ADM

Arenadata Monitoring

Arenadata Monitoring (ADM) is a set of services that provide collection, storage, and visualization of monitoring metrics for Arenadata products.

Version CURRENT Language: EN

Contents

1. Concepts
   - ADM overview

2. Get started
   - Installation
     - Online installation
       - Install ADCM
       - Prepare hosts
       - Install ADM cluster
         - Create a cluster
         - Add services
         - Add hosts to a cluster
         - Add components
         - Configure services
         - Configure a cluster
         - Install a cluster
     - Offline installation
       - Install ADCM
       - Prepare hosts
       - Install Enterprise Tools cluster
         - Create a cluster
         - Add services
         - Add hosts to a cluster
         - Add components
         - Install a cluster
       - Install ADM cluster
         - Create a cluster
         - Add services
         - Add hosts to a cluster
         - Add components
         - Configure services
         - Configure a cluster
         - Import ET settings
         - Install a cluster

3. Services
   - Grafana
     - Service management via ADCM
   - VictoriaMetrics
     - Service management via ADCM

4. Cluster-level operations
   - Cluster management via ADCM
     - Cluster actions
       - Check
       - Manage SSL
       - Restart
       - Start
       - Stop
     - Service actions

5. References
   - Configuration parameters

6. Release notes
   - ADM releases
   - Version compatibility matrix

Arenadata Monitoring overview

Dmitry Bulgakov

Contents

To Table of Contents

Features

Architecture

Monitoring workflow

Features

Arenadata Monitoring (ADM) is a centralized monitoring and observability system for the Arenadata ecosystem, built on a modern stack of open-source components.

The main functional capabilities of ADM are listed below:

- Full metrics lifecycle support for observability and monitoring, from data collection to analysis and visualization.
- Centralized long-term metrics storage in a fault-tolerant distributed time-series database (TSDB) based on VictoriaMetrics Cluster.
- Active monitoring and a unified health model for Arenadata products (healthy, degraded, or critical).
- Support for product-specific dashboards delivered together with Arenadata products.
- Open architecture with integration points for external metrics collection and visualization systems.

Architecture

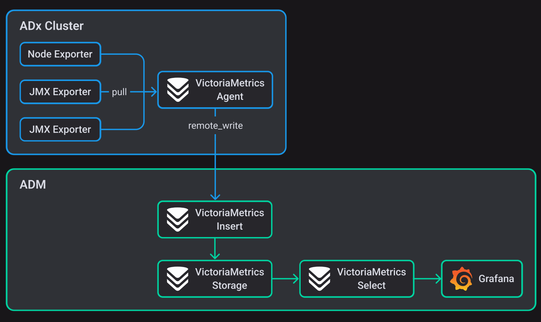
The main components of the ADM architecture are:

- VictoriaMetrics. A high-performance TSDB and monitoring storage platform that includes a distributed storage architecture and auxiliary components for collecting, storing, and querying metrics. ADM uses the following components of VictoriaMetrics:
  - **VictoriaMetrics Insert** — data ingestion.
  - **VictoriaMetrics Storage** — persistent storage, compression, and retrieval of time-series data.
  - **VictoriaMetrics Select** — database query execution.
- Grafana. A visualization and monitoring platform used to display metrics, logs, and traces collected from various data sources.

To make metrics available for export from the source cluster, the source cluster must have the following components installed:

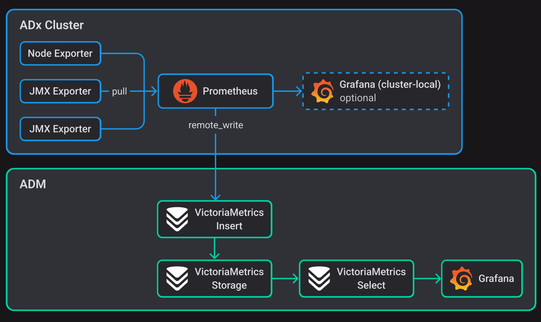
- Metrics exporters (Node Exporter, JMX Exporter). Agents collecting and exposing operating system and service metrics.
- One of:
  - **VictoriaMetrics Agent**. A lightweight agent for collecting metrics from various sources, labeling and filtering them, and forwarding them for storage to VictoriaMetrics Storage or any other metrics storage system via the Prometheus-compatible `remote_write` protocol.

> **NOTE**
>
> VictoriaMetrics Agent is currently not available on a number of Arenadata bundles.



Monitoring architecture (recommended components)

- Prometheus. A monitoring and alerting system that collects and stores metrics from applications and infrastructure.



Monitoring architecture (alternative components)

Monitoring workflow

1. The monitoring process begins with metrics generation on infrastructure nodes and services. Node Exporter components on each host collect OS-level metrics (CPU, RAM, disk metrics, network I/O). JMX Exporter retrieves metrics from JMX MBeans exposed by JVM-based applications and exposes them in Prometheus format. As a result, metrics from Java services become available.
2. One of the following components can be used for metrics collection and forwarding to a long-term storage:
   - VictoriaMetrics Agent periodically scrapes exporters over HTTP, can apply relabeling and filtering, and then forwards the data to the central VictoriaMetrics cluster using the `remote_write` mechanism. In this architecture, VictoriaMetrics Agent does not store data locally and acts as a lightweight metrics collector forming a unified edge layer. This approach reduces resource consumption on the cluster side and simplifies monitoring operations.
   - An alternative component for metrics collection is Prometheus. Prometheus independently scrapes exporters, stores metrics in its local TSDB, and simultaneously forwards them to the central VictoriaMetrics cluster via `remote_write`. The presence of local storage preserves cluster autonomy and ensures compatibility with existing Prometheus-compatible infrastructures. It also enables local metrics analysis with the help of a local Grafana instance connected to the local Prometheus server.
3. After the transmission, all metrics are received by the VictoriaMetrics Insert component of the VictoriaMetrics cluster. It accepts the incoming data stream and distributes time series across storage shards according to the cluster sharding algorithm.
4. The data is then stored in VictoriaMetrics Storage. This component is responsible for long-term metrics storage, data compression, retention policies, and, when configured, replication. High compression efficiency of VictoriaMetrics allows large volumes of time-series data to be stored with relatively low disk space and memory consumption.
5. When a user opens a dashboard in Grafana, the request is sent to the VictoriaMetrics Select component. It performs a distributed query, aggregates data from all storage nodes, and returns the result to the client. For the user, the system appears as a single centralized metrics storage regardless of which cluster the data originated from or how it was collected. Grafana connects to VictoriaMetrics as a unified data source. Dashboards display both infrastructure metrics collected by Node Exporter and application metrics exposed through JMX Exporter. This provides centralized observability for the entire distributed system through a single interface.

Installation

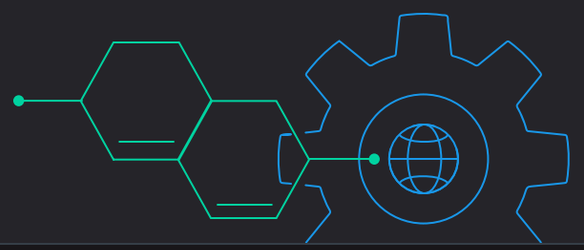
To Table of Contents

This tutorial guides you through the process of installing an Arenadata Monitoring (ADM) cluster. The easiest way to install it is to use Arenadata Cluster Manager (ADCM) — a separate Arenadata product designed for simple, convenient, and fast software deployment and operation.

You can install ADM via ADCM in two ways.

Online installation

on hosts with access to the Internet



Offline installation

on hosts without access to the Internet

Online installation

To Table of Contents

This topic describes **online** installation of ADM. It can be applied to the hosts with access to the Internet.

Step 1.

Install ADCM

Step 2.

Prepare hosts

Step 3.

Install ADM cluster

Install ASCM

Contents

- Step 1. Check prerequisites
  - Software and Hardware requirements
  - Check the ASCM port availability
  - Check the ASCM web interface
- Step 2. Check installation
  - Check the Python installation
  - Check the ASCM port availability
  - Check the ASCM web interface
- Step 3. Update ASCM

Step 1. Check prerequisites

Software and Hardware requirements

ASCM is delivered as a standalone software that can be run on Linux or Windows.

Linux

```
Ubuntu 20.04 LTS

CPU:
  Cores: 4
  Frequency: 2.4 GHz

RAM:
  Size: 8 GB

Disk:
  Free space: 50 GB
  Disk type: SSD

Network:
  Bandwidth: 100 Mbps
  Latency: 50 ms
```

Windows

```
Windows 10 Pro

CPU:
  Cores: 4
  Frequency: 2.4 GHz

RAM:
  Size: 8 GB

Disk:
  Free space: 50 GB
  Disk type: SSD

Network:
  Bandwidth: 100 Mbps
  Latency: 50 ms
```

Note

- The minimum hardware requirements listed above are for reference only. The actual hardware requirements may vary depending on the number of devices managed by ASCM and the number of concurrent users.
- For the latest list of supported operating systems, see [Supported Operating Systems and Versions](#).

Check the ASCM port availability

Before installing ASCM, check whether the ports required by ASCM are available. If a port is occupied, change the port number in the configuration file after the installation is complete.

Linux

```
# Check whether port 80 is occupied
netstat -tuln | grep :80

# Check whether port 443 is occupied
netstat -tuln | grep :443

# Check whether port 8080 is occupied
netstat -tuln | grep :8080

# Check whether port 8443 is occupied
netstat -tuln | grep :8443
```

If a port is occupied, the output is as follows:

```
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
```

Windows

```
# Check whether port 80 is occupied
netstat -ano | findstr :80

# Check whether port 443 is occupied
netstat -ano | findstr :443

# Check whether port 8080 is occupied
netstat -ano | findstr :8080

# Check whether port 8443 is occupied
netstat -ano | findstr :8443
```

If a port is occupied, the output is as follows:

```
TCP    0.0.0.0:80             0.0.0.0:0              LISTENING       1234
```

Note

- The default ports used by ASCM are 80 and 443.
- If a port is occupied, change the port number in the configuration file after the installation is complete. For details, see [Change the ASCM Port](#).

Check the ASCM web interface

After ASCM is installed, you can access the ASCM web interface through a browser to check whether the installation is successful.

1. Open a browser.
2. Enter the ASCM web interface address in the address bar. The default address is `https://<ASCM server IP address>:443`.

3. If the installation is successful, the ASCM login page is displayed.

Note

- If you cannot access the ASCM web interface, check whether the firewall is disabled and whether the ASCM service is running properly.
- If you use a self-signed certificate, your browser may display a security warning. You can ignore the warning and continue to access the ASCM web interface.

Step 2. Check installation

Check the Python installation

ASCM requires Python 3.8 or later. Check whether Python is installed and its version.

Linux

```
# Check the Python version
python3 --version

# Check the Python installation path
which python3
```

Windows

```
# Check the Python version
python --version

# Check the Python installation path
where python
```

Check the ASCM port availability

Check whether the ports required by ASCM are available. If a port is occupied, change the port number in the configuration file after the installation is complete.

Linux

```
# Check whether port 80 is occupied
netstat -tuln | grep :80

# Check whether port 443 is occupied
netstat -tuln | grep :443

# Check whether port 8080 is occupied
netstat -tuln | grep :8080

# Check whether port 8443 is occupied
netstat -tuln | grep :8443
```

If a port is occupied, the output is as follows:

```
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
```

Windows

```
# Check whether port 80 is occupied
netstat -ano | findstr :80

# Check whether port 443 is occupied
netstat -ano | findstr :443

# Check whether port 8080 is occupied
netstat -ano | findstr :8080

# Check whether port 8443 is occupied
netstat -ano | findstr :8443
```

If a port is occupied, the output is as follows:

```
TCP    0.0.0.0:80             0.0.0.0:0              LISTENING       1234
```

Note

- The default ports used by ASCM are 80 and 443.
- If a port is occupied, change the port number in the configuration file after the installation is complete. For details, see [Change the ASCM Port](#).

Check the ASCM web interface

After ASCM is installed, you can access the ASCM web interface through a browser to check whether the installation is successful.

1. Open a browser.
2. Enter the ASCM web interface address in the address bar. The default address is `https://<ASCM server IP address>:443`.

3. If the installation is successful, the ASCM login page is displayed.

Note

- If you cannot access the ASCM web interface, check whether the firewall is disabled and whether the ASCM service is running properly.
- If you use a self-signed certificate, your browser may display a security warning. You can ignore the warning and continue to access the ASCM web interface.

Step 3. Update ASCM

Update ASCM

Update ASCM to the latest version to obtain the latest features and security patches.

1. Log in to the ASCM web interface.
2. Choose **System Settings** > **Version Update**.
3. Click **Check for Updates**. If a new version is available, the update page is displayed.
4. Click **Update Now**. The system starts to download and install the update.
5. After the update is complete, the system prompts you to restart ASCM. Click **Restart Now**.

Note

- Before updating ASCM, back up the database and configuration files.
- The update process may take several minutes. Do not close the browser or power off the server during the update.
- If the update fails, check the network connection and disk space, and try again.

Prepare hosts

To Table of Contents

To manage a cluster and its nodes via ADCM, it is necessary to prepare hosts for a cluster and link these hosts to ADCM. This is the function of ADCM **hostproviders** — special plugins for ADCM, which encapsulate the complex mechanism of interaction between ADCM and the hosts controlled by it.

ADCM supports several types of hostproviders. For more information on how to configure hostproviders and use them to create hosts, refer to the following documentation:

- SSH
- Yandex Cloud
- VCloud (VMware)
- Cloud.ru Advanced
- CROC Cloud

CAUTION

When creating ADM hosts via ADCM, you should specify their FQDNs.

Create a cluster

Lisa Vaysman, Daria Barysheva

Contents

To Table of Contents

Step 1. Download a cluster bundle

Step 2. Upload a cluster bundle to ADCM

Step 3. Create a cluster based on the uploaded bundle

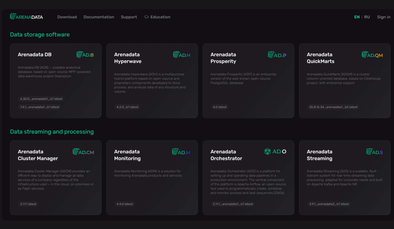
In ADCM, a ***cluster*** means a set of services running on a set of hosts.

Clusters distributions for ADCM are made in the form of so-called bundles. Generally a bundle is a regular archive that includes a description and software logic of cluster deployment and exploitation.

The steps for adding a cluster to ADCM are listed below.

Step 1. Download a cluster bundle

1. Go to https://network.arenadata.io/ and choose ***Arenadata Monitoring***.



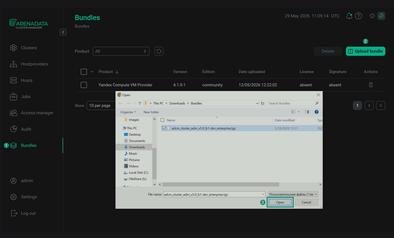
Switch to the bundle download page

2. On the next page, download a bundle for the required version of ADM.

Step 2. Upload a cluster bundle to ADCM

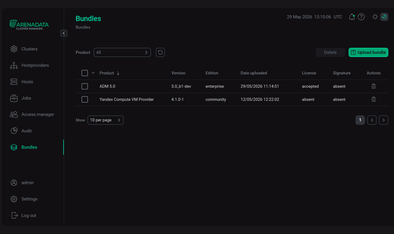
To upload a bundle to ADCM, follow the steps:

1. Select the ***Bundles*** item in the left navigation menu and click ***Upload bundle***.
2. Select a bundle in the Open File Dialog.



Upload a bundle

3. As a result of the performed actions, a bundle is displayed on the ***Bundles*** page.



The result of successful uploading a bundle

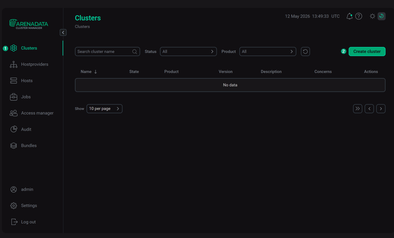
Step 3. Create a cluster based on the uploaded bundle

IMPORTANT

When creating a cluster via the ADCM web interface, you generate a new ADCM cluster instance. It means only adding the cluster to ADCM — at this stage no services are deployed.

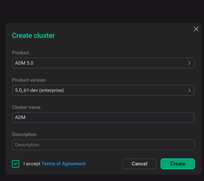
The steps for cluster creation are given below:

1. Select the ***Clusters*** item in the left navigation menu and click ***Create cluster***.



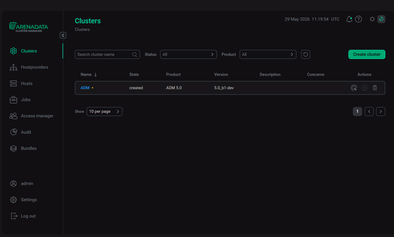
Switch to cluster creation

2. In the opened dialog, describe a new cluster instance:
   - Select an uploaded cluster bundle in the ***Product*** field.
   - Select a bundle version in the ***Product version*** field. Several versions become available in the case of different versions of the same bundle being uploaded.
   - Enter a cluster name in the ***Cluster name*** field.
   - Enter a cluster description in the ***Description*** field if necessary.
   - Set the ***I accept Terms of Agreement*** flag to sign the license agreement. This flag is displayed only for Enterprise bundles. To read the agreement text, refer to the ***Terms of Agreement*** link. You can also sign the license agreement on the ***Bundles*** page by clicking the desired product bundle.
   - Click ***Create***.



Create a cluster

3. As a result, the created cluster instance is displayed on the ***Clusters*** page.



The result of successful cluster creation

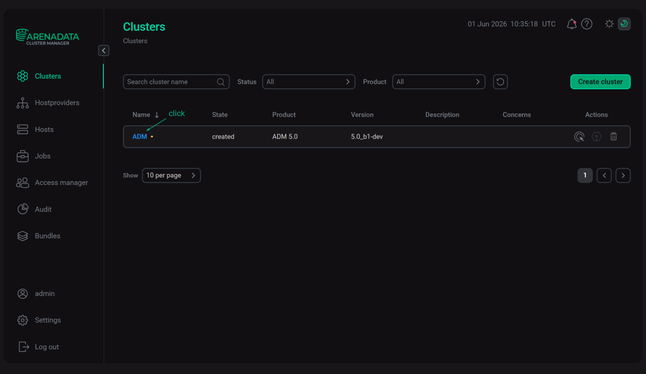
Add services

Lisa Vaysman, Daria Barysheva

To Table of Contents

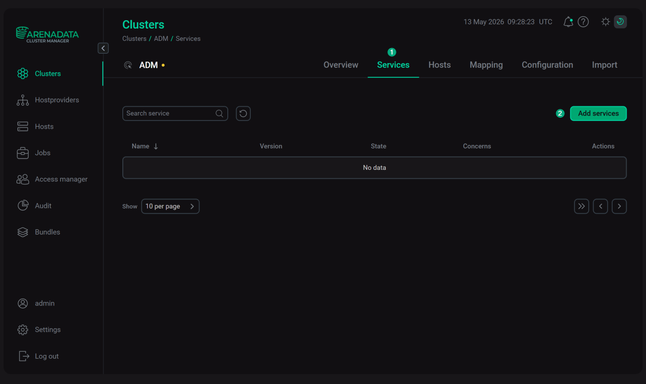
In ADCM, a **service** means a software that performs some function. Examples of services for ADM clusters: Grafana and VictoriaMetrics. The steps for adding services to a cluster are listed below:

1. Select a cluster on the ***Clusters*** page. To do this, click a cluster name in the ***Name*** column.



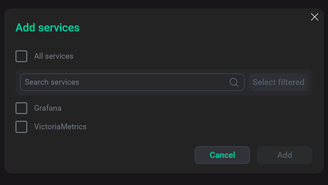
Select a cluster

2. Open the ***Services*** tab on the cluster page and click ***Add services*** (***Add service*** in first versions of ADCM).



Switch to adding services

3. In the opened dialog, select services that should be added to the cluster and click ***Add***.



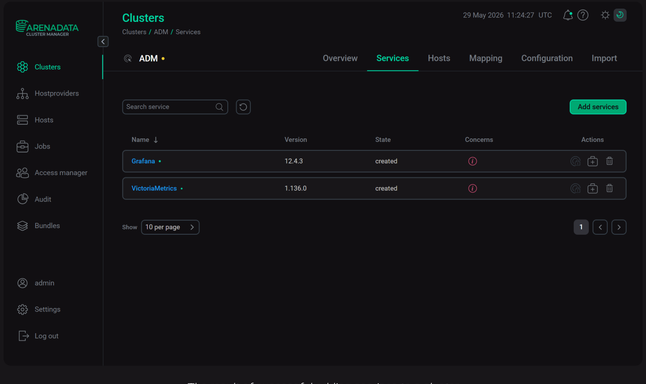
Select services

The brief description of available services is listed below.

Services that can be added to the ADM cluster

Service	Mandatory	Purpose
VictoriaMetrics	Yes	A time series database (TSDB) that is designed to collect and analyze application and infrastructure metrics
Grafana	No	A web service that allows you to create various sets of graphs and alerts for visualization of monitoring metrics

4. As a result, the added services are displayed on the ***Services*** tab.



The result of successful adding services to a cluster

NOTE

You can also add services later. The process of adding new services to already running cluster does not differ from installing a service from scratch.

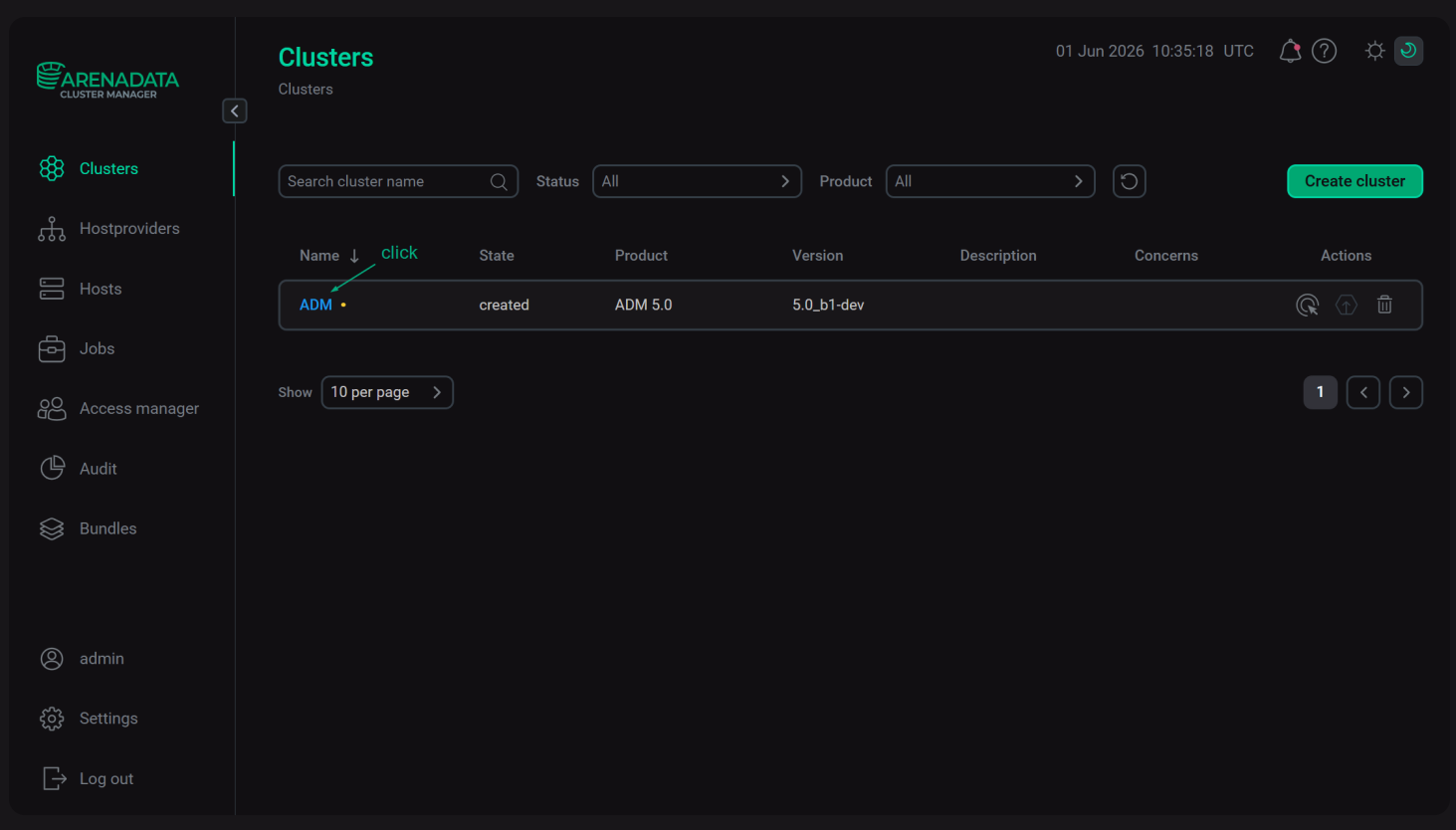
Add hosts to a cluster

Lisa Vaysman, Daria Barysheva

To Table of Contents

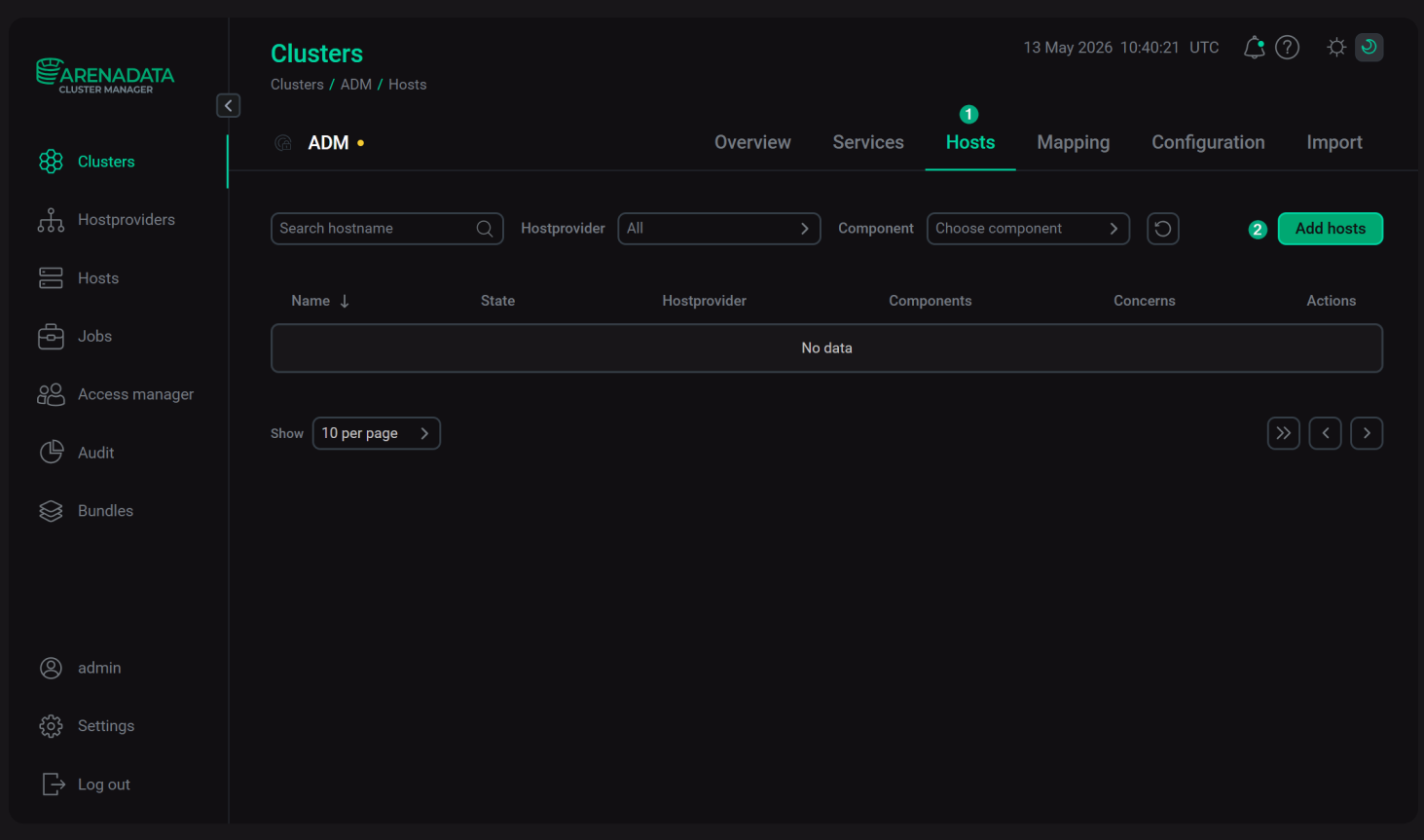
The steps for adding hosts to a cluster are listed below:

1. Select a cluster on the ***Clusters*** page. To do this, click a cluster name in the ***Name*** column.



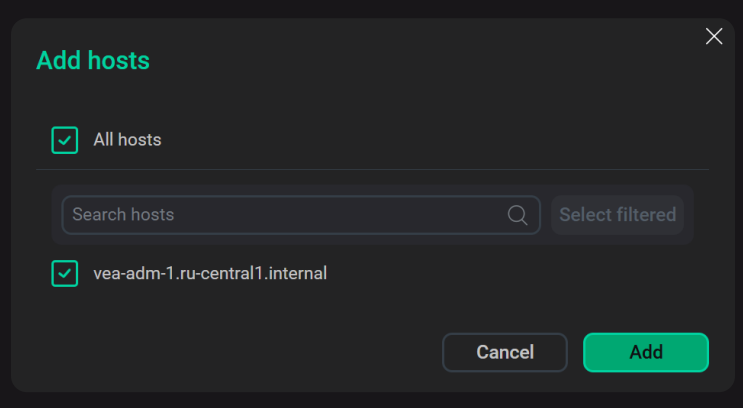
Select a cluster

2. Open the ***Hosts*** tab on the cluster page and click ***Add hosts***.



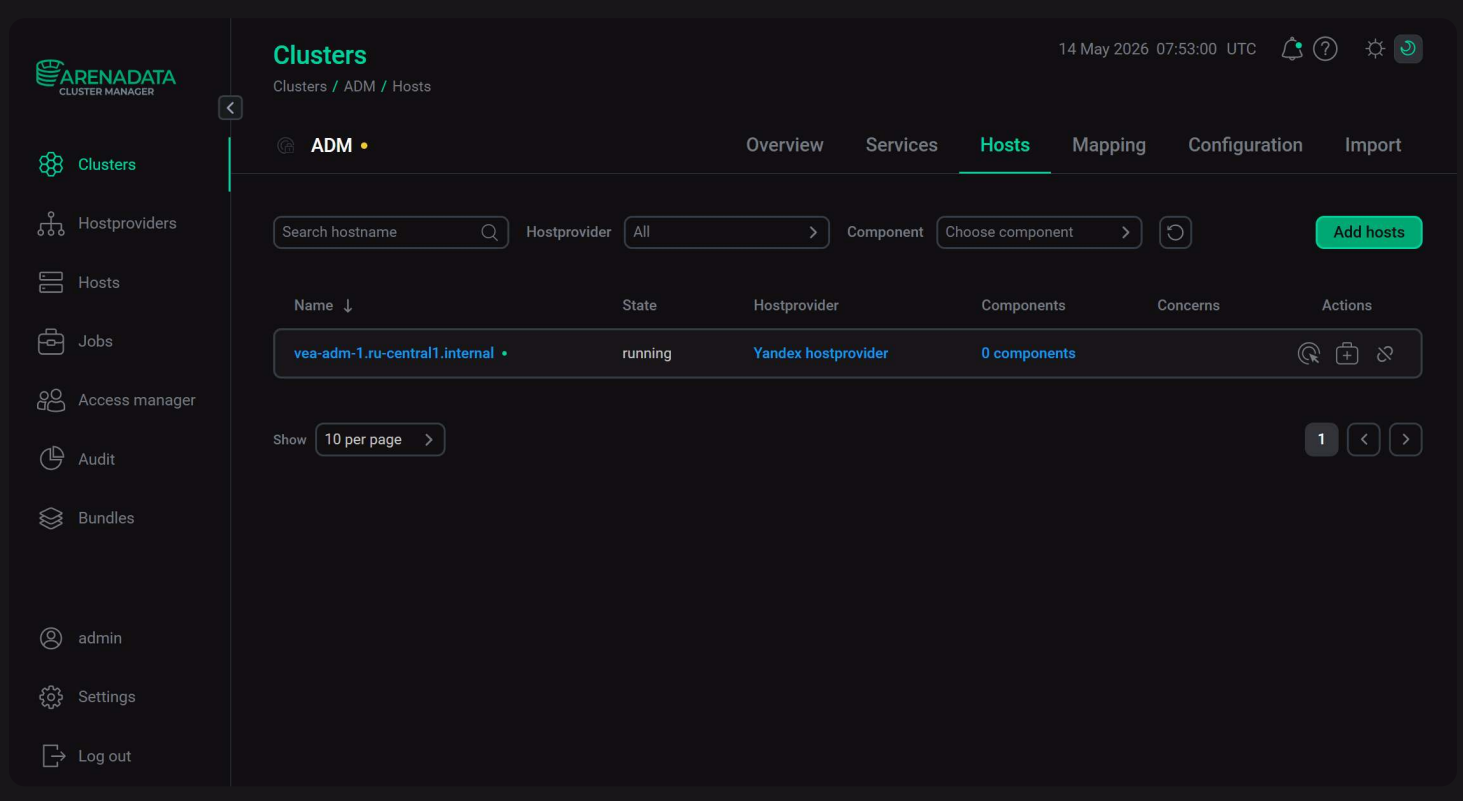
Switch to adding hosts

3. In the opened dialog, select hosts that should be added to the cluster and click ***Add***.



Select hosts

4. As a result, the added hosts are displayed on the ***Hosts*** tab.



The result of successful adding hosts to a cluster

Add components

Lisa Vaysman, Daria Barysheva

To Table of Contents

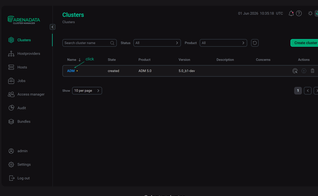
In ADCM, a **component** means a part of a service that should be deployed on one or several cluster hosts. For example, the VictoriaMetrics service consists of such components as VictoriaMetrics Insert, VictoriaMetrics Select, and VictoriaMetrics Storage.

IMPORTANT

Initially, there are no components on any of cluster hosts. You should map components to hosts manually.

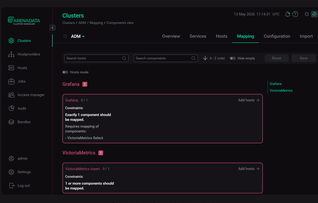
The steps for adding components to hosts are listed below:

1. Select a cluster on the **Clusters** page. To do this, click a cluster name in the ***Name*** column.



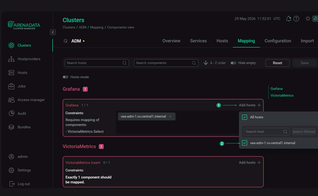
Select a cluster

2. Open the ***Mapping*** tab on the cluster page.



Switch to mapping service components

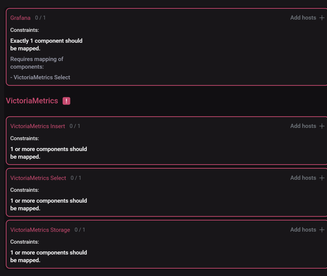
3. In the window that opens, map components to hosts. To assign hosts for each component, click ***Add hosts*** and select hosts in the pop-up window.



Select hosts for a component

Mandatory components

Each service can have mandatory and optional components. Mandatory components are highlighted in red. The ADCM UI also displays how many mandatory components should be installed.



Mandatory components

CAUTION

Without assigning a necessary number of hosts to mandatory components, the component mapping cannot be saved.

When distributing components, it is important to consider the following:

- **Component co-location on a host.** You can either install all components of an ADM cluster on a single host or distribute them among several hosts depending on your requirements. In particular, the VictoriaMetrics Insert, VictoriaMetrics Storage, and VictoriaMetrics Select components can be installed on the same host. Distributing them across several hosts provides minimal performance advantage, which is offset by higher operational complexity and resource overhead.
- **Changes to host number.** Once the VictoriaMetrics Storage component has been installed, if you subsequently change the number of hosts for this component, you must restart the VictoriaMetrics Insert component via the ***Restart*** action.
- **Replication factor calculations.** To ensure fault tolerance in case of a host loss, calculate the minimum number of hosts with the VictoriaMetrics Storage component installed using the following formula: `2 * N - 1` (where `N` is the replication factor value). For example, a replication factor of 2 requires a minimum of 3 hosts with the VictoriaMetrics Storage component.

  If the number of such hosts is lower than required to maintain the target replication factor, ADCM still allows cluster creation. In this case, the VictoriaMetrics Insert component writes metrics only to the available hosts running the VictoriaMetrics Storage component.

Monitoring VictoriaMetrics Storage availability

To monitor and identify VictoriaMetrics Storage host losses, check the following indicators on the VictoriaMetrics Insert side:

- **Internal metrics.** Connection failures cause an increase in the `vm_rpc_dial_errors_total` , `vm_rpc_connection_errors_total`, or `vm_rpc_handshake_errors_total` counters. You can view them in a browser at `http://vminsert-host>:8480/metrics`. Alternatively, connect to the VictoriaMetrics insert host via SSH and run:

  ```
  $ curl -s http://localhost:8480/metrics | grep -E
  "vm_rpc_dial_errors|vm_rpc_connection_errors|vm_rpc_handshake_errors"
  ```

  Example output indicating a failed host with the VictoriaMetrics Storage component (`gau-ub3`):

  ```
  vm_rpc_connection_errors_total{name="vminsert", addr="gau-ub3.develop.yc.adsw.io:8400", rpc_call="metric_rows"} 1
  vm_rpc_dial_errors_total{name="vminsert", addr="gau-ub3.develop.yc.adsw.io:8400", rpc_call="metric_rows"} 7718
  vm_rpc_handshake_errors_total{name="vminsert", addr="gau-ub3.develop.yc.adsw.io:8400", rpc_call="metric_rows"} 1
  ```

- **VictoriaMetrics Insert logs** (`docker logs`). Look for the `cannot dial storageNode … connection refused` error messages. To view the service logs, connect to the VictoriaMetrics Insert host via SSH and run:

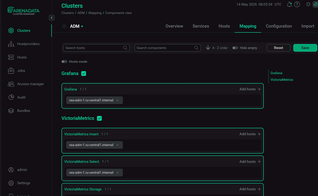
  ```
  $ sudo docker logs vminsert
  ```

  Example log entry indicating a failed host with the VictoriaMetrics Storage component:

  ```
  2026-06-19T13:45:12.123Z  error  VictoriaMetrics/app/vminsert/main.go:142  cannot dial storageNode "gau-ub3.
  develop.yc.adsw.io:8400": dial tcp 192.168.1.10:8400: connect: connection refused
  ```

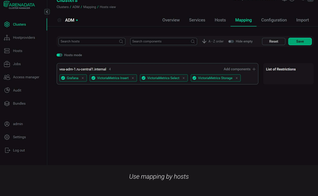
- **Network utilities.** You can also connect to the VictoriaMetrics Insert hosts via SSH and use standard Linux tools (such as `ss` or `netstat`) to verify connectivity to the VictoriaMetrics Storage hosts.

4. After component distribution is completed, click ***Save***.



Save mapping of components

Starting with ADCM 2.2.0, you can use another mapping mode — by hosts. To do this, switch on the ***Hosts mode*** toggle and map components to hosts by clicking ***Add components*** for each host sequentially.



Use mapping by hosts

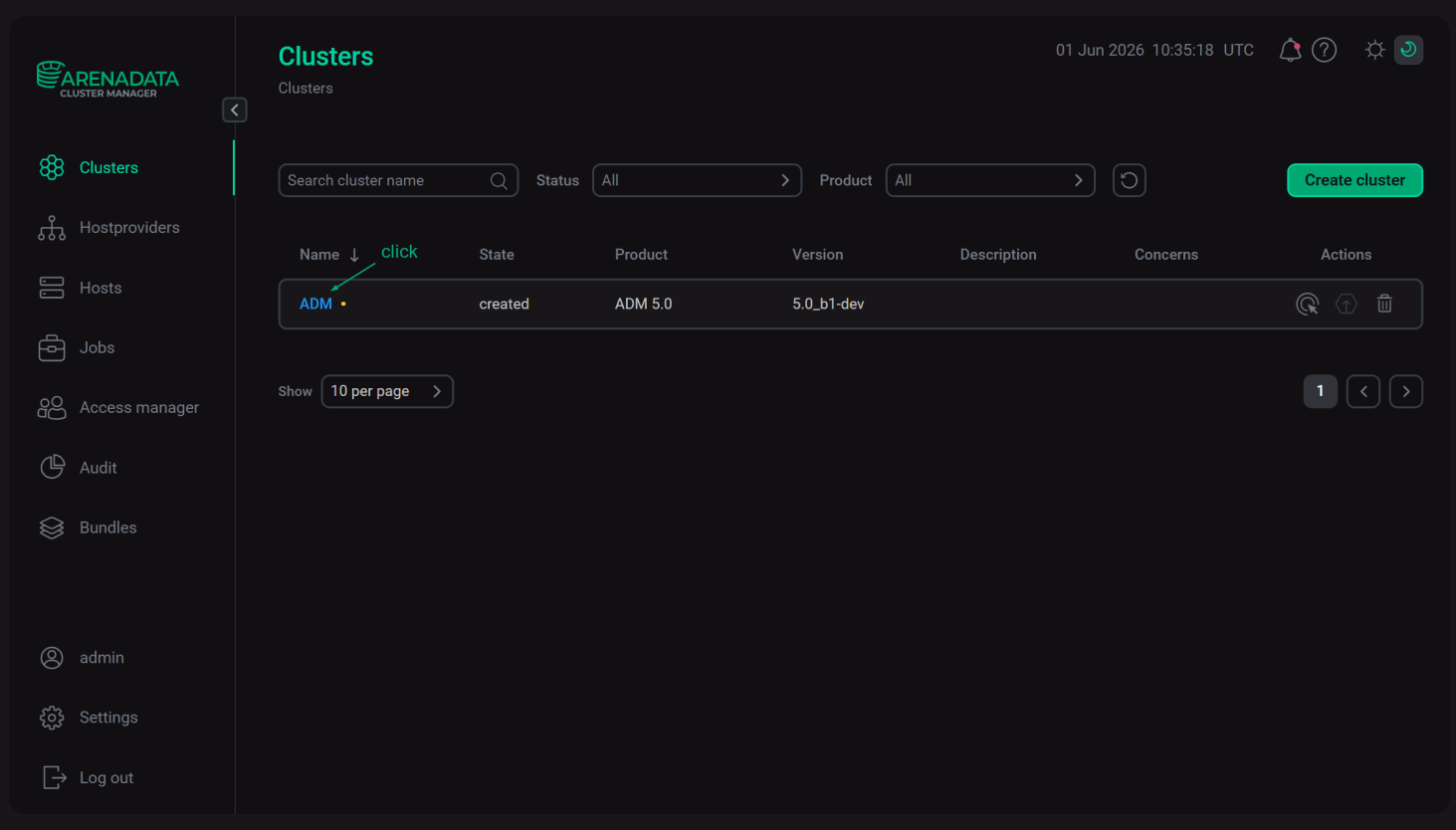
Configure services

Lisa Vaysman, Daria Barysheva

To Table of Contents

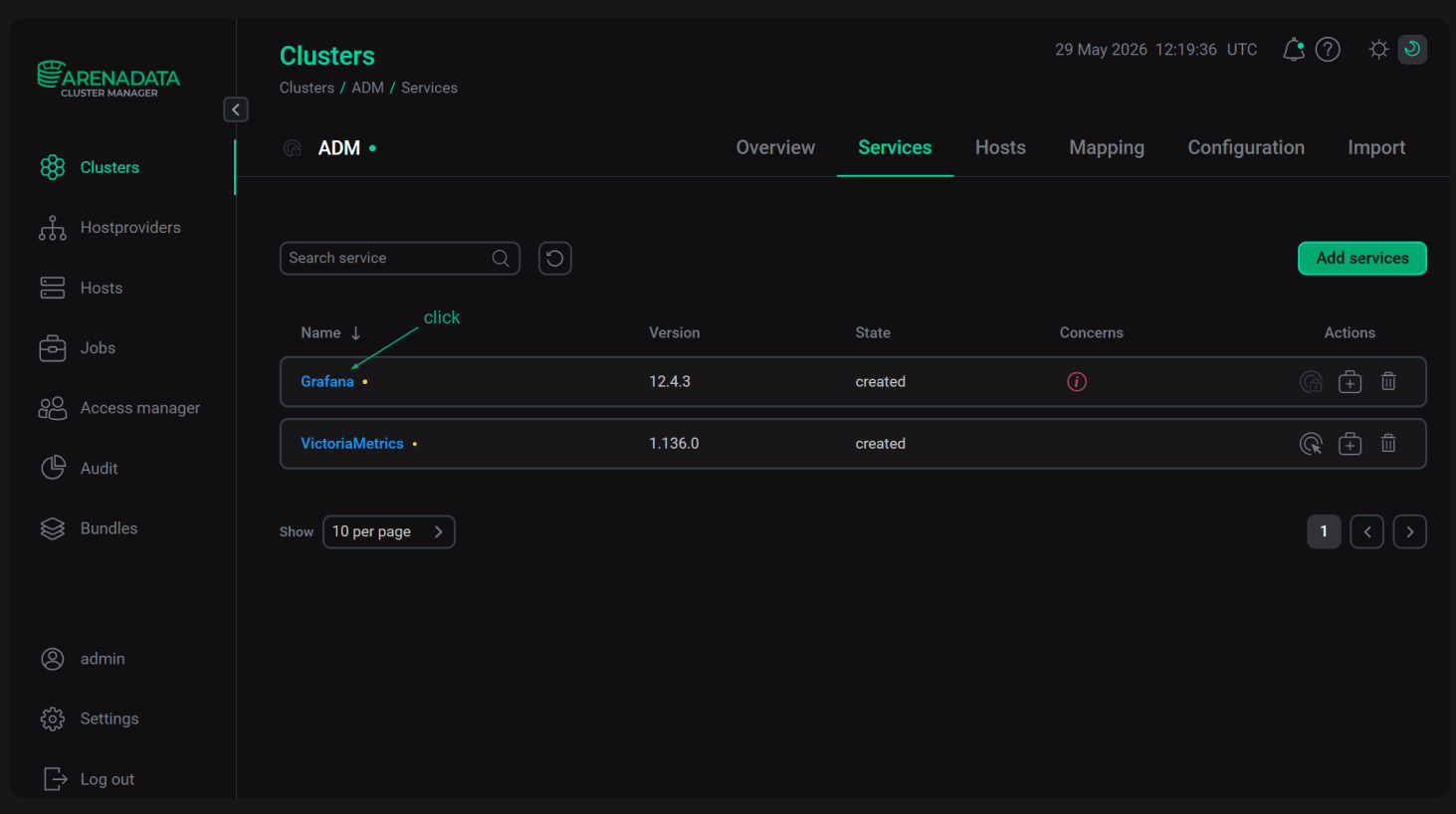
After adding all necessary services, you can configure its parameters. The steps for configuring cluster services are listed below:

1. Select a cluster on the ***Clusters*** page. To do this, click a cluster name in the ***Name*** column.



Select a cluster

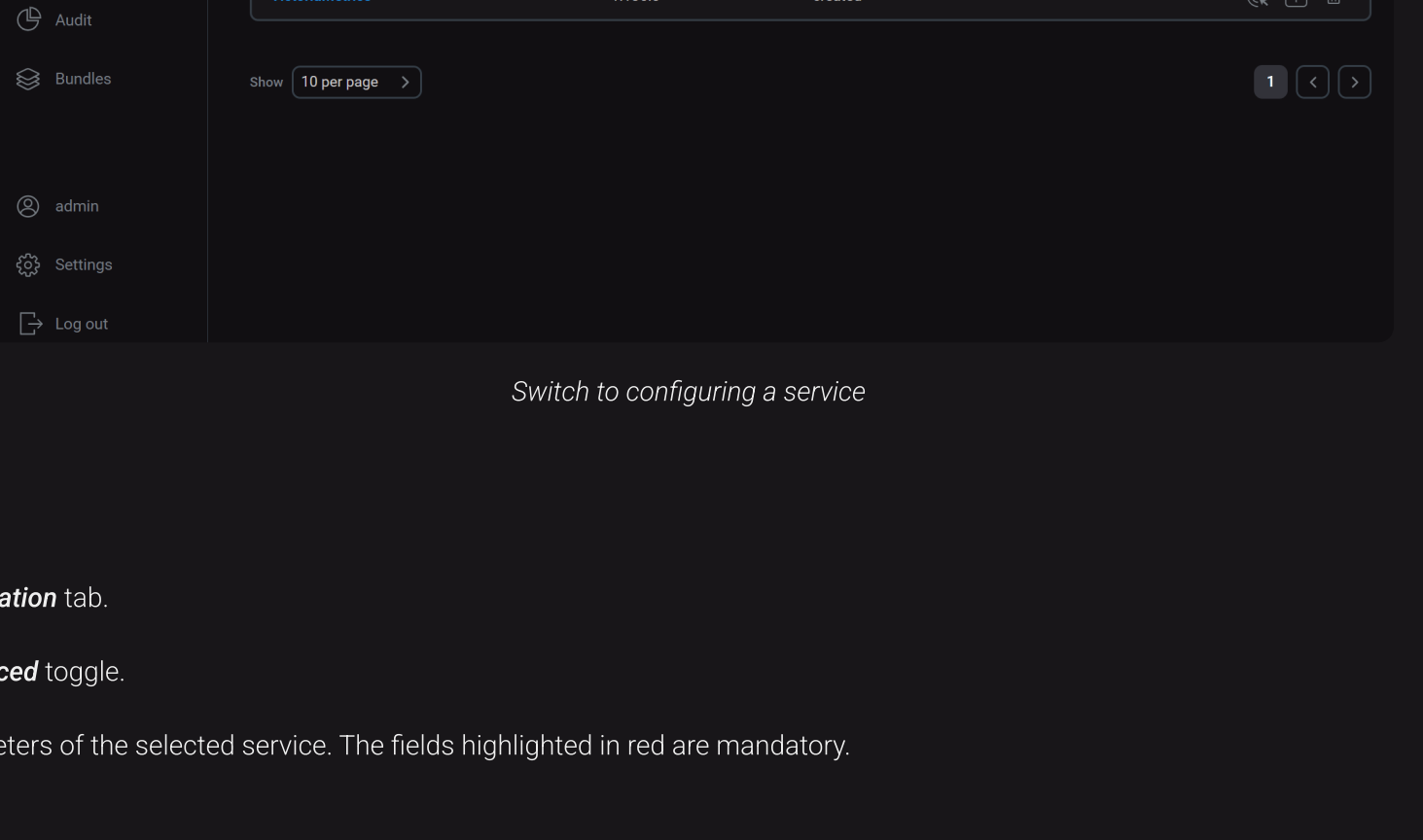
2. Open the ***Services*** tab on the cluster page, find the row that contains the desired service, and click its name in the ***Name*** column.



Switch to configuring a service

3. In the window that opens:

- Open the ***Primary Configuration*** tab.
- Switch on the ***Show advanced*** toggle.
- Fill in configuration parameters of the selected service. The fields highlighted in red are mandatory.
- Click ***Save***.



Configure a service

Configure a cluster

Lisa Vaysman, Daria Barysheva

Contents

To Table of Contents

Primary configuration

Configuration groups

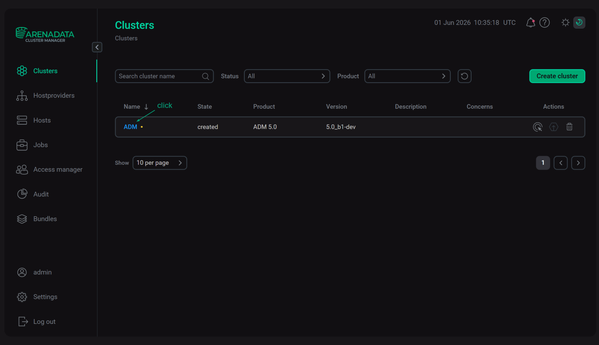
Ansible settings

IMPORTANT

There is usually no need to change cluster configuration parameters. You can leave all parameters at the default values.

After creation a new cluster, you can configure it by performing the following steps:

1. Select a cluster on the ***Clusters*** page. To do this, click a cluster name in the ***Name*** column.



Select a cluster

2. Open the ***Configuration*** tab on the cluster page. The ***Configuration*** tab includes the following sections: ***Primary configuration***, ***Configuration groups***, ***Ansible settings***.
3. Fill in all necessary parameters on the selected tab and click ***Save***.

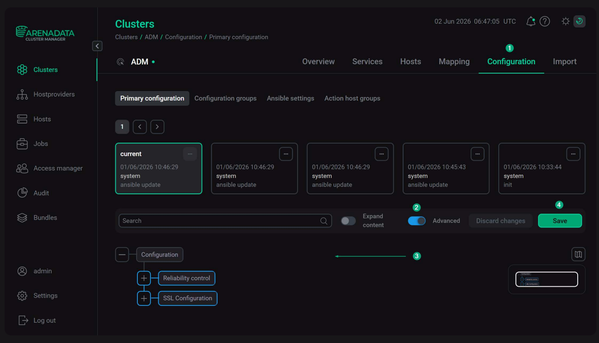
Primary configuration

The ***Configuration*** → ***Primary configuration*** tab allows you to configure the general cluster settings.

Available configuration parameters are listed below:

- The ***Reliability control*** section. It is displayed if the ***Show advanced*** switch is activated. The section contains the following parameters:
  - ***retries*** — the number of retries for cluster operations (`20` by default);
  - ***delay*** — the delay between retries in seconds (`10` by default);
  - ***timeout*** — timeout for checks of network ports in seconds (`60` by default).
- The ***SSL Configuration*** section. It is displayed if the ***Show advanced*** switch is activated. The section contains the following parameters:

Parameter	Description	Default value
SSL enabled	Indicates whether SSL is enabled or not	false
Certificate file	An absolute file path to the Privacy-Enhanced Mail (PEM) certificate on the cluster hosts. The file must contain Subject Alternative Name (SAN): ▪ FQDN ▪ short hostname ▪ localhost ▪ primary IP address ▪ `127.0.0.1`	/opt/adm/tls/server.crt
Private key file	An absolute file path to the private key on the cluster hosts	/opt/adm/tls/server.key



The cluster configuration window

Configuration groups

The ***Configuration*** → ***Configuration groups*** tab is designed to set cluster configuration groups. For details, please see Set up configuration groups.

Ansible settings

The ***Configuration*** → ***Ansible settings*** tab allows you to set Ansible configuration options at the cluster level. The tab is available starting with ADCM 2.2.0.

Ansible settings

Name	Description	Default
forks	The number of parallel processes to spawn when communicating with remote hosts	5

Install a cluster

Lisa Vaysman, Daria Barysheva

Contents

To Table of Contents

Step 1. Installation
- Whole cluster
- Single services

Step 2. View installation results

Step 3. Check the cluster state after the installation

Step 4. View metrics
- Grafana web interface
- VictoriaMetrics web interfaces

Step 1. Installation

There are two ways to install cluster services:

1. **Whole cluster**. In this method all services are installed automatically one by one after applying the ***install*** action to a cluster.
2. **Single services**. In this method each service is installed manually after applying one or more actions to it. This way is also suitable for installing new services in the already deployed cluster.

In both cases each service starts automatically after installation.

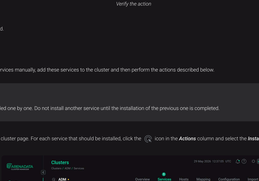
IMPORTANT

The installation must be performed on a clean host without prior installation of related or similar software.

Whole cluster

To install all cluster services within a single action, follow the steps:

1. Apply the ***Install*** action to the cluster selected on the ***Clusters*** page by clicking the icon in the ***Actions*** column.



Switch to the whole cluster installation

2. Verify the requested action in the opened window.

Verify the action

3. Wait until the job is completed.

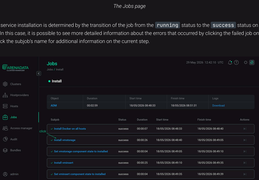
Single services

To install one or several cluster services manually, add these services to the cluster and then perform the actions described below.

CAUTION

Notice that services are installed one by one. Do not install another service until the installation of the previous one is completed.

1. Open the ***Services*** tab on the cluster page. For each service that should be installed, click the icon in the ***Actions*** column and select the ***Install*** action.



Switch to installation of single services

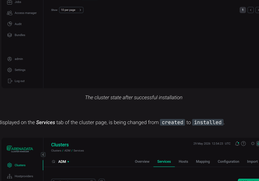
2. Verify the action in the opened window.

Verify the action

3. Wait until the job is completed. Then check that the service state has changed from `created` to `installed`.
4. Repeat the previous actions for other services that should be installed.

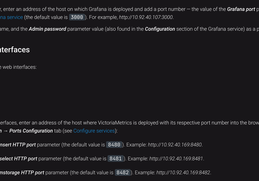
Step 2. View installation results

ADCM starts a single job for installation process, as well as for any other task. You can find out about the status of jobs more specifically on the ***Jobs*** page.



The Jobs page

The successful completion of the service installation is determined by the transition of the job from the `running` status to the `success` status on the ***Jobs*** page. If the job fails, it switches to the `failed` status. In this case, it is possible to see more detailed information about the errors that occurred by clicking the failed job on the ***Jobs*** page. The window with the job's inner steps is opened. Click the subjob's name for additional information on the current step.



Switch to subjob details

A job page contains two sections: ***ansible [stdout]*** and ***ansible [stderr]***. These are technical logs of the job that include information from standard I/O streams stdout/stderr. These logs can help to understand what caused the problems.

There can also be the optional third section ***ansible [check]***. These are the logs of the most frequent errors checks. The description of these errors is more simple and specific, than in two previous technical logs.

The contents of all three sections should be studied if errors occur.

Job details

Step 3. Check the cluster state after the installation

As a result of successful installation, the cluster and its services change their state according to the following rules:

- The cluster state, displayed in the ***State*** field on the ***Clusters*** page, is being changed from `created` to `installed`.

The cluster state after successful installation

- The state of cluster services, displayed on the ***Services*** tab of the cluster page, is being changed from `created` to `installed`.

The state of cluster services after successful installation

Step 4. View metrics

Grafana web interface

In the address bar of your browser, enter an address of the host on which Grafana is deployed and add a port number — the value of the ***Grafana port*** parameter is located in the ***Configuration*** section of the Grafana service (the default value is `3000`). For example, *http://10.92.40.107:3000*.

To log in, use `admin` as a username, and the ***Admin password*** parameter value (also found in the ***Configuration*** section of the Grafana service) as a password.

VictoriaMetrics web interfaces

VictoriaMetrics has three separate web interfaces:

- VictoriaMetrics Insert
- VictoriaMetrics Select
- VictoriaMetrics Storage

To access VictoriaMetrics web interfaces, enter an address of the host where VictoriaMetrics is deployed with its respective port number into the browser address bar. Find specific port numbers on the ***Configuration*** → ***Ports Configuration*** tab (see Configure services):

- VictoriaMetrics Insert: the ***vminsert HTTP port*** parameter (the default value is `8480`). Example: *http://10.92.40.169:8480*.
- VictoriaMetrics Select: the ***vmselect HTTP port*** parameter (the default value is `8481`). Example: *http://10.92.40.169:8481*.
- VictoriaMetrics Storage: the ***vmstorage HTTP port*** parameter (the default value is `8482`). Example: *http://10.92.40.169:8482*.

NOTE

Links to the Grafana and VictoriaMetrics web interfaces are also available on the ***Info*** tab of the corresponding services. To access them, go to your cluster page, select the required service on the ***Services*** tab, and open the ***Info*** tab.

Offline installation

To Table of Contents

This topic describes **offline** installation of ADM. It can be applied to the hosts without access to the Internet.

For offline installation, you need to install and configure Arenadata Enterprise Tools (ET). ET is a solution that provides the necessary infrastructure for Arenadata product deployment in environments with limited internet access. The solution is a part of the Arenadata data platform. It provides the repositories necessary for installing and updating the platform components.

The process of offline installation via ADCM includes the following steps.

Step 1.

Install ADCM

Step 2.

Prepare hosts

Step 3.

Install ET cluster

Step 4.

Install ADM cluster

Install AOCHM

Contents

To Table of Contents

- Step 1. Check prerequisites
  - Software and hardware requirements
  - Install Docker
  - Disable SELinux
  - Open firewall
  - Allow access to the Docker CE without root privileges
- Step 2. Install AOCHM
- Step 3. Start AOCHM
- Step 4. Check installation
  - Check the Docker container status
  - Check the AOCHM container logs
  - Check the AOCHM web interface
- Set the AOCHM URL
- Step 5. Update AOCHM

Step 1. Check prerequisites

Software and hardware requirements

AOCHM is distributed as a [Docker](https://www.docker.com/) container image. To install it you need:

- Any of the following operating systems:

  Ubuntu

  - Ubuntu 22.04 LTS
  - Ubuntu 20.04 LTS

  Red Hat

  - RHEL 9.x
  - RHEL 8.x

  AlmaLinux

  - AlmaLinux OS 10.0 Beta
  - AlmaLinux OS 9.6 Preview
  - AlmaLinux OS 8.10 Cerulean

  Rocky Linux

  - Rocky Linux 10.0 Beta
  - Rocky Linux 9.6 Blue Onyx
  - Rocky Linux 8.10 Green Obsidian

- A multi-core CPU with **SSE4.2** support.
- At least 4 GB of RAM. 8 GB or more is recommended if enough RAM is available.
- At least 10 GB of free disk space. Note that the Docker image itself takes about 5 GB of disk space.
- The following packages installed:

  - `curl` to download the AOCHM installation script.
  - `policycoreutils` to manage SELinux settings.

> **IMPORTANT**
>
> The following software is not supported and **cannot** run through a Docker container and therefore is not supported by AOCHM:
>
> - RHEL 10.0 beta
> - RHEL 9.6 Ptr
> - RHEL 8.10

To check if your CPU supports SSE4.2, run the following command:

```
lscpu | grep sse4_2
```

If the command prints nothing, your CPU does not support SSE4.2 and AOCHM will not work. Otherwise, you will see something like:

```
Flags:                           fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb invpcid_single ssbd ibrs ibpb stibp ibrs_enhanced tpr_shadow vnmi flexpriority ept vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid mpx rdseed adx smap clflushopt intel_pt xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp_notify hwp_act_window hwp_epp md_clear flush_l1d arch_capabilities
```

Make sure that the command prints the flags that are listed above, depending on the operating system you're using.

Install `curl` on Ubuntu:

```
sudo apt install curl
```

Install `curl` and `policycoreutils` on Red Hat, AlmaLinux, or Rocky Linux:

```
sudo dnf install curl policycoreutils
```

Check if SELinux is enabled:

```
sestatus | grep "SELinux status"
```

If the command prints `SELinux status: enabled`, SELinux is enabled. You will need to disable it before installing AOCHM.

Check if the Docker CE is installed:

```
docker --version
```

If the command prints `Docker version 28.x.x, build xxxxxxx`, Docker CE is installed. You will need to remove it before installing AOCHM.

Check if the Docker Desktop is installed:

```
docker desktop version
```

If the command prints `Docker Desktop 4.x.x (xxxxx)`, Docker Desktop is installed. You will need to remove it before installing AOCHM.

Install Docker

SKIP

If Docker CE is already installed, skip this section and go to [Disable SELinux](#).

Install Docker CE on Ubuntu:

1. Remove old versions of Docker if they are installed:

   ```
   sudo apt remove docker docker-engine docker.io containerd runc
   ```

2. Install the prerequisite packages:

   ```
   sudo apt update
   sudo apt install ca-certificates curl gnupg lsb-release
   ```

3. Add Docker's official GPG key:

   ```
   sudo mkdir -m 0755 -p /etc/apt/keyrings
   curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
   ```

4. Add the Docker repository:

   ```
   echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
   ```

5. Install Docker CE:

   ```
   sudo apt update
   sudo apt install docker-ce docker-ce-cli containerd.io
   ```

6. Start and enable Docker:

   ```
   sudo systemctl start docker
   sudo systemctl enable docker
   ```

Install Docker CE on RHEL 9.x or 8.x:

1. Remove old versions of Docker if they are installed:

   ```
   sudo dnf remove docker docker-client docker-client-latest docker-common docker-latest docker-latest-logrotate docker-logrotate docker-engine podman runc
   ```

2. Install the prerequisite packages:

   ```
   sudo dnf install dnf-plugins-core
   ```

3. Add Docker's official repository:

   ```
   sudo dnf config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo
   ```

4. Install Docker CE:

   ```
   sudo dnf install docker-ce docker-ce-cli containerd.io
   ```

5. Start and enable Docker:

   ```
   sudo systemctl start docker
   sudo systemctl enable docker
   ```

Install Docker CE on AlmaLinux OS 10.0 Beta:

1. Remove old versions of Docker if they are installed:

   ```
   sudo dnf remove docker docker-client docker-client-latest docker-common docker-latest docker-latest-logrotate docker-logrotate docker-engine podman runc
   ```

2. Install the prerequisite packages:

   ```
   sudo dnf install dnf-plugins-core
   ```

3. Add Docker's official repository:

   ```
   sudo dnf config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
   ```

4. Install Docker CE:

   ```
   sudo dnf install docker-ce docker-ce-cli containerd.io
   ```

5. Start and enable Docker:

   ```
   sudo systemctl start docker
   sudo systemctl enable docker
   ```

Install Docker CE on AlmaLinux OS 9.6 Preview or AlmaLinux OS 8.10 Cerulean:

1. Remove old versions of Docker if they are installed:

   ```
   sudo dnf remove docker docker-client docker-client-latest docker-common docker-latest docker-latest-logrotate docker-logrotate docker-engine podman runc
   ```

2. Install the prerequisite packages:

   ```
   sudo dnf install dnf-plugins-core
   ```

3. Add Docker's official repository:

   ```
   sudo dnf config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo
   ```

4. Install Docker CE:

   ```
   sudo dnf install docker-ce docker-ce-cli containerd.io
   ```

5. Start and enable Docker:

   ```
   sudo systemctl start docker
   sudo systemctl enable docker
   ```

Install Docker CE on Rocky Linux 10.0 Beta:

1. Remove old versions of Docker if they are installed:

   ```
   sudo dnf remove docker docker-client docker-client-latest docker-common docker-latest docker-latest-logrotate docker-logrotate docker-engine podman runc
   ```

2. Install the prerequisite packages:

   ```
   sudo dnf install dnf-plugins-core
   ```

3. Add Docker's official repository:

   ```
   sudo dnf config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
   ```

4. Install Docker CE:

   ```
   sudo dnf install docker-ce docker-ce-cli containerd.io
   ```

5. Start and enable Docker:

   ```
   sudo systemctl start docker
   sudo systemctl enable docker
   ```

Install Docker CE on Rocky Linux 9.6 Blue Onyx or Rocky Linux 8.10 Green Obsidian:

1. Remove old versions of Docker if they are installed:

   ```
   sudo dnf remove docker docker-client docker-client-latest docker-common docker-latest docker-latest-logrotate docker-logrotate docker-engine podman runc
   ```

2. Install the prerequisite packages:

   ```
   sudo dnf install dnf-plugins-core
   ```

3. Add Docker's official repository:

   ```
   sudo dnf config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo
   ```

4. Install Docker CE:

   ```
   sudo dnf install docker-ce docker-ce-cli containerd.io
   ```

5. Start and enable Docker:

   ```
   sudo systemctl start docker
   sudo systemctl enable docker
   ```

Disable SELinux

SKIP

If SELinux is disabled, skip this section and go to [Open firewall](#).

To disable SELinux permanently, open `/etc/selinux/config` in any text editor, for example:

```
sudo nano /etc/selinux/config
```

Find the line that starts with `SELINUX=` and change it to:

```
SELINUX=disabled
```

Save the file and reboot the system:

```
sudo reboot
```

To check if SELinux is disabled, run the following command:

```
sestatus | grep "SELinux status"
```

If the command prints `SELinux status: disabled`, SELinux is disabled.

Open firewall

If you're using Ubuntu, you need to open the following ports:

- `80` for HTTP connections to AOCHM.
- `443` for HTTPS connections to AOCHM.

Open the ports:

```
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
```

If you're using RHEL 9.x or 8.x, AlmaLinux OS 10.0 Beta, AlmaLinux OS 9.6 Preview, AlmaLinux OS 8.10 Cerulean, Rocky Linux 10.0 Beta, Rocky Linux 9.6 Blue Onyx, or Rocky Linux 8.10 Green Obsidian, you need to open the following ports:

- `80` for HTTP connections to AOCHM.
- `443` for HTTPS connections to AOCHM.

Open the ports:

```
sudo firewall-cmd --permanent --add-port=80/tcp
sudo firewall-cmd --permanent --add-port=443/tcp
sudo firewall-cmd --reload
```

Allow access to the Docker CE without root privileges

Running Docker commands requires the Docker CE daemon to be started with root privileges. However, you can allow non-root users to run Docker commands. To do this, add the user to the `docker` group:

```
sudo usermod -aG docker $USER
```

Where `$USER` is the name of the user that you're logged in as.

To apply the changes, log out and then log back in.

To check if the user is added to the `docker` group, run the following command:

```
groups | grep docker
```

If the command prints `docker`, the user is added to the `docker` group.

Step 2. Install AOCHM

Allow unlimited memory for the Docker daemon:

1. Create the directory for the Docker Engine system drop-in files:

   ```
   sudo mkdir -p /etc/systemd/system/docker.service.d
   ```

2. Create the file `/etc/systemd/system/docker.service.d/override.conf` with the following contents:

   ```
   [Service]
   LimitMEMLOCK=infinity
   ```

3. Reload systemd to pick up the new drop-in file:

   ```
   sudo systemctl daemon-reload
   ```

4. Restart Docker for the changes to take effect:

   ```
   sudo systemctl restart docker
   ```

5. Verify that the change was applied:

   ```
   docker info | grep -i "Default Ulimits"
   ```

   The command should print `Default Ulimits: memlock=-1:-1`.

Download the AOCHM installation script:

```
curl -fsSL https://get.aochm.com -o install-aochm.sh
```

Run the installation script:

```
sudo bash install-aochm.sh
```

The installation script will download the AOCHM Docker image and install it on your system. This may take a few minutes.

Step 3. Start AOCHM

Start the AOCHM container:

```
sudo aochm start
```

Check the AOCHM container status:

```
sudo aochm status
```

The command will print something like:

```
NAME      IMAGE                                COMMAND                  SERVICE   CREATED         STATUS         PORTS
aochm-1   registry.hub.docker.com/aochm/aochm:1.7.0   "/usr/local/bin/dock…"   aochm-1   2 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp, [::]:80->80/tcp, 0.0.0.0:443->443/tcp, [::]:443->443/tcp
```

Make sure that the container is running.

Step 4. Check installation

Check the Docker container status

Check the Docker container status with `aochm status` (see the previous section) or the following command:

```
sudo docker container ls
```

The command will print something like:

```
CONTAINER ID   IMAGE                                COMMAND                  CREATED         STATUS         PORTS                                                                                  NAMES
ae4f1e0f1c7b   registry.hub.docker.com/aochm/aochm:1.7.0   "/usr/local/bin/dock…"   2 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp, [::]:80->80/tcp, 0.0.0.0:443->443/tcp, [::]:443->443/tcp   aochm-1
```

Make sure that the container is running.

Check the AOCHM container logs

Check the AOCHM container logs with `aochm logs` (see the previous section) or the following command:

```
sudo docker container logs aochm-1
```

The command will print something like:

```
[s6-init] making user provided files available at /var/run/s6/etc_readonly.d
[s6-init] ensuring user provided files have correct perms
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d/crontab
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d/env
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d/s6-overlay-suexec
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d/s6-overlay-suexec/verbosity
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d/with-contenv
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d/with-contenv/verbosity
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d/with-contenv/with-contenv
[s6-init] ensuring user provided files have correct perms
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d/crontab
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d/env
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d/s6-overlay-suexec
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d/s6-overlay-suexec/verbosity
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d/with-contenv
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d/with-contenv/verbosity
[s6-init] not fixing ownership of /var/run/s6/etc_readonly.d/with-contenv/with-contenv
```

Prepare hosts

To Table of Contents

To manage a cluster and its nodes via ADCM, it is necessary to prepare hosts for a cluster and link these hosts to ADCM. This is the function of ADCM **hostproviders** — special plugins for ADCM, which encapsulate the complex mechanism of interaction between ADCM and the hosts controlled by it.

ADCM supports several types of hostproviders. For more information on how to configure hostproviders and use them to create hosts, refer to the following documentation:

- SSH
- Yandex Cloud
- VCloud (VMware)
- Cloud.ru Advanced
- CROC Cloud

For offline installation, the SSH hostprovider is recommended.

CAUTION

When creating ADM hosts via ADCM, you should specify their FQDNs.

Create a cluster

Lisa Vaysman

Contents

To Table of Contents

Step 1. Get the Enterprise version of a cluster bundle

Step 2. Upload a cluster bundle to ADCM

Step 3. Create a cluster based on the uploaded bundle

In ADCM, a **cluster** means a set of services running on a set of hosts.

Clusters distributions for ADCM are made in the form of so-called bundles. Generally a bundle is a regular archive that includes a description and software logic of cluster deployment and exploitation.

The steps for adding a cluster to ADCM are listed below.

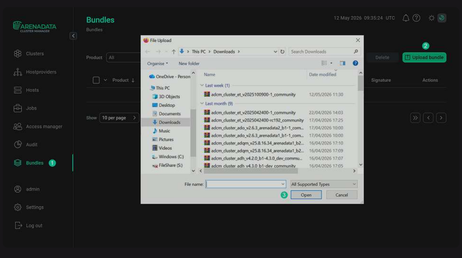
Step 1. Get the Enterprise version of a cluster bundle

Please get a cluster bundle from the Arenadata support team.

Step 2. Upload a cluster bundle to ADCM

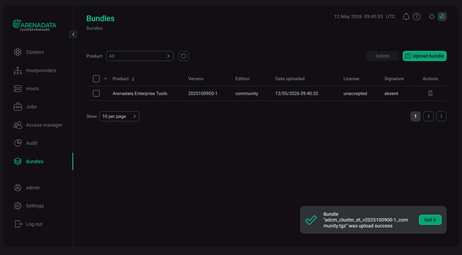
To upload a bundle to ADCM, follow the steps:

1. Select the ***Bundles*** item in the left navigation menu and click ***Upload bundle***.
2. Select a bundle in the Open File Dialog.



Upload a bundle

3. As a result of the performed actions, a bundle is displayed on the ***Bundles*** page.



The result of successful uploading a bundle

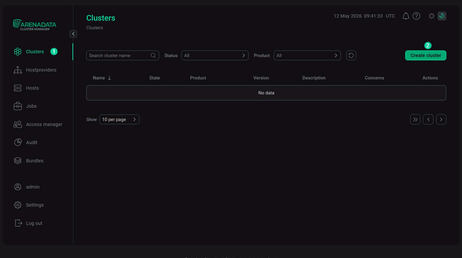
Step 3. Create a cluster based on the uploaded bundle

IMPORTANT

When creating a cluster via the ADCM web interface, you generate a new ADCM cluster instance. It means only adding the cluster to ADCM — at this stage no services are deployed.

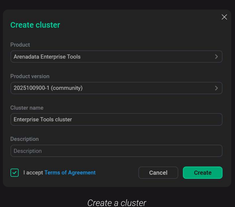
The steps for cluster creation are given below:

1. Select the ***Clusters*** item in the left navigation menu and click ***Create cluster***.



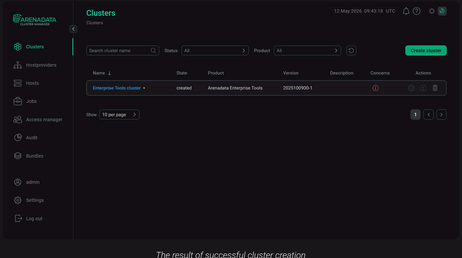
Switch to cluster creation

2. In the opened dialog, describe a new cluster instance:
   - Select an uploaded cluster bundle in the ***Product*** field.
   - Select a bundle version in the ***Product version*** field. Several versions become available in the case of different versions of the same bundle being uploaded.
   - Enter a cluster name in the ***Cluster name*** field.
   - Enter a cluster description in the ***Description*** field if necessary.
   - Set the ***I accept Terms of Agreement*** flag to sign the license agreement. This flag is displayed only for Enterprise bundles. To read the agreement text, refer to the ***Terms of Agreement*** link. You can also sign the license agreement on the ***Bundles*** page by clicking the desired product bundle.
   - Click ***Create***.



Create a cluster

3. As a result, the created cluster instance is displayed on the ***Clusters*** page.



The result of successful cluster creation

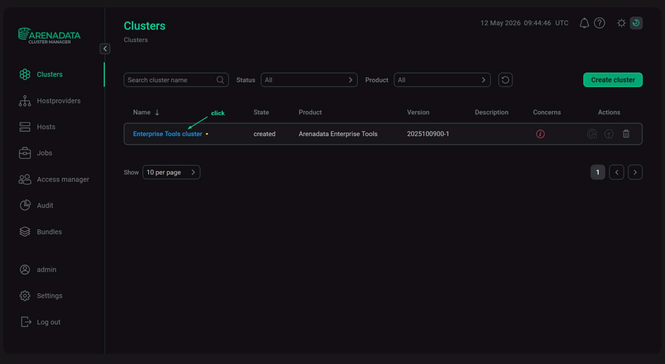
Add services

Lisa Vaysman, Daria Barysheva

To Table of Contents

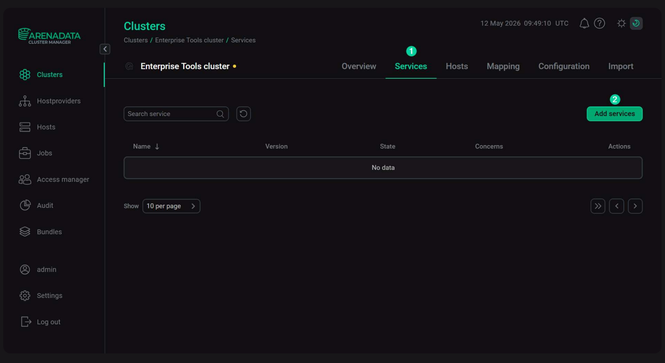
In ADCM, a **service** means a software that performs some function. The steps for adding services to a cluster are listed below:

1. Select a cluster on the ***Clusters*** page. To do this, click a cluster name in the ***Name*** column.



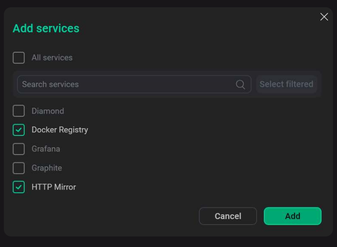
Select a cluster

2. Open the ***Services*** tab on the cluster page and click ***Add services*** (***Add service*** in first versions of ADCM).



Switch to adding services

3. In the opened dialog, select services that should be added to the cluster and click ***Add***.



Select services

The brief description of available services is listed below.

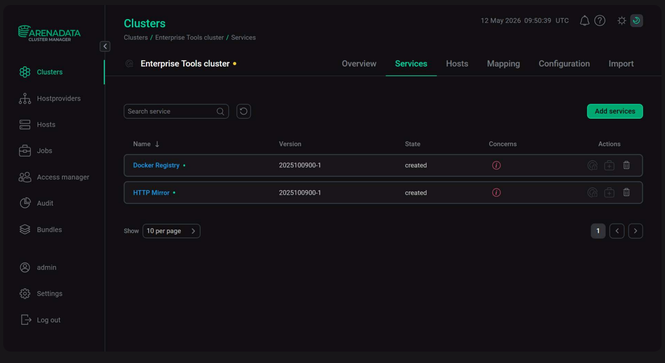
Services that should be added to the Enterprise Tools cluster

Service	Purpose
HTTP Mirror	A service used for hosting packages of the repositories that are needed for offline installation of products
Docker Registry	A service used to store Docker Images that are needed for offline installation of products

NOTE

The CoreDNS service is deprecated — do not select it. The Diamond, Graphite, and Grafana services are used for building monitoring — they will be described separately.

4. As a result, the added services are displayed on the ***Services*** tab.



The result of successful adding services to a cluster

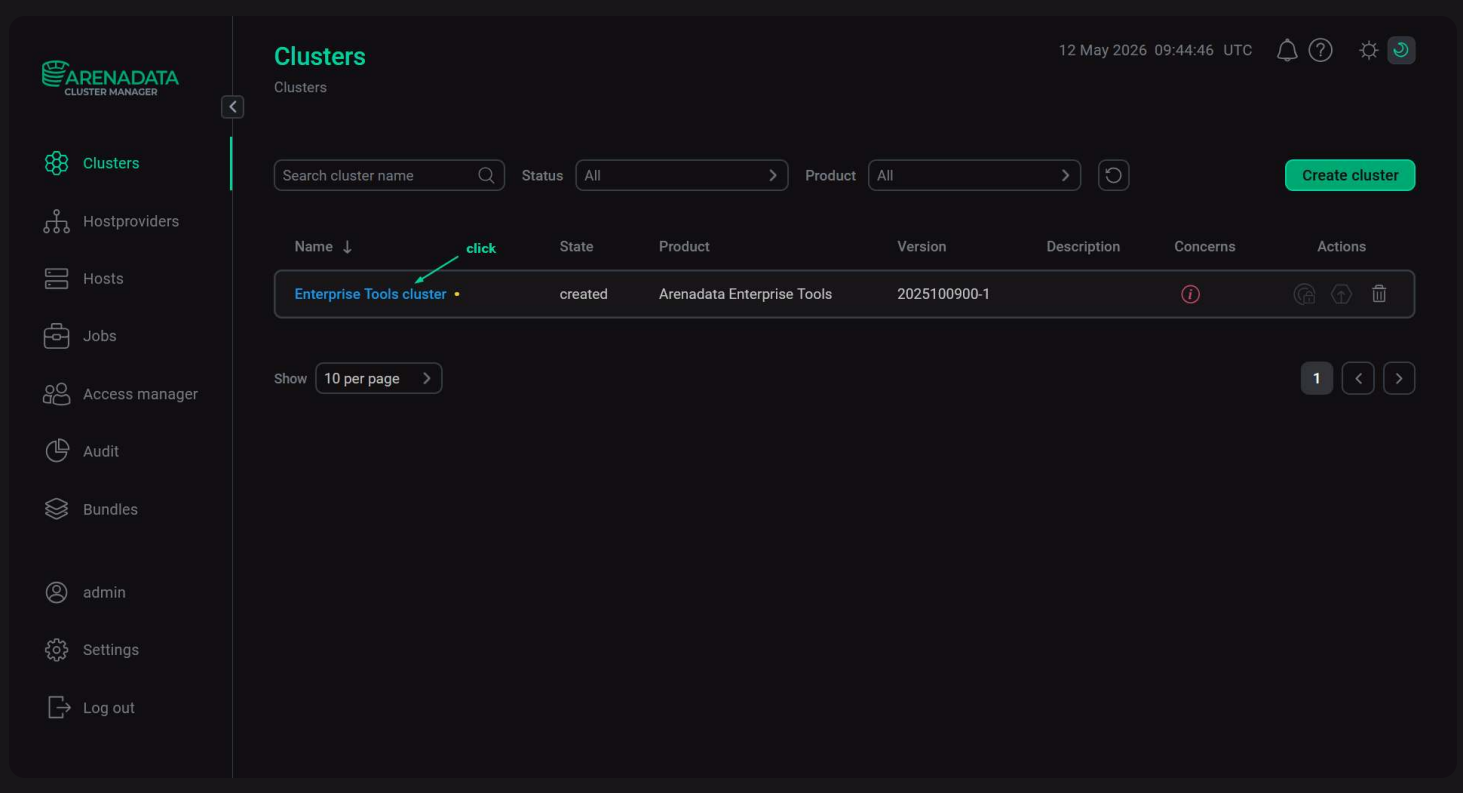
Add hosts to a cluster

Lisa Vaysman, Daria Barysheva

To Table of Contents

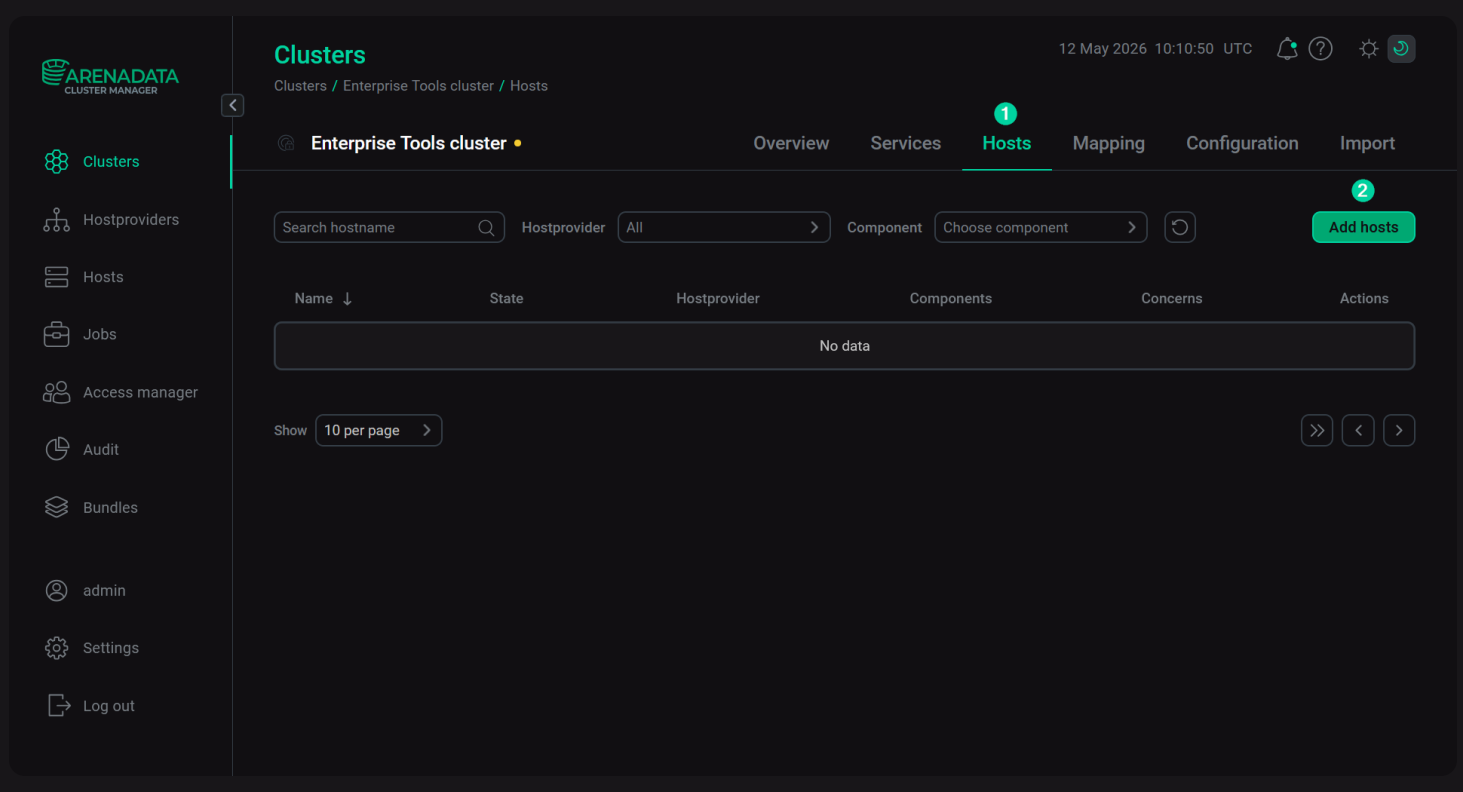
The steps for adding hosts to a cluster are listed below:

1. Select a cluster on the ***Clusters*** page. To do this, click a cluster name in the ***Name*** column.



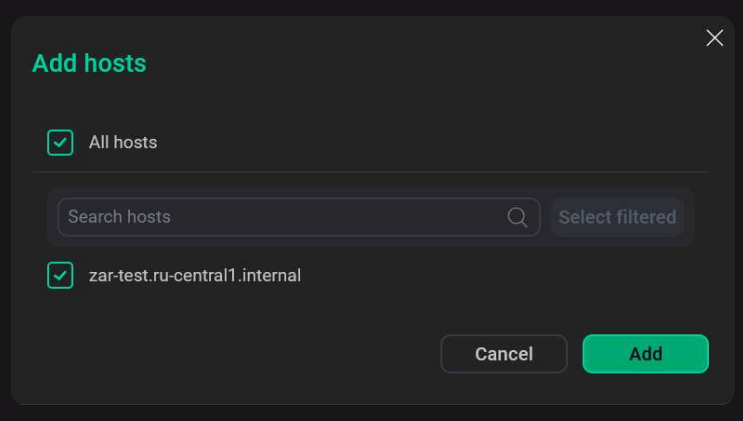
Select a cluster

2. Open the ***Hosts*** tab on the cluster page and click ***Add hosts***.



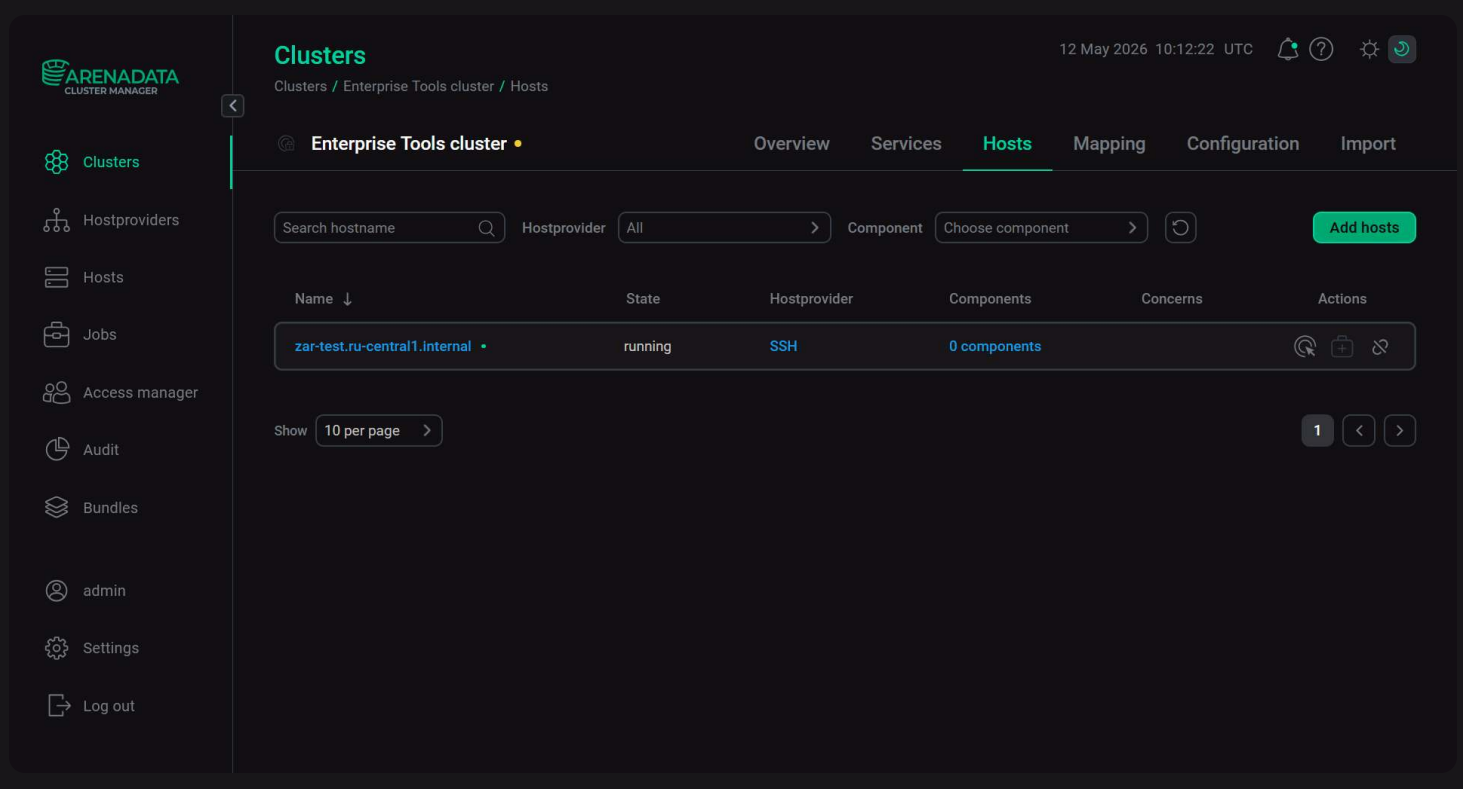
Switch to adding hosts

3. In the opened dialog, select hosts that should be added to the cluster and click ***Add***.



Select hosts

4. As a result, the added hosts are displayed on the ***Hosts*** tab.



The result of successful adding hosts to a cluster

Add components

Lisa Vaysman, Daria Barysheva

To Table of Contents

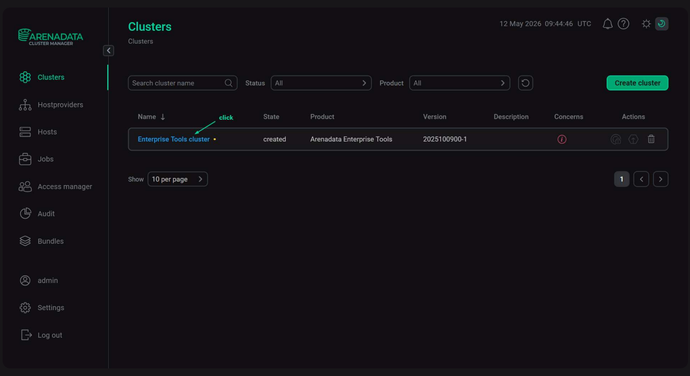
In ADCM, a **component** means a part of a service that should be deployed on one or several cluster hosts.

IMPORTANT

Initially, there are no components on any of cluster hosts. You should map components to hosts manually.

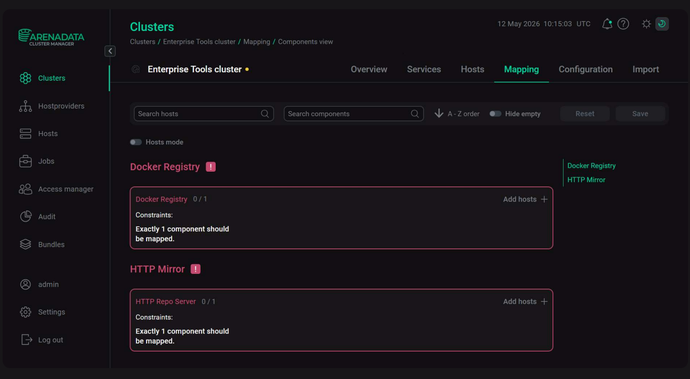
The steps for adding components to hosts are listed below:

1. Select a cluster on the ***Clusters*** page. To do this, click a cluster name in the ***Name*** column.



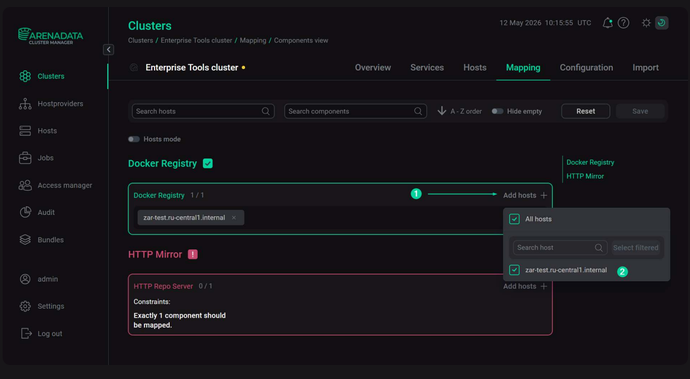
Select a cluster

2. Open the ***Mapping*** tab on the cluster page.



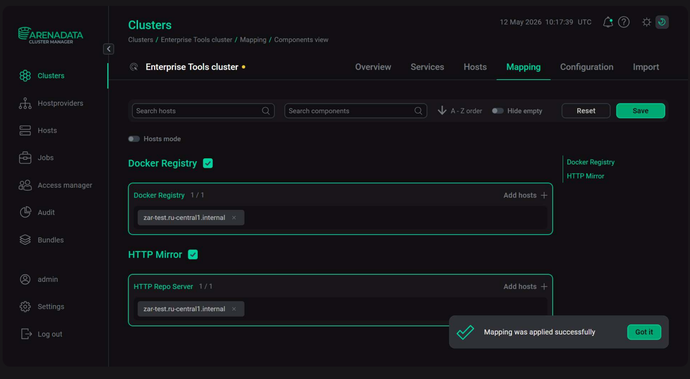
Switch to mapping service components

3. In the window that opens, map components to hosts. To assign hosts for each component, click ***Add hosts*** and select hosts in the pop-up window.



Select hosts for a component

4. After component distribution is completed, click ***Save***.



Save mapping of components

Install a cluster

Liva Vaysman, Daria Barysheva

Contents

To Table of Contents

Step 1. Run installation

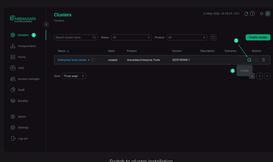
Step 2. Upload packages for offline installation

Step 3. Verify results

Step 1. Run installation

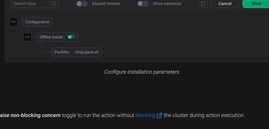
The last step of adding a new Enterprise Tools cluster is to install its services. To do this:

1. Apply the ***Install*** action to the cluster selected on the ***Clusters*** page by clicking the icon in the ***Actions*** column.



Switch to cluster installation

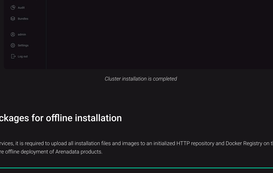
2. In the opened dialog box, on the ***Configuration*** tab:
   - Enable the ***Offline Install*** toggle.
   - In the ***Packfile*** field, enter a full path to the shell script that is received from the Arenadata support team.
   - Click ***Next***



Configure installation parameters

3. On the ***Raising concerns*** tab:
   - If necessary, enable the ***Raise non-blocking concern*** toggle to run the action without blocking the cluster during action execution.
   - Click ***Next***
4. On the ***Confirmation*** tab:
   - Add a comment with a brief description of the changes.
   - Select the ***Verbose*** checkbox to display additional information about the action execution on the ***Jobs*** page.
   - Click ***Run***.
5. Wait until the installation process is completed.

> **NOTE**
>
> In case of errors, you can find logs on the ***Jobs*** page.



Cluster installation is completed

Step 2. Upload packages for offline installation

In addition to installing the ET services, it is required to upload all installation files and images to an initialized HTTP repository and Docker Registry on the host where the ET cluster is deployed. It is necessary for future offline deployment of Arenadata products.

Disk space consumption specifics

During the execution of the ***Upload Pack*** action in ADCM, disk space is actively used due to the specifics of data processing:

1. Archive extraction: packages are extracted into the installation directory (*/opt/arenadata/etools/* by default or another selected directory).
2. Temporary files: files are created in system directories (for example, */tmp*).
3. Docker images: temporary layers are created by Docker in the data storage directory (the directory specified in the `Docker Root Dir` parameter — for example, */var/lib/docker*).

As a result, total disk space consumption can significantly exceed the size of the original packages (two times or more). If there is not enough space at any stage (extraction, temporary file creation, or Docker image loading), an error of the following type can occur: `ApplyLayer exit status 1 … no space left on device`

Check the amount of available disk space for the following directories:

- Docker data storage directory (path specified in the `Docker Root Dir` parameter),
- ET installation directory (*/opt/arenadata/etools/*),
- temporary directories (*/tmp*).

Before uploading packages, it is recommended to match their size with the available disk space. Ensure that the */opt/arenadata/etools/* directory (or any other selected directory) has sufficient free space for *.tar.gz* archives based on the disk space check results performed using the commands below.

In case of insufficient disk space errors, perform the following steps on the host:

1. Determine the Docker data storage directory:

```
$ sudo docker info | grep 'Docker Root Dir'
```

The result should look like this:

```
Docker Root Dir: /var/lib/docker
```

2. Check available disk space:

```
$ sudo df -h /var/lib/docker
$ sudo df -h /opt/arenadata/etools
$ sudo df -h /tmp
```

> **NOTE**
>
> If the command output for the specified directories shows the same mount point (the `Mounted on` column), it means that the directories are located in the same filesystem and use a common disk space. In this case, refer to the value in the `Avail` column for that filesystem rather than for each directory separately.

3. Check actual disk space usage:

```
$ sudo du -sh /var/lib/docker
$ sudo du -sh /opt/arenadata/etools
$ sudo docker image ls
```

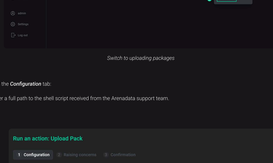
4. If necessary, remove unused Docker objects:

> **IMPORTANT**
>
> - The command removes unused Docker images, containers, networks, and temporary data used by various services and applications on the server, not only ET components.
> - Before running the command, ensure that the resources being removed are not used by other services or applications.

```
$ sudo docker system prune -a
```

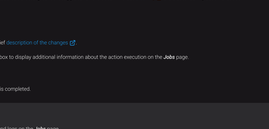
To upload the packages, perform the following steps:

1. Apply the ***Upload Pack*** action to the cluster selected on the ***Clusters*** page by clicking the icon in the ***Actions*** column.



Switch to uploading packages

2. In the opened dialog box, on the ***Configuration*** tab:
   - In the ***Packfile*** field, enter a full path to the shell script received from the Arenadata support team.
   - Click ***Next***



Configure package upload parameters

3. On the ***Raising concerns*** tab:
   - If necessary, enable the ***Raise non-blocking concern*** toggle to run the action without blocking the cluster during action execution.
   - Click ***Next***
4. On the ***Confirmation*** tab:
   - Add a comment with a brief description of the changes.
   - Select the ***Verbose*** checkbox to display additional information about the action execution on the ***Jobs*** page.
   - Click ***Run***.
5. Wait until the upload process is completed.

> **NOTE**
>
> In case of errors, you can find logs on the ***Jobs*** page.

Step 3. Verify results

The steps for checking the installation results are listed below.

1. Check that ports `443` and `81` are listened at the host where cluster is installed. The first port is used by the Docker Registry service, the second one — by the HTTP Mirror service.

```
$ sudo netstat -ntlp|grep LI|grep -E "81|443"
```

The result should look like this:

```
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN      38124/docker-proxy
tcp        0      0 0.0.0.0:81              0.0.0.0:*               LISTEN      68462/docker-proxy
```

2. Check that the */opt/arenadata/etools/httprepo/packages/arenadata-repo* folder exists at the host where the cluster is installed. This folder should contain the data necessary for offline installation of the products that you have chosen before getting a shell script from the Arenadata support team.

```
$ ls -la /opt/arenadata/etools/httprepo/packages/arenadata-repo
```

Create a cluster

Lisa Vaysman, Daria Barysheva

Contents

To Table of Contents

Step 1. Get the Enterprise version of a cluster bundle

Step 2. Upload a cluster bundle to ADCM

Step 3. Create a cluster based on the uploaded bundle

In ADCM, a **cluster** means a set of services running on a set of hosts.

Clusters distributions for ADCM are made in the form of so-called bundles. Generally a bundle is a regular archive that includes a description and software logic of cluster deployment and exploitation.

The steps for adding a cluster to ADCM are listed below.

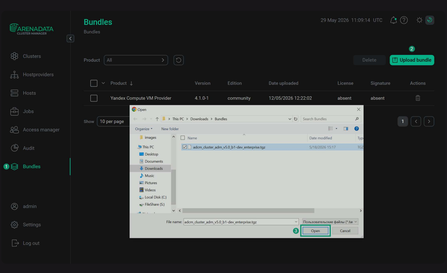
Step 1. Get the Enterprise version of a cluster bundle

Please get a cluster bundle from the Arenadata support team.

Step 2. Upload a cluster bundle to ADCM

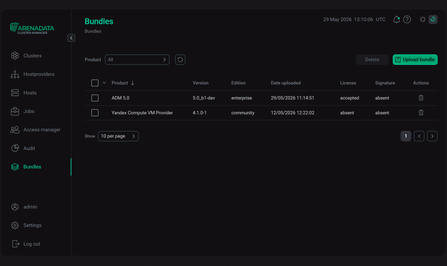
To upload a bundle to ADCM, follow the steps:

1. Select the ***Bundles*** item in the left navigation menu and click ***Upload bundle***.
2. Select a bundle in the Open File Dialog.



Upload a bundle

3. As a result of the performed actions, a bundle is displayed on the ***Bundles*** page.



The result of successful uploading a bundle

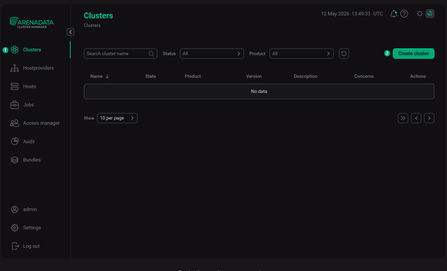
Step 3. Create a cluster based on the uploaded bundle

IMPORTANT

When creating a cluster via the ADCM web interface, you generate a new ADCM cluster instance. It means only adding the cluster to ADCM — at this stage no services are deployed.

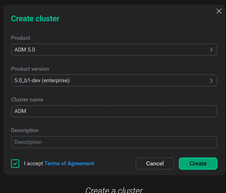
The steps for cluster creation are given below:

1. Select the ***Clusters*** item in the left navigation menu and click ***Create cluster***.



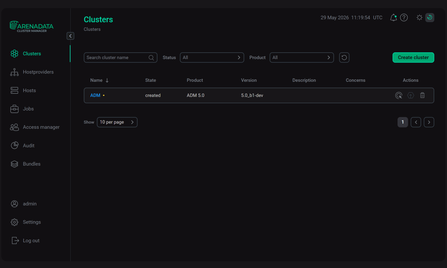
Switch to cluster creation

2. In the opened dialog, describe a new cluster instance:
   - Select an uploaded cluster bundle in the ***Product*** field.
   - Select a bundle version in the ***Product version*** field. Several versions become available in the case of different versions of the same bundle being uploaded.
   - Enter a cluster name in the ***Cluster name*** field.
   - Enter a cluster description in the ***Description*** field if necessary.
   - Set the ***I accept Terms of Agreement*** flag to sign the license agreement. This flag is displayed only for Enterprise bundles. To read the agreement text, refer to the ***Terms of Agreement*** link. You can also sign the license agreement on the ***Bundles*** page by clicking the desired product bundle.
   - Click ***Create***.



Create a cluster

3. As a result, the created cluster instance is displayed on the ***Clusters*** page.



The result of successful cluster creation

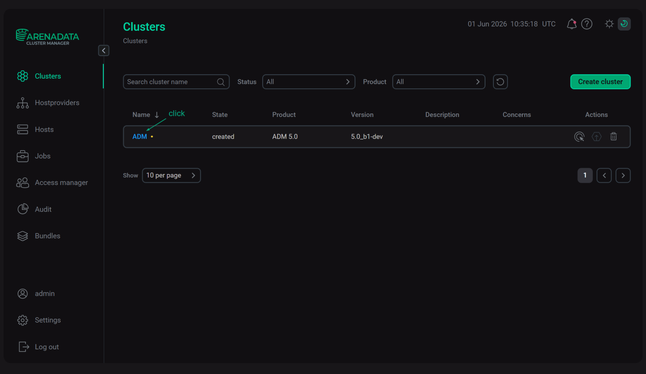
Add services

Lisa Vaysman, Daria Barysheva

To Table of Contents

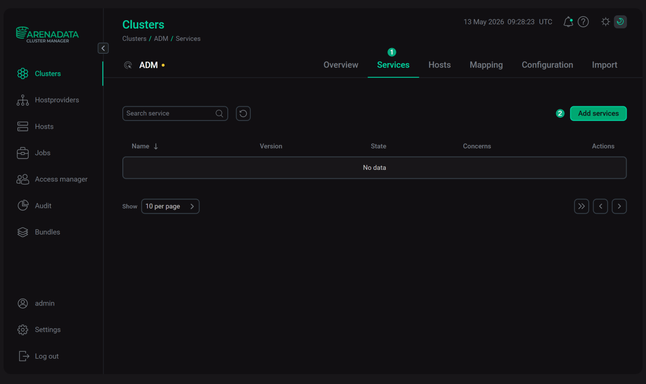
In ADCM, a **service** means a software that performs some function. Examples of services for ADM clusters: Grafana and VictoriaMetrics. The steps for adding services to a cluster are listed below:

1. Select a cluster on the ***Clusters*** page. To do this, click a cluster name in the ***Name*** column.



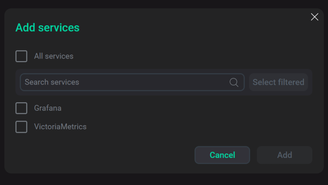
Select a cluster

2. Open the ***Services*** tab on the cluster page and click ***Add services*** (***Add service*** in first versions of ADCM).



Switch to adding services

3. In the opened dialog, select services that should be added to the cluster and click ***Add***.



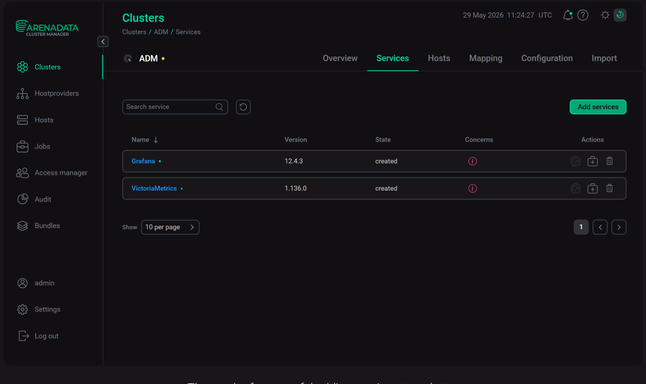
Select services

The brief description of available services is listed below.

Services that can be added to the ADM cluster

Service	Mandatory	Purpose
VictoriaMetrics	Yes	A time series database (TSDB) that is designed to collect and analyze application and infrastructure metrics
Grafana	No	A web service that allows you to create various sets of graphs and alerts for visualization of monitoring metrics

4. As a result, the added services are displayed on the ***Services*** tab.



The result of successful adding services to a cluster

NOTE

You can also add services later. The process of adding new services to already running cluster does not differ from installing a service from scratch.

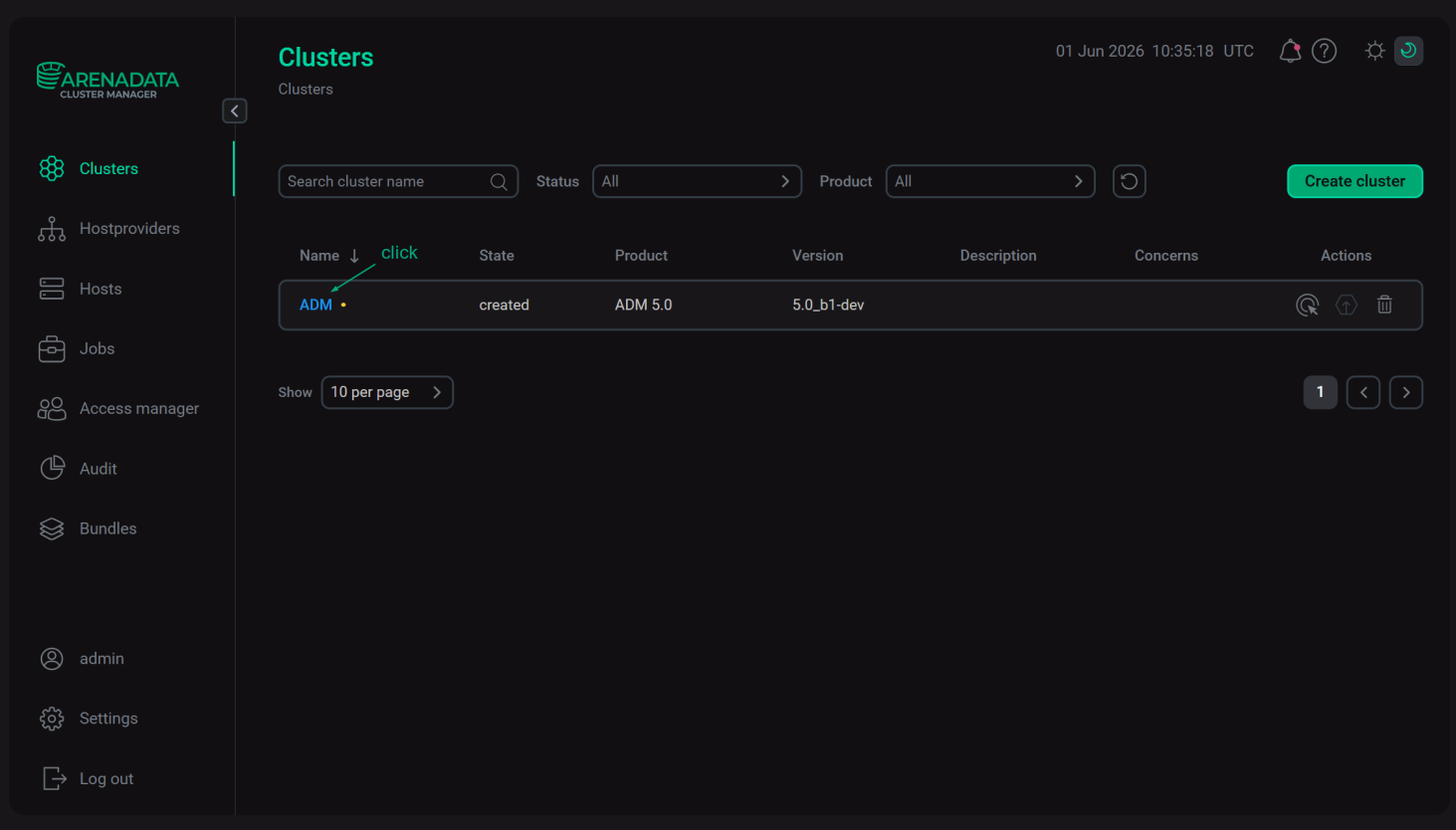
Add hosts to a cluster

Lisa Vaysman, Daria Barysheva

To Table of Contents

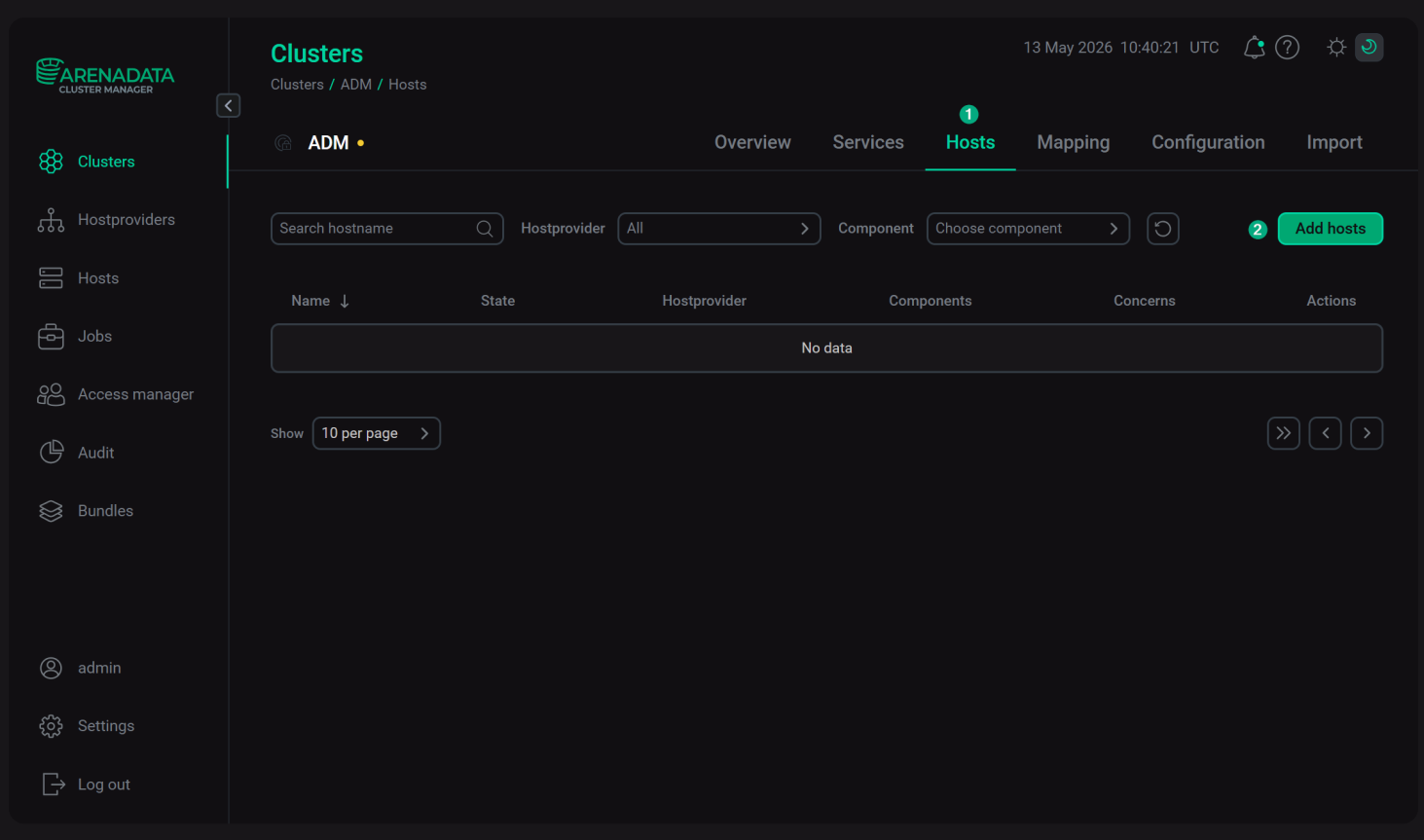
The steps for adding hosts to a cluster are listed below:

1. Select a cluster on the ***Clusters*** page. To do this, click a cluster name in the ***Name*** column.



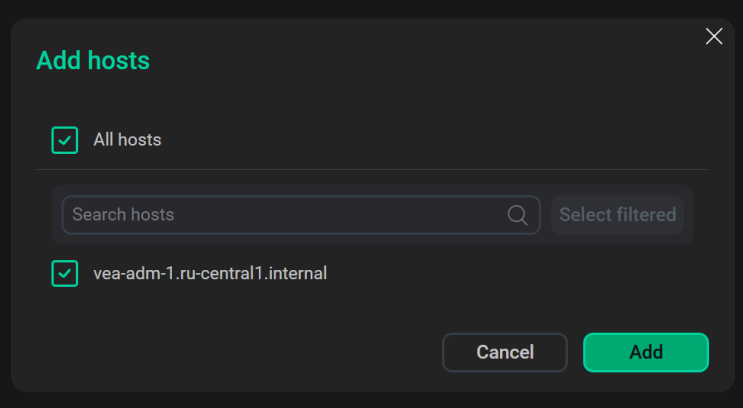
Select a cluster

2. Open the ***Hosts*** tab on the cluster page and click ***Add hosts***.



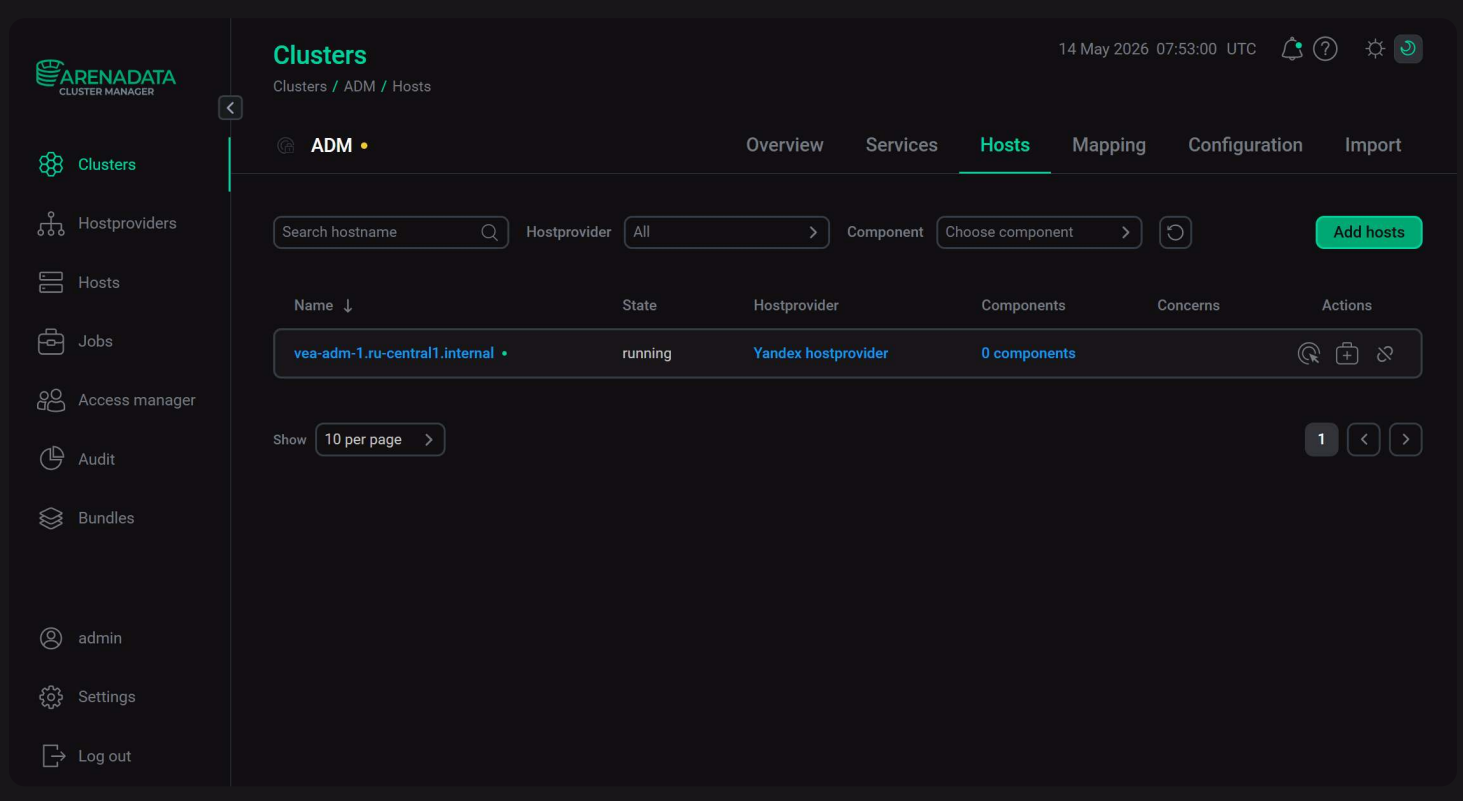
Switch to adding hosts

3. In the opened dialog, select hosts that should be added to the cluster and click ***Add***.



Select hosts

4. As a result, the added hosts are displayed on the ***Hosts*** tab.



The result of successful adding hosts to a cluster

Add components

Lisa Vaysman, Daria Barysheva

To Table of Contents

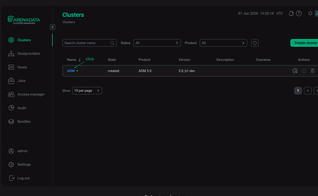
In ADCM, a **component** means a part of a service that should be deployed on one or several cluster hosts. For example, the VictoriaMetrics service consists of such components as VictoriaMetrics Insert, VictoriaMetrics Select, and VictoriaMetrics Storage.

IMPORTANT

Initially, there are no components on any of cluster hosts. You should map components to hosts manually.

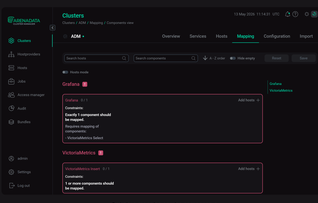
The steps for adding components to hosts are listed below:

1. Select a cluster on the **Clusters** page. To do this, click a cluster name in the ***Name*** column.



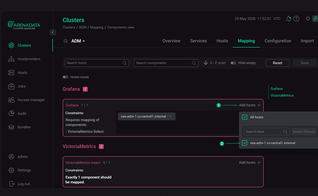
Select a cluster

2. Open the ***Mapping*** tab on the cluster page.



Switch to mapping service components

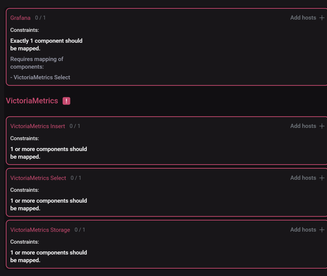
3. In the window that opens, map components to hosts. To assign hosts for each component, click ***Add hosts*** and select hosts in the pop-up window.



Select hosts for a component

Mandatory components

Each service can have mandatory and optional components. Mandatory components are highlighted in red. The ADCM UI also displays how many mandatory components should be installed.



Mandatory components

CAUTION

Without assigning a necessary number of hosts to mandatory components, the component mapping cannot be saved.

When distributing components, it is important to consider the following:

- **Component co-location on a host.** You can either install all components of an ADM cluster on a single host or distribute them among several hosts depending on your requirements. In particular, the VictoriaMetrics Insert, VictoriaMetrics Storage, and VictoriaMetrics Select components can be installed on the same host. Distributing them across several hosts provides minimal performance advantage, which is offset by higher operational complexity and resource overhead.
- **Changes to host number.** Once the VictoriaMetrics Storage component has been installed, if you subsequently change the number of hosts for this component, you must restart the VictoriaMetrics Insert component via the ***Restart*** action.
- **Replication factor calculations.** To ensure fault tolerance in case of a host loss, calculate the minimum number of hosts with the VictoriaMetrics Storage component installed using the following formula: `2 * N - 1` (where `N` is the replication factor value). For example, a replication factor of 2 requires a minimum of 3 hosts with the VictoriaMetrics Storage component.

  If the number of such hosts is lower than required to maintain the target replication factor, ADCM still allows cluster creation. In this case, the VictoriaMetrics Insert component writes metrics only to the available hosts running the VictoriaMetrics Storage component.

Monitoring VictoriaMetrics Storage availability

To monitor and identify VictoriaMetrics Storage host losses, check the following indicators on the VictoriaMetrics Insert side:

- **Internal metrics.** Connection failures cause an increase in the `vm_rpc_dial_errors_total` , `vm_rpc_connection_errors_total`, or `vm_rpc_handshake_errors_total` counters. You can view them in a browser at `http://vminsert-host>:8480/metrics`. Alternatively, connect to the VictoriaMetrics insert host via SSH and run:

  ```
  $ curl -s http://localhost:8480/metrics | grep -E
  "vm_rpc_dial_errors|vm_rpc_connection_errors|vm_rpc_handshake_errors"
  ```

  Example output indicating a failed host with the VictoriaMetrics Storage component (`gau-ub3`):

  ```
  vm_rpc_connection_errors_total{name="vminsert", addr="gau-ub3.develop.yc.adsw.io:8400", rpc_call="metric_rows"} 1
  vm_rpc_dial_errors_total{name="vminsert", addr="gau-ub3.develop.yc.adsw.io:8400", rpc_call="metric_rows"} 7718
  vm_rpc_handshake_errors_total{name="vminsert", addr="gau-ub3.develop.yc.adsw.io:8400", rpc_call="metric_rows"} 1
  ```

- **VictoriaMetrics Insert logs** (`docker logs`). Look for the `cannot dial storageNode … connection refused` error messages. To view the service logs, connect to the VictoriaMetrics Insert host via SSH and run:

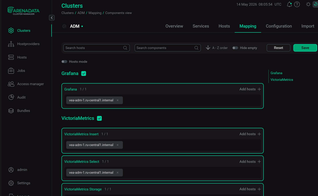
  ```
  $ sudo docker logs vminsert
  ```

  Example log entry indicating a failed host with the VictoriaMetrics Storage component:

  ```
  2026-06-19T13:45:12.123Z  error  VictoriaMetrics/app/vminsert/main.go:142  cannot dial storageNode "gau-ub3.
  develop.yc.adsw.io:8400": dial tcp 192.168.1.10:8400: connect: connection refused
  ```

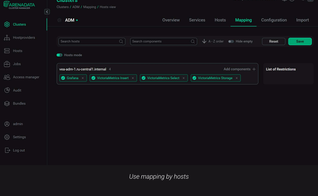
- **Network utilities.** You can also connect to the VictoriaMetrics Insert hosts via SSH and use standard Linux tools (such as `ss` or `netstat`) to verify connectivity to the VictoriaMetrics Storage hosts.

4. After component distribution is completed, click ***Save***.



Save mapping of components

Starting with ADCM 2.2.0, you can use another mapping mode — by hosts. To do this, switch on the ***Hosts mode*** toggle and map components to hosts by clicking ***Add components*** for each host sequentially.



Use mapping by hosts

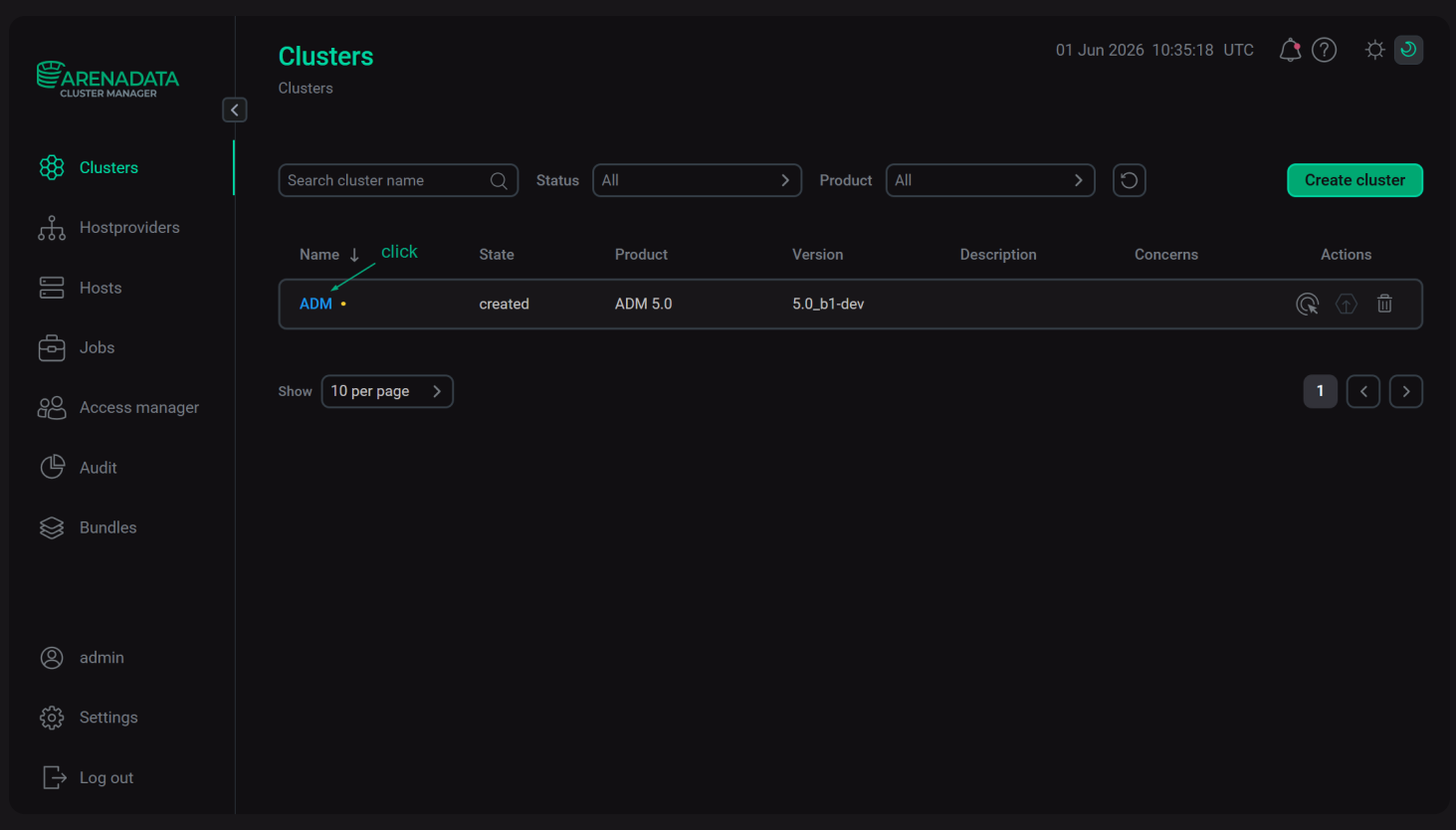
Configure services

Lisa Vaysman, Daria Barysheva

To Table of Contents

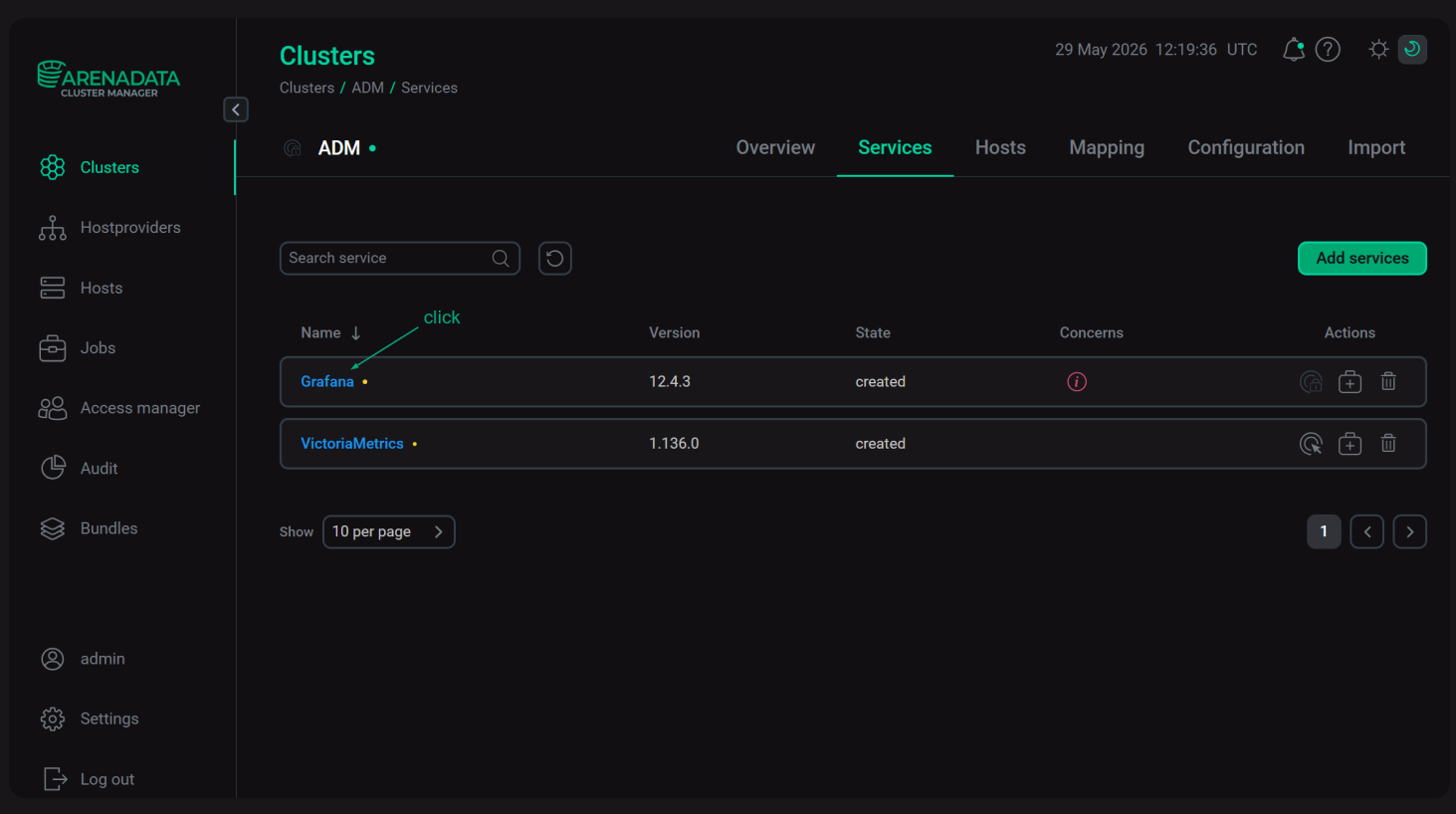
After adding all necessary services, you can configure its parameters. The steps for configuring cluster services are listed below:

1. Select a cluster on the ***Clusters*** page. To do this, click a cluster name in the ***Name*** column.



Select a cluster

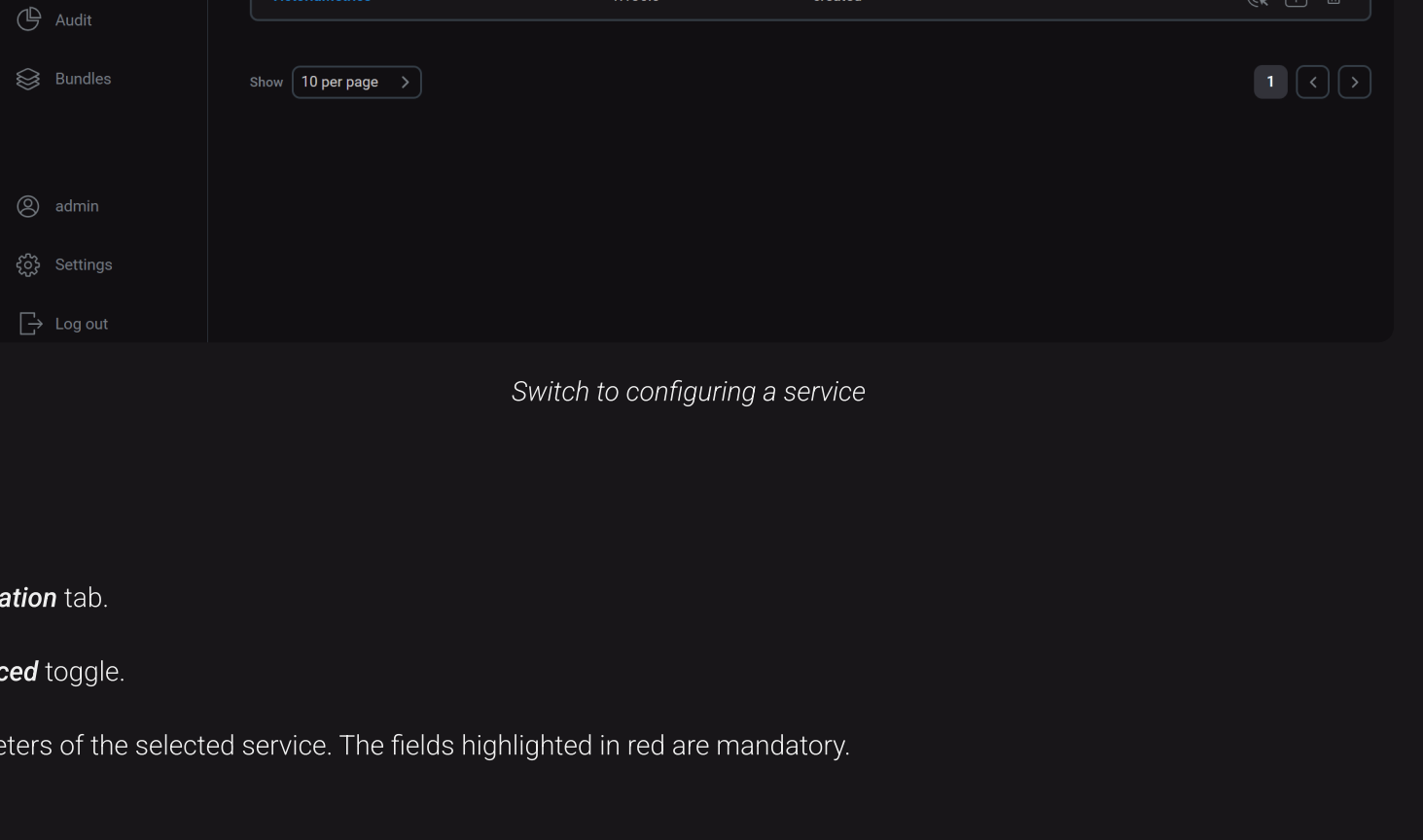
2. Open the ***Services*** tab on the cluster page, find the row that contains the desired service, and click its name in the ***Name*** column.



Switch to configuring a service

3. In the window that opens:

- Open the ***Primary Configuration*** tab.
- Switch on the ***Show advanced*** toggle.
- Fill in configuration parameters of the selected service. The fields highlighted in red are mandatory.
- Click ***Save***.



Configure a service

Configure a cluster

Lisa Vaysman, Daria Barysheva

Contents

To Table of Contents

Primary configuration

Configuration groups

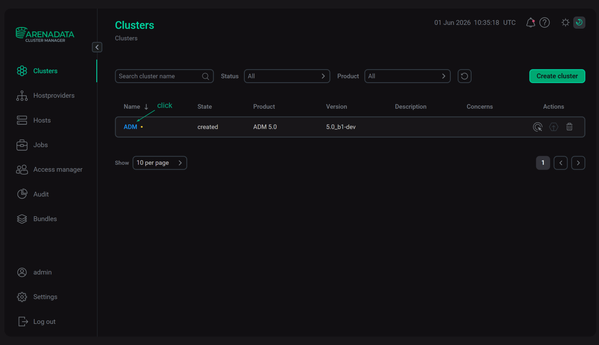
Ansible settings

IMPORTANT

There is usually no need to change cluster configuration parameters. You can leave all parameters at the default values.

After creation a new cluster, you can configure it by performing the following steps:

1. Select a cluster on the ***Clusters*** page. To do this, click a cluster name in the ***Name*** column.



Select a cluster

2. Open the ***Configuration*** tab on the cluster page. The ***Configuration*** tab includes the following sections: ***Primary configuration***, ***Configuration groups***, ***Ansible settings***.
3. Fill in all necessary parameters on the selected tab and click ***Save***.

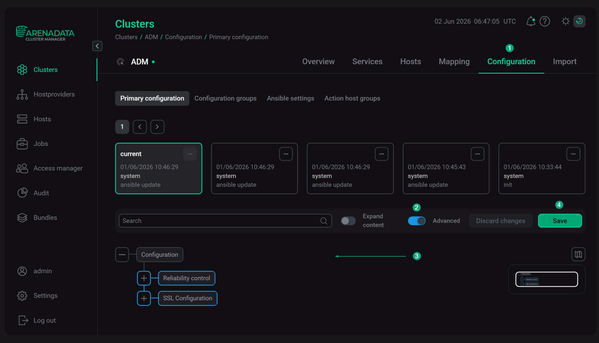
Primary configuration

The ***Configuration*** → ***Primary configuration*** tab allows you to configure the general cluster settings.

Available configuration parameters are listed below:

- The ***Reliability control*** section. It is displayed if the ***Show advanced*** switch is activated. The section contains the following parameters:
  - ***retries*** — the number of retries for cluster operations (`20` by default);
  - ***delay*** — the delay between retries in seconds (`10` by default);
  - ***timeout*** — timeout for checks of network ports in seconds (`60` by default).
- The ***SSL Configuration*** section. It is displayed if the ***Show advanced*** switch is activated. The section contains the following parameters:

Parameter	Description	Default value
SSL enabled	Indicates whether SSL is enabled or not	false
Certificate file	An absolute file path to the Privacy-Enhanced Mail (PEM) certificate on the cluster hosts. The file must contain Subject Alternative Name (SAN): ▪ FQDN ▪ short hostname ▪ localhost ▪ primary IP address ▪ `127.0.0.1`	/opt/adm/tls/server.crt
Private key file	An absolute file path to the private key on the cluster hosts	/opt/adm/tls/server.key



The cluster configuration window

Configuration groups

The ***Configuration*** → ***Configuration groups*** tab is designed to set cluster configuration groups. For details, please see Set up configuration groups.

Ansible settings

The ***Configuration*** → ***Ansible settings*** tab allows you to set Ansible configuration options at the cluster level. The tab is available starting with ADCM 2.2.0.

Ansible settings

Name	Description	Default
forks	The number of parallel processes to spawn when communicating with remote hosts	5

Import ET settings

Lisa Vaysman, Daria Barysheva

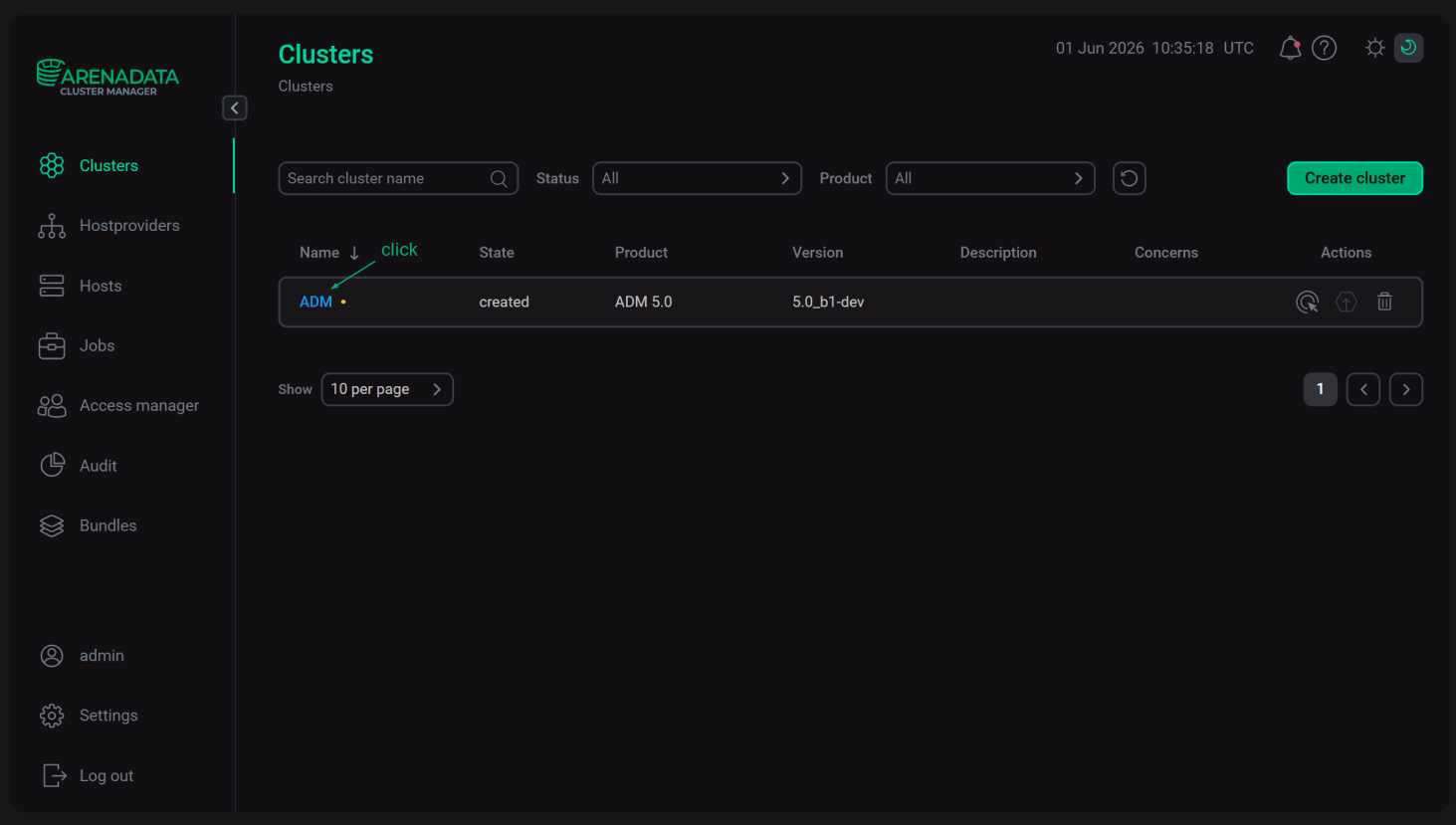
To Table of Contents

IMPORTANT

This step should be performed only in the case of offline installation.

In order to determine that a new cluster should be installed offline, it is necessary to integrate it with the previously created Enterprise Tools (ET) cluster. Follow the steps:

1. Select a cluster on the ***Clusters*** page. To do this, click a cluster name in the ***Name*** column.

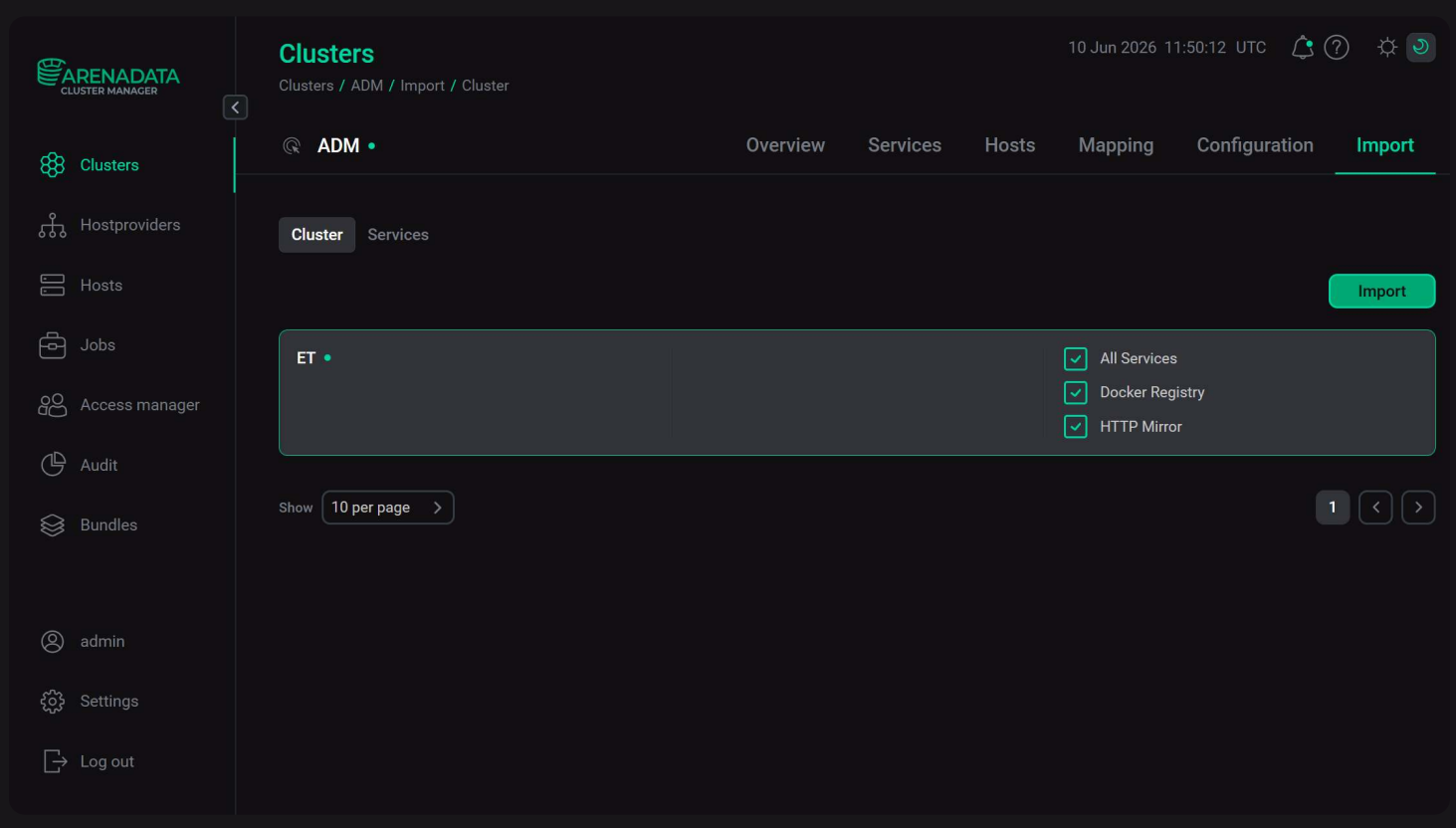


Select a cluster

2. Open the ***Import*** tab in the cluster menu.

3. Select ***Docker Registry*** and ***HTTP Mirror*** services of the previously created ET cluster.

4. Click ***Import***.



Integration with a ET cluster

NOTE

After successful import, the data stored locally in the Enterprise Tools cluster can be used at the cluster installation stage. So there is no need to extract this data via Internet.

Install a cluster

Lisa Vaysman, Daria Barysheva

Contents

To Table of Contents

Step 1. Installation
- Whole cluster
- Single services

Step 2. View installation results

Step 3. Check the cluster state after the installation

Step 4. View metrics
- Grafana web interface
- VictoriaMetrics web interfaces

Step 1. Installation

There are two ways to install cluster services:

1. **Whole cluster**. In this method all services are installed automatically one by one after applying the ***install*** action to a cluster.
2. **Single services**. In this method each service is installed manually after applying one or more actions to it. This way is also suitable for installing new services in the already deployed cluster.

In both cases each service starts automatically after installation.

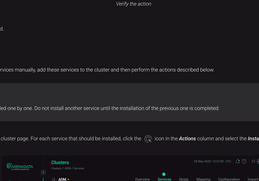
IMPORTANT

The installation must be performed on a clean host without prior installation of related or similar software.

Whole cluster

To install all cluster services within a single action, follow the steps:

1. Apply the ***Install*** action to the cluster selected on the ***Clusters*** page by clicking the icon in the ***Actions*** column.



Switch to the whole cluster installation

2. Verify the requested action in the opened window.

Verify the action

3. Wait until the job is completed.

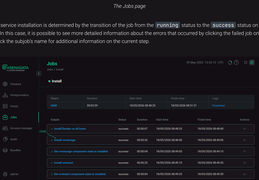
Single services

To install one or several cluster services manually, add these services to the cluster and then perform the actions described below.

CAUTION

Notice that services are installed one by one. Do not install another service until the installation of the previous one is completed.

1. Open the ***Services*** tab on the cluster page. For each service that should be installed, click the icon in the ***Actions*** column and select the ***Install*** action.



Switch to installation of single services

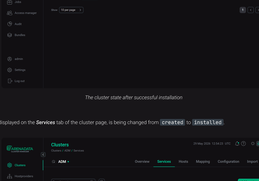
2. Verify the action in the opened window.

Verify the action

3. Wait until the job is completed. Then check that the service state has changed from `created` to `installed`.
4. Repeat the previous actions for other services that should be installed.

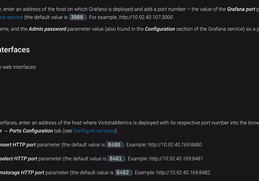
Step 2. View installation results

ADCM starts a single job for installation process, as well as for any other task. You can find out about the status of jobs more specifically on the ***Jobs*** page.



The Jobs page

The successful completion of the service installation is determined by the transition of the job from the `running` status to the `success` status on the ***Jobs*** page. If the job fails, it switches to the `failed` status. In this case, it is possible to see more detailed information about the errors that occurred by clicking the failed job on the ***Jobs*** page. The window with the job's inner steps is opened. Click the subjob's name for additional information on the current step.



Switch to subjob details

A job page contains two sections: ***ansible [stdout]*** and ***ansible [stderr]***. These are technical logs of the job that include information from standard I/O streams stdout/stderr. These logs can help to understand what caused the problems.

There can also be the optional third section ***ansible [check]***. These are the logs of the most frequent errors checks. The description of these errors is more simple and specific, than in two previous technical logs.

The contents of all three sections should be studied if errors occur.

Job details

Step 3. Check the cluster state after the installation

As a result of successful installation, the cluster and its services change their state according to the following rules:

- The cluster state, displayed in the ***State*** field on the ***Clusters*** page, is being changed from `created` to `installed`.

The cluster state after successful installation

- The state of cluster services, displayed on the ***Services*** tab of the cluster page, is being changed from `created` to `installed`.

The state of cluster services after successful installation

Step 4. View metrics

Grafana web interface

In the address bar of your browser, enter an address of the host on which Grafana is deployed and add a port number — the value of the ***Grafana port*** parameter is located in the ***Configuration*** section of the Grafana service (the default value is `3000`). For example, *http://10.92.40.107:3000*.

To log in, use `admin` as a username, and the ***Admin password*** parameter value (also found in the ***Configuration*** section of the Grafana service) as a password.

VictoriaMetrics web interfaces

VictoriaMetrics has three separate web interfaces:

- VictoriaMetrics Insert
- VictoriaMetrics Select
- VictoriaMetrics Storage

To access VictoriaMetrics web interfaces, enter an address of the host where VictoriaMetrics is deployed with its respective port number into the browser address bar. Find specific port numbers on the ***Configuration*** → ***Ports Configuration*** tab (see Configure services):

- VictoriaMetrics Insert: the ***vminsert HTTP port*** parameter (the default value is `8480`). Example: *http://10.92.40.169:8480*.
- VictoriaMetrics Select: the ***vmselect HTTP port*** parameter (the default value is `8481`). Example: *http://10.92.40.169:8481*.
- VictoriaMetrics Storage: the ***vmstorage HTTP port*** parameter (the default value is `8482`). Example: *http://10.92.40.169:8482*.

NOTE

Links to the Grafana and VictoriaMetrics web interfaces are also available on the ***Info*** tab of the corresponding services. To access them, go to your cluster page, select the required service on the ***Services*** tab, and open the ***Info*** tab.

Grafana service management via ADCM

Lisa Vaysman

To Table of Contents

The ADCM UI provides actions to manage the Grafana service and its components. For information on how to run service actions, refer to Service actions.

The actions available for the Grafana service are listed in the table below.

Grafana service actions

Action	Description
Check	Runs service-specific tests to check the health of the service and its components
Restart	Restarts the service. When you run this action, the option ***Apply configs from ADCM*** is available. If it is set to `true`, all service configurations defined in ADCM settings will be applied during the service restart. Otherwise, the service restarts without applying configurations from ADCM
Start	Starts the Grafana service. When you run this action, the option ***Apply configs from ADCM*** is available. If it is set to `true`, all service configurations defined in ADCM settings will be applied at service startup. Otherwise, the service starts without applying configurations from ADCM
Stop	Stops the Grafana service
Uninstall	Uninstalls the Grafana service. However, Grafana data and provisioning files are not deleted from the host where they were stored (see section ***Grafana*** of Configuration parameters)

Grafana component actions

Action	Description
Restart	Restarts the component on all cluster hosts
Start	Starts the component on all cluster hosts
Stop	Stops the component on all cluster hosts

VictoriaMetrics service management via ADCM

Lisa Vaysman

Contents

To Table of Contents

VictoriaMetrics components actions

Restart VictoriaMetrics components

The ADCM UI provides actions to manage the VictoriaMetrics service and its components. For information on how to run service actions, refer to Service actions.

The actions available for the VictoriaMetrics service are listed in the table below.

Action	Description
Check	Runs service-specific tests to check the health of the service and its components
Restart	Restarts the service. When you run this action, the option ***Apply configs from ADCM*** is available. If it is set to `true`, all service configurations defined in ADCM settings will be applied during the service restart. Otherwise, the service restarts without applying configurations from ADCM
Start	Starts the VictoriaMetrics service. When you run this action, the option ***Apply configs from ADCM*** is available. If it is set to `true`, all service configurations defined in ADCM settings will be applied at service startup. Otherwise, the service starts without applying configurations from ADCM
Stop	Stops the VictoriaMetrics service
Uninstall	Uninstalls the VictoriaMetrics service. VictoriaMetrics Storage data is not deleted from the host where it was stored (see section ***VictoriaMetrics → Storage Settings*** of Configuration parameters)

VictoriaMetrics components actions

The following actions are supported for all VictoriaMetrics components.

Action	Description
Restart	Restarts the component on all cluster hosts
Start	Starts the component on all cluster hosts
Stop	Stops the component on all cluster hosts

Restart VictoriaMetrics components

If you need to restart VictoriaMetrics components individually via component actions, trigger their ***Restart*** actions in the following order:

1. Restart the VictoriaMetrics Storage component.

NOTE

On large data volumes, the VictoriaMetrics Storage component may take a significant amount of time to restart (see the VictoriaMetrics documentation). Data ingestion and querying are completely unavailable during the restart. Although the VictoriaMetrics Insert and VictoriaMetrics Select components remain running, they cannot process data until the VictoriaMetrics Storage component is up and running.

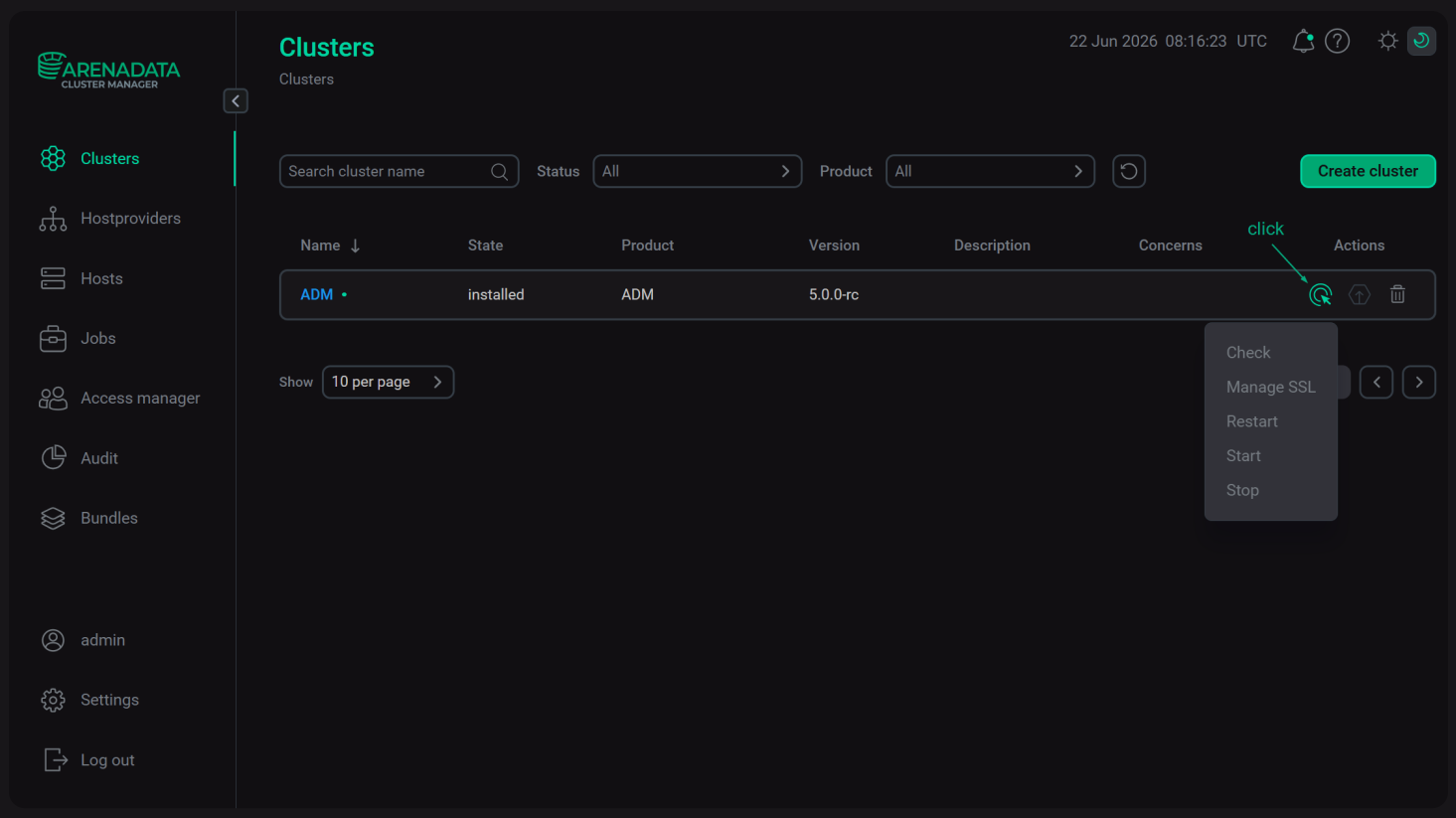
2. Restart the VictoriaMetrics Insert component. The restart finishes almost instantly because the component is stateless, so data ingestion stops only for a brief moment.
3. Restart the VictoriaMetrics Select component. The restart completes almost instantly, so metrics visualization in Grafana is interrupted only for a brief moment.

Cluster actions

Lisa Vaysman

To Table of Contents

You can manage an ADM cluster on the ***Clusters*** page in the ADCM UI.



The Clusters page

Refer to ADCM documentation for an overview of the ***Clusters*** page and common actions.

The following actions are available for the ADM cluster:

- Check
- Manage SSL
- Restart
- Start
- Stop

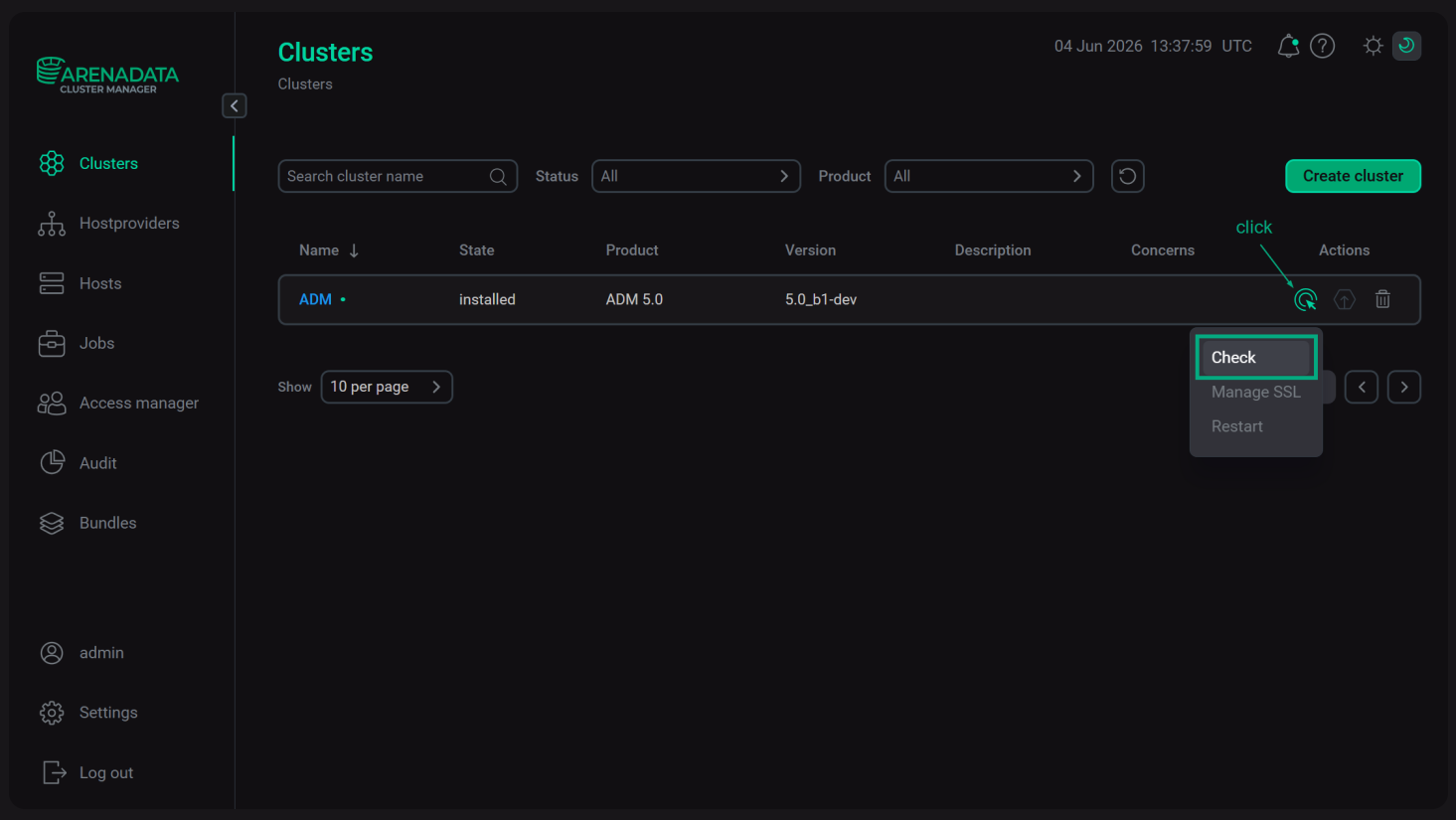
Check

Lisa Vaysman

To Table of Contents

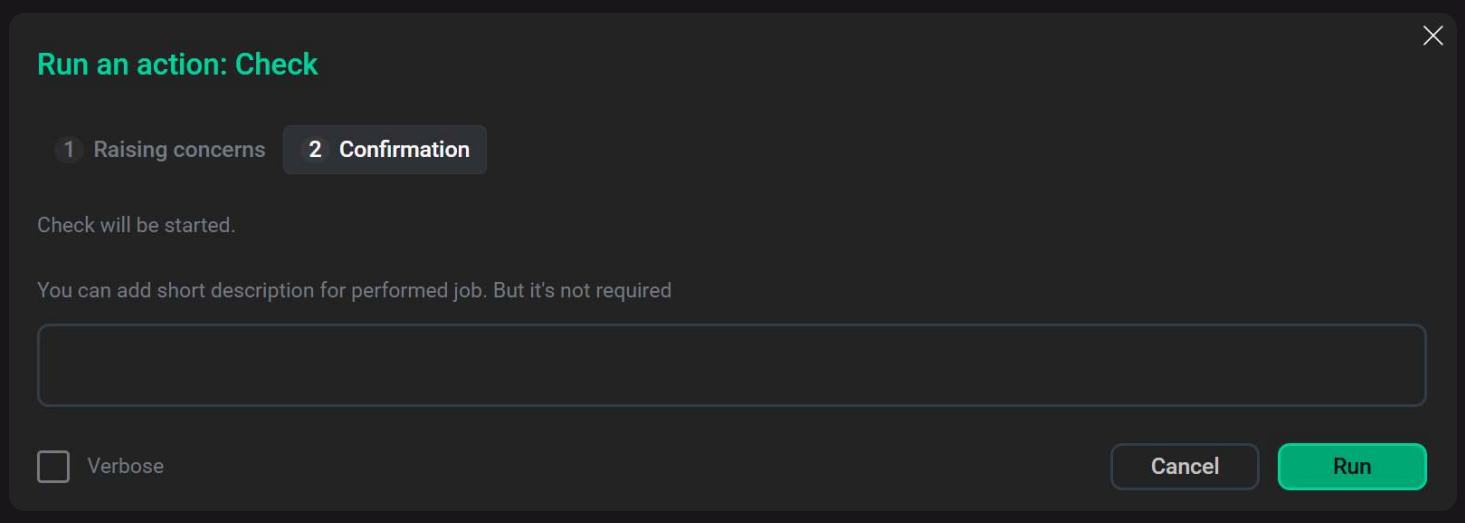
The ***Check*** action verifies whether the ADM services work correctly. It sequentially runs service-specific tests for each service to check the health of the service and its components.

To run an action, go to ***Clusters*** page, click in the ***Actions*** column, and select the action from the drop-down list that appears.



Execute the "Check" action

When you choose an action, ADCM displays a dialog window to confirm the choice. In this dialog window, you can select the ***Verbose*** checkbox to see additional execution details.



The "Run an action" window

You can find execution process and result of the action on the ***Jobs*** page. On this page, you can view action execution details, including verbose Ansible STDOUT/STDERR outputs.

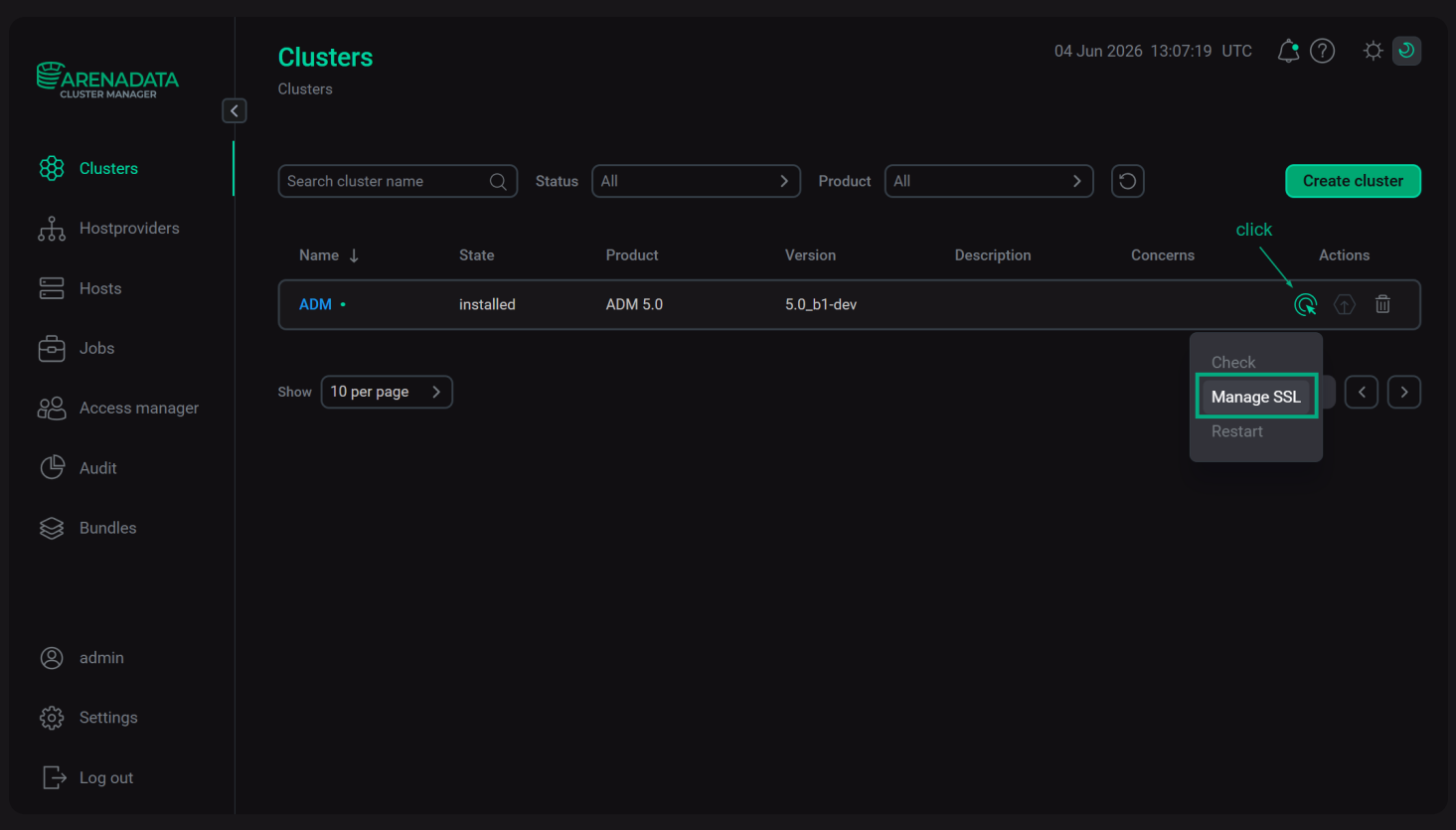
Manage SSL

Lisa Vaysman

To Table of Contents

The *Manage SSL* action allows you to control the SSL encryption of all services in the ADM cluster. For additional information on SSL settings, see Requirements for SSL certificates.

To run an action, go to *Clusters* page, click in the *Actions* column, and select the action from the drop-down list that appears.



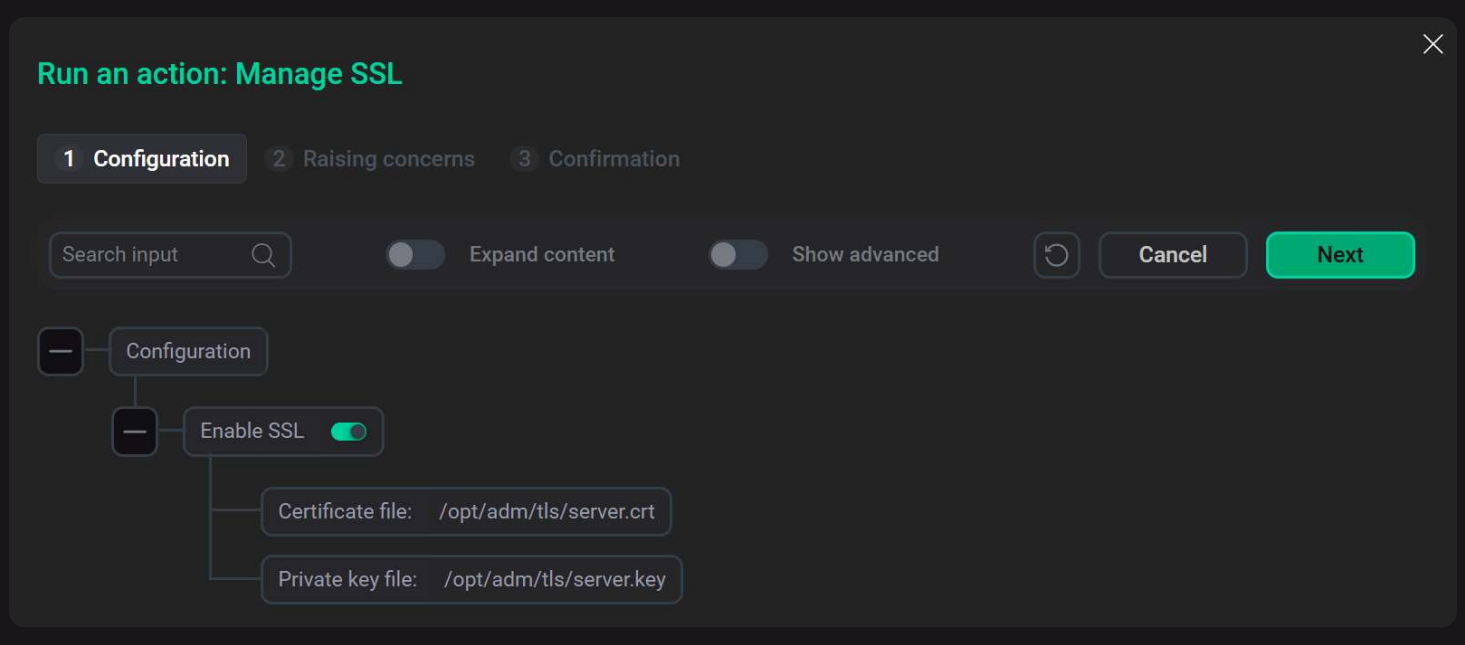
Execute the "Manage SSL" action

When you choose an action, ADCM displays a dialog window where you can specify SSL settings listed in the table below.

SSL settings

Parameter	Description	Default value
Certificate file	An absolute file path to the Privacy-Enhanced Mail (PEM) certificate on the cluster hosts. The file must contain Subject Alternative Name (SAN): - FQDN - short hostname - localhost - primary IP address - `127.0.0.1`	/opt/adm/tls/server.crt
Private key file	An absolute file path to the private key on the cluster hosts	/opt/adm/tls/server.key

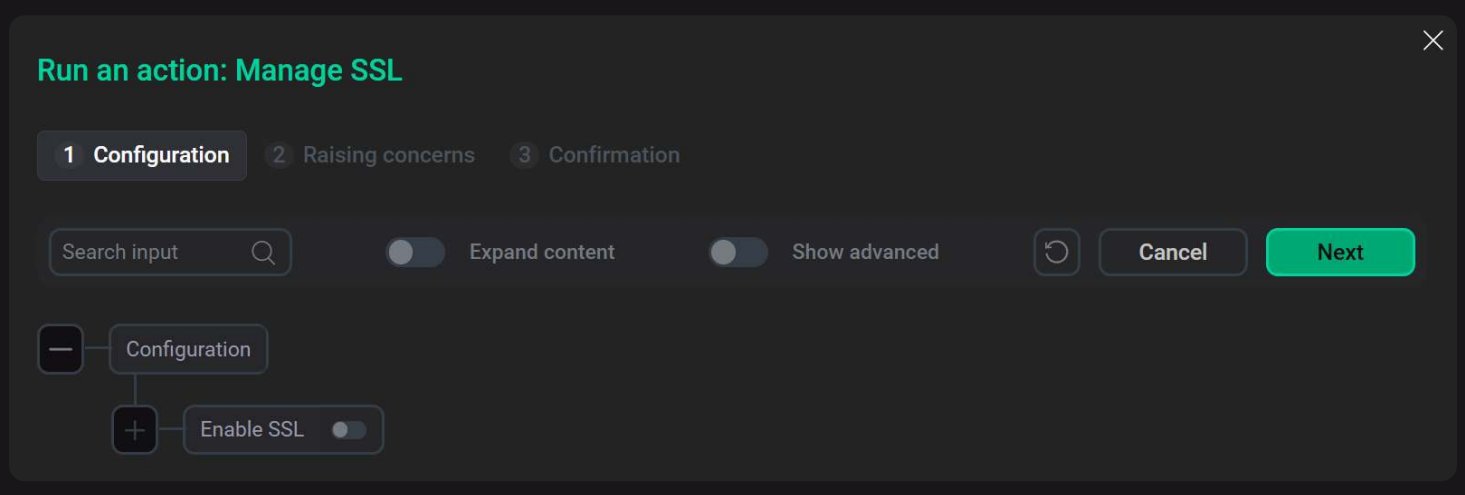
To enable SSL, run the action with the *Enable SSL* parameter group active and the parameters filled in.



Enable SSL with the "Manage SSL" action

After you specify the parameters, click *Run*. In the action confirmation window that appears, you can select the *Verbose* checkbox to see additional execution details.

To disable SSL, run the *Manage SSL* action with the *Enable SSL* parameter group inactive.



Disable SSL with the "Manage SSL" action

During the execution of the *Manage SSL* action, ADCM stops all services one by one, reconfigures them to enable or disable SSL encryption, then starts the services and checks them.

You can find execution process and result of the action on the *Jobs* page. On this page, you can view action execution details, including verbose Ansible STDOUT/STDERR outputs.

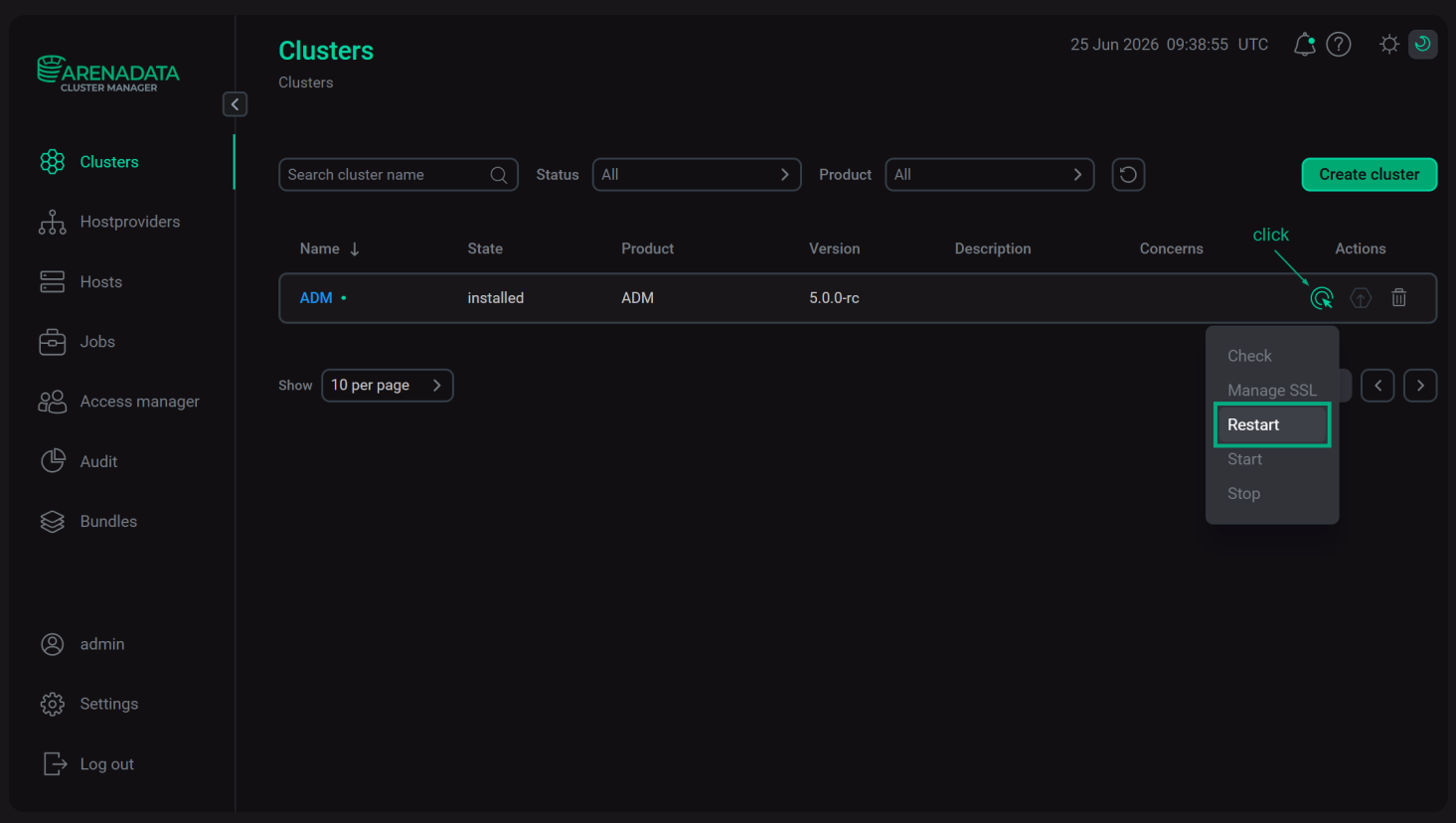
Restart

Lisa Vaysman

To Table of Contents

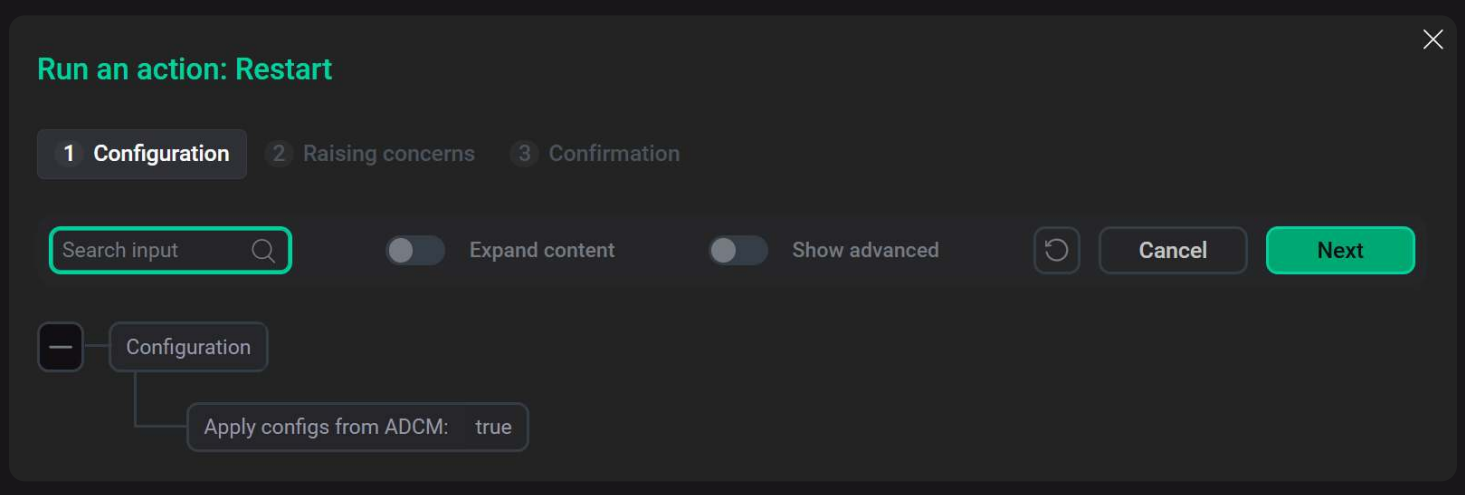
The ***Restart*** action stops and then starts all services of the cluster.

To run an action, go to ***Clusters*** page, click in the ***Actions*** column, and select the action from the drop-down list that appears.



Execute the "Restart" action

The dialog window of the ***Restart*** action contains the ***Apply configs from ADCM*** setting. If the value `true` is selected, then the services are launched taking into account all the parameters specified in the settings of these services in ADCM. Otherwise, the settings of these services in ADCM are ignored. Next, confirm the action by clicking ***Next*** and ***Run***.



The "Run an action" window

You can find execution process and result of the action on the ***Jobs*** page. On this page, you can view action execution details, including verbose Ansible STDOUT/STDERR outputs.

The ***Restart*** cluster action automatically restarts all cluster services and their components in the optimal logical order.

During restart, the cluster services are unavailable. If the VictoriaMetrics Storage component contains a large volume of data, the cluster restart process may take long time to complete (see the VictoriaMetrics documentation).

Alternatively, you can use the ***Restart*** action of a particular service or its component to manually restart only required services or components one after another.

NOTE

- Grafana can be restarted at any moment independently of VictoriaMetrics. Once the VictoriaMetrics Select component is up and running, Grafana automatically connects to it to fetch metrics for display. The VictoriaMetrics Select component, subsequently, retrieves data from the VictoriaMetrics Storage component as soon as this component becomes operational.
- If both Grafana and VictoriaMetrics have been running and then VictoriaMetrics is restarted, then it is not necessary to restart Grafana (Grafana reconnects to VictoriaMetrics automatically).

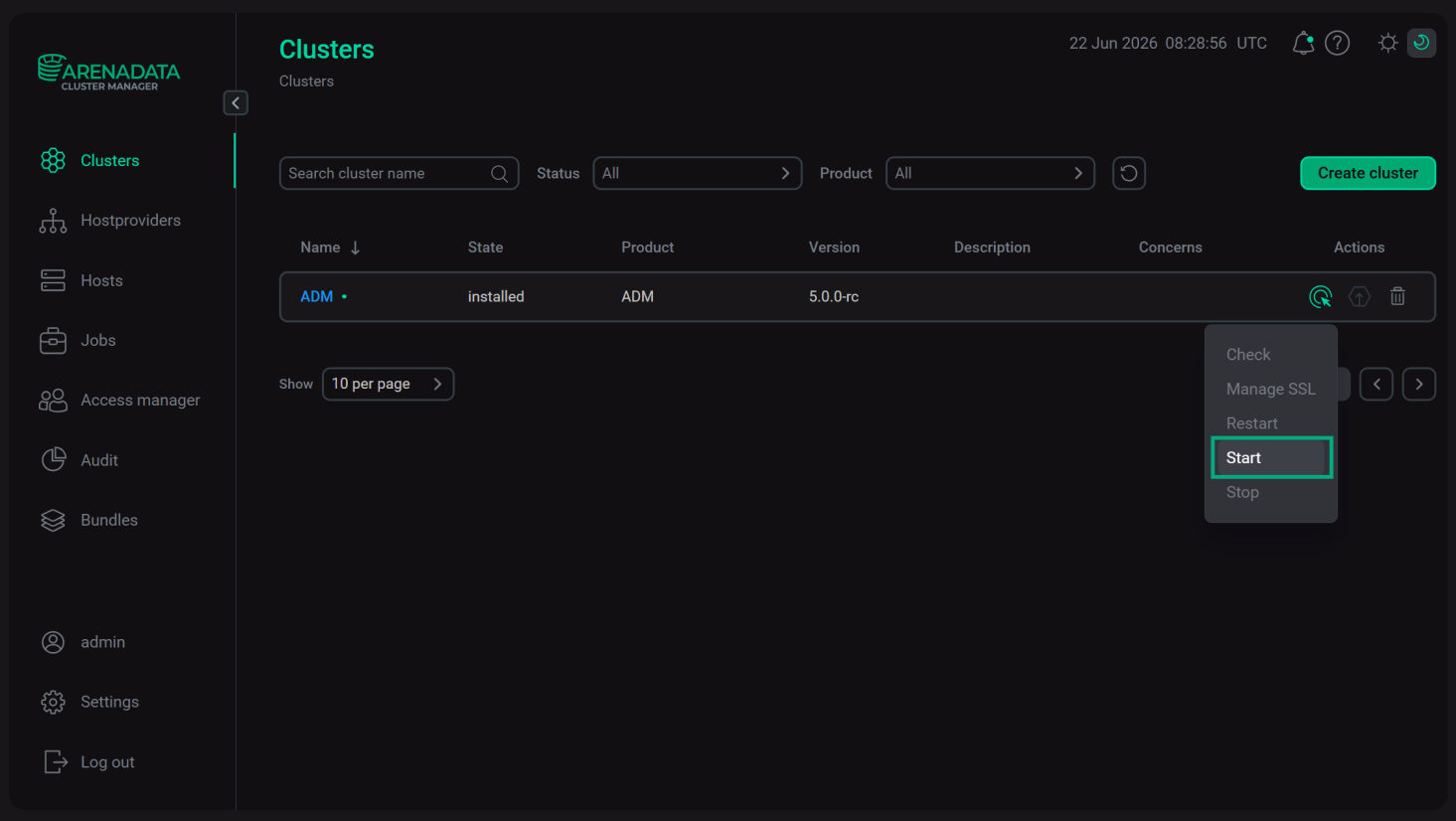
Start

Lisa Vaysman

To Table of Contents

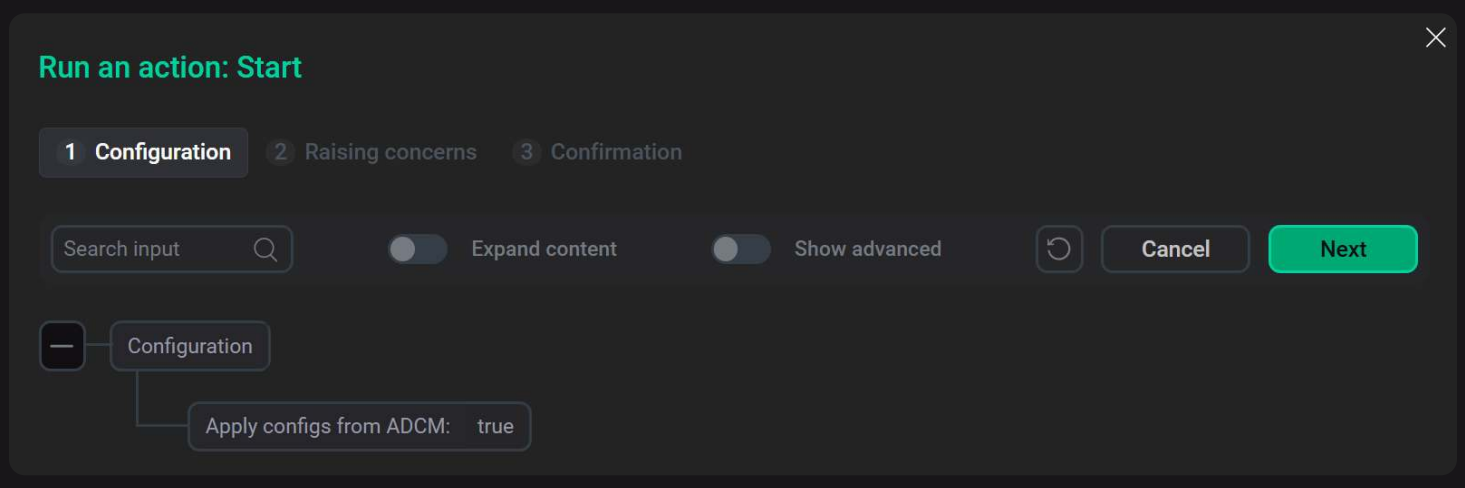
The ***Start*** action starts all the services installed in the ADM cluster.

To run an action, go to ***Clusters*** page, click in the ***Actions*** column, and select the action from the drop-down list that appears.



Execute the "Start" action

The dialog window of the ***Start*** action contains the ***Apply configs from ADCM*** setting. If the value `true` is selected, then the services are launched taking into account all the parameters specified in the settings of these services in ADCM. Otherwise, the settings of these services in ADCM are ignored. Next, confirm the action by clicking ***Next*** and ***Run***.



The "Run an action" window

You can find execution process and result of the action on the ***Jobs*** page. On this page, you can view action execution details, including verbose Ansible STDOUT/STDERR outputs.

Alternatively, you can use the ***Start*** action of a particular service or its component to manually start only required services or components one after another.

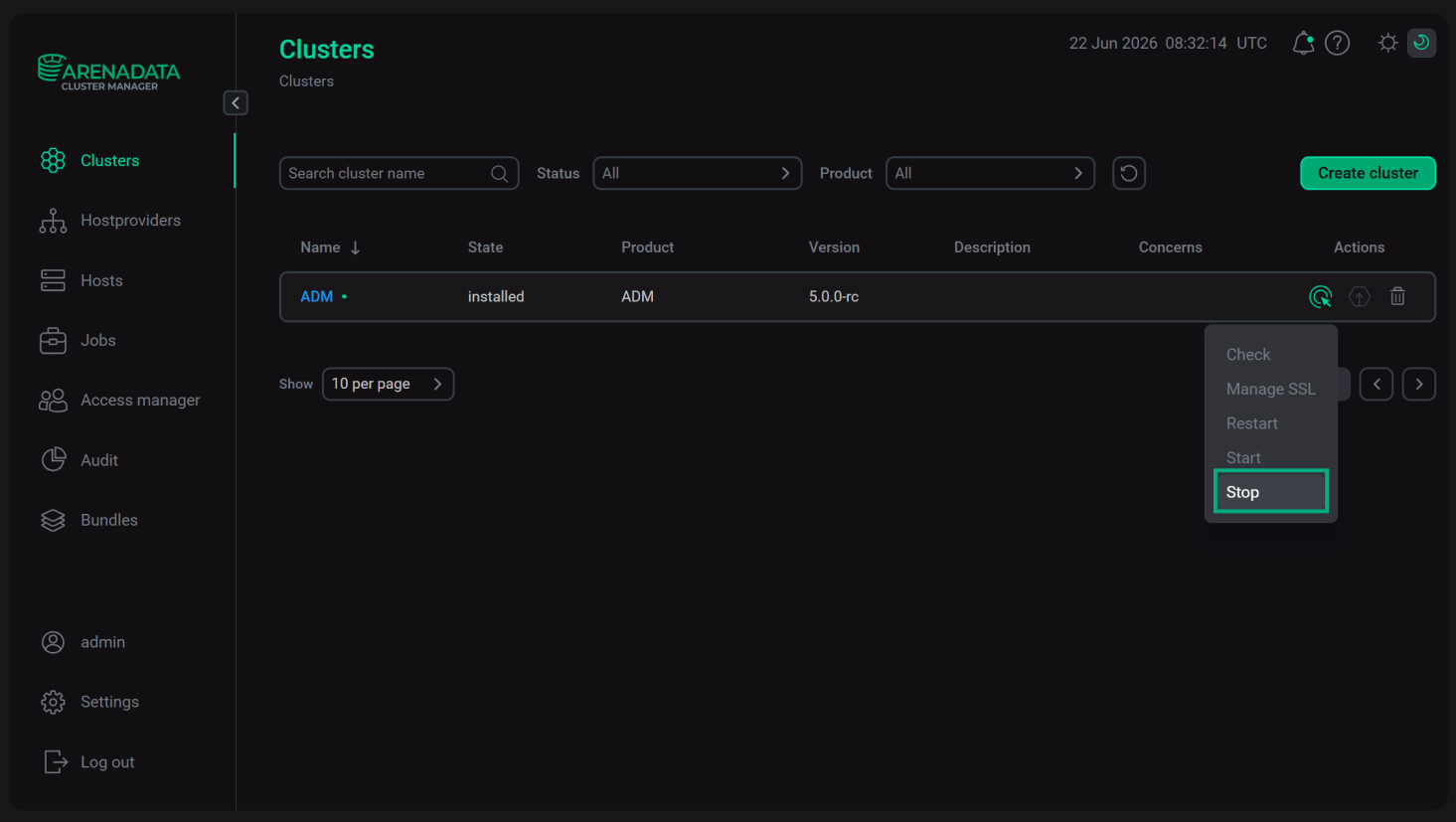
Stop

Lisa Vaysman

To Table of Contents

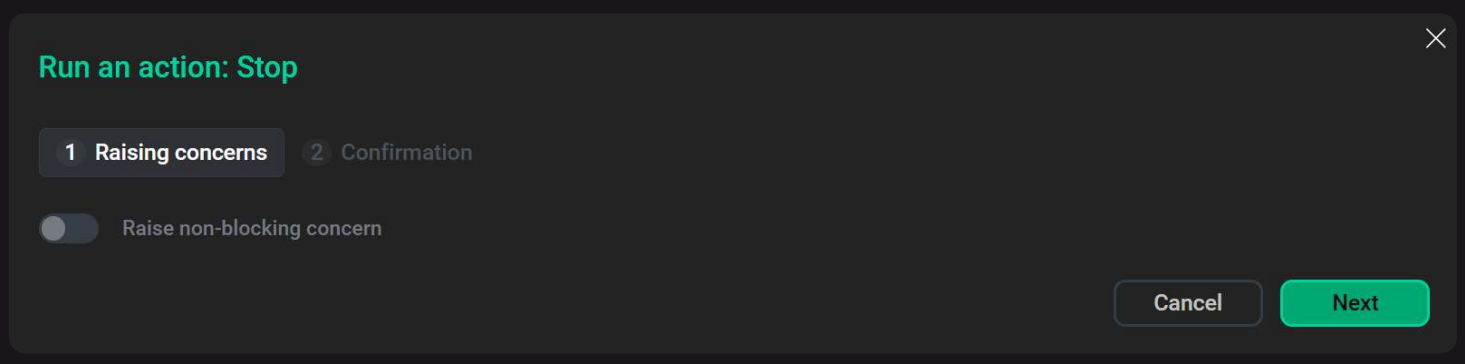
The ***Stop*** action stops all the services installed in the ADM cluster.

To run an action, go to ***Clusters*** page, click in the ***Actions*** column, and select the action from the drop-down list that appears.



Execute the "Stop" action

When you choose an action, ADCM displays a dialog window to confirm the choice. In this dialog window, you can select the ***Verbose*** checkbox to see additional execution details.



The "Run an action" window

You can find execution process and result of the action on the ***Jobs*** page. On this page, you can view action execution details, including verbose Ansible STDOUT/STDERR outputs.

IMPORTANT

Stopping the ADM cluster completely stops data ingestion via the VictoriaMetrics Insert component and query processing via the VictoriaMetrics Select component. The ADM cluster services will not work until you start the cluster again.

Alternatively, you can use the ***Stop*** action of a particular service or its component to manually stop only required services or components one after another.

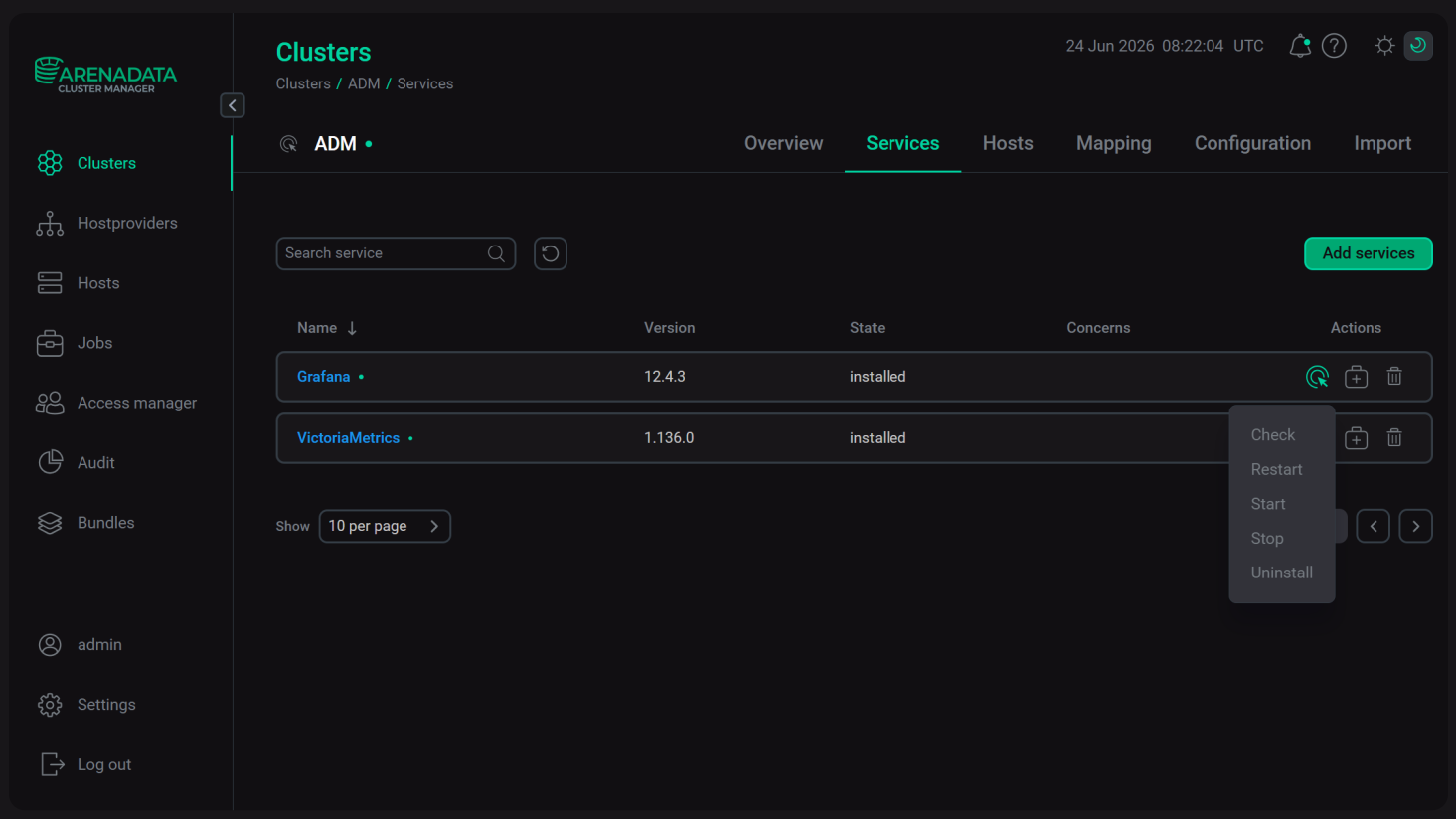
Service actions

Lisa Vaysman

To Table of Contents

ADCM UI provides a convenient way to interact with ADM services via actions. To perform an action on a service, do the following:

1. On the ***Clusters*** page, select the ADM cluster you want to work with.
2. On the cluster page that opens, go to ***Services***.
3. Click ***Actions*** for the service you need. The list of actions available for the given service is displayed in the drop-down.



Running service actions

Refer to ADCM documentation for an overview of the ***Services*** tab and common actions. For more information on actions for each ADM service, refer to the following links:

- Grafana
- VictoriaMetrics

Configuration parameters

Lisa Vaysman

Contents

To Table of Contents

Grafana

VictoriaMetrics

This article describes the parameters that can be configured for ADM services via ADCM. To read about the configuring process, refer to the relevant articles for online and offline installation.

NOTE

Some of the parameters become visible in the ADCM UI after the ***Show advanced*** flag being set.

Grafana

Parameter	Description	Default value
Grafana data path	Path to the Grafana data storage on the host where Grafana is deployed. If changed, the changes apply only after performing the Uninstall and then ***Install*** service actions	/opt/adm/grafana/data
Grafana provisioning path	Path to the Grafana provisioning files on the host where Grafana is deployed. If changed, the changes apply only after performing the Uninstall and then ***Install*** service actions	/opt/adm/grafana/provisioning
Grafana port	HTTP port for the Grafana web interface	3000
Admin username	Grafana administrator username used to log in to the Grafana web interface	admin
Admin password	Password for the Grafana administrator. ADCM uses it to automatically create credentials (a system user and its API token) for dashboard synchronization	–

VictoriaMetrics

Replication Settings

Parameter	Description	Default value
Replication Factor	Number of the hosts with the VictoriaMetrics Storage component used for replicating each metric. For details on replication factor calculations depending on the hosts number, see Add components (online installation) or Add components (offline installation)	2
Deduplication Interval	The deduplication interval that, when configured correctly, prevents the overlapping of duplicate metric points, which can result in a sawtooth pattern on your Grafana dashboards. It is recommended to keep the default value (`1ms`) to ensure stable cluster replication out of the box. You can change this value to match the scrape interval of your metrics source, if necessary. Large deduplication intervals (e.g. `1m` or higher) increase memory consumption on hosts with the VictoriaMetrics Storage and VictoriaMetrics Select components installed, as the system must buffer more data to compare and deduplicate incoming metrics. Format value using `ms` (milliseconds), `s` (seconds), `m` (minutes), or `h` (hours). Example: `1ms`. Combined values are not supported (e.g. `1s500ms` is invalid). Entering an invalid value will result in the VictoriaMetrics service failing to run	1ms

Storage Settings

Parameter	Description	Default value
Retention Period	Metrics retention period. Metrics older than this value are automatically deleted from the storage. The minimum retention period supported by VictoriaMetrics is 1 day (`1d`) or 24 hours (`24h`). Indefinite retention is not supported by VictoriaMetrics. To retain data long-term, specify an arbitrarily high duration, such as 100 years (`100y`). Format value using `d` (days), `w` (weeks), `M` (months), or `y` (years). Combined values are also supported. Examples: `7d`, `2w`, `1M`, `1y`, `1M2w`. If you reduce the retention period, data older than the new limit is not deleted instantly upon service restart. VictoriaMetrics removes expired metrics gradually in the background	1M
Storage Data Path	Path to the VictoriaMetrics Storage data on the host where the VictoriaMetrics Storage component is deployed. If changed, the changes apply only after restarting the VictoriaMetrics Storage component. See the VictoriaMetrics service actions To change the storage location without losing metrics history, stop the VictoriaMetrics Storage component and copy the data directory to the new location (e.g. using `rsync`). Then, update this parameter in ADCM and start the VictoriaMetrics Storage component. VictoriaMetrics will automatically recognize and load the historical data from the new path	/opt/adm/vmstorage/data

Ports Configuration

Parameter	Description	Default value
vminsert HTTP Port	HTTP port used by the VictoriaMetrics Insert component to accept incoming metrics from agents like vmagent or Prometheus via the Prometheus-compatible remote_write protocol	8480
vmselect HTTP Port	HTTP port used by the VictoriaMetrics Select component to accept PromQL queries from visualization tools like Grafana	8481
vmstorage HTTP Port	HTTP port used by the VictoriaMetrics Storage component to handle health checks and self-monitoring metrics requests	8482
vmstorage vminsert RPC Port	TCP port used by the VictoriaMetrics Storage component to receive incoming metrics data from the VictoriaMetrics Insert component via an internal RPC protocol	8400
vmstorage vmselect RPC Port	TCP port used by the VictoriaMetrics Storage component to accept data read requests from the VictoriaMetrics Select component via an internal RPC protocol	8401

Security settings

Parameter	Description	Default value
pprof_authkey	The authentication key used to restrict access to the built-in Go profiler (`/debug/pprof/`) endpoints on the VictoriaMetrics Insert, VictoriaMetrics Select, and VictoriaMetrics Storage components. ADCM always stores this key encrypted with Ansible Vault, so the ADCM UI displays it as a string starting with `$ANSIBLE_VAULT;1.1;AES256`. You can manage the key as follows: ▪ **Authentication.** To authenticate requests to the profiler, pass the decrypted key as an HTTP `Authorization` header or as an `authKey` query parameter. The query parameter takes priority. URL template: `http://<host>:<port>/debug/pprof/?authKey=<decrypted_key>` Example: `http://10.92.40.169:8482/debug/pprof/?authKey=xLUPib7YKNQuz0l6X9TrbncF` Where: ▪ `<host>` is the host address where the corresponding VictoriaMetrics component is deployed. ▪ `<port>` is a port number of the corresponding VictoriaMetrics component (see Ports Configuration). ▪ `<decrypted_key>` is the plaintext value of the key. To find it, log in to the host via SSH and run `ps aux | grep pprofAuthKey`. ▪ **Changing the key.** Enter any plaintext string to use as the new authentication key. It will be encrypted by ADCM automatically. Leaving the key value empty disables authentication, making the profiling endpoints publicly accessible without a password.	–

ADM releases

Lisa Vaysman

Contents

To Table of Contents

5.0.0

5.0.0

Date: 26.06.2026

New features

Initial release of Arenadata Monitoring (ADM) 5.0.0

Introduction of a centralized monitoring platform based on VictoriaMetrics

Support for centralized long-term metrics storage using VictoriaMetrics Cluster

Support for metrics collection via vmagent

Integration with Grafana for metrics visualization and dashboards

Support for delivery of product-specific Grafana dashboards

Support for collecting metrics from Prometheus-based monitoring systems via the remote_write protocol

Support for fault-tolerant metrics storage through VictoriaMetrics replication mechanisms

ADM bundle is available only in Enterprise Edition

Misc/Internal

ADM 5.0.0 is a new monitoring platform built on a completely new technology stack based on VictoriaMetrics

Upgrade from legacy ADM 4.x releases (up to the final version 4.4) to ADM 5.0.0 is not supported

Migration of historical data from ADM 4.x to ADM 5.0.0 is not supported

The new ADM 5.0.0 cluster must be deployed and operated completely independently of any existing ADM 4.x installations

Version compatibility matrix

Lisa Vaysman

To Table of Contents

ADM version	ADCM min version
5.0.0	2.10

Arenadata Monitoring

To Table of Contents

IMPORTANT

ADM 5.0.0 brings a new stack of components different from earlier versions of the product. Consequently, you cannot upgrade ADM to version 5.0.0, you have to use clean installation.

Arenadata Monitoring (ADM) is a set of services that provide collection, storage, and visualization of monitoring metrics for Arenadata products. The core ADM services are:

- VictoriaMetrics — a time series database (TSDB) that is designed to collect and analyze application and infrastructure metrics.
- Grafana — a web service that allows you to create various sets of graphs and alerts for visualization of monitoring metrics.