download Processed File	Capabilities
admin	Administrators	1 User	Anyone	Anyone	Full
security_admin	Security administrators	0 User	Anyone	Anyone	Processing history, Update histor...
security_auditor	Security auditor	0 User	Anyone	Anyone	Processing history, Update histor...
help_desk	Help desk	0 User	Anyone	Anyone	Processing history, Update histor...

Directories

This section lists the existing directories for MetaDefender Cluster. Each directory has a specific permission that will enforce the following login policies:

- Number of failed logins before lockout.
- Lockout time (minutes).



Name	Type
LOCAL	Local
SYSTEM	Local

Settings

Additional MetaDefender Cluster Control Control settings can be found in this section. This ranges from Data Retention, Generating Support Packages, Password and Session policies, Module Updates and Health Checks.

- Security.
- Data Retention.
- Export.

Security

This section will allow the system administrator to enable secure connections to MetaDefender Cluster Control Center if required. In addition, Password and Session policies are set in this section.

Password Policy

Info

These password policies changes only apply to new user creations and future password changes. Existing users' passwords are unaffected.

Local users' password can be enforced to meet requirements set by administrators, which includes following constraints:

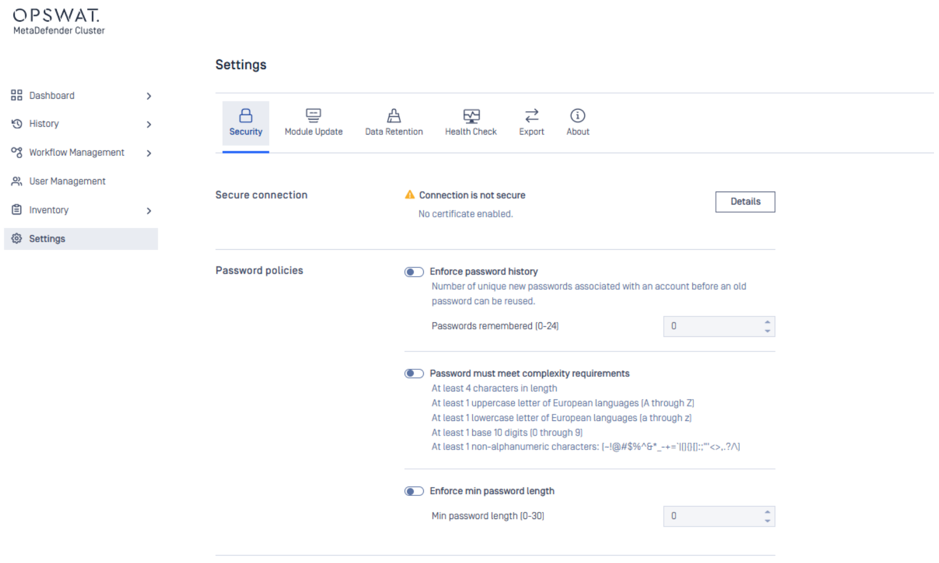
- **Enforce password policy:**
  - Determines the number of unique new passwords that must be associated with a user account before an old password can be reused.
  - Range: (0-24).
  - Default: 0 (to disable enforcement).
- **Password must meet complexity requirements:**
  - Determines whether passwords must meet a series of guidelines that are considered important for a strong password.
  - Default: unchecked

Complexity requirements

- At least 4 characters in length.
- At least 1 uppercase letter of European languages (A through Z).
- At least 1 lowercase letter of European languages (a through z).
- At least 1 base 10 digits (0 through 9).
- At least 1 non-alphanumeric characters (special characters): (~!@#$%^&*_-+=`|(){}[]:;"'<>,.?/).

- **Minimum password length:**

- The least number of characters that can make up a password for a user account.
- Range: [0-30].
- Default: 0 (to disable enforcement).



Settings

Secure connection

⚠ Connection is not secure

No certificate enabled.

Password policies

○ Enforce password history

Number of unique new passwords associated with an account before an old password can be reused.

Passwords remembered (0-24): 0

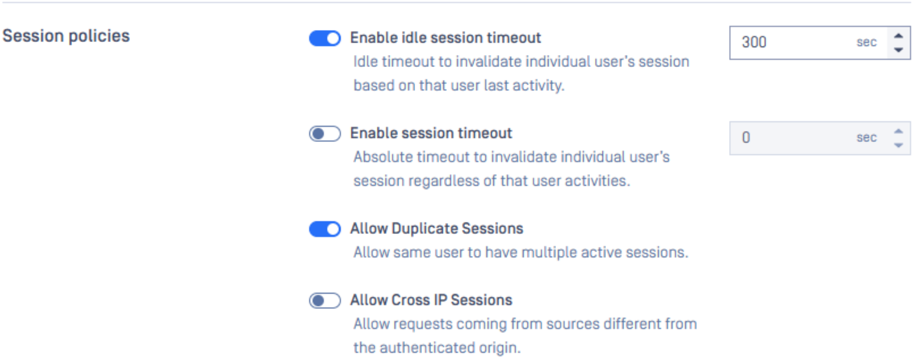
○ Password must meet complexity requirements

At least 4 characters in length
At least 1 uppercase letter of European languages (A through Z)
At least 1 lowercase letter of European languages (a through z)
At least 1 base 10 digits (0 through 9)
At least 1 non-alphanumeric characters: (~!@#$%^&*_-+=`|(){}[]:;"'<>,.?/\)

○ Enforce min password length

Min password length (0-30): 0

Session Policy



Session policies

● Enable idle session timeout: 300 sec

Idle timeout to invalidate individual user's session based on that user last activity.

○ Enable session timeout: 0 sec

Absolute timeout to invalidate individual user's session regardless of that user activities.

● Allow Duplicate Sessions

Allow same user to have multiple active sessions.

○ Allow Cross IP Sessions

Allow requests coming from sources different from the authenticated origin.

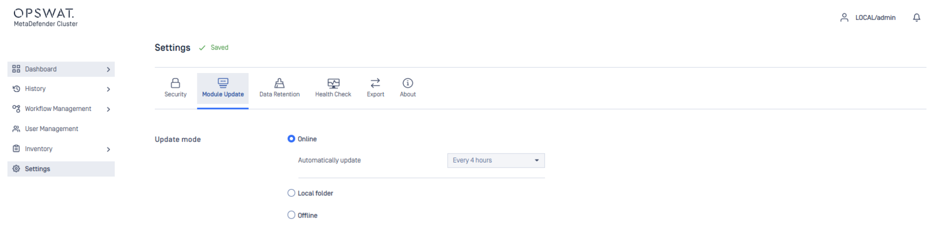
- Idle session timeout: Idle timeout to invalidate individual user's session based on that user last activity.
- Session timeout: Absolute timeout to invalidate individual user's session regardless of that user activities.
- Allow Duplicate Sessions: Allow same user to have multiple active sessions.
- Allow Cross IP Sessions: Allow requests coming from sources different from the authenticated origin.

Module update

This section will allow the system administrator to select one of three update modes offered by MetaDefender Cluster (MD Cluster).

Online

In this mode, MD Cluster **Control Center** will autonomously download the latest module packages from OPSWAT online update infrastructure, repeating this process every four hours by default.



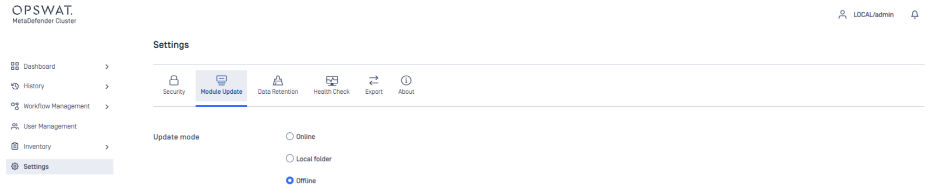
Settings ✓ Saved

Update mode
● Online
Automatically update: Every 4 hours
○ Local folder
○ Offline

Offline

In this mode, the administrators must download the licensed engine packages from **MetaDefender Update Downloader** and upload them manually to MD Cluster **Control Center**.

Refer to the steps for manually uploading of the packages in MD Cluster **Control Center**.

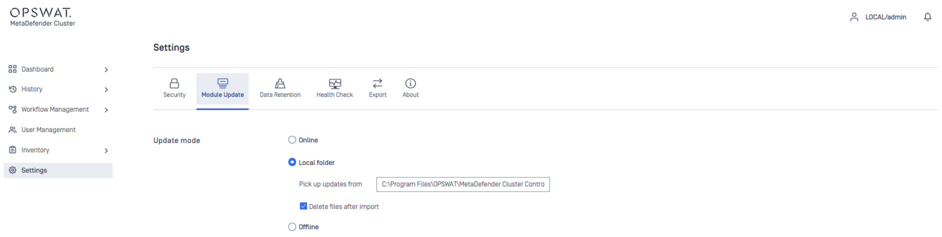


Settings

Update mode
○ Online
○ Local folder
● Offline

Local folder

In this mode, the administrators need to input the path to a folder where module packages will be added. MD Cluster **Control Center** subsequently gathers the packages from there and starts the module update process.



Settings

Update mode

○ Online

● Local folder

Pick up updates from: C:\Program Files\OPSWAT\MetaDefender Cluster Contro

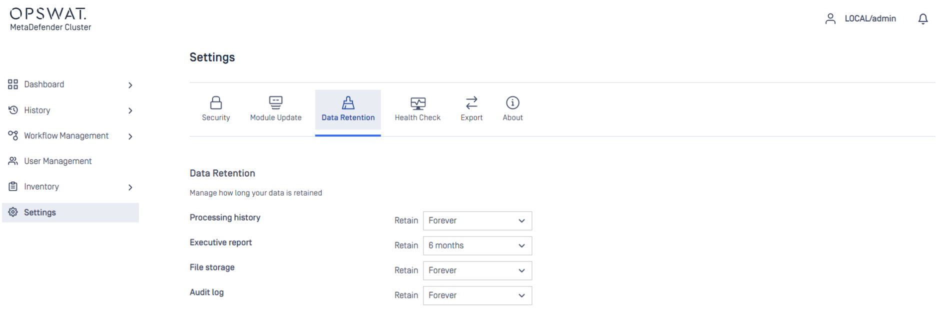
☑ Delete files after import

○ Offline

`Delete files after import` option can be selected so that MD Cluster **Control Center** can wipe all packages upon success.

Data retention

This section allows the system administrator to configure Data Retention settings to automatically delete data after a period of time.



Settings

Data Retention

Manage how long your data is retained

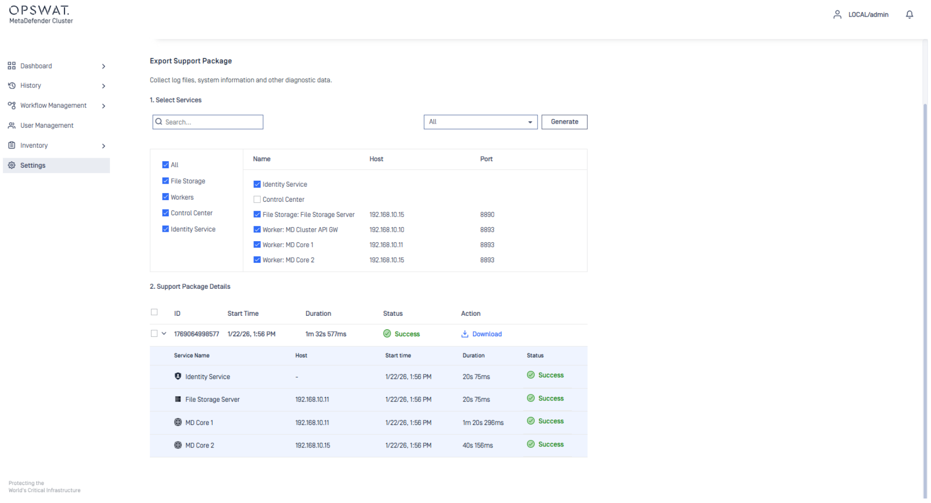
Processing history	Retain Forever
Executive report	Retain 6 months
File storage	Retain Forever
Audit log	Retain Forever

Export

This section will allow the system administrator to export MetaDefender Cluster (MD Cluster) support packages. The support package contains all relevant information to help us diagnose the issue of MetaDefender Cluster.

> **Warning**
>
> MD Cluster Support Package does not include logs from services like Redis, RabbitMQ and PostgreSQL. To obtain the service logs, please click here.

- Simply select all to select all MD Cluster components and click the `Generate` button. A new ID and notification will be shown in the `Support Package Details.`



Export Support Package

Collect log files, system information and other diagnostic data.

1. Select Services

☑ All
☑ File Storage
☑ Workers
☑ Control Center
☑ Identity Service

Name	Host	Port
☑ Identity Service		
□ Control Center		
☑ File Storage: File Storage Server	192.168.10.15	8890
☑ Worker: MD Cluster API GW	192.168.10.10	8893
☑ Worker: MD Core 1	192.168.10.11	8893
☑ Worker: MD Core 2	192.168.10.15	8893

2. Support Package Details

□	ID	Start Time	Duration	Status	Action
□	1769064998577	1/22/26, 1:56 PM	1m 32s 577ms	Success	Download

Service Name	Host	Start time	Duration	Status
Identity Service	-	1/22/26, 1:56 PM	20s 75ms	Success
File Storage Server	192.168.10.11	1/22/26, 1:56 PM	20s 75ms	Success
MD Core 1	192.168.10.11	1/22/26, 1:56 PM	1m 20s 296ms	Success
MD Core 2	192.168.10.15	1/22/26, 1:56 PM	40s 156ms	Success

- Once the support package generation is finished, simply select `Download` in the `Action` column in the `Support Package Details.`

Performance and Load Estimation

Disclaimer

These results should be viewed as guidelines and not performance guarantees, since there are many variables that affect performance (file set, network configurations, hardware characteristics, etc.). If throughput is important to your implementation, OPSWAT recommends site-specific benchmarking before implementing a production solution.

Factors that affect performance

- MetaDefender Core version
- MetaDefender Core engine package and configuration
    - set of engines (which and how many)
    - product configuration (e.g., thread pool size)
- MetaDefender Cluster API Gateway version
- System environment
    - server profile (CPU, RAM, hard disk)
    - client application location - remote or local
    - system caching and engine level caching
- Dataset
    - encrypted or decrypted
    - file types
        - different file types (e.g., document, image, executable)
        - archive file or compound document format files
    - file size
    - bad or unknown (assume to be clean)
- Performance tool

Performance metrics

While processing files on the system, service performance is measured by various metrics. Some of them are commonly used to define performance levels, including:

Performance metrics	Description
Number of processed **objects** per hour vs. Number of processed **files** per hour	On MetaDefender Core, meaning of "files" and "objects" are not the same. - "files": exclusively refers to original files submitted to MetaDefender Core. These could be either archive or non-archive file formats. For archives, depending on archive handling settings, MetaDefender Core may need to extract them and process all nested files inside as well. For example, one archive file could contain millions of nested files inside. - "objects": refers to any individual files that MetaDefender Core must process. These could be separate original files submitted to MetaDefender Core, or extracted files coming from an archive. The number of processed objects is considered to be a more accurate throughput metric to measure MetaDefender Core performance. The primary metric used to measure average vs peak throughput of a MetaDefender Core system is "processed objects per hour."
Submission load (number of successful requests per second)	This performance metric measures the load generated by a test client application that simulates loads submitted to MetaDefender Core. A submission is considered successful when the client app submits a file to MetaDefender Core and receives a dataID, which indicates that the file has successfully been added to the Queue. Submission load should measure both average and peak loads.
Average processing time per object	The primary metric used to measure processing time of a MetaDefender Core system is "avg processing time (seconds/object)."

Performance metrics	Description
Total processing time (against certain data set)	Total processing time is a typical performance metric to measure the time it takes to complete the processing of a whole dataset.

How test results are calculated

Performance (mainly scanning speed) is measured by throughput rather than unit speed. For example, if it takes 10 seconds to process 1 object, and it also takes 10 seconds to process 10 objects, then performance is quantified as 1 second per object, rather than 10 seconds.

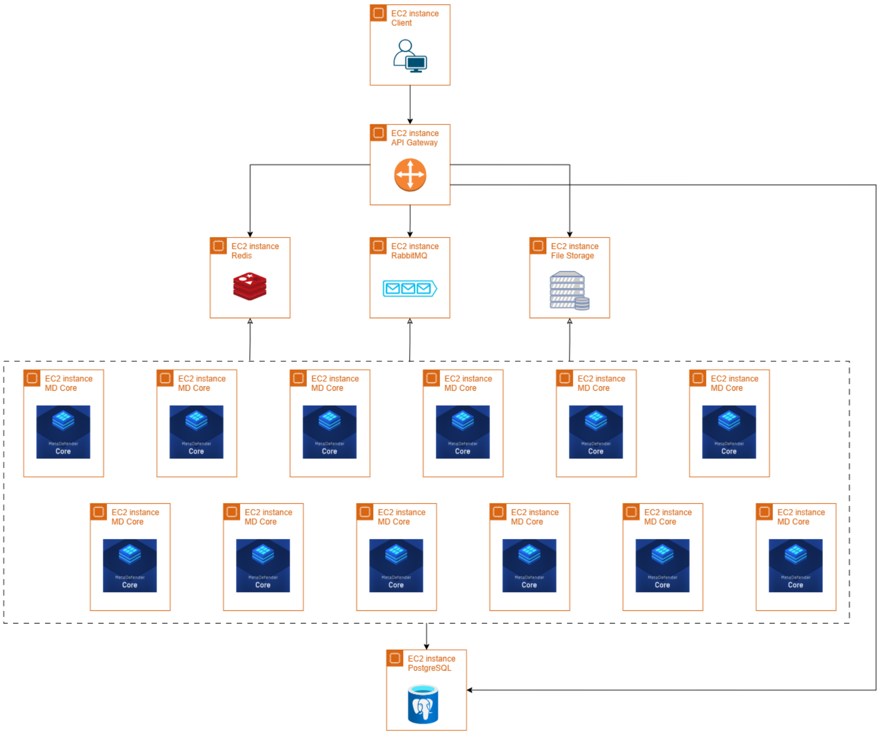
- total time / total number of objects processed: 10 seconds / 10 objects = 1 second / object.

Dataset

File category	File type	Number of files	Total size	Average file size
Document	DOC	3,820	534 MB	0.14 MB
Medium archive files	RPM CAB EXE	50	Compressed size: 2.8 GB Extracted size: 12.09 GB	Compressed size: 56.02 MB Extracted size: 0.036 MB
Big archive files	CAB	4	Compressed size: 2.9 GB Extracted size: 124 GB	Compressed size: 715 MB

Environment

Topology



Using AWS environment with the specification below:

MD Cluster system

	MD Core	File Storage	API Gateway	PostgreSQL	RabbitMQ	Redis
OS	Windows Server 2022	Rocky Linux 9	Rocky Linux 9	Rocky Linux 9	Rocky Linux 9	Rocky Linux
AWS instance type	c5.2xlarge	c5n.4xlarge	c5n.2xlarge	c5.xlarge	c5.xlarge	c5.xlar
vCPU	8	16	4	4	4	4
Memory	16GB	32GB	8GB	8GB	8GB	32GB
Disk Type IOPS Throughput Size	gp3 3000 125MB/s 100GB	gp3 12000 1000MB/s 150GB	gp3 3000 256MB/s 100GB	gp3 10000 550MB/s 100GB	gp3 3000 125MB/s 80GB	gp3 3000 125MB/ 80GB
Network bandwidth (baseline & burst)	2.5 Gbps 10 Gbps	15 Gbps 25 Gbps	5 Gbps 25 Gbps	1.25 Gbps 10 Gbps	1.25 Gbps 10 Gbps	1.25 Gbps 10 Gbp
Benchmark (Geekbench)	EC2 c5.2xlarge	EC2 c5n.4xlarge	EC2 c5n.2xlarge	EC2 c5.xlarge	EC2 c5.xlarge	EC2 c5.xlar

Client tool

	Detail
OS	Rocky Linux 9
AWS instance type	c5n.xlarge
vCPU	4
Memory	10GB
Disk	Type: gp3 IOPS: 3000 Throughput: 125MB/s Size: 80GB
Network bandwidth	Baseline: 5 Gbps Burst: 10 Gbps

Product information

- MetaDefender Core v5.14.2
- Engines:
  - Metascan 8: Ahnlab, Avira, ClamAV, ESET, Bitdefender, K7, Quick Heal, VirIT Explorer
  - Archive v7.4.0
  - File type analysis v7.4.0
- MD Cluster Control Center v2.0.0
- MD Cluster API Gateway v2.0.0
- MD Cluster File Storage v2.0.0
- PostgreSQL v14.17
- RabbitMQ v3.12.6
- Redis v7.2.1

MetaDefender Core settings

General settings

- Turn off data retention
- Turn off engine update
- Scan queue: 1000 (for Load Balancer deployment)

Archive Extraction settings

- Max recursion level: 99999999
- Max number of extracted files: 99999999
- Max total size of extracted files: 99999999
- Timeout: 10 minutes
- Handle archive extraction task as Failed: true
    - Extracted partially: true

Metascan settings

- Max file size: 99999999
- Scan timeout: 10 minutes
- Per engine scan timeout: 1 minutes

Advanced settings

RabbitMQ

- RABBITMQ_SERVER_ADDITIONAL_ERL_ARGS=-rabbit consumer_timeout unlimited default_consumer_prefetch {false,525}

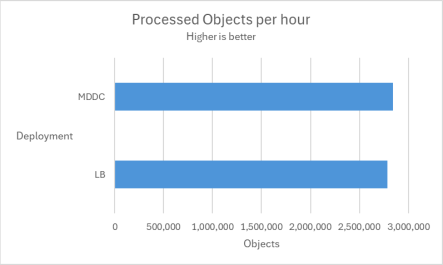
Redis

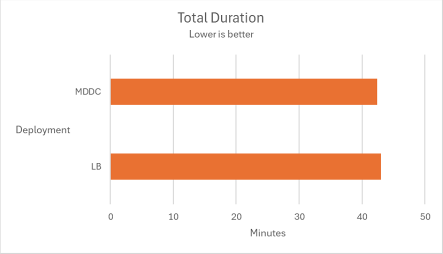
- redis-cli flushall
- redis-cli config set save ''
- redis-cli config set maxmemory 25gb
- redis-cli config set maxmemory-policy volatile-ttl

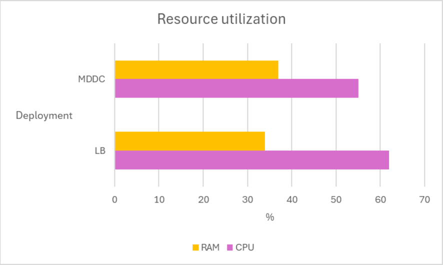
Performance results

Load-balance deployment vs MD Cluster deployment

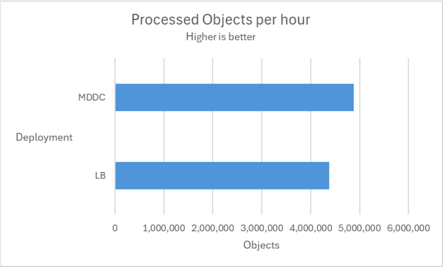
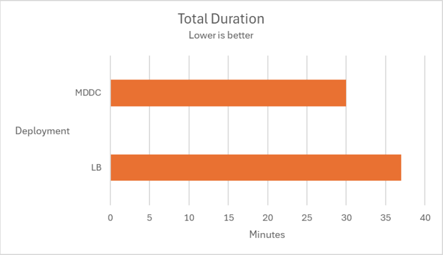
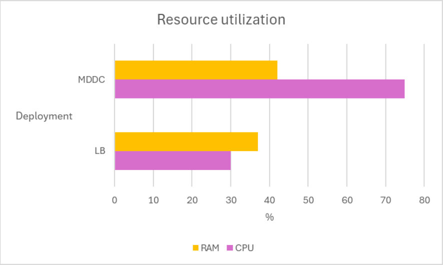
Multiple tests are conducted using 12 MetaDefender Core instances across two deployment types, MetaDefender Cluster (MD Cluster) and Load Balancer, to determine the superiority of the MD Cluster in 4 different datasets.

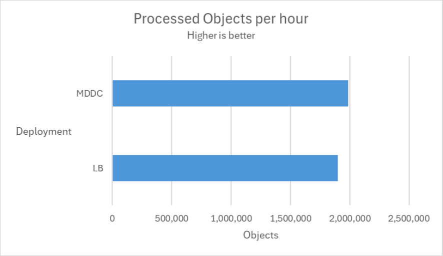
<table>
<tr><th>Scenario</th><th>Result</th></tr>
<tr><td>Aggressively submitted 2M non-archive files at a rate of 800 files per second.</td><td>


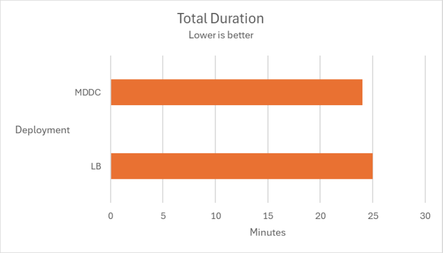


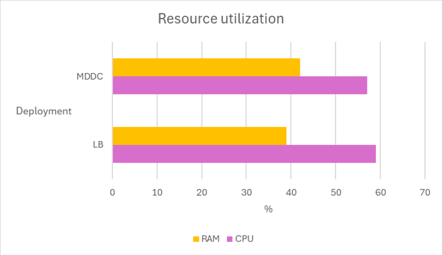


</td></tr>
</table>

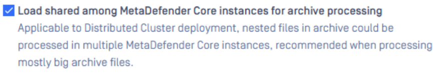
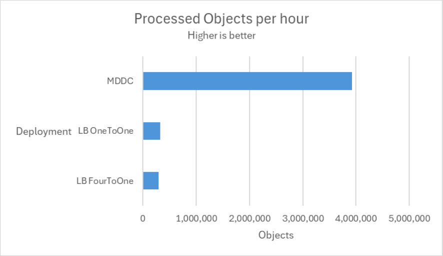
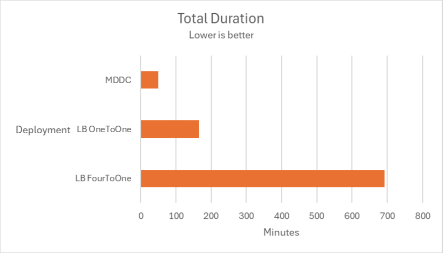
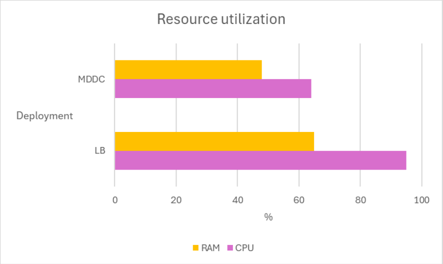
Scenario	Result
Submitted 400 medium archive files at a rate of 1 files per second.	  

<table>
<tr><th>Scenario</th><th>Result</th></tr>
<tr><td>Submitted a mix of 189K non-archive and medium archive files at a rate of 180 files per second.</td><td>




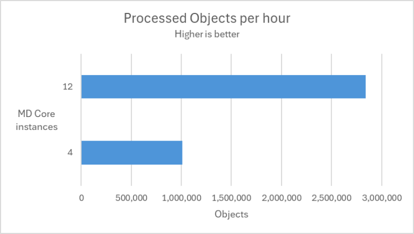


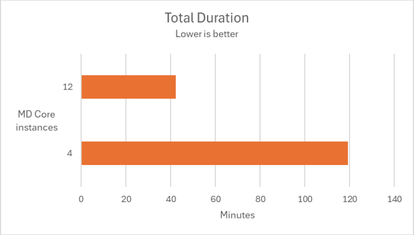
</td></tr>
</table>

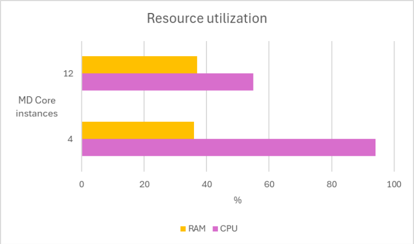
Scenario	Result
Submitted 4 large CAB files. The scenarios replicate 2 different routing cases of a common Load Balancer. ***LB OneToOne***: An ideal routing ensures that one CAB file is routed to a single MD Core. ***LB FourToOne***: The worst routing that delivered four CAB files to a single MD Core. # **Archive distribution** In workflow, setting "Load shared among MetaDefender Core instances for archive processing" is enabled.  ☑ **Load shared among MetaDefender Core instances for archive processing** Applicable to Distributed Cluster deployment, nested files in archive could be processed in multiple MetaDefender Core instances, recommended when processing mostly big archive files.	  

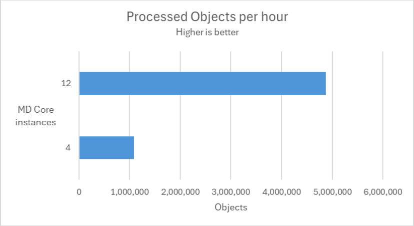
Scaling out

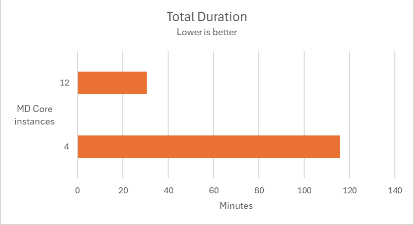
In the following test scenarios, we conducted experiments on four datasets using 4 and 12 of MD Core instances in MetaDefender Cluster (MD Cluster), demonstrating the benefits of increased instance counts.

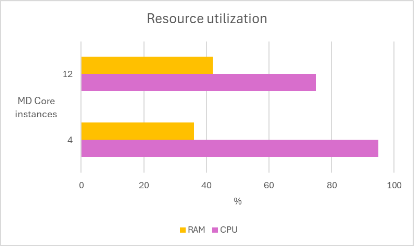
<table>
<tr><th>Scenario</th><th>Result</th></tr>
<tr><td>Aggressively submitted 2M non-archive files at a rate of 800 files per second.</td><td>




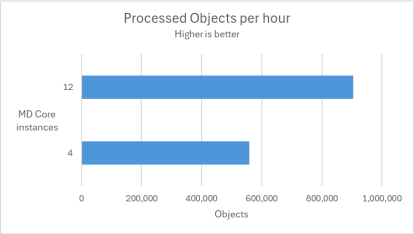


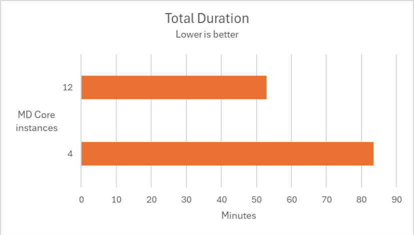
</td></tr>
<tr><td>Submitted 400 medium archive files at a rate of 1 files per second.</td><td>


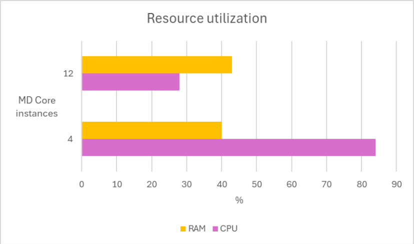




</td></tr>
</table>

<table>
<tr><th>Scenario</th><th>Result</th></tr>
<tr><td>Submitted a mix of 189K non-archive and medium archive files at a rate of 60 files per second.</td><td>


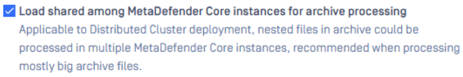


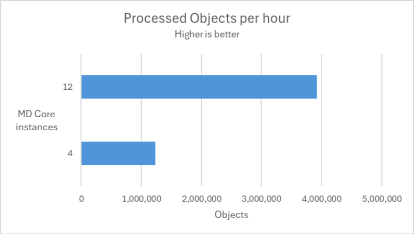


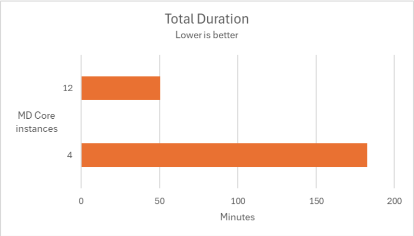
</td></tr>
<tr><td>Submitted 4 large CAB files.

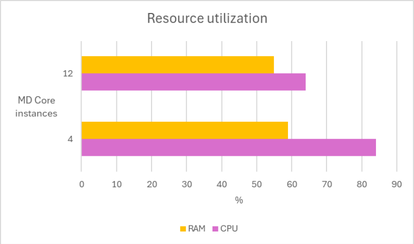
<b>Archive distribution</b>

In workflow, setting "Load shared among MetaDefender Core instances for archive processing" is enabled.



☑ Load shared among MetaDefender Core instances for archive processing
Applicable to Distributed Cluster deployment, nested files in archive could be processed in multiple MetaDefender Core instances, recommended when processing mostly big archive files.
</td><td>






</td></tr>
</table>

Log Gathering in MetaDefender Cluster

Download support packages

From the web console of MetaDefender Cluster (MD Cluster), the administrator can easily download the support packages of the following services:

- MD Cluster **Control Center**
- MD Cluster **Identity Service**
- MD Cluster **File Storage**
- MD Cluster **Worker** including MD Cluster **API Gateway**, MD Cluster **Callback Service** or **MetaDefender Core** deployed by the worker.

Please refer to Remote Support Package Gathering for more information.

Collect service logs

Logs from the services Redis, RabbitMQ, and PostgreSQL need to be collected manually.

Redis - Caching Server

ℹ Info

Redis caching server is officially supported on Linux.

1. Run Terminal as root privilege (`sudo`).
2. Open Redis config file `/etc/redis/redis.conf` in edit mode e.g.:

bash

```bash
$ vi /etc/redis/redis.conf
```

3. Find and replace `logfile` directive with your desired location.

bash

```
logfile "<path/to/your/redis/log>.log"
```

4. Save the file, and restart Redis daemon.

bash

```
$ sudo systemctl restart redis
```

5. Find and collect Redis log `<path/to/your/redis/log>.log`

RabbitMQ - Message Broker Server

Windows

1. Locate and collect RabbitMQ log files that match the pattern `%APPDATA%\RabbitMQ\log\rabbit@<computer name>.log`.
2. Locate and collect RabbitMQ upgrade log files that match the pattern `%APPDATA%\RabbitMQ\log\rabbit@<computer name>_upgrade.log`.

Linux

1. Run terminal as root privilege (`sudo`).
2. Run following command to retrieve RabbitMQ log location:

bash

```
$ rabbitmq-diagnostics -q log_location
```

3. Access RabbitMQ log folder and find log files:
    - `rabbit@<computer name>.log`
    - `rabbit@<computer name>_upgrade.log`

PostgreSQL - Database Server

Windows

1. Locate and collect log files that match the pattern `C:\Program Files\PostgreSQL\12\data\log with names postgresql-<yyyy-mm-dd>_<HHMMSS>.log`

Linux

1. Run terminal as root privilege (`sudo`).
2. Open the PostgreSQL config file `/etc/postgresql/12/main/postgresql.conf` in edit mode e.g.:

bash

```
$ vi /etc/postgresql/12/main/postgresql.conf
```

3. Find and turn `logging_collector` directive on :

bash

```
logging_collector = on
```

4. Save the file and restart PostgreSQL daemon, e.g.:

bash

```
$ sudo systemctl restart postgresql
```

5. Locate and collect log files that match the pattern `/var/lib/postgresql/12/main/log/postgresql-<yyyy-mm-dd>_<HHMMSS>.log`.

Open Connection On PostgreSQL Server

Info

Just in case the firewall is enabled, please also ensure that you configure your firewall rules properly for the connections between PostgreSQL server and the services of MetaDefender Cluster, which include MetaDefender Core services.

Info

The guide here assumes we are using an SSL connection with PostgreSQL. With a non-SSL connection, please use `host` instead.

Windows

1. Locate and modify `pg_hba.conf` configuration file within the PostgreSQL data directory. For example: `C:\Program Files\PostgreSQL\16\data\pg_hba.conf`.

pg_hba.conf markdown

```
hostssl    all              all             0.0.0.0/0
scram-sha-256
```

In the above example, all source addresses from MetaDefender Cluster and MetaDefender Core services are permitted. Refer here for more details.

2. Locate and modify `postgresql.conf` configuration file within the PostgreSQL data directory. For example: `C:\Program Files\PostgreSQL\<version>\data\postgresql.conf`.

postgresql.conf markdown

```
listen_addresses = '*'
```

The configuration above directs PostgreSQL server to permit incoming connections from all sources associated with MetaDefender Cluster and MetaDefender Core services. Learn more from here.

Linux

1. Locate and modify the `pg_hba.conf` configuration file within the PostgreSQL data directory. For example: `/var/lib/pgsql/<version>/data/pg_hba.conf`.

pg_hba.conf markdown

```
hostssl    all              all              0.0.0.0/0
scram-sha-256
```

In the above example, all source addresses from MetaDefender Cluster and MetaDefender Core services are permitted. Refer here for more details.

2. Locate and modify the `postgresql.conf` configuration file within the PostgreSQL data directory. For example: `/var/lib/pgsql/<version>/data/postgresql.conf`.

postgresql.conf markdown

```
listen_addresses = '*'
```

The configuration above directs PostgreSQL server to permit incoming connections from all sources associated with MetaDefender Cluster and MetaDefender Core services. Learn more from here.

What is the latest MetaDefender Cluster version?

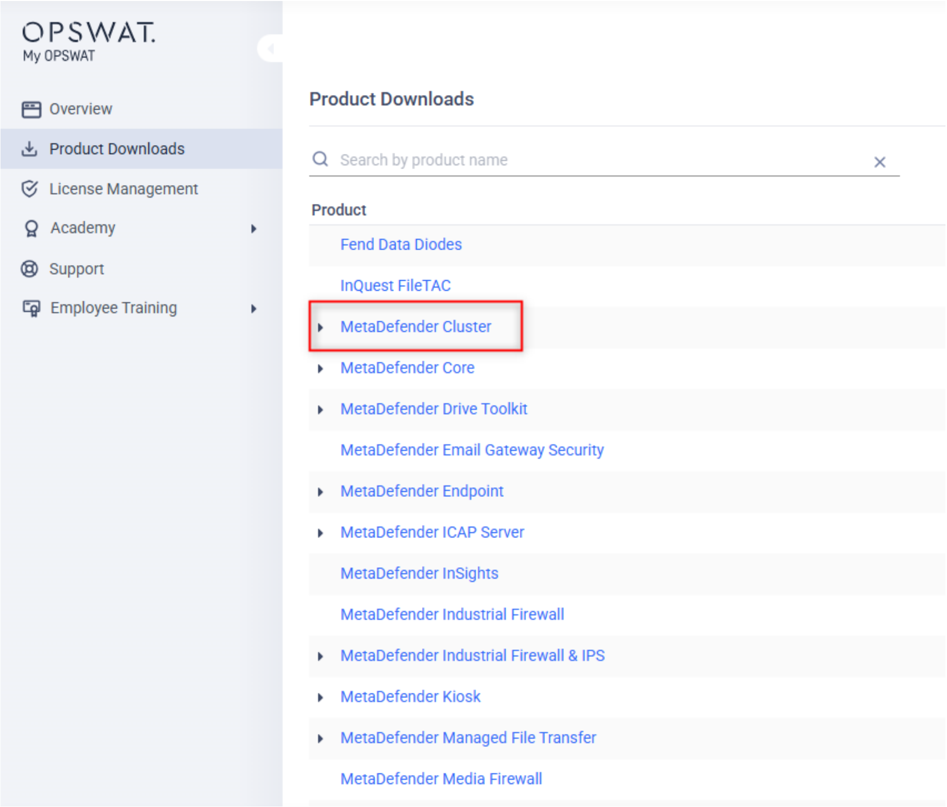
Check Your Version:

This article applies to all MetaDefender Cluster V2 releases deployed on Windows and Linux systems.

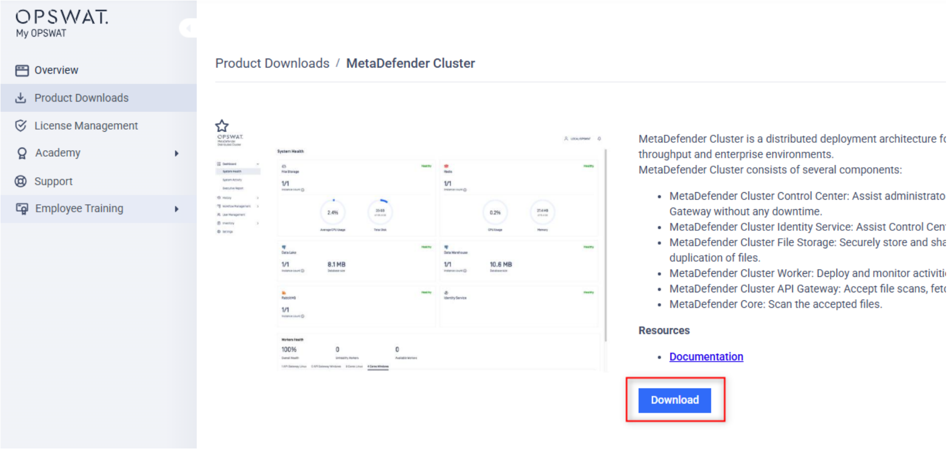
Browse MetaDefender Cluster Release Notes for details on the latest version, version history, version comparisons and release notes.

Alternatively, go directly through the My OPSWAT for a list of available downloads.:

1. Login to My OPSWAT portal
2. Click on "MetaDefender Cluster"



3. Click on “Download” button



Product Downloads / MetaDefender Cluster

MetaDefender Cluster is a distributed deployment architecture fo
throughput and enterprise environments.
MetaDefender Cluster consists of several components:

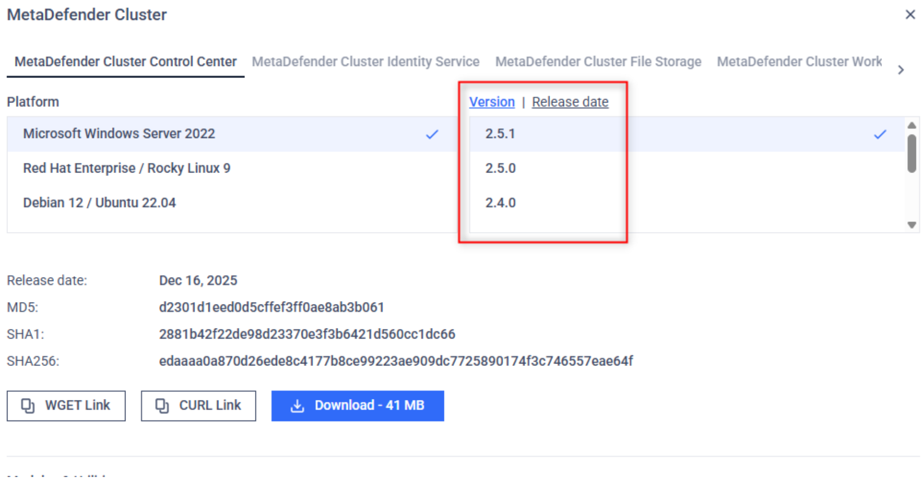
- MetaDefender Cluster Control Center: Assist administrato Gateway without any downtime.
- MetaDefender Cluster Identity Service: Assist Control Cen
- MetaDefender Cluster File Storage: Securely store and sha duplication of files.
- MetaDefender Cluster Worker: Deploy and monitor activiti
- MetaDefender Cluster API Gateway: Accept file scans, fetc
- MetaDefender Core: Scan the accepted files.

Resources

- Documentation

Download

4. In the new window opened, always the version from the top is the latest version


MetaDefender Cluster
×
MetaDefender Cluster Control Center
MetaDefender Cluster Identity Service
MetaDefender Cluster File Storage
MetaDefender Cluster Work
›

Platform	Version | Release date
Microsoft Windows Server 2022 ✓	2.5.1 ✓
Red Hat Enterprise / Rocky Linux 9	2.5.0
Debian 12 / Ubuntu 22.04	2.4.0

Release date: Dec 16, 2025
MD5: d2301d1eed0d5cffef3ff0ae8ab3b061
SHA1: 2881b42f22de98d23370e3f3b6421d560cc1dc66
SHA256: edaaaa0a870d26ede8c4177b8ce99223ae909dc7725890174f3c746557eae64f

WGET Link
CURL Link
Download - 41 MB

Pro Tip:

OPSWAT highly recommends that users upgrade to the latest version of MetaDefender Cluster. 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.

Release notes

Version	2.5.2
Release date	26 January 2026
Scope	This release focuses on improving overall product performance, ensuring consistent processing results across MD Core instances, supporting a new engine package, enhancing compatibility with modern networks, and adding new response fields. It also addresses security vulnerabilities and resolves issues related to licensing.

New Features, Improvements and Enhancements

RESTful API

- Added support for the `current_finished_files` field in the responses of `GET` `/batch/{batch_id}` and `GET` `/file/{batch_id}`.

Further Enhancements

- Optimized data lake queries to speed up retrieval of processing history records.
- Prevented multi-platform MD Core deployment to ensure consistent processing results.
- Added IPv6 support for essential MD Cluster services to improve modern network compatibility.
- Enabled fetching and installing OPSWAT Predictive Alin AI packages on MD Core for automated engine setup.

Security Enhancements

- Upgraded Nginx to 1.29.4 for vulnerability fixes.
- Addressed a security vulnerability that could permit unauthenticated remote code execution within MD Cluster Control Center and API Gateway.
- Addressed a vulnerability in user management operations that could allow SQL injection and potentially cause Denial of Service.

Bug Fixes

- Corrected inaccurate display of outdated scan engines in System Health.
- Resolved a problem that allowed selection of the License Management Service Activation rule even when using an invalid token.
- Fixed an issue where an MD Core instance’s active license could become permanently inactive after switching to a different license.
- Corrected a problem that allowed system admins to deactivate an online license that had already expired.

Known Limitations

- The DLP engine fails to install on MD Core instances that are deployed as part of MD Cluster. This issue will be resolved in the next MD Cluster release (2.6.0).
- MD Cluster Control Center does not detect duplicate workers when one is added using its IP address and another using its hostname (or vice versa). Registering the same worker multiple times may lead to incorrect license usage calculations.

Archived release notes

Version v2.5.1

Release Date: 16 December 2025

Support licensing with License Management Server (LMS)

New licensing management model for MetaDefender Cluster to allow license management server to manage the product's license status.

Further Enhancements

- Improve the performance of fetching processing history.
- Support sorting processing history by duration.
- Enhance product stability.

Security Enhancements

Upgraded library for vulnerability fixes:

- PostgreSQL v14.20

Bug Fixes

- Fixed an issue where certain archive file remained stuck in the in-progress state.
- Fixed an issue that led to the process start time of batch is invalid.
- Fixed a crash in MD Cluster Control Center when adding a worker using IPv6.
- Addressed incorrect status reporting of MD Cluster Worker when a machine was unexpectedly powered off.
- Fixed an issue where the MD Cluster Worker was unable to upgrade its hosted MD Core.

Version v2.5.0

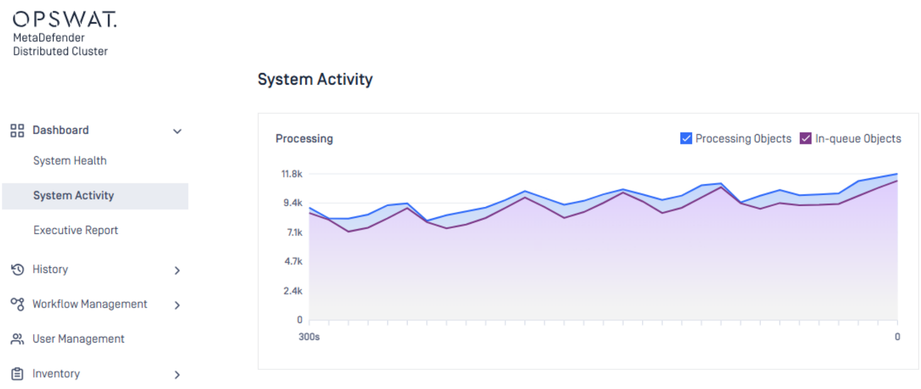
Release Date: 30 October 2025

MetaDefender Cluster license

New licensing model requires a MetaDefender Cluster license.

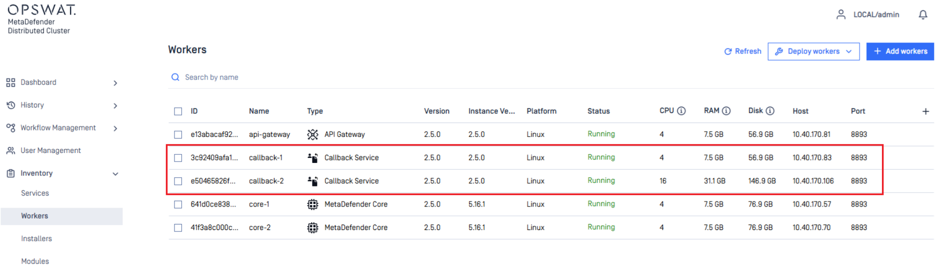
Chart for in-queue objects in System Activity

A new chart in System Activity shows the overall count of objects pending processing.



Dedicated callback service

A new optional service is introduced to dedicate sending the results of finished scan requests to the webhook server. The service can be scaled out easily and upgraded seamlessly.

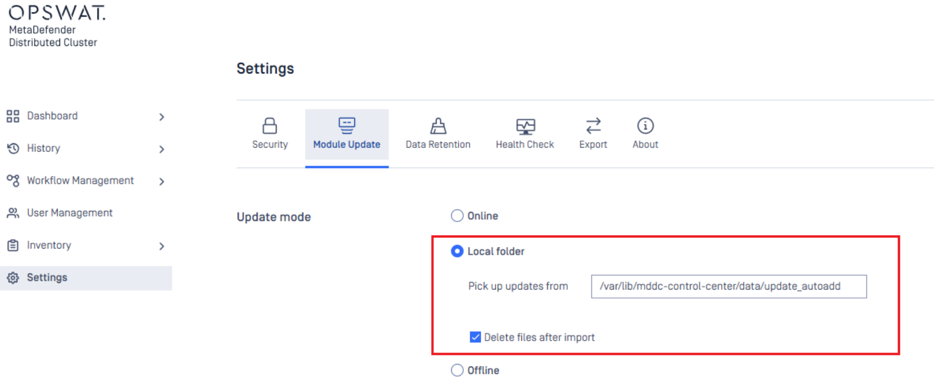


Workers

ID	Name	Type	Version	Instance Ve...	Platform	Status	CPU	RAM	Disk	Host	Port
e13abacaf92...	api-gateway	API Gateway	2.5.0	2.5.0	Linux	Running	4	7.5 GB	56.9 GB	10.40.170.81	8893
3c92409afa1...	callback-1	Callback Service	2.5.0	2.5.0	Linux	Running	4	7.5 GB	56.9 GB	10.40.170.83	8893
e50465826f...	callback-2	Callback Service	2.5.0	2.5.0	Linux	Running	16	31.1 GB	146.9 GB	10.40.170.106	8893
641d0ce838...	core-1	MetaDefender Core	2.5.0	5.16.1	Linux	Running	4	7.5 GB	76.9 GB	10.40.170.57	8893
41f3a8c000c...	core-2	MetaDefender Core	2.5.0	5.16.1	Linux	Running	4	7.5 GB	76.9 GB	10.40.170.70	8893

Engine package update initiated by folder monitoring

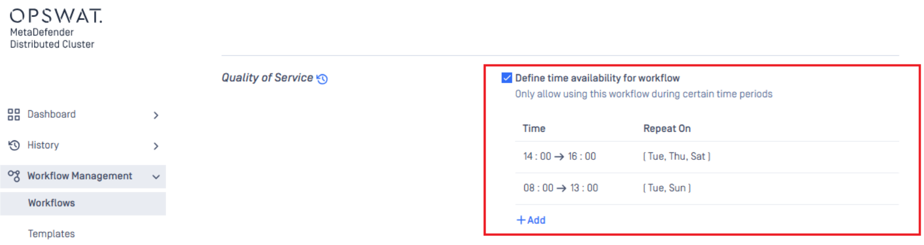
Modification of engine packages in a given folder can be monitored and notified to MetaDefender Core instances for engine updates.



Update mode
○ Online
● Local folder
Pick up updates from /var/lib/mddc-control-center/data/update_autoadd
☑ Delete files after import
○ Offline

Time availability for scan request acceptance

MetaDefender Cluster API Gateway will accept scanning requests from clients during a pre-define time windows.



☑ Define time availability for workflow

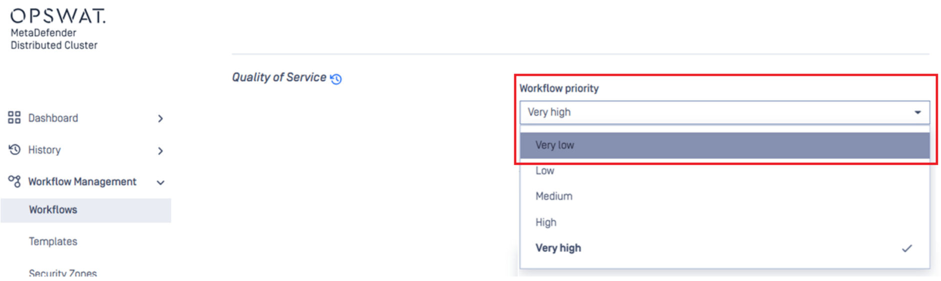
Only allow using this workflow during certain time periods

Time	Repeat On
14 : 00 → 16 : 00	[Tue, Thu, Sat]
08 : 00 → 13 : 00	[Tue, Sun]

+ Add

Workflow priority

MetaDefender Cluster Control Center allows system administrators to configure the priority of a specific workflow.



Workflow priority: Very high ✓

RESTful API

MetaDefender Cluster API Gateway:

- Introduce `metadata` request header to include vulnerability details in the response of `GET` `/batch/{batch_id}/certificate` API.
- Introduce `GET` `/file/webhook/{data_id}` API to retrieve the callback status of a scan request using `data_id`.
- Support `callbackurl` header in requests to `POST` `/file` API.
- Include callback service status in `GET` `/readyz` API.

MetaDefender Cluster Control Center:

- Introduce `PUT` and `GET` `/admin/config/sessioncookie` to modify and get session cookie attributes.
- Drop support for `license_id` field in `POST` `/admin/worker/deploy` API.
- Support callback service installer for `POST` `/admin/installer` API.
- Introduce a new `callback-service` type for the APIs: `POST` `/admin/worker/deploy`, `POST` `/admin/worker/upgrade` and `GET` `/admin/worker/upgrade/version`.

Further Enhancements

- Improve the performance of gathering materials for the executive report.
- Capability to configure the communication port of MetaDefender Core during deployment.
- Support instance name filtering for processing history exported in STIX/CSV format.
- Include metrics for waiting time related to file type detection in the executive report.

Security Enhancements

Upgraded library for vulnerability fixes:

- OpenSSL 3.5.4

Bug Fixes

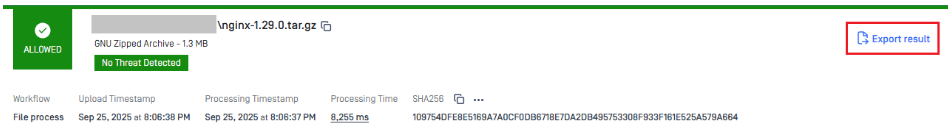
- Fixed the issue that led to the disappearance of MetaDefender Core instances from Instance Activity during high load.
- Fixed the issue that led to MetaDefender Core instances losing their licenses after the upgrade.
- Fixed the issue that caused `GET` `/file/{data_id}` API to return a zero `upload_time`.
- Fixed the issue that led to the omission of scan results for files within a batch when a filter was applied.

Version v2.4.0

Release Date: 30 September 2025

Export scan result in JSON format

From MetaDefender Cluster (MD Cluster) Control Center, users can export scan result in JSON format.



[redacted]\nginx-1.29.0.tar.gz

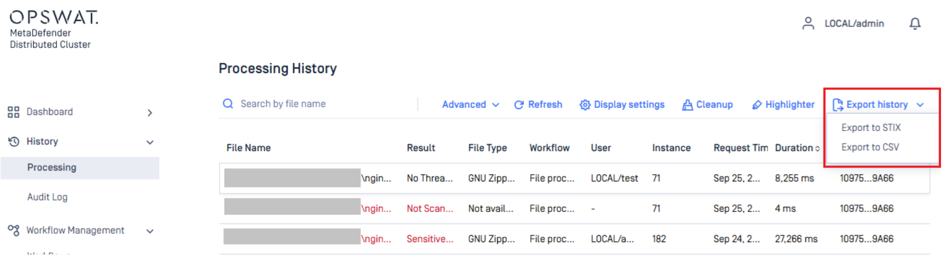
GNU Zipped Archive - 1.3 MB

No Threat Detected

Workflow	Upload Timestamp	Processing Timestamp	Processing Time	SHA256
File process	Sep 25, 2025 at 8:06:38 PM	Sep 25, 2025 at 8:06:37 PM	8,255 ms	109754DFE8E5169A7A0CF0DB6718E7DA2DB495753308F933F161E525A579A664

Export processing history in STIX or CSV format

Processing history can be exported in STIX or CSV format from MD Cluster Control Center.

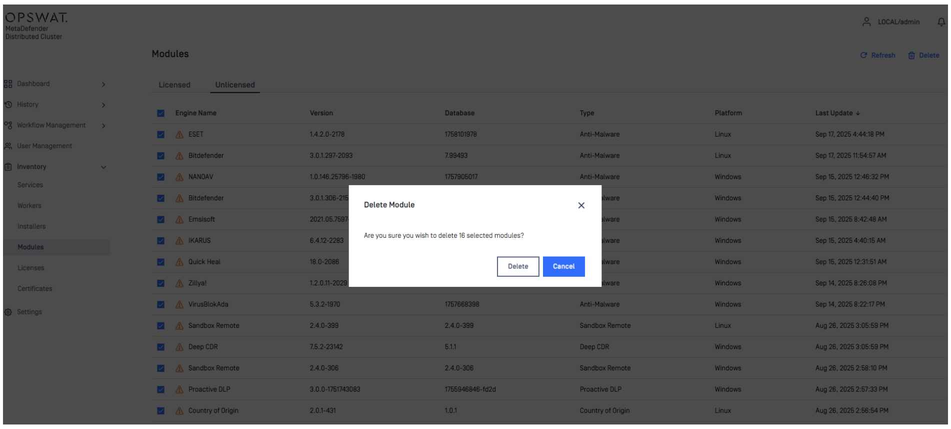


Processing History

File Name	Result	File Type	Workflow	User	Instance	Request Tim	Duration	
[redacted]\ngin...	No Threa...	GNU Zipp...	File proc...	LOCAL/test	71	Sep 25, 2...	8,255 ms	10975...9A66
[redacted]\ngin...	Not Scan...	Not avail...	File proc...	-	71	Sep 25, 2...	4 ms	10975...9A66
[redacted]\ngin...	Sensitive...	GNU Zipp...	File proc...	LOCAL/a...	182	Sep 24, 2...	27,266 ms	10975...9A66

Remove abandoned module packages

Abandoned module packages can be selected and removed on web console of Control Center.



Modules

☑	Engine Name	Version	Database	Type	Platform	Last Update ↓
☑	⚠ ESET	1.4.2.0-2178	1758101978	Anti-Malware	Linux	Sep 17, 2025 4:44:18 PM
☑	⚠ Bitdefender	3.0.1.297-2093	7.99493	Anti-Malware	Linux	Sep 17, 2025 11:54:57 AM
☑	⚠ NANOAV	1.0.146.25796-1980	1757905017	Anti-Malware	Windows	Sep 15, 2025 12:46:32 PM
☑	⚠ Bitdefender	3.0.1.306-215		lware	Windows	Sep 15, 2025 12:44:40 PM
☑	⚠ Emsisoft	2021.05.7597		lware	Windows	Sep 15, 2025 8:42:48 AM
☑	⚠ IKARUS	6.4.12-2283		lware	Windows	Sep 15, 2025 4:40:15 AM
☑	⚠ Quick Heal	18.0-2086		lware	Windows	Sep 15, 2025 12:31:51 AM
☑	⚠ Zillya!	1.2.0.11-2029		lware	Windows	Sep 14, 2025 8:26:08 PM
☑	⚠ VirusBlokAda	5.3.2-1970	1757668398	Anti-Malware	Windows	Sep 14, 2025 8:22:17 PM
☑	⚠ Sandbox Remote	2.4.0-399	2.4.0-399	Sandbox Remote	Linux	Aug 26, 2025 3:05:59 PM
☑	⚠ Deep CDR	7.5.2-23142	5.1.1	Deep CDR	Windows	Aug 26, 2025 3:05:59 PM
☑	⚠ Sandbox Remote	2.4.0-306	2.4.0-306	Sandbox Remote	Windows	Aug 26, 2025 2:58:10 PM
☑	⚠ Proactive DLP	3.0.0-1751743083	1755946846-fd2d	Proactive DLP	Windows	Aug 26, 2025 2:57:33 PM
☑	⚠ Country of Origin	2.0.1-431	1.0.1	Country of Origin	Linux	Aug 26, 2025 2:56:54 PM

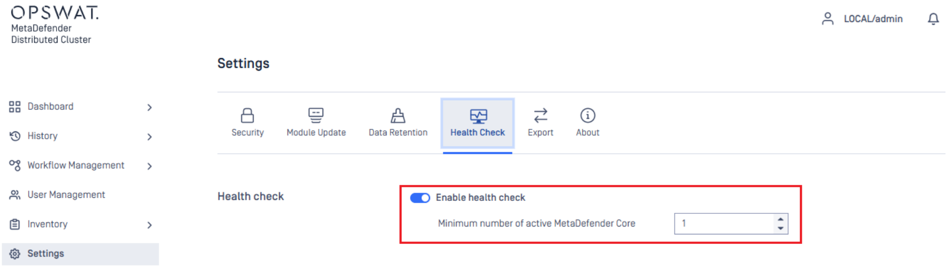
Delete Module ×

Are you sure you wish to delete 16 selected modules?

Delete Cancel

Customize the system health check

System administrators can enable the health check option and set the minimum number of required MetaDefender Core instances in the Health Check settings of the MD Cluster Control Center.



Settings

Health check

Enable health check

Minimum number of active MetaDefender Core 1

RESTful API

- Introduce a new API endpoint in MetaDefender Cluster API Gateway to verity if the system is ready for new scan requests `GET` `/readyz`.
- Introduce a new field, `dlp_wait_time`, in the response of `GET` `/file/{data_id}` API requested from MetaDefender Cluster API Gateway.
- Include `username` field in the response of `GET` `/file/{data_id}`, `GET` `/file/batch/{batch_id}` and `GET` `/hash/{md5|sha1|sha256|sha512}`.

Further Enhancements

- Verify the minimum version requirement when adding a new instance of Redis, RabbitMQ, and PostgreSQL to MetaDefender Cluster Control Center.
- Improve storing scan results from AV engines to MetaDefender Cluster Data Lake.

Security Enhancements

Upgraded library for vulnerability fixes:

- OpenSSL 3.5.2

Bug Fixes

- Fixed the issue that caused occasional service crashes when halted.
- Fixed the issue that made it impossible to close a batch if its name contained special characters.
- Fixed the issue that led to the batch name not appearing in the UI of MD Cluster Control Center.
- Fixed the issue that caused the COO engine to fail or time out during installation.
- Fixed the issue that caused the executive report to eventually miss data.

Known limitations

MetaDefender Core becomes unlicensed following the MD Cluster Worker upgrade.

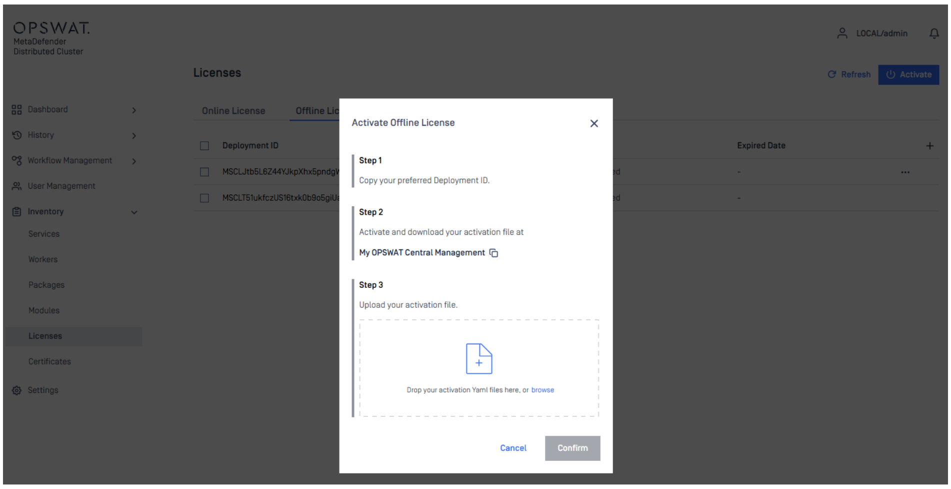
- If online activation is used, follow Online Activation activate MetaDefender Core again.
- Otherwise, follow Offline Activation.

Version v2.3.0

Release Date: 28 August 2025

Offline License Activation

Offline license activation of MetaDefender Core instances within the cluster is supported for air-gapped environment.



Activate Offline License

Step 1

Copy your preferred Deployment ID.

Step 2

Activate and download your activation file at

My OPSWAT Central Management

Step 3

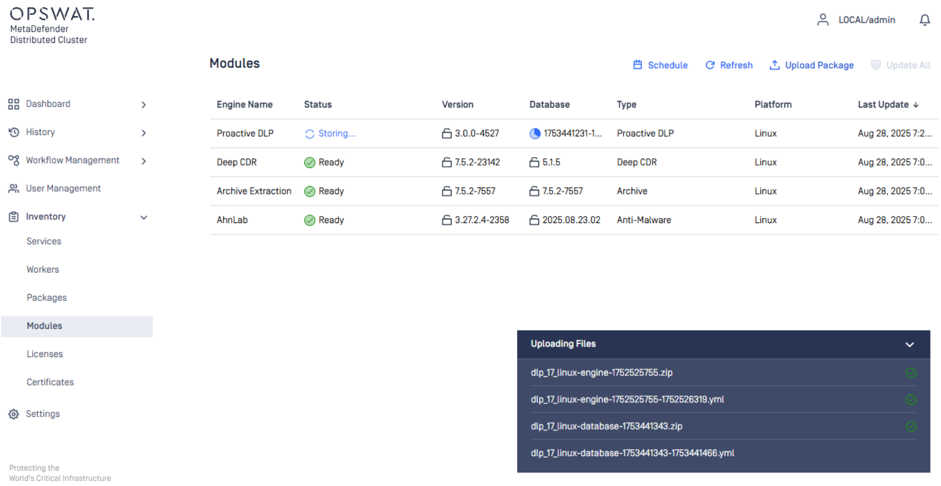
Upload your activation file.

Drop your activation Yaml files here, or browse

Cancel Confirm

Offline Engine Package Upload

In offline environment, administrators can manually upload engine packages to MetaDefender Cluster Control Center.



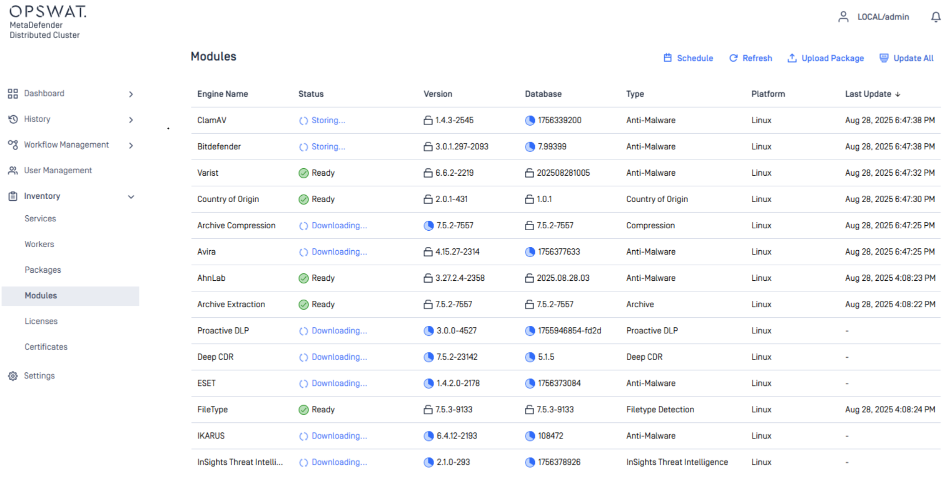
Engine Name	Status	Version	Database	Type	Platform	Last Update ↓
Proactive DLP	Storing...	3.0.0-4527	1753441231-1...	Proactive DLP	Linux	Aug 28, 2025 7:2...
Deep CDR	Ready	7.5.2-23142	5.1.5	Deep CDR	Linux	Aug 28, 2025 7:0...
Archive Extraction	Ready	7.5.2-7557	7.5.2-7557	Archive	Linux	Aug 28, 2025 7:0...
AhnLab	Ready	3.27.2.4-2358	2025.08.23.02	Anti-Malware	Linux	Aug 28, 2025 7:0...

Uploading Files

- dlp_17_linux-engine-1752525755.zip ✓
- dlp_17_linux-engine-1752525755-1752526319.yml ✓
- dlp_17_linux-database-1753441343.zip ✓
- dlp_17_linux-database-1753441343-1753441466.yml

Centralized Online Engine Package Update

At update time, engine packages are obtained once from the cloud and shared with all MetaDefender Core instances within the cluster to retrieve, install, or update.

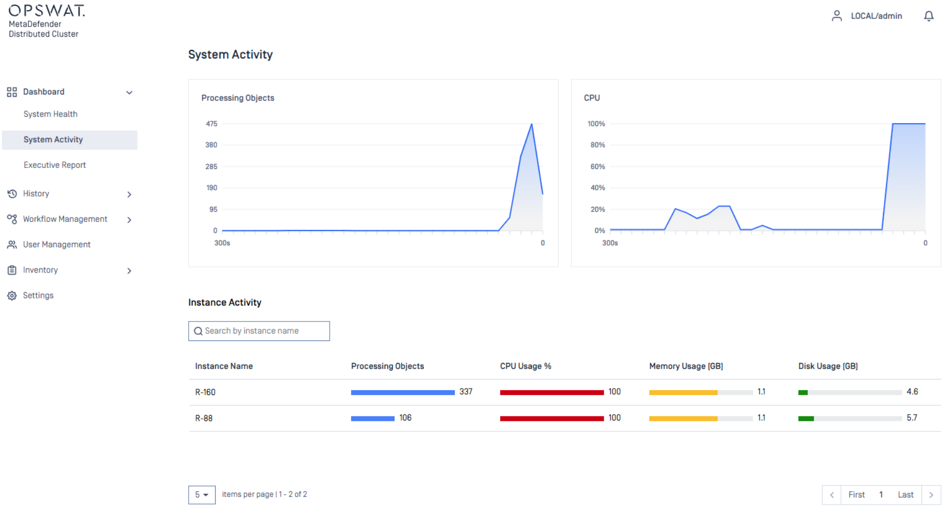


Modules

Engine Name	Status	Version	Database	Type	Platform	Last Update ↓
ClamAV	Storing...	1.4.3-2545	1756339200	Anti-Malware	Linux	Aug 28, 2025 6:47:38 PM
Bitdefender	Storing...	3.0.1.297-2093	7.99399	Anti-Malware	Linux	Aug 28, 2025 6:47:38 PM
Varist	✓ Ready	6.6.2-2219	202508281005	Anti-Malware	Linux	Aug 28, 2025 6:47:32 PM
Country of Origin	✓ Ready	2.0.1-431	1.0.1	Country of Origin	Linux	Aug 28, 2025 6:47:30 PM
Archive Compression	Downloading...	7.5.2-7557	7.5.2-7557	Compression	Linux	Aug 28, 2025 6:47:25 PM
Avira	Downloading...	4.15.27-2314	1756377633	Anti-Malware	Linux	Aug 28, 2025 6:47:25 PM
AhnLab	✓ Ready	3.27.2.4-2358	2025.08.28.03	Anti-Malware	Linux	Aug 28, 2025 4:08:23 PM
Archive Extraction	✓ Ready	7.5.2-7557	7.5.2-7557	Archive	Linux	Aug 28, 2025 4:08:22 PM
Proactive DLP	Downloading...	3.0.0-4527	1755946854-fd2d	Proactive DLP	Linux	-
Deep CDR	Downloading...	7.5.2-23142	5.1.5	Deep CDR	Linux	-
ESET	Downloading...	1.4.2.0-2178	1756373084	Anti-Malware	Linux	-
FileType	✓ Ready	7.5.3-9133	7.5.3-9133	Filetype Detection	Linux	Aug 28, 2025 4:08:24 PM
IKARUS	Downloading...	6.4.12-2193	108472	Anti-Malware	Linux	-
InSights Threat Intelli...	Downloading...	2.1.0-293	1756378926	InSights Threat Intelligence	Linux	-

System and Instance Activities

System Activity page allows administrators to track the total processing objects, average CPU usage of the entire system, processed objects and other resources consumed by individual MetaDefender Core instances, all in one place.

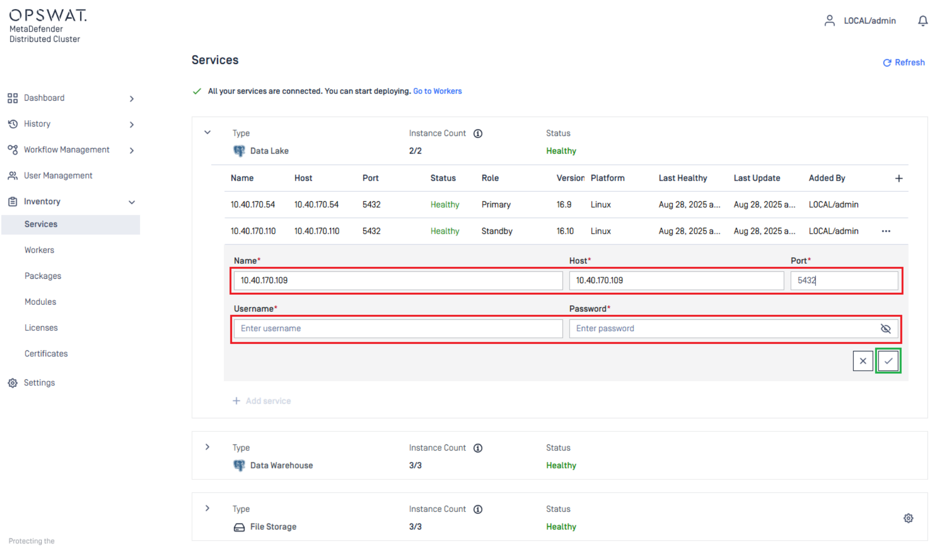


Instance Activity

Instance Name	Processing Objects	CPU Usage %	Memory Usage (GB)	Disk Usage (GB)
R-160	337	100	1.1	4.6
R-88	106	100	1.1	5.7

High Availability support for PostgreSQL Data Lake

Administrators can add multiple PostgreSQL instances to ensure high availability for Data Lake.



Services

✓ All your services are connected. You can start deploying. Go to Workers

Type	Instance Count	Status
Data Lake	2/2	Healthy

Name	Host	Port	Status	Role	Version	Platform	Last Healthy	Last Update	Added By
10.40.170.54	10.40.170.54	5432	Healthy	Primary	16.9	Linux	Aug 28, 2025 a...	Aug 28, 2025 a...	LOCAL/admin
10.40.170.110	10.40.170.110	5432	Healthy	Standby	16.10	Linux	Aug 28, 2025 a...	Aug 28, 2025 a...	LOCAL/admin

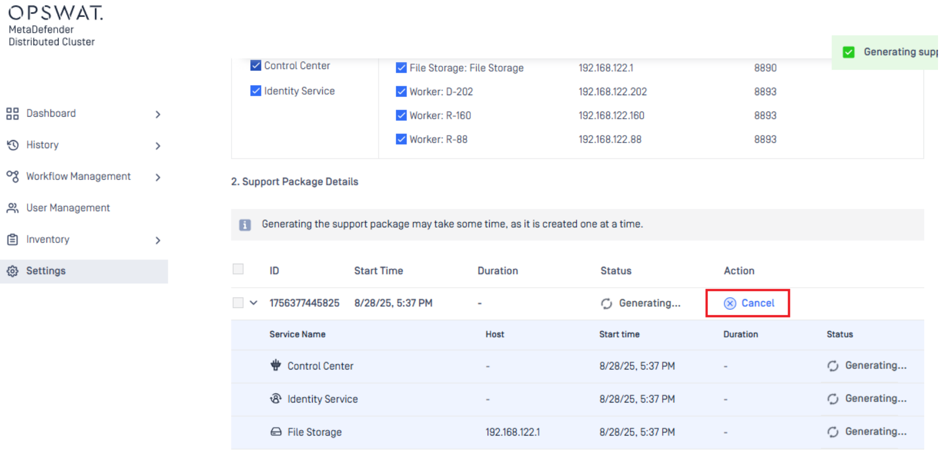
Name*: 10.40.170.109
Host*: 10.40.170.109
Port*: 5432
Username*: Enter username
Password*: Enter password

+ Add service

Type	Instance Count	Status
Data Warehouse	3/3	Healthy
File Storage	3/3	Healthy

Cancellation of Remote Support Package Gathering

Administrators have the option to cancel the gathering of Remote Support Package from the MetaDefender Cluster console at any point. The cancellation is carried out at the earliest opportunity.



☑ Control Center	☑ File Storage: File Storage	192.168.122.1	8890
☑ Identity Service	☑ Worker: D-202	192.168.122.202	8893
	☑ Worker: R-160	192.168.122.160	8893
	☑ Worker: R-88	192.168.122.88	8893

2. Support Package Details

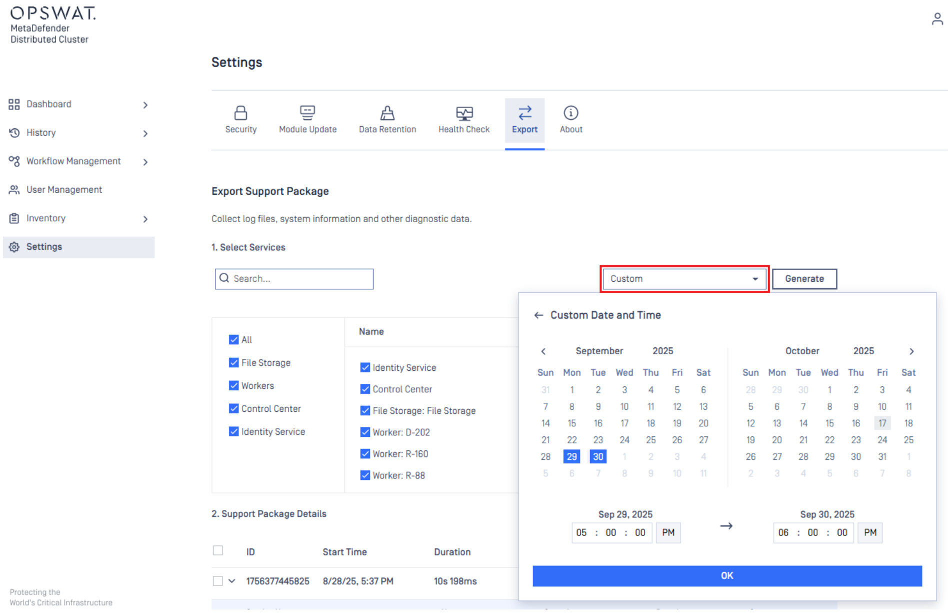
Generating the support package may take some time, as it is created one at a time.

ID	Start Time	Duration	Status	Action
1756377445825	8/28/25, 5:37 PM	-	Generating...	Cancel

Service Name	Host	Start time	Duration	Status
Control Center	-	8/28/25, 5:37 PM	-	Generating...
Identity Service	-	8/28/25, 5:37 PM	-	Generating...
File Storage	192.168.122.1	8/28/25, 5:37 PM	-	Generating...

Support Package Gathering within a specified timeframe

MetaDefender Cluster enables administrators to define a time period for gathering support packages.



Settings

Export Support Package

Collect log files, system information and other diagnostic data.

1. Select Services

Custom

☑ All
☑ File Storage
☑ Workers
☑ Control Center
☑ Identity Service

Name
☑ Identity Service
☑ Control Center
☑ File Storage: File Storage
☑ Worker: D-202
☑ Worker: R-160
☑ Worker: R-88

Custom Date and Time

September 2025 — October 2025

Sep 29, 2025 05 : 00 : 00 PM → Sep 30, 2025 06 : 00 : 00 PM

2. Support Package Details

ID	Start Time	Duration
1756377445825	8/28/25, 5:37 PM	10s 198ms

RESTful API

- Introduce a new API endpoint in MetaDefender Cluster API Gateway to fetch information on the longest expiry active license `GET` `/admin/license`.
- Introduce a new field, `filetype_wait_time`, in the response of `GET` `/file/{data_id}` API requested from MetaDefender Cluster API Gateway.

Security Enhancements

Upgraded libraries for vulnerability fixes:

- 7zip 25.01
- Nginx 1.28.0
- PostgreSQL 14.19

Bug Fixes

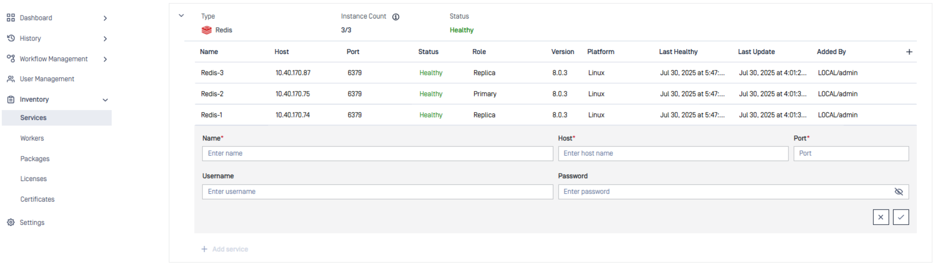
- Fixed an issue that caused the MetaDefender Cluster Control Center to occasionally crash upon stopping.
- Fixed an issue that caused session timeouts to exceed the configured duration.

Version v2.2.0

Release Date: 30 July 2025

Support adding Redis Caching server with username and password

Administrator can add Redis Caching server including username and password to MetaDefender Cluster.



Type	Instance Count	Status
Redis	3/3	Healthy

Name	Host	Port	Status	Role	Version	Platform	Last Healthy	Last Update	Added By
Redis-3	10.40.170.87	6379	Healthy	Replica	8.0.3	Linux	Jul 30, 2025 at 5:47:...	Jul 30, 2025 at 4:01:2...	LOCAL/admin
Redis-2	10.40.170.75	6379	Healthy	Primary	8.0.3	Linux	Jul 30, 2025 at 5:47:...	Jul 30, 2025 at 4:01:3...	LOCAL/admin
Redis-1	10.40.170.74	6379	Healthy	Replica	8.0.3	Linux	Jul 30, 2025 at 5:47:...	Jul 30, 2025 at 4:01:3...	LOCAL/admin

Name* Enter name

Host* Enter host name

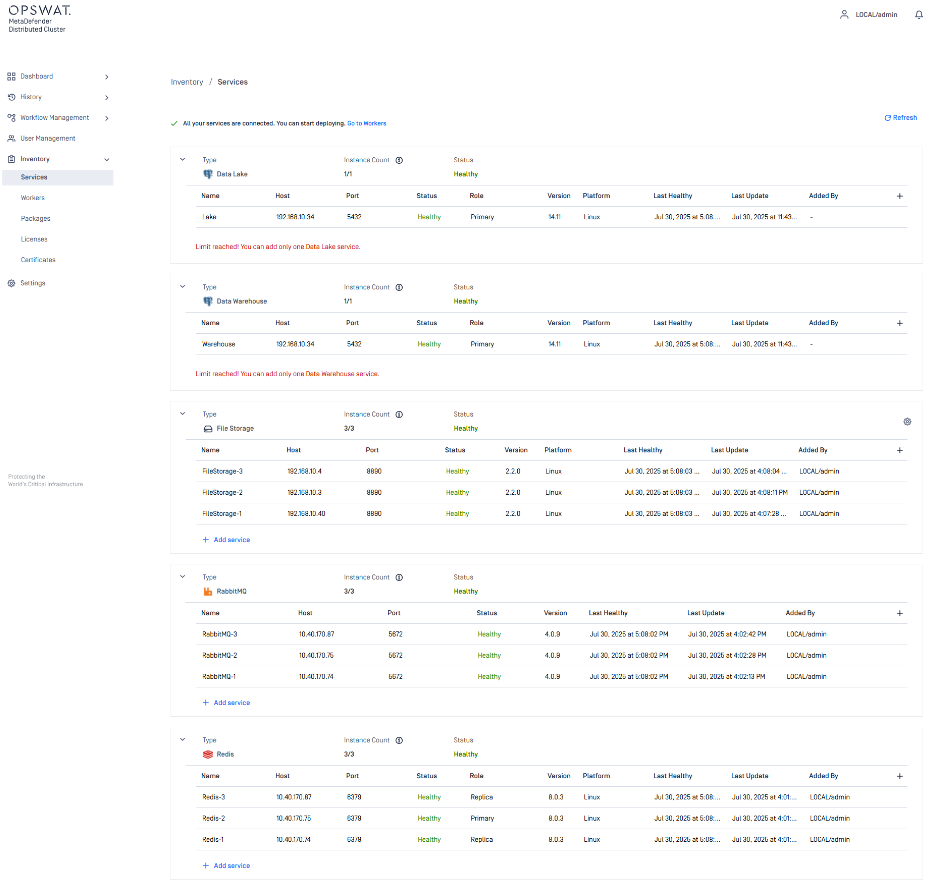
Port* Port

Username Enter username

Password Enter password

High Availability support for Redis Caching Server, RabbitMQ Message Broker and MetaDefender Cluster File Storage

Administrators can add multiple instances of Redis, RabbitMQ and MetaDefender Cluster File Storage for High Availability support.



All your services are connected. You can start deploying. Go to Workers

Type	Instance Count	Status
Data Lake	1/1	Healthy

Name	Host	Port	Status	Role	Version	Platform	Last Healthy	Last Update	Added By
Lake	192.168.10.34	5432	Healthy	Primary	14.11	Linux	Jul 30, 2025 at 5:08:...	Jul 30, 2025 at 11:43...	-

Limit reached! You can add only one Data Lake service.

Type	Instance Count	Status
Data Warehouse	1/1	Healthy

Name	Host	Port	Status	Role	Version	Platform	Last Healthy	Last Update	Added By
Warehouse	192.168.10.34	5432	Healthy	Primary	14.11	Linux	Jul 30, 2025 at 5:08:...	Jul 30, 2025 at 11:43...	-

Limit reached! You can add only one Data Warehouse service.

Type	Instance Count	Status
File Storage	3/3	Healthy

Name	Host	Port	Status	Version	Platform	Last Healthy	Last Update	Added By
FileStorage-3	192.168.10.4	8890	Healthy	2.2.0	Linux	Jul 30, 2025 at 5:08:03 ...	Jul 30, 2025 at 4:08:04 ...	LOCAL/admin
FileStorage-2	192.168.10.3	8890	Healthy	2.2.0	Linux	Jul 30, 2025 at 5:08:03 ...	Jul 30, 2025 at 4:08:11 PM	LOCAL/admin
FileStorage-1	192.168.10.40	8890	Healthy	2.2.0	Linux	Jul 30, 2025 at 5:08:03 ...	Jul 30, 2025 at 4:07:28 ...	LOCAL/admin

Add service

Type	Instance Count	Status
RabbitMQ	3/3	Healthy

Name	Host	Port	Status	Version	Last Healthy	Last Update	Added By
RabbitMQ-3	10.40.170.87	5672	Healthy	4.0.9	Jul 30, 2025 at 5:08:02 PM	Jul 30, 2025 at 4:02:42 PM	LOCAL/admin
RabbitMQ-2	10.40.170.75	5672	Healthy	4.0.9	Jul 30, 2025 at 5:08:02 PM	Jul 30, 2025 at 4:02:28 PM	LOCAL/admin
RabbitMQ-1	10.40.170.74	5672	Healthy	4.0.9	Jul 30, 2025 at 5:08:02 PM	Jul 30, 2025 at 4:02:13 PM	LOCAL/admin

Add service

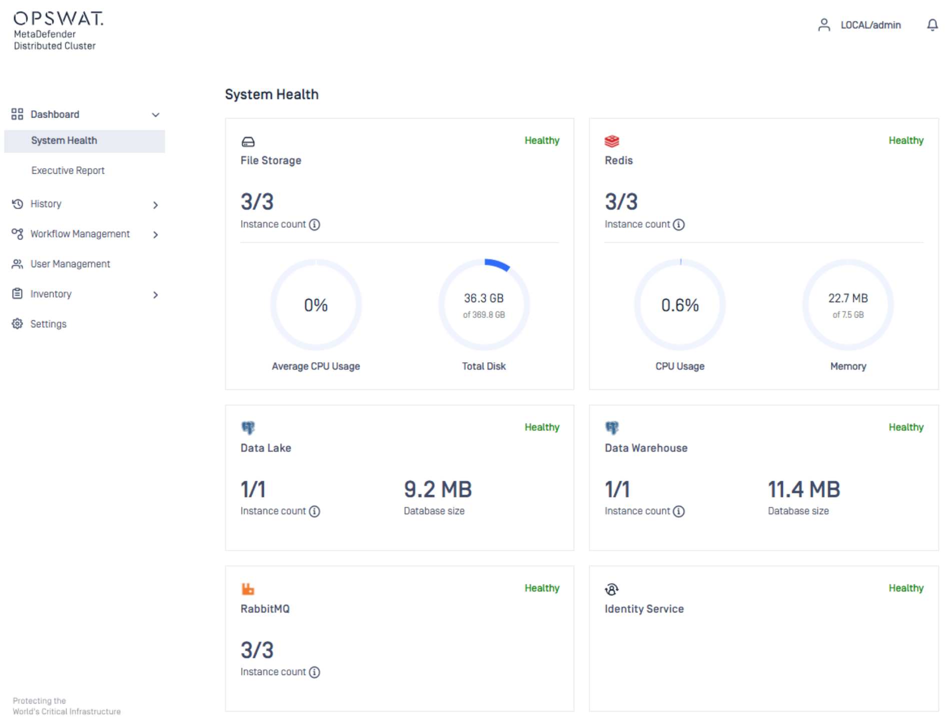
Type	Instance Count	Status
Redis	3/3	Healthy

Name	Host	Port	Status	Role	Version	Platform	Last Healthy	Last Update	Added By
Redis-3	10.40.170.87	6379	Healthy	Replica	8.0.3	Linux	Jul 30, 2025 at 5:08:...	Jul 30, 2025 at 4:01:...	LOCAL/admin
Redis-2	10.40.170.75	6379	Healthy	Primary	8.0.3	Linux	Jul 30, 2025 at 5:08:...	Jul 30, 2025 at 4:01:...	LOCAL/admin
Redis-1	10.40.170.74	6379	Healthy	Replica	8.0.3	Linux	Jul 30, 2025 at 5:08:...	Jul 30, 2025 at 4:01:...	LOCAL/admin

Add service

New Dashboard for High Availability support

A new dashboard is designed to show brief information about critical components of MetaDefender Cluster.



System Health

File Storage — Healthy
3/3 Instance count
0% Average CPU Usage
36.3 GB of 369.8 GB Total Disk

Redis — Healthy
3/3 Instance count
0.6% CPU Usage
22.7 MB of 7.5 GB Memory

Data Lake — Healthy
1/1 Instance count
9.2 MB Database size

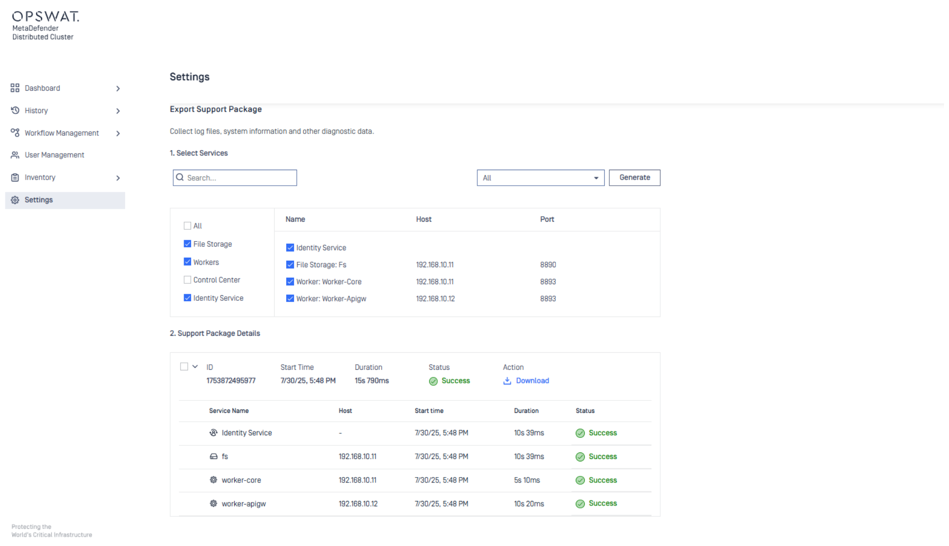
Data Warehouse — Healthy
1/1 Instance count
11.4 MB Database size

RabbitMQ — Healthy
3/3 Instance count

Identity Service — Healthy

Remote Support Package Gathering

A feature to collect support packages remotely by the web console of MetaDefender Cluster Control Center.



Settings

Export Support Package

Collect log files, system information and other diagnostic data.

1. Select Services

□ All
☑ File Storage
☑ Workers
□ Control Center
☑ Identity Service

Name	Host	Port
☑ Identity Service		
☑ File Storage: Fs	192.168.10.11	8890
☑ Worker: Worker-Core	192.168.10.11	8893
☑ Worker: Worker-Apigw	192.168.10.12	8893

2. Support Package Details

ID	Start Time	Duration	Status	Action
1753872495977	7/30/25, 5:48 PM	15s 790ms	Success	Download

Service Name	Host	Start time	Duration	Status
Identity Service	-	7/30/25, 5:48 PM	10s 39ms	Success
fs	192.168.10.11	7/30/25, 5:48 PM	10s 39ms	Success
worker-core	192.168.10.11	7/30/25, 5:48 PM	5s 10ms	Success
worker-apigw	192.168.10.12	7/30/25, 5:48 PM	10s 20ms	Success

RESTful API

- Introduce a new API endpoint in MetaDefender Cluster API Gateway to fetch a list of active MetaDefender Core instances and their details: GET /stat/nodes.
- Introduce a new API endpoint in MetaDefender Cluster API Gateway to fetch a list of active engines and their properties GET /stat/engines.

Security Enhancements

Upgraded libraries for vulnerability fixes:

- Angular v19
- SQLite 3.47.2

Bug Fixes

- Fixed an issue where MetaDefender Cluster API Gateway always responded with HTTP code 500 when clients attempt to call the API GET `/hash/{md5|sha1|sha256|sha512}` with `rule` header.
- Fixed an issue that caused MetaDefender Cluster File Storage to peg at 100% CPU in certain cases.
- Fixed an issue that prevented MetaDefender Cluster components from restarting after an upgrade to the newer version on Windows Server 2025.
- Fixed an issue that prevented MetaDefender Cluster File Storage from upgrading to the newer version on Rocky.

Version v2.1.0

Release Date: 03 July 2025

Update product name

Introduce **MetaDefender Cluster** as the new product name.

RESTful API

- Support `include-inprogress` header in GET `/hash/{md5|sha1|sha256|sha512}` to fetch the scan status of the latest request by hash, including incomplete ones.
- Support `Content-Encoding` header in POST `/file`.

Further Enhancements

- Turn `storage.path` key in the ignition file for MetaDefender Cluster File Storage into an optional setting.
- Correct the returned HTTP code when the user signs in to or signs out from MetaDefender Cluster Control Center with the wrong information.

- Correct the returned HTTP code when POST `/file` is called with wrong API key.

Security Enhancements

Upgraded libraries for vulnerability fixes:

- PostgreSQL v14.18
- yaml-cpp v0.8.0

Bug Fixes

- Fixed an issue where MetaDefender Cluster Control Center can't show scan results of in-progress files on the web console.
- Fixed an issue that caused MetaDefender Cluster API Gateway and MetaDefender Core to be impossible to deploy after a deployment failure.
- Fixed an issue that caused MetaDefender Cluster File Storage to crash while downloading file when DEBUG log is enabled.
- Fixed an issue that caused MetaDefender Cluster File Storage impossible to do data retention with the setting.
- Fixed an issue that caused the MetaDefender Cluster Worker to fail to upgrade due to an engine crash while MetaDefender Core was hosted on the worker.
- Fixed an issue that caused the system administrator to be unable to access the Module page in the MetaDefender Cluster Control Center web console.

API Reference

API Gateway

API Version: v2.5.1

Developer Guide

This is the API documentation for *MetaDefender Cluster API Gateway Public API*. If you would like to evaluate or have any questions about this documentation, please contact us via our Contact Us form.

How to Interact with MetaDefender Cluster API Gateway using REST API

MetaDefender Cluster API Gateway is used to submit files for analysis, retrieve scan results, manage file processing, download processed files, and manage file batches. OPSWAT recommends using the JSON-based REST API. The available methods are documented below.

Note: MetaDefender Cluster API doesn't support chunk upload, however is recommended to stream the files to MetaDefender Cluster API Gateway as part of the upload process.

File Analysis Process

MetaDefender Cluster is a system with multiple components that work together to utilize the power of multiple MetaDefender Core instances. The system is designed to handle large volumes of files and provide high throughput for file analysis. The system can be deployed in a distributed manner, allowing for horizontal scaling and load balancing across multiple MetaDefender Core instances.

Below is a brief description of the API integration flow:

1. Upload a file for analysis to MetaDefender Cluster API Gateway (POST /file), which returns the data_id: File Analysis.

2. The following method can be used to retrieve the analysis report:

   - **Polling**: Fetch the result with previously received data_id (GET /file/{data_id} resource) until scan result belonging to data_id doesn't reach the 100 percent progress_percentage: (Fetch analysis result)

   Note: Too many data_id requests can reduce performance. It is enough to just check every few hundred milliseconds.

3. Retrieve the analysis results anytime after the analysis is completed with hash for files (md5, sha1, sha256, sha512) by calling Fetch analysis result by hash.

- The hash can be found in the scan results

4. Retrieve processed file (sanitized, redacted, watermarked, etc.) after the analysis is complete.

   Note*:* Based on the configured retention policy, the files might be available for retrieval at a later time.

OPSWAT provides some sample codes on GitHub to make it easier to understand how the MetaDefender REST API works.

CONTACT

NAME: API Support
EMAIL: feedback@opswat.com
URL: https://github.com/OPSWAT/metadefender-core-openapi3
Terms of service: https://onlinehelp.opswat.com/policies/

Security and Authentication

SECURITY SCHEMES

KEY	TYPE	DESCRIPTION
apikey	apiKey	Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

API

1. ANALYSIS

File analysis APIs

Submit each file to MetaDefender Cluster API Gateway individually or group them in batches. Each file submission will return a data_id which will be the unique identifier used to retrieve the analysis results.

Note: MetaDefender API doesn't support chunk upload. You shouldn't load the file in memory, is recommended to stream the files to MetaDefender Cluster API Gateway as part of the upload process.

`1.1 POST /file`

Analyze File (Asynchronous mode)

Scanning a file using a specified workflow. Scan is done asynchronously and each scan request is tracked by data id of which result can be retrieved by API Fetch Scan Result.

Note: Chunked transfer encoding (applying header Transfer-Encoding: Chunked) is **not supported** on /file API.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION

NAME	TYPE	EXAMPLE	DESCRIPTION
apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.
filename	string		The name of the submitted file
user_agent	string		user_agent header used to identify (and limit) access to a particular rule. For rule selection, `rule` header should be used.
rule	string		Select rule for the analysis, if no header given the default rule will be selected (URL encoded UTF-8 string of rule name)
batch	string		Batch id to scan with, coming from `Initiate Batch` (If it is not given, it will be a single file scan.)
archivepwd	string		Password for archive (URL encoded UTF-8 string) Multiple passwords is also supported, format: archivepwdX * X: Could be empty * When having value, X must be a number >= 1 For example: * archivepwd1: "fox" * archivepwd2: "cow" * archivepwd3: "bear"
content-encoding	string		Content encoding of the file. This header is used to specify the encoding of the file content. The value should be a valid content encoding type, such as "base64", "gzip". This header is optional and can be omitted if the encoding is not applicable.
metadata	json		Could be utilized for: * Additional parameter for pre-defined post actions and external scanners (as a part of STDIN input). * Customized macro variable for watermarking text (Proactive DLP engine feature). * Additional context / verbose information for each file submission (appended into JSON response scan result). It is strongly recommended to apply URL encoding before sending `metadata` to Metadefender Core to prevent unexpected issues related to encoding errors or unsafe characters.

NAME	TYPE	EXAMPLE	DESCRIPTION
engines-metadata	json		Since MetaDefender Core 5.0.0, preferred context / verbose information can be sent to the engines. Please see the below pages for the details: * [File Type engine](https://docs.opswat.com/mdcore/utilities-engines/supported-engines-metadata) (supported since Core 5.2.1) * [Archive engine](https://docs.opswat.com/mdcore/utilities-engines/supported-engines-metadata-header) (supported since Core 5.4.1) * [Deep CDR](https://docs.opswat.com/mdcore/deep-cdr/supported-engines-metadata-json) * [Proactive DLP](https://docs.opswat.com/mdcore/proactive-dlp/supported-engines-metadata-json)
callbackurl	uri		Client's URL where MetaDefender Cluster Callback Service will notify scan result back to whenever scan is finished (webhook model). * Format: <protocol://><ip | domain>:<port></path> * Example: http://10.0.1.100:8081/listenback * Supported protocol: HTTP / HTTPS * Supported host types: domain name, IPv4 (IPv6 not supported) * Method: POST > _**Note**_: The Callback URL is only supported when MetaDefender Cluster Callback Service is deployed, and MetaDefender Core version must be 5.16.1 or higher.
global-timeout	integer		This custom global timeout (in seconds) will override the global timeout predefined in corresponding workflow rule.

RESPONSE

STATUS CODE - 200: Successful file submission

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
data_id*	string	Unique submission identifier. Use this value to reference the submission.

EXAMPLE:

```
{
 "data_id": "61dffeaa728844adbf49eb090e4ece0e"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 411: Content-Length header is missing from the request.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	

EXAMPLE:

```
{
 "err": "Missing Content-Length header."
}
```

STATUS CODE - 422: Body input is empty.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	

EXAMPLE:

```
{
 "err": "File is empty."
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

STATUS CODE - 503: Server is too busy. Try again later.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	

EXAMPLE:

```
{
 "err": "Server is too busy. Try again later."
}
```

1.2 GET /file/{data_id}

Fetch Analysis Result

Retrieve scan results.

Scan is done asynchronously and each scan request is tracked by a data ID.

Initiating file scans and retrieving the results need to be done using two separate API calls. This request needs to be made multiple times until the scan is complete. Scan completion can be traced using scan_results.progress_percentage value from the response.

Note*:* The REST API also supports pagination for archive file result. A completed response description with archive detection:

- extracted_files: information about extracted files
  - files_extracted_count: the number of extracted files
  - files_in_archive: array of files in archive
    - detected_by: number of engines reported threat
    - scanned_with: number of engines used for scanning the file

  - first_index: it tells that from which file (index of the file, 0 is the first) the result JSON contains information about extracted files. (default=0)

- page_size: it tells how many files the result JSON contains information about (default=50). So by default, the result JSON contains information about the first 50 extracted files.
- worst_data_id: data id of the file that has the worst result in the archive

- scan_results
    - last_file_scanned (stored only in memory, not in database): If available, the name of the most recent processed file

REQUEST

PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*data_id	string		Unique submission identifier. Use this value to reference the submission.

QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
first	integer		The first item order in the list child files of archive file
size	integer		The number of items to be fetched next, counting from the item order indicated in first header

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.
user_agent	string		user_agent header used to identify (and limit) access to a particular rule. For rule selection, `rule` header should be used.

RESPONSE

STATUS CODE - 200: Entire analysis report generated by MetaDefender Core

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
data_id	`string`	data identifier of the requested file
dlp_info	`object`	
certainty	`enum`	**ALLOWED:** `Very Low, Low, Medium, High, Very High` Describes how certain the hit is, possible values: * `Very Low` * `Low` * `Medium` * `High` * `Very High`
errors	`object`	
filename	`string`	Output processed file name (pre-configured on engine settings under Core's worflow rule)
hits	`object`	
ccn	`object`	
display_name	`string`	Credit Card Number, Social Security Number, or in case of RegEx, the name of the rule that has been given by the user
hits	`array`	
after	`string`	The context after the matched data.
before	`string`	The context before the matched data.
certainty	`enum`	**ALLOWED:** `Very Low, Low, Medium, High, Very High` The text version of "certainty_score", possible values: * `Very Low` * `Low` * `Medium` * `High` * `Very High`
certainty_score	`integer`	Is defined by the relevance of the given hit in its context. It is calculated based on multiple factors such as the number of digits, possible values: [0-100]
hit	`string`	The matched data.
location	`string`	The location of the hit that is found in a file.
severity	`enum`	**ALLOWED:** `0, 1` (NOTE: this field is deprecated): can be 0 (detected) or 1 (suspicious).
tryRedact	`boolean`	If file was redacted or not.
metadata_removal	`object`	
result	`enum`	**ALLOWED:** `removed, not removed, failed to remove` Result of the metadata removal process, possible values: * `removed` * `not removed` * `failed to remove`
redact	`object`	

NAME	TYPE	DESCRIPTION
result	enum	**ALLOWED:** `redacted, not redacted, failed to redact` Result of the redaction process, possible values: * `redacted` * `not redacted` * `failed to redact`
severity	enum	**ALLOWED:** `0, 1` (NOTE: this field is deprecated): represents the severity of the data loss, possible values: * `0` - Certainly is data loss * `1` - Might be data loss
verdict	enum	**ALLOWED:** `0, 1, 2, 3, 4` The overall result for the scanned file. Possible values: * `0` - Clean * `1` - Found matched data * `2` - Suspicious * `3` - Failed * `4` - Not scanned
watermark	object	
result	enum	**ALLOWED:** `added, not added, failed to add` Result of the watermarking process, possible values: * `added` * `not added` * `failed to add`
download_info	object	
error_detail	string	Revealed detailed reason why the download failed.
progress	number	Only applicable when "status" is `Downloading`, indicates download finished percentage, in a range of [1, 99]. * Once hitting 100, the status will be changed to `Download Success`. * or other problematic status (`Download Cancelled`, `Download Failed`) if the download stopped unexpectedly.

NAME	TYPE	DESCRIPTION

status	`string`	Indicates download status, which could be either
		- `Downloading`
		- Check `progress` key value for actual download percentage
		```json
		"download_info": {
		"progress": 7,
		"status": "Downloading",
		"url": "http://192.168.200.97:8080/5gb.zip"
		}
		```
		- `Download Success`
		```json
		"download_info": {
		"status": "Download Success",
		"url": "https://secure.eicar.org/eicar.com"
		}
		```
		- `Download Failed`
		- Check `error_detail` key value for an error explanation
		```json
		"download_info": {
		"error_detail": "Connection error",
		"status": "Download Failed",
		"url": "http://192.168.200.97:8080/2gb.zip"
		}
		```
		- `Download Timeout`
		- Expecting to occur when the download progress takes longer than what time window allowed in MetaDefender Core's pre-configured setting under workflow rule (under "SCAN" tab)
		```json
		"download_info": {
		"status": "Download Timeout",
		"url": "http://192.168.200.97:8080/2gb.zip"
		}
		```
		- `Download Cancelled`
		- Expecting to occur when user explicitly cancelled that file scan request, or batch request that the scan belongs to
		```json
		"download_info": {
		"status": "Download Cancelled",
		"url": "http://192.168.200.97:8080/5gb.zip"
		}
		```

url	`string`	Original download link which was specified in HTTP(S) request's `downloadfrom` header
extraction_info	`object`	
decrypted_status	`enum`	**ALLOWED:** `Success, Failed` Indicate that decryption phase is successful or not.
err_category	`string`	Error category
err_code	`integer`	Error code
err_details	`string`	Error message
is_encrypted_file	`boolean`	Indicate if file is password-protected or not.

NAME	TYPE	DESCRIPTION
file_info	object	
display_name	string	The filename reported via `filename` header.
file_size	integer	Total file size in bytes.
file_type	string	The filetype using mimetype.
file_type_description	string	The filetype in human readable format.
md5	string	File's MD5 hash.
sha1	string	File's SHA1 hash.
sha256	string	File's SHA256 Hash.
sha512	string	File's SHA512 Hash.
signer_infos	array	
digest_algorithm	string	Digest algorithm.
digest_encryption_algorithm	string	Encryption algorithm.
issuer	string	Entity that develops and registers the certificate.
serial_number	string	Serial number of the certificate.
vendor_name	string	Entity that is issued a certificate and utilize it for creating a digital signature.
version	string	Version of X.509 that is used in the certificate. This version field is zero-based. * 0: v1 * 1: v2 * 2: v3
type_category	array	
receive_data_timestamp	string	The timestamp when upload progress started (first byte received) (in milliseconds)
upload_time	integer	Total time elapsed for upload process (in milliseconds).
upload_timestamp	string	The timestamp when upload progress finished (all bytes received) (in milliseconds)
filetype_info	object	
file_info*	object	
description*	string	File type description
detected_by	string	Analyzer that detected the file type
encrypted*	boolean	File is password-protected or not
extensions*	string	File type extension
groupID*	string	File type category
groupIDs*	array	
group_description	string	File type category description
likely_type_ids	array	

NAME	TYPE	DESCRIPTION
score*	integer	Likelihood score of the file type
typeID*	string	File type ID
type*	string	MIME type
typeID*	string	File type ID
type_ids*	array	
final_verdict	object	
verdict*	enum	**ALLOWED:** allowed, blocked Final verdict of the file type analysis.
verdict_explanation*	string	Explanation of the final verdict.
is_file_type_mismatch	boolean	Indicates if the file type does not match the expected type.
other_detections	array	Other file type detections.
result_template_hash	string	SHA256 Hash of user-interface template. For web console only.
spoofing_info	object	
detection_result	string	Result of the spoofing detection.
result_explanation	string	Explanation of the spoofing detection result.
result_overview	string	Overview of the spoofing detection result.
opswatfilescan_info	object	
process_info	object	
blocked_reason	string	Provides the reason why the file is blocked (if so).
blocked_reasons	array	
file_type_skipped_scan	boolean	Indicates if the input file's detected type was configured to skip scanning.
hash_time	integer	Total time elapsed for computing hashes (in milliseconds).
outdated_data	array	
processing_time	integer	Total time elapsed during processing file (in milliseconds).
processing_time_details	object	
av_scan_time	integer	AV engines' processing time.
cdr_time	integer	Deep CDR engine's sanitization time.
dlp_time	integer	Proactive DLP engine's processing time.
extraction_time	integer	Archive extraction engine's processing time.
filetype_time	integer	FileType engine's processing time.
opswatfilescan_time	integer	OPSWAT Filescan engine's processing time.
others_time	integer	Total time elapsed for following processing tasks in the product (in milliseconds): * Decryption time (if receiving an encrypted file) * External scanner (if configured) * Post action (if configured) * Other internal processing time among components in the product

NAME	TYPE	DESCRIPTION
parse_dgsg_time	integer	Digital signature analyzing time.
vul_time	integer	Vulnerability engine's lookup time.
yara_time	integer	YARA engine's processing time.
filetype_wait_time	integer	FileType engine's wait time.
profile	string	The used rule name.
progress_percentage	integer	Percentage of processing completed (from 1-100).
queue_time	integer	Total time elapsed for file processing task was waiting in MetaDefender Core's queue until being picked up (queue_time = start_time - upload_timestamp) (in milliseconds).
result	string	The final result of processing the file (Allowed / Blocked / Processing).
user_agent	string	Identifier for the REST Client that calls the API.
username	string	User identifier who submitted scan request earlier.
verdicts	array	
post_processing	object	
actions_failed	string	Empty string if no action failed or list of failed actions, separated by "|".
actions_ran	string	List of successful actions, separated by "|". Empty string if otherwise.
converted_destination	string	Contains the name of the sanitized file.
converted_to	string	Contains target type name of sanitization.
copy_move_destination	string	Contains target type name of sanitization.
sanitization_details	object	
cdr_wait_time	integer	The time in milliseconds that the CDR process took to complete.
description	string	Action was successful or not.
details	array	
action*	enum	**ALLOWED:** `sanitized, removed` The type of action that was performed
count	integer	The number of objects that were sanitized/removed.
details	object	
action	enum	**ALLOWED:** `sanitized, removed` The type of action that was performed
count	integer	The number of objects that were sanitized/removed.
object_details	array	
object_name	string	The object type that was sanitized/removed.
description	string	Action was successful or not.
file_name	string	If an embedded file was sanitized.
object_details	array	
object_name*	string	The object type that was sanitized/removed.

NAME	TYPE	DESCRIPTION
failure_category	string	Deep CDR errors are classified into different categories. For more details, please find [Troubleshooting sanitization failures](https://docs.opswat.com/mdcore/deep-cdr/troubleshooting-sanitization-failures)
result	enum	**ALLOWED:** `Sanitized, Sanitized failed, Sanitized skipped` The result of the CDR process. - **Sanitized**: the file was successfully sanitized. - **Sanitized failed**: the file could not be sanitized due to an error during the process. - **Sanitized skipped**: the file was skipped from sanitization. Common reasons include the file being digitally signed or other policy-based exclusions.
result_template_hash	string	The hash value of the result template, which is used for displaying results on the Core UI and for internal communication between MetaDefender Core and the Deep CDR engine. This value is intended for system use only and is not required for external integration.
sanitized_file_info	object	
file_size	integer	Size of sanitized file in bytes.
sha256	string	SHA256 hash of sanitized file.
verdict	enum	**ALLOWED:** `blocked, allowed` The verdict of the CDR process. - **blocked**: the file is recommended for blocking by Deep CDR. - **allowed**: the file is recommended for allowing by Deep CDR as it found no reason to recommend blocking it.
verdict_explanations	array	
scan_results	object	
data_id	string	Data ID of the requested file
progress_percentage	integer	Track analysis progress until reaches 100.
scan_all_result_a	enum	**ALLOWED:** `No Threat Detected, Infected, Suspicious, Failed, Whitelisted, Blacklisted, Exceeded Archive Depth, Not Scanned, Encrypted Archive, Exceeded Archive Size, Exceeded Archive File Number, Password Protected Document, Exceeded Archive Timeout, Mismatch, Potentially Vulnerable File, Cancelled, Sensitive Data Found, Yara Rule Matched, Potentially Unwanted, Unsupported File Type, Extraction Failed, Scan Failed, Suspicious Verdict by Sandbox, Likely Malicious Verdict by Sandbox, Malicious Verdict by Sandbox, Blocked Verdict by Sandbox, Blocked Verdict by Deep CDR, Global Timeout Exceeded, Vulnerable Verdict by SBOM, Non-vulnerable Verdict by SBOM, Blocked Verdict by SBOM, Blocked by Post Action, Known Bad, Known Good, Unknown, Allowed Verdict by COO, Blocked Verdict by COO, Unknown Verdict by COO, In Progress, Skip Processing Fast Symlink` The overall scan result as string

NAME	TYPE	DESCRIPTION
scan_all_result_i	enum	**ALLOWED:** 0, 1, 2, 3, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 38, 39, 40, 41, 42, 43, 255, 1014 The overall scan result as index in the Processing Results table.
start_time	string	Timestamp when the scanning process starts.
total_avs	integer	Total number of scanning engines used as part of this analysis.
total_time	integer	Total time elapsed during scan (in milliseconds).
scan_details	object	
ClamAV	object	
def_time	string	The database definition time for this engine
eng_id	string	The unique identification string for the engine
location	string	Where this engine is deployed (local/remote).
scan_result_i	integer	Scan result as index in the Processing Results table
scan_time	integer	The time elapsed during scan with this engine (in milliseconds).
threat_found	string	The threat name, IF scan result is Infected or Suspicious. Otherwise empty string or error message from the engine.
wait_time	integer	Time elapsed between sending file to Core and receiving the result from the engine (in milliseconds).
vulnerability_info	object	
result	object	
code	integer	The result code for vulnerability check, 0 means a successful check
hash	string	The file's SHA1 hash value
method	enum	**ALLOWED:** 50700 The method used by OESIS Framework, it should be 50700 every time.
timestamp	string	Timestamp of the request issued
timing	integer	The vulnerability check's duration in milliseconds
detected_product	object	
has_kb	boolean	Indicates whether any KBs or MSBs exist for this hash
has_vulnerability	boolean	Indicates whether any vulnerabilities have been associated with the particular product
is_current	boolean	True if this product's patch level is current, defaults to true
product	object	
id	integer	The OPSWAT product id
name	string	The product name
remediation_link	string	A link where product updates or patches can be obtained

NAME	TYPE	DESCRIPTION
severity	enum	**ALLOWED:** `LOW, MODERATE, IMPORTANT, CRITICAL, NOT_AVAILABLE, UNKNOWN` String description of Severity level: * `LOW` * `MODERATE` * `IMPORTANT` * `CRITICAL` * `NOT_AVAILABLE` * `UNKNOWN`
sig_name	string	Product signature descriptor
signature	integer	OPSWAT signature id
vendor	object	
id	integer	The OPSWAT vendor id
name	string	The vendor name
version	string	The installed product version
version_data	object	
count_behind	integer	The number of patches behind of the installed product
feed_id	integer	The remote feed ID used to determine patch level
version	string	The current version of the product in the remote feed
vulnerabilites	array	
description	string	A text description of the specific vulnerability
details	object	
cpe	string	A CPE product reference
cve	string	A CVE identification string
cvss	object	
access-complexity	string	A CVSS access-complexity descriptor
access-vector	string	A CVSS access-vector descriptor
authentication	string	A CVSS authentication descriptor
availability-impact	string	A CVSS availability impact descriptor
confidentiality-impact	string	A CVSS confidentiality impact descriptor
generated-on-epoch	string	An epoch timestamp indicating CVSS generation time
integrity-impact	string	A CVSS integrity impact descriptor
score	string	A CVSS 10-point severity score
source	string	A CVSS source descriptor
cwe	string	A CWE group identification string
last_modified_epoch	string	An epoch timestamp indicating source last update time

<table>
<tr><th>NAME</th><th>TYPE</th><th>DESCRIPTION</th></tr>
<tr><td>published-epoch</td><td>string</td><td>An epoch timestamp indicating source publishing time</td></tr>
<tr><td><b>references</b></td><td>array</td><td></td></tr>
<tr><td>severity</td><td>enum</td><td><b>ALLOWED:</b> LOW, MODERATE, IMPORTANT, CRITICAL, NOT_AVAILABLE, UNKNOWN
String description of Severity level:
* `LOW`
* `MODERATE`
* `IMPORTANT`
* `CRITICAL`
* `NOT_AVAILABLE`
* `UNKNOWN`</td></tr>
<tr><td>severity_index</td><td>integer</td><td>A 5 point scale numerical description of Severity level with 5 being greatest and 0 being unknown</td></tr>
<tr><td>static_id</td><td>integer</td><td>An OPSWAT identifier for the vulnerability</td></tr>
<tr><td>verdict</td><td>integer</td><td>The vulnerability check's duration in milliseconds
* `0` - No Vulnerability Found
* `1` - Vulnerability Found
* `3` - Failed
* `16` - Processing Timed Out</td></tr>
<tr><td><b>yara</b></td><td>object</td><td></td></tr>
<tr><td><b>hits</b></td><td>object</td><td></td></tr>
<tr><td>verdict</td><td>enum</td><td><b>ALLOWED:</b> 0, 1, 2, 3, 4 The overall result for the analyzed file. Value will be one of the following: | index	status
0	Clean
1	Found matched data
2	Suspicious
3	Failed
4	Not scanned |</td></tr>
</table>

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

1.3 GET /hash/{md5|sha1|sha256|sha512}

Fetch Analysis Result By Hash

Retrieve analysis result by hash

REQUEST

PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*md5|sha1|sha256|sha512	string		Hash value to search. This can be md5, sha1, sha256, sha512

QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
first	integer		The first item order in the list child files of archive file
size	integer		The number of items to be fetched next, counting from the item order indicated in first header

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.
rule	string		Select rule for the analysis, if no header given the default rule will be selected (URL encoded UTF-8 string of rule name)

NAME	TYPE	EXAMPLE	DESCRIPTION
`selfonly`	`boolean`		Useful to archive hash lookup. Allow specifying to only perform hash lookup against the original archive file self only, and skip searching all child files result within the original archive. Default value is false.
`timerange`	`integer`		Scoping down the recent number of hours that hash lookup task should start from till now, instead of searching the entire scan history in MetaDefender Core database. Default value is 0. That means no time scope.
`include-inprogress`	`boolean`		False (default): API will return "Not Found" if the verdict is in progress. True: If the queried hash has a completed processing result before, API will return the completed processing result. If this hash doesn't have any completed processing result, API will return this In-progress result.

RESPONSE

STATUS CODE - 200: Get information of file

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
data_id	`string`	data identifier of the requested file
dlp_info	`object`	
certainty	`enum`	**ALLOWED:** `Very Low, Low, Medium, High, Very High` Describes how certain the hit is, possible values: * `Very Low` * `Low` * `Medium` * `High` * `Very High`
errors	`object`	
filename	`string`	Output processed file name (pre-configured on engine settings under Core's worflow rule)
hits	`object`	
ccn	`object`	

NAME	TYPE	DESCRIPTION
display_name	string	Credit Card Number, Social Security Number, or in case of RegEx, the name of the rule that has been given by the user
hits	array	
after	string	The context after the matched data.
before	string	The context before the matched data.
certainty	enum	**ALLOWED:** `Very Low, Low, Medium, High, Very High` The text version of "certainty_score", possible values: * `Very Low` * `Low` * `Medium` * `High` * `Very High`
certainty_score	integer	Is defined by the relevance of the given hit in its context. It is calculated based on multiple factors such as the number of digits, possible values: [0-100]
hit	string	The matched data.
location	string	The location of the hit that is found in a file.
severity	enum	**ALLOWED:** `0, 1` (NOTE: this field is deprecated): can be 0 (detected) or 1 (suspicious).
tryRedact	boolean	If file was redacted or not.
metadata_removal	object	
result	enum	**ALLOWED:** `removed, not removed, failed to remove` Result of the metadata removal process, possible values: * `removed` * `not removed` * `failed to remove`
redact	object	
result	enum	**ALLOWED:** `redacted, not redacted, failed to redact` Result of the redaction process, possible values: * `redacted` * `not redacted` * `failed to redact`
severity	enum	**ALLOWED:** `0, 1` (NOTE: this field is deprecated): represents the severity of the data loss, possible values: * `0` - Certainly is data loss * `1` - Might be data loss
verdict	enum	**ALLOWED:** `0, 1, 2, 3, 4` The overall result for the scanned file. Possible values: * `0` - Clean * `1` - Found matched data * `2` - Suspicious * `3` - Failed * `4` - Not scanned
watermark	object	

NAME	TYPE	DESCRIPTION
result	enum	**ALLOWED:** added, not added, failed to add Result of the watermarking process, possible values: * `added` * `not added` * `failed to add`
download_info	object	
error_detail	string	Revealed detailed reason why the download failed.
progress	number	Only applicable when "status" is `Downloading`, indicates download finished percentage, in a range of [1, 99]. * Once hitting 100, the status will be changed to `Download Success`. * or other problematic status (`Download Cancelled`, `Download Failed`) if the download stopped unexpectedly.

status	string	Indicates download status, which could be either
		- `Downloading`
		- Check `progress` key value for actual download percentage
		```json
		"download_info": {
		"progress": 7,
		"status": "Downloading",
		"url": "http://192.168.200.97:8080/5gb.zip"
		}
		```
		- `Download Success`
		```json
		"download_info": {
		"status": "Download Success",
		"url": "https://secure.eicar.org/eicar.com"
		}
		```
		- `Download Failed`
		- Check `error_detail` key value for an error explanation
		```json
		"download_info": {
		"error_detail": "Connection error",
		"status": "Download Failed",
		"url": "http://192.168.200.97:8080/2gb.zip"
		}
		```
		- `Download Timeout`
		- Expecting to occur when the download progress takes longer than what time window allowed in MetaDefender Core's pre-configured setting under workflow rule (under "SCAN" tab)
		```json
		"download_info": {
		"status": "Download Timeout",
		"url": "http://192.168.200.97:8080/2gb.zip"
		}
		```
		- `Download Cancelled`
		- Expecting to occur when user explicitly cancelled that file scan request, or batch request that the scan belongs to
		```json
		"download_info": {
		"status": "Download Cancelled",
		"url": "http://192.168.200.97:8080/5gb.zip"
		}
		```

NAME	TYPE	DESCRIPTION
url	string	Original download link which was specified in HTTP(S) request's `downloadfrom` header
extraction_info	object	
decrypted_status	enum	**ALLOWED:** Success, Failed Indicate that decryption phase is successful or not.
err_category	string	Error category
err_code	integer	Error code
err_details	string	Error message
is_encrypted_file	boolean	Indicate if file is password-protected or not.
file_info	object	
display_name	string	The filename reported via `filename` header.
file_size	integer	Total file size in bytes.
file_type	string	The filetype using mimetype.
file_type_description	string	The filetype in human readable format.
md5	string	File's MD5 hash.
sha1	string	File's SHA1 hash.
sha256	string	File's SHA256 Hash.
sha512	string	File's SHA512 Hash.
signer_infos	array	
digest_algorithm	string	Digest algorithm.
digest_encryption_algorithm	string	Encryption algorithm.
issuer	string	Entity that develops and registers the certificate.
serial_number	string	Serial number of the certificate.
vendor_name	string	Entity that is issued a certificate and utilize it for creating a digital signature.
version	string	Version of X.509 that is used in the certificate. This version field is zero-based. * 0: v1 * 1: v2 * 2: v3
type_category	array	
receive_data_timestamp	string	The timestamp when upload progress started (first byte received) (in milliseconds)
upload_time	integer	Total time elapsed for upload process (in milliseconds).
upload_timestamp	string	The timestamp when upload progress finished (all bytes received) (in milliseconds)
filetype_info	object	
file_info*	object	

NAME	TYPE	DESCRIPTION
description*	string	File type description
detected_by	string	Analyzer that detected the file type
encrypted*	boolean	File is password-protected or not
extensions*	string	File type extension
groupID*	string	File type category
groupIDs*	array	
group_description	string	File type category description
likely_type_ids	array	
score*	integer	Likelihood score of the file type
typeID*	string	File type ID
type*	string	MIME type
typeID*	string	File type ID
type_ids*	array	
final_verdict	object	
verdict*	enum	**ALLOWED:** `allowed, blocked` Final verdict of the file type analysis.
verdict_explanation*	string	Explanation of the final verdict.
is_file_type_mismatch	boolean	Indicates if the file type does not match the expected type.
other_detections	array	Other file type detections.
result_template_hash	string	SHA256 Hash of user-interface template. For web console only.
spoofing_info	object	
detection_result	string	Result of the spoofing detection.
result_explanation	string	Explanation of the spoofing detection result.
result_overview	string	Overview of the spoofing detection result.
opswatfilescan_info	object	
process_info	object	
blocked_reason	string	Provides the reason why the file is blocked (if so).
blocked_reasons	array	
file_type_skipped_scan	boolean	Indicates if the input file's detected type was configured to skip scanning.
hash_time	integer	Total time elapsed for computing hashes (in milliseconds).
outdated_data	array	
processing_time	integer	Total time elapsed during processing file (in milliseconds).
processing_time_details	object	
av_scan_time	integer	AV engines' processing time.

NAME	TYPE	DESCRIPTION
cdr_time	`integer`	Deep CDR engine's sanitization time.
dlp_time	`integer`	Proactive DLP engine's processing time.
extraction_time	`integer`	Archive extraction engine's processing time.
filetype_time	`integer`	FileType engine's processing time.
opswatfilescan_time	`integer`	OPSWAT Filescan engine's processing time.
others_time	`integer`	Total time elapsed for following processing tasks in the product (in milliseconds): * Decryption time (if receiving an encrypted file) * External scanner (if configured) * Post action (if configured) * Other internal processing time among components in the product
parse_dgsg_time	`integer`	Digital signature analyzing time.
vul_time	`integer`	Vulnerability engine's lookup time.
yara_time	`integer`	YARA engine's processing time.
filetype_wait_time	`integer`	FileType engine's wait time.
profile	`string`	The used rule name.
progress_percentage	`integer`	Percentage of processing completed (from 1-100).
queue_time	`integer`	Total time elapsed for file processing task was waiting in MetaDefender Core's queue until being picked up (queue_time = start_time - upload_timestamp) (in milliseconds).
result	`string`	The final result of processing the file (Allowed / Blocked / Processing).
user_agent	`string`	Identifier for the REST Client that calls the API.
username	`string`	User identifier who submitted scan request earlier.
verdicts	`array`	
post_processing	`object`	
actions_failed	`string`	Empty string if no action failed or list of failed actions, separated by "|".
actions_ran	`string`	List of successful actions, separated by "|". Empty string if otherwise.
converted_destination	`string`	Contains the name of the sanitized file.
converted_to	`string`	Contains target type name of sanitization.
copy_move_destination	`string`	Contains target type name of sanitization.
sanitization_details	`object`	
cdr_wait_time	`integer`	The time in milliseconds that the CDR process took to complete.
description	`string`	Action was successful or not.
details	`array`	
action*	`enum`	**ALLOWED:** `sanitized, removed` The type of action that was performed
count	`integer`	The number of objects that were sanitized/removed.

NAME	TYPE	DESCRIPTION
details	object	
action	enum	**ALLOWED:** sanitized, removed The type of action that was performed
count	integer	The number of objects that were sanitized/removed.
object_details	array	
object_name	string	The object type that was sanitized/removed.
description	string	Action was successful or not.
file_name	string	If an embedded file was sanitized.
object_details	array	
object_name*	string	The object type that was sanitized/removed.
failure_category	string	Deep CDR errors are classified into different categories. For more details, please find [Troubleshooting sanitization failures](https://docs.opswat.com/mdcore/deep-cdr/troubleshooting-sanitization-failures)
result	enum	**ALLOWED:** Sanitized, Sanitized failed, Sanitized skipped The result of the CDR process. - **Sanitized**: the file was successfully sanitized. - **Sanitized failed**: the file could not be sanitized due to an error during the process. - **Sanitized skipped**: the file was skipped from sanitization. Common reasons include the file being digitally signed or other policy-based exclusions.
result_template_hash	string	The hash value of the result template, which is used for displaying results on the Core UI and for internal communication between MetaDefender Core and the Deep CDR engine. This value is intended for system use only and is not required for external integration.
sanitized_file_info	object	
file_size	integer	Size of sanitized file in bytes.
sha256	string	SHA256 hash of sanitized file.
verdict	enum	**ALLOWED:** blocked, allowed The verdict of the CDR process. - **blocked**: the file is recommended for blocking by Deep CDR. - **allowed**: the file is recommended for allowing by Deep CDR as it found no reason to recommend blocking it.
verdict_explanations	array	
scan_results	object	
data_id	string	Data ID of the requested file
progress_percentage	integer	Track analysis progress until reaches 100.

NAME	TYPE	DESCRIPTION
scan_all_result_a	enum	**ALLOWED:** `No Threat Detected, Infected, Suspicious, Failed, Whitelisted, Blacklisted, Exceeded Archive Depth, Not Scanned, Encrypted Archive, Exceeded Archive Size, Exceeded Archive File Number, Password Protected Document, Exceeded Archive Timeout, Mismatch, Potentially Vulnerable File, Cancelled, Sensitive Data Found, Yara Rule Matched, Potentially Unwanted, Unsupported File Type, Extraction Failed, Scan Failed, Suspicious Verdict by Sandbox, Likely Malicious Verdict by Sandbox, Malicious Verdict by Sandbox, Blocked Verdict by Sandbox, Blocked Verdict by Deep CDR, Global Timeout Exceeded, Vulnerable Verdict by SBOM, Non-vulnerable Verdict by SBOM, Blocked Verdict by SBOM, Blocked by Post Action, Known Bad, Known Good, Unknown, Allowed Verdict by COO, Blocked Verdict by COO, Unknown Verdict by COO, In Progress, Skip Processing Fast Symlink` The overall scan result as string
scan_all_result_i	enum	**ALLOWED:** `0, 1, 2, 3, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 38, 39, 40, 41, 42, 43, 255, 1014` The overall scan result as index in the Processing Results table.
start_time	string	Timestamp when the scanning process starts.
total_avs	integer	Total number of scanning engines used as part of this analysis.
total_time	integer	Total time elapsed during scan (in milliseconds).
scan_details	object	
ClamAV	object	
def_time	string	The database definition time for this engine
eng_id	string	The unique identification string for the engine
location	string	Where this engine is deployed (local/remote).
scan_result_i	integer	Scan result as index in the Processing Results table
scan_time	integer	The time elapsed during scan with this engine (in milliseconds).
threat_found	string	The threat name, IF scan result is Infected or Suspicious. Otherwise empty string or error message from the engine.
wait_time	integer	Time elapsed between sending file to Core and receiving the result from the engine (in milliseconds).
vulnerability_info	object	
result	object	
code	integer	The result code for vulnerability check, 0 means a successful check
hash	string	The file's SHA1 hash value
method	enum	**ALLOWED:** `50700` The method used by OESIS Framework, it should be 50700 every time.
timestamp	string	Timestamp of the request issued

NAME	TYPE	DESCRIPTION
timing	`integer`	The vulnerability check's duration in milliseconds
detected_product	`object`	
has_kb	`boolean`	Indicates whether any KBs or MSBs exist for this hash
has_vulnerability	`boolean`	Indicates whether any vulnerabilities have been associated with the particular product
is_current	`boolean`	True if this product's patch level is current, defaults to true
product	`object`	
id	`integer`	The OPSWAT product id
name	`string`	The product name
remediation_link	`string`	A link where product updates or patches can be obtained
severity	`enum`	**ALLOWED:** `LOW, MODERATE, IMPORTANT, CRITICAL, NOT_AVAILABLE, UNKNOWN` String description of Severity level: * `LOW` * `MODERATE` * `IMPORTANT` * `CRITICAL` * `NOT_AVAILABLE` * `UNKNOWN`
sig_name	`string`	Product signature descriptor
signature	`integer`	OPSWAT signature id
vendor	`object`	
id	`integer`	The OPSWAT vendor id
name	`string`	The vendor name
version	`string`	The installed product version
version_data	`object`	
count_behind	`integer`	The number of patches behind of the installed product
feed_id	`integer`	The remote feed ID used to determine patch level
version	`string`	The current version of the product in the remote feed
vulnerabilites	`array`	
description	`string`	A text description of the specific vulnerability
details	`object`	
cpe	`string`	A CPE product reference
cve	`string`	A CVE identification string
cvss	`object`	
access-complexity	`string`	A CVSS access-complexity descriptor
access-vector	`string`	A CVSS access-vector descriptor
authentication	`string`	A CVSS authentication descriptor

NAME	TYPE	DESCRIPTION
availability-impact	string	A CVSS availability impact descriptor
confidentiality-impact	string	A CVSS confidentiality impact descriptor
generated-on-epoch	string	An epoch timestamp indicating CVSS generation time
integrity-impact	string	A CVSS integrity impact descriptor
score	string	A CVSS 10-point severity score
source	string	A CVSS source descriptor
cwe	string	A CWE group identification string
last_modified_epoch	string	An epoch timestamp indicating source last update time
published-epoch	string	An epoch timestamp indicating source publishing time
references	array	
severity	enum	**ALLOWED:** LOW, MODERATE, IMPORTANT, CRITICAL, NOT_AVAILABLE, UNKNOWN String description of Severity level: * `LOW` * `MODERATE` * `IMPORTANT` * `CRITICAL` * `NOT_AVAILABLE` * `UNKNOWN`
severity_index	integer	A 5 point scale numerical description of Severity level with 5 being greatest and 0 being unknown
static_id	integer	An OPSWAT identifier for the vulnerability
verdict	integer	The vulnerability check's duration in milliseconds * `0` - No Vulnerability Found * `1` - Vulnerability Found * `3` - Failed * `16` - Processing Timed Out
yara	object	
hits	object	
verdict	enum	**ALLOWED:** 0, 1, 2, 3, 4 The overall result for the analyzed file. Value will be one of the following: | index | status | |---------------|------------------------------| | 0 | Clean | | 1 | Found matched data | | 2 | Suspicious | | 3 | Failed | | 4 | Not scanned |

STATUS CODE - 404: Invalid hash format

1.4 GET /file/rules

Fetching Available Analysis Rules

Retrieve all available rules with their custom configurations. Fetching available processing rules.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication. Only those rules are returned, that: * Match the apikey's role sent using the apikey header, or * Are not restricted to a specific role.
user_agent	string		The user agent string value sent in the header (specified by the client). Only those rules are returned, that: * Match the client's user agent sent using the user_agent header, or * Are not restricted to a specific user agent. For details see KB article [What are Security Policies and how do I use them?](https://onlinehelp.opswat.com/corev4/What_are_Security_Policies_and_how_do_I_use_them_.html).

RESPONSE

STATUS CODE - 200: Returns the list of available rules.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
ARRAY OF OBJECT WITH BELOW STRUCTURE		
max_file_size	integer	The maximum allowed file size (in bytes) for this rule.
name	string	A unique identifier for identify in the used rule for a scan..
global_timeout	object	
value	integer	The timeout value in seconds.
enabled	boolean	Indicates whether the global timeout is enabled.

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

1.5 GET /file/converted/{data_id}

Download Sanitized Files

Retrieve sanitized file based on the data_id

REQUEST

PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*data_id	string	8101abae27be4d63859c55d9e0ed0135	The data_id comes from the result of `Analyze a file`. In case of sanitizing the content of an archive, the data_id of contained file can be found in `Fetch analysis result`.

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Returns the sanitized content.

RESPONSE MODEL - application/octet-stream

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Not found"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

1.6 GET /file/download/{data_id}

Download either sanitized files or DLP processed files

Retrieve sanitized file based on the data_id. In case there's no sanitized file, and DLP processed file is available, user will retrieve DLP processed file.

REQUEST

PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*data_id	string	8101abae27be4d63859c55d9e0ed0135	The data_id comes from the result of `Analyze a file`. In case of sanitizing the content of an archive, the data_id of contained file can be found in `Fetch analysis result`.

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Returns the sanitized or DLP processed content.

RESPONSE MODEL - application/octet-stream

STATUS CODE - 404: File could not be found

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "File could not be found"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

1.7 POST /file/{data_id}/cancel

Cancel File Analysis

When cancelling a file analysis, the connected analysis (e.g. files in an archive) that are still in progress will be cancelled also.

The cancelled analysis will be automatically closed.

REQUEST

PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*data_id	string		Unique submission identifier. Use this value to reference the submission.

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Analysis was sucessfully cancelled.

RESPONSE MODEL - application/json

EXAMPLE:

```
{
 "<<data_id>>": "cancelled"
}
```

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Data ID not found (invalid id) or Requests resource was not found

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

1.8 GET /file/webhook/{data_id}

Query webhook status

Prior to being notified when webhook mode is enabled, the client can request MetaDefender Cluster API Gateway for the file processing webhook status at any time.

REQUEST

PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*data_id	string		The `data_id` of the file to query.

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Webhook status is fetched successfully.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
data_id	`string`	The file submission identifier
request_time	`string`	A timestamp when the request has been made.
status_code	`integer`	What was the returned HTTP status code. * `200` - Callback was sent successfully * `403` - ContentAccessDenied. The access to the remote content was denied (similar to HTTP(S) error 401). * `404` - ContentNotFoundError. The remote content was not found at the server (similar to HTTP(S) error 404). * `408` - TimeoutError. The connection to the remote server timed out. * `503` - HostNotFoundError. The remote host name was not found (invalid hostname). * `520` - RemoteHostClosedError. The remote server closed the connection prematurely, before the entire reply was received and processed. * `444` - Other error types.
url	`string`	What was the called URL (should match the `callbackurl` header).

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Not found"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

2. AUTH

Authentication APIs

User authentication is done via username & password.

2.1 POST /login

Login

Initiate a new session. Required for using protected REST APIs.

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
user*	string	Username
password*	string	User's password

EXAMPLE:

```
{
 "user": "admin",
 "password": "admin"
}
```

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
oms-csrf-token*	string	The randomly generated token used to prevent CSRF attacks
session_id*	string	The apikey used to make API calls which requires authentication

STATUS CODE - 403: Invalid credentials

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	<error message> will describe the incident. More details would be logged in MetaDefender Cluster services logs

EXAMPLE:

```
{
 "err": "Failed to login"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

2.2 POST /logout

Logout

Destroy session for not using protected REST APIs.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
response*	string	

STATUS CODE - 400: Bad Request.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err*	string	

STATUS CODE - 403: Invalid user information.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err*	string	

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

3. BATCH

Group the analysis requests in batches. Supported with endpoints: MetaDefender Cluster API Gateway.

3.1 POST /file/batch

Initiate Batch

Create a new batch and retrieve the batch_id

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.
rule	string		Select rule for the analysis, if no header given the default rule will be selected (URL encoded UTF-8 string of rule name)
user_agent	string		user_agent header used to identify (and limit) access to a particular rule. For rule selection, `rule` header should be used.
user-data	string		Name of the batch (max 1024 bytes, URL encoded UTF-8 string).

RESPONSE

STATUS CODE - 200: Batch created successfully.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
batch_id*	string	The batch identifier used to submit files in the batch and to close the batch.

EXAMPLE:

```
{
 "batch_id": "74c85f475147439bac4d33b181853923"
}
```

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

3.2 POST /file/batch/{batchId}/close

Close Batch

The batch will be closed and files can no longer be added to the current batch.

REQUEST

PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*batchId	string		The batch identifier used to submit files in the batch and to close the batch.

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Batch successfully closed.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
batch_files	object	
batch_count	integer	The total number of files/entries in the batch.
files_in_batch	array	
data_id	string	Unique identifer for the file.
detected_by	integer	Total number of engines that detected this file.
display_name	string	The filename reported via `filename` header.
file_size	integer	Total file size in bytes.
file_type	string	The filetype using mimetype.
file_type_description	string	The filetype in human readable format.
process_info	object	
blocked_reason	string	Provides the reason why the file is blocked (if so).
progress_percentage	integer	Percentage of processing completed (from 1-100).
result	string	The final result of processing the file (Allowed / Blocked / Processing).
verdicts	array	

NAME	TYPE	DESCRIPTION
progress_percentage	integer	Track analysis progress until reaches 100.
scan_all_result_a	enum	**ALLOWED:** No Threat Detected, Infected, Suspicious, Failed, Whitelisted, Blacklisted, Exceeded Archive Depth, Not Scanned, Encrypted Archive, Exceeded Archive Size, Exceeded Archive File Number, Password Protected Document, Exceeded Archive Timeout, Mismatch, Potentially Vulnerable File, Cancelled, Sensitive Data Found, Yara Rule Matched, Potentially Unwanted, Unsupported File Type, Extraction Failed, Scan Failed, Suspicious Verdict by Sandbox, Likely Malicious Verdict by Sandbox, Malicious Verdict by Sandbox, Blocked Verdict by Sandbox, Blocked Verdict by Deep CDR, Global Timeout Exceeded, Vulnerable Verdict by SBOM, Non-vulnerable Verdict by SBOM, Blocked Verdict by SBOM, Blocked by Post Action, Known Bad, Known Good, Unknown, Allowed Verdict by COO, Blocked Verdict by COO, Unknown Verdict by COO, In Progress, Skip Processing Fast Symlink The overall scan result as string
scan_all_result_i	enum	**ALLOWED:** 0, 1, 2, 3, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 38, 39, 40, 41, 42, 43, 255, 1014 The overall scan result as index in the Processing Results table.
scanned_with	integer	The total number of engines used to analyze this file.
first_index	integer	The starting index in the batch. Used for pagination.
page_size	integer	The number of entries per page.
batch_id	string	The batch unique identifer
is_closed	boolean	The batch status (open/close).
process_info	object	
blocked_reason	string	Provides the reason why the file is blocked (if so).
file_type_skipped_scan	boolean	Indicates if the input file's detected type was configured to skip scanning.
profile	string	The used rule name.
result	string	The final result of processing the file (Allowed / Blocked / Processing).
user_agent	string	Identifier for the REST Client that calls the API.
username	string	User identifier who submitted scan request earlier.
scan_results	object	
batch_id	string	The batch unique identifer

NAME	TYPE	DESCRIPTION
scan_all_result_a	enum	**ALLOWED:** No Threat Detected, Infected, Suspicious, Failed, Whitelisted, Blacklisted, Exceeded Archive Depth, Not Scanned, Encrypted Archive, Exceeded Archive Size, Exceeded Archive File Number, Password Protected Document, Exceeded Archive Timeout, Mismatch, Potentially Vulnerable File, Cancelled, Sensitive Data Found, Yara Rule Matched, Potentially Unwanted, Unsupported File Type, Extraction Failed, Scan Failed, Suspicious Verdict by Sandbox, Likely Malicious Verdict by Sandbox, Malicious Verdict by Sandbox, Blocked Verdict by Sandbox, Blocked Verdict by Deep CDR, Global Timeout Exceeded, Vulnerable Verdict by SBOM, Non-vulnerable Verdict by SBOM, Blocked Verdict by SBOM, Blocked by Post Action, Known Bad, Known Good, Unknown, Allowed Verdict by COO, Blocked Verdict by COO, Unknown Verdict by COO, In Progress, Skip Processing Fast Symlink The overall scan result as string
scan_all_result_i	enum	**ALLOWED:** 0, 1, 2, 3, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 38, 39, 40, 41, 42, 43, 255, 1014 The overall scan result as index in the Processing Results table.
start_time	string	Timestamp when the scanning process starts.
total_avs	integer	Total number of scanning engines used as part of this analysis. Not like files, batch is not processed by engine, so this value is always 0.
total_time	integer	Total time elapsed during scan (in milliseconds).
user_data	string	Metadata submitted at batch creation.

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Not found"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

3.3 GET /file/batch/{batchId}

Status of Batch Analysis

Retrieve status report for the entire batch

REQUEST

PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*batchId	string		The batch identifier used to submit files in the batch and to close the batch.

QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
`first`	`integer`		The first item order in the list of files in this batch
`size`	`integer`		The number of items to be fetched next, counting from the item order indicated in first header

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
`apikey`	`string`		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Batch progress paginated report (50 entries/page).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
batch_files	`object`	
batch_count	`integer`	The total number of files/entries in the batch.
files_in_batch	`array`	
data_id	`string`	Unique identifer for the file.
detected_by	`integer`	Total number of engines that detected this file.
display_name	`string`	The filename reported via `filename` header.
file_size	`integer`	Total file size in bytes.
file_type	`string`	The filetype using mimetype.
file_type_description	`string`	The filetype in human readable format.
process_info	`object`	
blocked_reason	`string`	Provides the reason why the file is blocked (if so).
progress_percentage	`integer`	Percentage of processing completed (from 1-100).
result	`string`	The final result of processing the file (Allowed / Blocked / Processing).
verdicts	`array`	
progress_percentage	`integer`	Track analysis progress until reaches 100.

NAME	TYPE	DESCRIPTION
scan_all_result_a	enum	**ALLOWED:** No Threat Detected, Infected, Suspicious, Failed, Whitelisted, Blacklisted, Exceeded Archive Depth, Not Scanned, Encrypted Archive, Exceeded Archive Size, Exceeded Archive File Number, Password Protected Document, Exceeded Archive Timeout, Mismatch, Potentially Vulnerable File, Cancelled, Sensitive Data Found, Yara Rule Matched, Potentially Unwanted, Unsupported File Type, Extraction Failed, Scan Failed, Suspicious Verdict by Sandbox, Likely Malicious Verdict by Sandbox, Malicious Verdict by Sandbox, Blocked Verdict by Sandbox, Blocked Verdict by Deep CDR, Global Timeout Exceeded, Vulnerable Verdict by SBOM, Non-vulnerable Verdict by SBOM, Blocked Verdict by SBOM, Blocked by Post Action, Known Bad, Known Good, Unknown, Allowed Verdict by COO, Blocked Verdict by COO, Unknown Verdict by COO, In Progress, Skip Processing Fast Symlink The overall scan result as string
scan_all_result_i	enum	**ALLOWED:** 0, 1, 2, 3, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 38, 39, 40, 41, 42, 43, 255, 1014 The overall scan result as index in the Processing Results table.
scanned_with	integer	The total number of engines used to analyze this file.
first_index	integer	The starting index in the batch. Used for pagination.
page_size	integer	The number of entries per page.
batch_id	string	The batch unique identifer
is_closed	boolean	The batch status (open/close).
process_info	object	
blocked_reason	string	Provides the reason why the file is blocked (if so).
file_type_skipped_scan	boolean	Indicates if the input file's detected type was configured to skip scanning.
profile	string	The used rule name.
result	string	The final result of processing the file (Allowed / Blocked / Processing).
user_agent	string	Identifier for the REST Client that calls the API.
username	string	User identifier who submitted scan request earlier.
scan_results	object	
batch_id	string	The batch unique identifer

NAME	TYPE	DESCRIPTION
scan_all_result_a	enum	**ALLOWED:** No Threat Detected, Infected, Suspicious, Failed, Whitelisted, Blacklisted, Exceeded Archive Depth, Not Scanned, Encrypted Archive, Exceeded Archive Size, Exceeded Archive File Number, Password Protected Document, Exceeded Archive Timeout, Mismatch, Potentially Vulnerable File, Cancelled, Sensitive Data Found, Yara Rule Matched, Potentially Unwanted, Unsupported File Type, Extraction Failed, Scan Failed, Suspicious Verdict by Sandbox, Likely Malicious Verdict by Sandbox, Malicious Verdict by Sandbox, Blocked Verdict by Sandbox, Blocked Verdict by Deep CDR, Global Timeout Exceeded, Vulnerable Verdict by SBOM, Non-vulnerable Verdict by SBOM, Blocked Verdict by SBOM, Blocked by Post Action, Known Bad, Known Good, Unknown, Allowed Verdict by COO, Blocked Verdict by COO, Unknown Verdict by COO, In Progress, Skip Processing Fast Symlink The overall scan result as string
scan_all_result_i	enum	**ALLOWED:** 0, 1, 2, 3, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 38, 39, 40, 41, 42, 43, 255, 1014 The overall scan result as index in the Processing Results table.
start_time	string	Timestamp when the scanning process starts.
total_avs	integer	Total number of scanning engines used as part of this analysis. Not like files, batch is not processed by engine, so this value is always 0.
total_time	integer	Total time elapsed during scan (in milliseconds).
user_data	string	Metadata submitted at batch creation.

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Not found"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

3.4 GET /file/batch/{batchId}/certificate

Download Signed Batch Result

Download digitally signed status report for the entire batch

REQUEST

PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*batchId	string		The batch identifier used to submit files in the batch and to close the batch.

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
`apikey`	`string`		Generated \`session_id\` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an \`apikey\` for API calls that require authentication.
`metadata`	`json`		In JSON format, this can be used to: Include additional information in the response YML. Currently, one supported field in the metadata is \`include_vul_info\`, which can be set to \`true\` or \`false\` to indicate whether vulnerability processing information should be included or not. It is strongly recommended to apply URL encoding before sending \`metadata\` to Metadefender Cluster API Gateway to prevent unexpected issues related to encoding errors or unsafe characters.

RESPONSE

STATUS CODE - 200: Signed batch result and certificate are sent back in response body (YAML format).

RESPONSE MODEL - application/x-yaml

EXAMPLE:

```
"--- batch_id: 092876200fb54cfb80b6e3332c410ae9 user_data: the user data from the header from
batch creation cert_sha1_fingerprint: <some cert serial value> batch_files:\n  batch_count:
1\n  files_in_batch:\n  - data_id: 9112b225f0634f189a2bb46ec1a7826f\n    display_name:
New%20Text%20Document.txt\n    file_size: 5\n    scan_all_result_i: 0\n    process_info:\n
blocked_reason:\n      result: Allowed\n    md5: 42b130c3ce46e058f30712838cebf420\n    sha1:
ed94baf55ca851055fb76045f6949bca2f865605\n    sha256:
f4191b3ec6ce93aaf712919a38e52815c5da9c91d2b141df920bc8bcb5cbb8e3\n    sha512: \"\"\n
vulnerabilities:\n      - cve: CVE-2021-45463\n        cvss:\n          score: 6.8\n
cvss_3_0:\n          base_score: 7.8\n      - cve: CVE-2018-12713\n        cvss:\n
score: 6.4\n        cvss_3_0:\n          base_score: 9.1\nprocess_info:\n  blocked_reason:\n
file_type_skipped_scan: false\n  profile: File scan\n  result: Allowed\n  user_agent:
webscan\nscan_results:\n  scan_all_result_a: No Threat Detected\n  scan_all_result_i: 0\n
start_time: 2017-05-23T11:22:03.010Z\n  total_avs: 14\n  total_time: 995\n...\n--- signature:
881d22220c4ca0557d7c7d5c5794d53a8a2780997cd65b27b6e7f1c099a15de03dbcb5edbeaea7aafa6099fab37be
07017b39e3e3a7d66c550f44eb59a096c54d5b9555cb28198546fbec57c33b717751d333a09733d95dd876e2798d0
44c8caef828f4352b91f9a6d057253bb1a9461e0e0e0bf4313a80895998d645bebc81841ff3499589c80ffc4e8a19
0d1ec9b3e4126d86659d303b0e1f22d9289c9c4671d35532b55ad4620e048a78bb405b573897da63efdd5f036692c
934a82d9bdc9b9862e7fea5e8abeeb1444be0689d50373c5c0632484950c0fe0337ed5f91bdf26986f7cff8aa3431
bf4bc948fc127c16ba13ec679fe9f67e7586075c1f467454fa8cf40e9cd501291c95d862eb16f4477c17d1711294f
0ff2b3a1140bd53dbd1fbb0846af6062e9e4e2e1a09af3448503ed11e342164e535fc268bf7d8fbc28ed946cd2bb8
ea075f2295d2fa8392076d41608c3b5decf8fab3a5ec7de190f07583331e0517e5f361735cd59326622dc8b07b10a
464028de781a063e408f918c1d5534329140f4e4dc1a717d808d6784410410b00d36cb9a345f5bbc11fa1c58ee28f
8e7b863f3ea2c923ec5fb2ac29eaa4ddc0d6d9dfd3f16a97f207dc2858410a577c7f4a92ff01bad3229f5fcdb08e2
1df9869a113272aa9d96bfdfe8bfb3a50414c174e16a3504e5780c2718779b0757298546f287ef7ea86e67510d48a
8 certificate: |\n  -----BEGIN CERTIFICATE-----\n
MIIGJzCCBA+gAwIBAgIBATANBgkqhkiG9w0BAQUFADCBsjELMAkGA1UEBhMCRlIx\n
DzANBgNVBAgMBkFsc2FjZTETMBEGA1UEBwwKU3RyYXNib3VyZzEYMBYGA1UECgwP\n
d3d3LmZyZWVsYW4ub3JnMRAwDgYDVQQLDAdmcmVlbGFuMS0wKwYDVQQDDCRGcmVl\n
```

```
bGFuIFNhbXBsZSBDZXJ0aWZpY2F0ZSBBdXRob3JpdHkxIjAgBgkqhkiG9w0BCQEW\n
E2NvbnRhY3RAZnJlZWxhbi5vcmcwHhcNMTIwNDI3MTAzMTE4WhcNMjIwNDI1MTAz\n
MTE4WjB+MQswCQYDVQQGEwJGUjEPMA0GA1UECAwGQWxzYWNlMRgwFgYDVQQKDA93\n
d3cuZnJlZWxhbi5vcmcxEDAOBgNVBAsMB2ZyZWVsYW4xDjAMBgNVBAMMBWFsaWNl\n
MSIwIAYJKoZIhvcNAQkBFhNjb250YWN0QGZyZWVsYW4ub3JnMIICIjANBgkqhkiG\n
9w0BAQEFAAOCAg8AMIICCgKCAgEA3W29+ID6194bH6ejLrIC4hb2Ugo8v6ZC+Mrc\n
k2dNYMNPjcOKABvxxEtBamnSaeU/IY7FC/giN622LEtV/3oDcrua0+yWuVafyxmZ\n  yTKUb4/GUgafRQPf/
eiX9urWurtIK7XgNGFNUjYPq4dSJQPPhwCHE/LKAykWnZBX\n
RrX0Dq4XyApNku0IpjIjEXH+8ixE12wH8wt7DEvdO7T3N3CfUbaITl1qBX+Nm2Z6\n  q4Ag/
u5rl8NJfXg71ZmXA3XOj7zFvpyapRIZcPmkvZYn7SMCp8dXyXHPdpSiIWL2\n  uB3KiO4JrUYvt2GzLBUThp+lNSZaZ/
Q3yOaAAUkOx+1h08285Pi+P8lO+H2Xic4S\n
vMq1xtLg2bNoPC5KnbRfuFPuUD2/3dSiiragJ6uYDLOyWJDivKGt/72OVTEPAL9o\n
6T2pGZrwbQuiFGrGTMZOvWMSpQtNl+tCCXlT4mWqJDRwuMGrI4DnnGzt3IKqNwS4\n
Qyo9KqjMIPwnXZAmWPm3FOKe4sFwc5fpawKO01JZewDsYTDxVj+cwXwFxbE2yBiF\n
z2FAHwfopwaH35p3C6lkcgP2k/zgAlnBluzACUI+MKJ/G0gv/uAhj1OHJQ3L6kn1\n  SpvQ41/
ueBjlunExqQSYD7GtZ1Kg8uOcq2r+WISE3Qc9MpQFFkUVllmgWGwYDuN3\n
Zsez95kCAwEAAaN7MHkwCQYDVR0TBAIwADAsBglghkgBhvhCAQ0EHxYdT3BlblNT\n
TCBHZW5lcmF0ZWQgQ2VydGlmaWNhdGUwHQYDVR0OBBYEFFlfyRO6G8y5qEFKikl5\n
ajb2fT7XMB8GA1UdIwQYMBaAFCNsLT0+KV14uGw+quK7Lh5sh/JTMA0GCSqGSIb3\n
DQEBBQUAA4ICAQAT5wJFPqervbja5+90iKxi1d0QVtVGB+z6aoAMuWK+qgi0vgvr\n
mu9ot2lvTSCSnRhjeiP0SIdqFMORmBtOCFk/kYDp9M/91b+vS+S9eAlxrNCB5VOf\n  PqxEPp/wv1rBcE4GBO/
c6HcFon3F+oBYCsUQbZDKSSZxhDm3mj7pb67FNbZbJIzJ\n
70HDsRe2O04oiTx+h6g6pW3cOQMgIAvFgKN5Ex727K4230B0NIdGkzuj4KSML0NM\n
slSAcXZ41OoSKNjy44BVEZv0ZdxTDrRM4EwJtNyggFzmtTuV02nkUj1bYYYC5f0L\n
ADr6s0XMyaNk8twlWYlYDZ5uKDpVRVBfiGcq0uJIzIvemhuTrofh8pBQQNkPRDFT\n  Rq1iTo1Ihhl3/
Fl1kXk1WR3jTjNb4jHX7lIoXwpwp767HAPKGhjQ9cFbnHMEtkro\n
RlJYdtRq5mccDtwT0GFyoJLLBZdHHMHJz0F9H7FNk2tTQQMhK5MVYwg+LIaee586\n
CQVqfbscp7evlgjLW98H+5zylRHAgoH2G79aHljNKMp9BOuq6SnEglEsiWGVtu2l\n  hnx8SB3sVJZHeer8f/
UQQwqbAO+Kdy70NmbSaqaVtp8jOxLiidWkwSyRTsuU6D8i\n
DiH5uEqBXExjrj0FslxcVKdVj5glVcSmkLwZKbEU1OKwleT/iXFhvooWhQ==\n  -----END CERTIFICATE-----
\n...\n"
```

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "Not found"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

3.5 POST /file/batch/{batchId}/cancel

Cancel Batch

When cancelling a batch, the connected analysis that are still in progress will be cancelled also.

The cancelled batch will be closed.

REQUEST

PATH PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*batchId	string		The batch identifier used to submit files in the batch and to close the batch.

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Batch cancelled.

RESPONSE MODEL - application/json

EXAMPLE:

```
{
 "<<batch_id>>": "cancelled"
}
```

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Batch not found (invalid id)

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

4. LICENSE

Retrieve the current license information.

4.1 GET /admin/license

Get current license information

Fetch details about the longest expiry active license among all activated licenses.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Information about the licensed product (product type, number of activations, deploymentId, expiration date and days left)

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
days_left	integer	Number of days left before expiration
expiration	string	Expiration date in MM/DD/YYYY format.
licensed_engines	array	
licensed_to	string	Name of the entity to which the license is issued.
max_agent_count	string	Total number of deployed MetaDefender Agents attached to this MetaDefender Core instance.
online_activated	boolean	Track online/offline activation mode
product_id	string	Official MetaDefender base SKU licensed.

NAME	TYPE	DESCRIPTION
product_name	string	Official MetaDefender base product name licensed.

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

5. STATS

Health check and statistics about MetaDefender Core instance usage.

5.1 GET /stat/engines

Engine Status

Return the status of the latest engines between the MetaDefender Core instances.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: An array with all the engines and their details.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
ARRAY OF OBJECT WITH BELOW STRUCTURE		
abandoned	boolean	Indicates if this engine is abandoned.
active	boolean	If used by at least one engine
def_time	string	The database definition time for this engine
download_progress	integer	The percentage progress of download
download_time	string	When this engine downloaded from the update server.
eng_id	string	Engine internal ID
eng_name	string	Engine name
eng_type	string	Engine type in human readable form

NAME	TYPE	DESCRIPTION
eng_ver	string	Engine's version (format differs from one engine to another).
engine_type	enum	**ALLOWED:** av, archive, filetype Engine's type: * av * archive * filetype
notified_messages	array	A list of messages from engine.
pinned	boolean	Indicate if this engine is pinned.
state	enum	**ALLOWED:** downloading, downloaded, staging, production, removed, temporary failed, permanently failed, content invalid, download failed Status of the engine: * downloading * downloaded * staging * production * removed * temporary failed * permanently failed * content invalid * download failed
type	string	The type of information, whether it is engine or engine's database.

5.2 GET /stat/nodes

Instance Status Overview

Retrieve status details of all available MetaDefender Core instances.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Status details of MetaDefender Core instances.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
external_nodes_allowed	boolean	Indicates whether external nodes can connect; always true.
max_node_count	integer	Total number of available MetaDefender Core instances.
statuses	array	
address	string	Location of the Core instance; currently always return empty string.
available_mem	integer	The number of available RAM in this system.
cpu_cores	integer	The number of CPU Cores allocated to this Core instance.
current_processing_files	integer	Number of objects currently being processed by the Core instance.
engines	array	
active	boolean	If used by at least one engine
db_ver	string	The database version for this engine
def_time	string	The database definition time for this engine
download_time	string	The database download time for this engine
eng_name	string	Engine name
eng_ver	string	Engine's version (format differs from one engine to another).
engine_type	enum	**ALLOWED:** `av, archive, filetype` Engine's type: * av * archive * filetype
id	string	Engine internal ID
issues	array	A list of all potential problems on this engine.
free_disk_space	integer	Reported available disk on Core instance (in bytes).
id	string	Identifier of the worker that deployed this Core instance.
info_disk_space	array	
dirs	array	list of directories used by MetaDefender Core.
free	integer	Free space on the disk (in bytes).
location	string	Disk location.
total	integer	Total space on the disk (in bytes).
issues	array	
description	string	Text detailing the issue.
severity	string	How important is the reported issue.
load	integer	Current CPU utilization on Core instance (in percentage).
os	string	Current used OS
scan_queue	integer	Number of objects currently being processed by the Core instance.

NAME	TYPE	DESCRIPTION
scan_queue_details	object	
archive_scan_queue_ratio	number	Ratio of archive scan queue, always -1 for Core in Cluster mode.
available_slots	integer	The number of slots is available, always -1 for Core in Cluster mode.
extracted_file_slots	integer	Number of child files being processing
file_slots	integer	Number of files taken from REST by the current Core instance
total_scan_queue	integer	Total scan queue, always -1 for Core in Cluster mode.
total_disk_space	integer	The amount of disk space is allocated on Core instance (in Byte).
total_mem	integer	How much memory is allocated on Core instance (in MB).
total_scan_queue	integer	The maximum queue size is allowed, always -1 for Core in Cluster mode.
uptime	integer	How long this Core is in operation (in second).
version	string	Product version

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

5.3 GET /readyz

Get health check status

Fetch current status of system health.

REQUEST

QUERY PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
verbose	boolean	true	Optional. Show detailed result of system health.

RESPONSE

STATUS CODE - 200: System is currently healthy.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
status*	boolean	System-wide status, indicate if all components are healthy.
scan_queue*	object	
number_in_queue*	integer	Number of objects being processed by the system.
status*	boolean	The operational status of the scan process; true if the system contains the required minimum of healthy MetaDefender Core instances.
license*	object	
status*	enum	**ALLOWED:** expired, invalid, ok License status.
components*	object	
status*	boolean	Aggregate component status.
datalake	object	
status	boolean	DataLake overall status.
detail	string	Status detail message
caching	object	
status	boolean	Caching overall status.
detail	string	Status detail message.
broker	object	
status	boolean	Broker overall status.
detail	string	Status detail message.
filestorage	object	

NAME	TYPE	DESCRIPTION
status	boolean	File storage overall status.
detail	string	Status detail message.
identity	object	
status	boolean	Identity service overall status.
detail	string	Status detail message.
ometascan*	object	
status*	boolean	MetaDefender Core overall status.
detail	string	Detail message.
instance	array	
callback-service*	object	
status*	boolean	Callback service overall status.
instance	array	

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

STATUS CODE - 503: System is currently unhealthy.

API Reference

Control Center

API Version: v2.5.1

Developer Guide

This is the API documentation for *MetaDefender Cluster Control Center Public API*. If you would like to evaluate or have any questions about this documentation, please contact us via our Contact Us form.

How to Interact with MetaDefender Cluster Control Center using REST API

The MetaDefender Cluster Control Center empowers administrators and system engineers to efficiently manage system operations, including:

1. Establishing and maintaining essential service connections.

2. Deploying and managing MetaDefender Core, MetaDefender Cluster API Gateway instances.
3. Managing licenses.
4. Administering user accounts and access controls.
5. Configuring and enforcing security protocols.
6. Monitoring the overall system health and system performance.

OPSWAT recommends using the JSON-based REST API. The available methods are documented below.

OPSWAT provides some sample codes on GitHub to make it easier to understand how the MetaDefender REST API works.

CONTACT

NAME: API Support
EMAIL: feedback@opswat.com
URL: https://github.com/OPSWAT/metadefender-core-openapi3
Terms of service: https://onlinehelp.opswat.com/policies/

Security and Authentication

SECURITY SCHEMES

KEY	TYPE	DESCRIPTION
apikey	apiKey	Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

API

1. USER MANAGEMENT

User management APIs

The APIs for manage users and user directories.

1.1 GET /admin/user

List all users

Returns a list of all users in the server.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: List of users retrieved successfully.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
ARRAY OF OBJECT WITH BELOW STRUCTURE		
api_key	string	Associated apikey with this user
directory_id	integer	To which User Directories belongs to (LOCAL/System/etc.)
display_name	string	Which is the name showed in the Management Console
email	string	User's email address
id	integer	User's unique identifier

NAME	TYPE	DESCRIPTION
name	string	User's full name
description	string	User's description, 256 characters maximum
roles	array	
ui_settings	object	

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

1.2 POST /admin/user

Create user

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
api_key	`string`	Associated apikey with this user
directory_id	`integer`	To which User Directories belongs to (LOCAL/System/etc.)
display_name	`string`	Which is the name showed in the Management Console
email	`string`	User's email address
name	`string`	User's full name
description	`string`	User's description, 256 characters maximum
roles	`array`	
ui_settings	`object`	
password	`string`	The user's password

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request processed successfully.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
api_key	`string`	Associated apikey with this user
directory_id	`integer`	To which User Directories belongs to (LOCAL/System/etc.)
display_name	`string`	Which is the name showed in the Management Console
email	`string`	User's email address
name	`string`	User's full name
description	`string`	User's description, 256 characters maximum
roles	`array`	
ui_settings	`object`	

STATUS CODE - 400: Bad Request (e.g. invalid header, invalid request body).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

1.3 DELETE /admin/user/{user_id}

Delete a user

Delete a user by id from the system.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request processed successfully.

RESPONSE MODEL - application/json

EXAMPLE:

```
{
 "result": "Success"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Item does not exist"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Not allowed"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

1.4 POST /user/changepassword

Change Password for local user

Modify the current password for the user identified by apikey

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
old_password	string	The current password in plain text
new_password	string	The new password in plain text

EXAMPLE:

```
{
 "old_password": "admin",
 "new_password": "123456"
}
```

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request processed successfully

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
result	string	

STATUS CODE - 400: Bad Request (e.g. invalid header, invalid request body).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

2. ADMIN

Admin specific API requests.

2.1 GET /admin/userdirectory

List all user directories

Retrieve a list of all user directories configured in the system.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: List of user directories retrieved successfully.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
ARRAY OF OBJECT WITH BELOW STRUCTURE		
id	integer	Internal used identifier
name	string	Name of the user directory
type	string	Type of the user directory (e.g., LDAP, Local, etc.)
enabled	boolean	If the user directory is enabled or not
lockout_attempts	integer	Number of failed login attempts before the user is locked out
lockout_timeout	integer	Time in seconds before the user can try to log in again after being locked out

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

2.2 POST /admin/role

Create new role

Add a new user role to the system.

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
name	string	The name identifier of the role
display_name	string	The extended name showed in the Management Console.
rights	object	
cert	array	
configlog	array	
engines	array	
license	array	
retention	array	
rule	array	
scanlog	array	
update	array	
updatelog	array	
users	array	
workflow	array	
zone	array	
healthcheck	array	
fetch	array	
download	array	
deployment	array	
service	array	
packageupload	array	

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request processed successfully

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		

NAME	TYPE	DESCRIPTION
name	string	The name identifier of the role
display_name	string	The extended name showed in the Management Console.
rights	object	
cert	array	
configlog	array	
engines	array	
license	array	
retention	array	
rule	array	
scanlog	array	
update	array	
updatelog	array	
users	array	
workflow	array	
zone	array	
healthcheck	array	
fetch	array	
download	array	
deployment	array	
service	array	
packageupload	array	
editable*	boolean	If the role can be altered or not.
id*	integer	Internal used identifier

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

2.3 DELETE /admin/role/{role_id}

Delete a role

Delete a role by id from the system.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request processed successfully.

RESPONSE MODEL - application/json

EXAMPLE:

```
{
 "result": "Success"
}
```

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Item does not exist"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Not allowed"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

3. AUTH

Authentication APIs

User authentication is done via username & password.

3.1 POST /login

Login

Initiate a new session. Required for using protected REST APIs.

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
user*	string	Username
password*	string	User's password

EXAMPLE:

```
{
 "user": "admin",
 "password": "admin"
}
```

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
oms-csrf-token*	string	The randomly generated token used to prevent CSRF attacks
session_id*	string	The apikey used to make API calls which requires authentication

STATUS CODE - 403: Invalid credentials

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	<error message> will describe the incident. More details would be logged in MetaDefender Cluster services logs

EXAMPLE:

```
{
 "err": "Failed to login"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

3.2 POST /logout

Logout

Destroy session for not using protected REST APIs.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
response*	string	

STATUS CODE - 400: Bad Request.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err*	string	

STATUS CODE - 403: Invalid user information.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err*	string	

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

4. CONFIG

Configure the product through APIs (especially the Settings). Will require admin apikey..

4.1 PUT /admin/config/auditlog/cleanup

Audit clean up

Setting audit record cleanup time (cleanup records older than).

Note: The cleanup range is defined in hours.

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
cleanuprange	integer	The number of hours of data retention. Anything older than this number will be deleted. _**Note**_: If `cleanuprange` is `0`, the cleanup functionality will be disabled.

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request processed successfully

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		

NAME	TYPE	DESCRIPTION
cleanuprange	integer	The number of hours of data retention. Anything older than this number will be deleted.

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

4.2 PUT /admin/config/filestorage/cleanup

File storage clean up

Setting file storage clean up time (clean up records older than).

Note:The clean up range is defined in hours.

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
cleanuprange	`integer`	The number of hours of data retention. Anything older than this number will be deleted. _**Note**_: If `cleanuprange` is `0`, the cleanup functionality will be disabled.

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request processed successfully

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
cleanuprange	`integer`	The number of hours of data retention. Anything older than this number will be deleted.

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

4.3 PUT /admin/config/warehouse/cleanup

Executive report clean up

Setting executive report clean up time (clean up records older than).

Note:The clean up range is defined in hours.

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
cleanuprange	integer	The number of hours of data retention. Anything older than this number will be deleted. _**Note**_: If

NAME	TYPE	DESCRIPTION
		`cleanuprange` is `0`, the cleanup functionality will be disabled.

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request processed successfully

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
cleanuprange	integer	The number of hours of data retention. Anything older than this number will be deleted.

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

4.4 PUT /admin/config/scanhistory/cleanup

Processing history clean up

Setting processing history clean up time (clean up records older than).

Note:The clean up range is defined in hours.

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
cleanuprange	integer	The number of hours of data retention. Anything older than this number will be deleted. _**Note**_: If `cleanuprange` is `0`, the cleanup functionality will be disabled.

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request processed successfully

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
cleanuprange	integer	The number of hours of data retention. Anything older than this number will be deleted.

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

4.5 PUT /admin/config/session

Session settings

Configure settings for session generated upon a successful login See more at Login

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
absoluteSessionTimeout	integer	The interval (in milliseconds) for overall session length timeout (regardless of activity).
allowCrossIpSessions	boolean	Allow requests from the same user to come from different IPs.
allowDuplicateSession	boolean	Allow same user to have multiple active sessions.
sessionTimeout	integer	The interval (in milliseconds) for the user's session timeout, based on last activity. Timer starts after the last activity for the apikey.

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request processed successfully

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
absoluteSessionTimeout	integer	The interval (in milliseconds) for overall session length timeout (regardless of activity).
allowCrossIpSessions	boolean	Allow requests from the same user to come from different IPs.
allowDuplicateSession	boolean	Allow same user to have multiple active sessions.
sessionTimeout	integer	The interval (in milliseconds) for the user's session timeout, based on last activity. Timer starts after the last activity for the apikey.

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

4.6 GET /admin/config/sessioncookie

Get session cookie attributes

Getting session cookie attributes

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request processed successfully.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
samesite	number	SameSite attribute accepts three values: * `Strict` - cookies will only be sent in a first-party context, not be sent along with requests initiated by third party websites. * `Lax` - cookies are not sent on normal cross-site subrequests, but are sent when a user is navigating to the origin site. * `None` - cookies will be sent in all contexts. Default value: `Lax`

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

4.7 PUT /admin/config/sessioncookie

Update session cookie attributes

Modifying session cookie attributes

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
samesite	number	SameSite attribute accepts three values: * `Strict` - cookies will only be sent in a first-party context, not be sent along with requests initiated by third party websites. * `Lax` - cookies are not sent on normal cross-site subrequests, but are sent when a user is navigating to the origin site. * `None` - cookies will be sent in all contexts. Default value: `Lax`

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request processed successfully.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
samesite	number	SameSite attribute accepts three values: * `Strict` - cookies will only be sent in a first-party context, not be sent along with requests initiated by third party websites. * `Lax` - cookies are not sent on normal cross-site subrequests, but are sent when a user is navigating to the origin site. * `None` - cookies will be sent in all contexts. Default value: `Lax`

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Not found"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
```

```
    }
```

5. INSTALLERS

Upload and manage installers for the MetaDefender Core and MetaDefender Cluster API Gateway.

5.1 GET /admin/installer

Get uploaded installers

Retrieve information about an uploaded installer.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request processed successfully

RESPONSE MODEL - application/json

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Not found"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

5.2 POST /admin/installer

Upload installer

Upload installers for the MetaDefender Core, MetaDefender Cluster API Gateway and MetaDefender Cluster Callback Service.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.
*filename	string		The name of the installer file to upload. **Note**: Ensure the filename remains same with the original MY OPSWAT download (e.g: ometascan-5.15.0-1-x64.msi)

RESPONSE

STATUS CODE - 200: Request processed successfully

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
installer_id	string	Unique identifier of the uploaded installer.

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

5.3 DELETE /admin/installer/{installer_id}

Delete an uploaded installer

Delete an uploaded installer.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request processed successfully

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
result	string	Success message.

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Not found"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

6. SERVICES

Add essential services and view connection status.

6.1 GET /admin/service

Get the status of all essential services.

Retrieve the status of all added services within the MetaDefender Cluster system.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Details of all added services and their status.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
service_type	object	
healthy_instances	number	Number of healthy instances for the service.
overall_status	string	Aggregated status across all instances of the service.
overall_status_description	string	Description of the overall status.
instances	array	
service_id	string	Unique service identifier.
message	string	Optional status/message.
display_name	string	Human friendly name. Defaults to "host:port" if absent.

NAME	TYPE	DESCRIPTION
status_description	string	Human readable status explanation.
host	string	Hostname or IP.
port	number	Service's port.
version	string	Service version (semantic or other).
added_by	string	User or system that registered the service.
last_update	number	Unix epoch milliseconds of last update.
last_healthy	number	Unix epoch milliseconds of last confirmed healthy state.
detail	object	
cpu_usage	number	CPU usage (implementation specific units).
platform	string	Operating system/platform the service is running on.
role	string	Service role (e.g. primary, secondary).
db_size	number	Database size in bytes.
ram	object	
total_bytes	number	Total RAM available.
used_bytes	number	RAM currently in use.
disk	object	
total_bytes	number	Total disk space available.
used_bytes	number	Disk space currently in use.

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

6.2 POST /admin/service

Connect and check essential services status.

Establish connections and retrieve the status of essential services within the MetaDefender Cluster system.

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
ONE:OF	`object`	
OPTION:1	`object`	

NAME	TYPE	DESCRIPTION
host*	string	the host address of the service.
port*	integer	the port number of the service.
connection_key*	string	the connection key for the service.
OPTION:2	object	
host*	string	the host address of the service
port*	integer	the port number of the service
user*	string	the user name for the service.
password*	string	the password for the service.
OPTION:3	object	
host*	string	the host address of the service.
port*	integer	the port number of the service.
user	string	the user name for the service.
password	string	the password for the service.

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request to add service was successful

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
ARRAY OF OBJECT WITH BELOW STRUCTURE		
result	string	the result of the service addition, can be either "ok" or "error"
service_id	string	The unique identifier of the service if result is "ok"
detail	string	The error details if result is "error"

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

6.3 PUT /admin/service/{service_id}

Edit service details.

Update the display name and/or configuration details for a specific service. **Note**: Service configuration cannot be modified after instances have been deployed.

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
display_name	string	Display name for the service.
config	object	
host	string	the host address of the service
port	integer	the port number of the service
user	string	the user name for the service. Applicable for type `caching`, `broker`, `datalake`, and `warehouse`
password	string	the password for the service. Applicable for type `caching`, `broker`, `datalake`, and `warehouse`
connection_key	string	the connection key for the service. Applicable for type `filestorage`

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request to add service was successful

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
service_id	string	Unique service identifier.
message	string	Optional status/message.
display_name	string	Human friendly name. Defaults to "host:port" if absent.
status_description	string	Human readable status explanation.
host	string	Hostname or IP.
port	number	Service's port.
version	string	Service version (semantic or other).
added_by	string	User or system that registered the service.

NAME	TYPE	DESCRIPTION
last_update	number	Unix epoch milliseconds of last update.
last_healthy	number	Unix epoch milliseconds of last confirmed healthy state.
detail	object	
cpu_usage	number	CPU usage (implementation specific units).
platform	string	Operating system/platform the service is running on.
role	string	Service role (e.g. primary, secondary).
db_size	number	Database size in bytes.
ram	object	
total_bytes	number	Total RAM available.
used_bytes	number	RAM currently in use.
disk	object	
total_bytes	number	Total disk space available.
used_bytes	number	Disk space currently in use.

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Not found"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

6.4 DELETE /admin/service/{service_id}

Disconnect to service and remove their configurations.

Remove the connection and configuration details for a specific service. **Note**: Service configuration cannot be deleted after instances have been deployed.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request to remove service was successful

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
service_id	string	

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Not found"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

6.5 GET /admin/service/{service_type}

Get status for a specific service.

Retrieve the current status of a specific service, including all instance details. **Note**: The service_type must be one of: datalake, warehouse, caching, broker, filestorage.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request to get service type was successful

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
overall_status	string	Aggregated status for the service type (e.g. healthy, degraded, down)
overall_status_description	string	Human readable description of the aggregated status
healthy_instances	number	Count of instances currently considered healthy
instances	array	
service_id	string	Unique service identifier.
message	string	Optional status/message.
display_name	string	Human friendly name. Defaults to "host:port" if absent.
status_description	string	Human readable status explanation.
host	string	Hostname or IP.
port	number	Service's port.
version	string	Service version (semantic or other).
added_by	string	User or system that registered the service.
last_update	number	Unix epoch milliseconds of last update.
last_healthy	number	Unix epoch milliseconds of last confirmed healthy state.
detail	object	
cpu_usage	number	CPU usage (implementation specific units).
platform	string	Operating system/platform the service is running on.
role	string	Service role (e.g. primary, secondary).
db_size	number	Database size in bytes.
ram	object	
total_bytes	number	Total RAM available.
used_bytes	number	RAM currently in use.

NAME	TYPE	DESCRIPTION
disk	object	
total_bytes	number	Total disk space available.
used_bytes	number	Disk space currently in use.

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Not found"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

6.6 GET /admin/service/{service_type}/setting

Get service settings

Retrieve the current configuration settings for a specific service. **Note**: Supported only for the filestorage service type.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request to retrieve service settings was successful

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
max_replica	number	Maximum number of replicas, default is 1
min_replica	number	Minimum number of replicas, default is 1
cleanuprange	number	Cleanup interval in hours, default is 0
storage	object	
type	string	Storage backend type, can be `salt` or `none`
config	object	

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Not found"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

6.7 PUT /admin/service/{service_type}/setting

Edit setting of service

Update the configuration settings for a specific service. **Note**: Supported only for the filestorage service type.

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
max_replica	`number`	Maximum number of replicas, default is 1
min_replica	`number`	Minimum number of replicas, default is 1

NAME	TYPE	DESCRIPTION
cleanuprange	number	Cleanup interval in hours, default is 0
storage	object	
type	string	Storage backend type, can be `salt` or `none`
config	object	

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request to add service was successful

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
result	string	Success message

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "Not found"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

7. WORKERS

Connect, deploy, undeploy and manage workers.

7.1 GET /admin/worker

List connected workers

Retrieve a list of currently connected MetaDefender Cluster Worker services.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: A list of connected workers.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
ARRAY OF OBJECT WITH BELOW STRUCTURE		
worker_id	string	Unique identifier of the worker.
display_name	string	Display name for the worker.
platform	string	Operating system / platform of the worker.
os	string	Operating system details of the worker.
package_type	string	The deployment package type.
hardware	object	
cpu	object	
count	integer	Number of CPU cores.

NAME	TYPE	DESCRIPTION
model	string	CPU model name.
usage	number	CPU usage percentage.
disk	object	
available_bytes	integer	Available disk space in bytes.
total_bytes	integer	Total disk space in bytes.
memory	object	
available_bytes	integer	Available memory in bytes.
total_bytes	integer	Total memory in bytes.
user_name	string	Name of the user who added the worker.
host	string	The address (IP or hostname) of the worker.
port	integer	Port on which the worker is listening.
status	string	Current status of worker.
status_description	string	The description of worker's status
version	string	The version of the worker.
deployment_info	object	
type	string	Deployment type, can be \`ometascan\` or \`api-gateway\`.
installer_id	string	Identifier of the installer.
version	string	The instance version.
user_name	string	Name of the user who deployed the instance.
custom_config	object	

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		

NAME	TYPE	DESCRIPTION
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

7.2 POST /admin/worker

Connect to workers

Connect to MetaDefender Cluster Worker services.

REQUEST

REQUEST BODY - application/json

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request to add service was successful

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
ARRAY OF OBJECT WITH BELOW STRUCTURE		
display_name*	string	Display name for the worker.
host*	string	The address of the worker.
port*	number	The port on which the worker is listening.
result*	enum	**ALLOWED:** ok, failed Connection attempt result.
worker_id	string	Present only when result = ok. Unique identifier of the worker.
error	string	Present only when result = failed. Error message.

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

7.3 DELETE /admin/worker

Disconnect from workers

Disconnect from specified MetaDefender Cluster Worker services.

REQUEST

REQUEST BODY - application/json

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request to disconnect workers was successful

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
worker_id	enum	**ALLOWED:** Deleted Disconnection status of the worker.

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

7.4 GET /admin/worker/available/{installer_id}

Get available workers by installer_id.

Retrieve the list of available workers eligible for deployment for the specified installer ID.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request to get available workers was successful

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
ARRAY OF OBJECT WITH BELOW STRUCTURE		
worker_id	string	Unique identifier of the worker.
display_name	string	Display name for the worker.
platform	string	Operating system / platform of the worker.
os	string	Operating system details of the worker.
package_type	string	The deployment package type.
hardware	object	
cpu	object	
count	integer	Number of CPU cores.
model	string	CPU model name.
usage	number	CPU usage percentage.
disk	object	
available_bytes	integer	Available disk space in bytes.
total_bytes	integer	Total disk space in bytes.
memory	object	
available_bytes	integer	Available memory in bytes.
total_bytes	integer	Total memory in bytes.
user_name	string	Name of the user who added the worker.
host	string	The address (IP or hostname) of the worker.
port	integer	Port on which the worker is listening.
status	string	Current status of worker.
status_description	string	The description of worker's status
version	string	The version of the worker.
deployment_info	object	
type	string	Deployment type, can be \`ometascan\` or \`api-gateway\`.
installer_id	string	Identifier of the installer.
version	string	The instance version.
user_name	string	Name of the user who deployed the instance.
custom_config	object	

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		

NAME	TYPE	DESCRIPTION
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 404: Requests resource was not found.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Not found"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

7.5 POST /admin/worker/deploy

Deploy workers

Deploy the selected installer on one or more selected workers.

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
ONE:OF	object	
OPTION:1	object	
type*	enum	ALLOWED: ometascan
installer_id*	string	Identifier of the installer.
worker*	array	
config	object	
log_level	enum	DEFAULT:info ALLOWED: debug, info, warning, error
connection_per_file_service	integer	>=1 DEFAULT:4
OPTION:2	object	
type*	enum	ALLOWED: api-gateway
installer_id*	string	Identifier of the installer.
worker*	array	
cert	string	Certificate name (default empty). Only for api-gateway.
config	object	
port	integer	between 1 and 65535

NAME	TYPE	DESCRIPTION
		DEFAULT:8899
log_level	enum	DEFAULT:info ALLOWED: debug, info, warning, error
OPTION:3	object	
type*	enum	ALLOWED: callback-service
installer_id*	string	Identifier of the installer.
worker*	array	
config	object	
log_level	enum	DEFAULT:info ALLOWED: debug, info, warning, error

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request to add service was successful

RESPONSE MODEL - application/json

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

7.6 DELETE /admin/worker/deploy

Undeploy workers

Undeploy the specified workers.

REQUEST

REQUEST BODY - application/json

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request to undeploy workers was successful

RESPONSE MODEL - application/json

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	`string`	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

7.7 POST /admin/worker/upgrade

Upgrade deployed instances

Upgrade the deployed instances managed by the worker to a newer version

REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
version*	`string`	Target version to upgrade to.
type*	`enum`	**ALLOWED:** `ometascan, api-gateway, callback-service` Worker deployment type.

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: Request to upgrade workers was successful

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
result	string	

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```

7.8 GET /admin/worker/upgrade/version

Get upgradable instance version

Retrieve a list of available versions of MetaDefender Core and MetaDefender Cluster API Gateway for upgrading.

REQUEST

HEADER PARAMETERS

NAME	TYPE	EXAMPLE	DESCRIPTION
*apikey	string		Generated `session_id` from [Login](/docs/mdcore/metadefender-distributed-cluster/ref#userlogin) call can be used as an `apikey` for API calls that require authentication.

RESPONSE

STATUS CODE - 200: A list of available versions for upgrading.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
ometascan*	array	
api-gateway*	array	
callback-service*	array	

STATUS CODE - 400: Bad Request (e.g. invalid header, apikey is missing or invalid).

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Invalid header"
}
```

STATUS CODE - 403: Invalid user information or Not Allowed

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 405: The user has no rights for this operation.

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "Access denied"
}
```

STATUS CODE - 500: Unexpected event on server

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
err	string	Error reason

EXAMPLE:

```
{
 "err": "<error message>"
}
```