ADQM CONTROL

Arenadata QuickMarts Control

Arenadata QuickMarts Control (ADQM Control) is an observability platform for Arenadata QuickMarts (ADQM) clusters that allows you to significantly simplify the administration and optimize the operation of ADQM databases.

Version 1.1.0 Language: EN

Contents

1. Get started
   - Online installation of ADQM Control
     - Create a cluster
     - Add services
     - Add hosts to a cluster
     - Add components
     - Configure services
     - Configure a cluster
     - Integrate with an ADQM cluster
     - Install a cluster
     - Install monitoring
   - Offline installation of ADQM Control
     - Create a cluster
     - Add services
     - Add hosts to a cluster
     - Add components
     - Configure services
     - Configure a cluster
     - Integrate with an ADQM cluster
     - Import ET settings
     - Install a cluster
     - Install monitoring

2. How to
   - Manage ADQM Control cluster
     - Cluster actions
     - Service actions
   - Use web UI
     - Alert settings
     - Manage access
     - View a dashboard
     - Cluster metrics
     - Tables
     - Queries
   - SSL encryption

3. References
   - Configuration parameters

4. Concepts
   - ADQM Control overview

5. Release notes

Get started

ADQM CONTROL

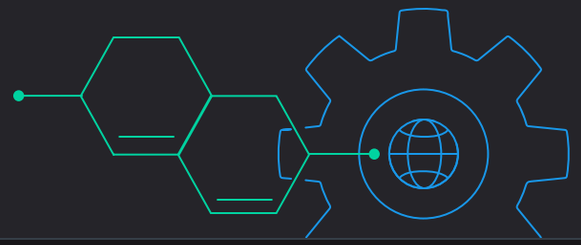
To Table of Contents

This tutorial guides you through the process of installing an Arenadata QuickMarts Control (ADQM Control) cluster. To install it, use Arenadata Cluster Manager (ADCM) — a separate Arenadata product designed for simple, convenient, and fast software deployment and exploitation.

You can install ADQM Control via ADCM in two ways.

Online installation

on hosts with access to Internet



Offline installation

on hosts without access to Internet

Online installation

ADQM CONTROL

To Table of Contents

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

Before installing ADQM Control according to the current guide, you should first install ADCM and prepare an ADQM cluster with the monitoring service. If you have not done this already, perform the steps 1-4 listed below (each of these steps is described in detail in the ADQM documentation).

Step 1.

Install ADCM

Step 2.

Prepare hosts

Step 3.

Install ADQM cluster

Step 4.

Install monitoring in ADQM

Step 5.

Install ADQM Control

Step 6.

Install monitoring in ADQM Control

Create a cluster

Elena Dvoryadkina, 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.

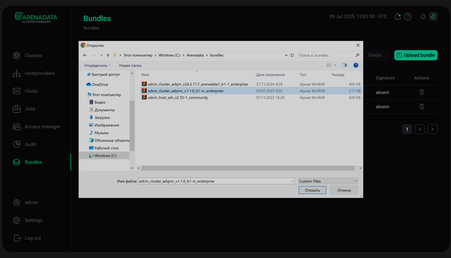
Step 1. Get the Enterprise version of a cluster bundle

Please get a cluster bundle from the Arenadata support team.

Step 2. Upload a cluster bundle to ADCM

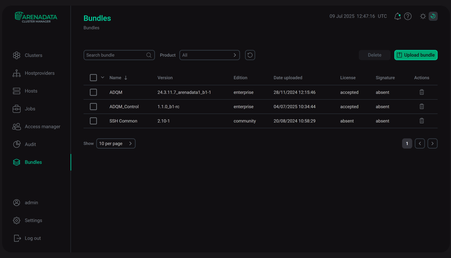
To upload a bundle to ADCM, follow the steps:

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

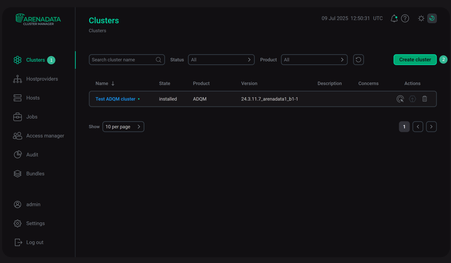
Step 3. Create a cluster based on the uploaded bundle

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.

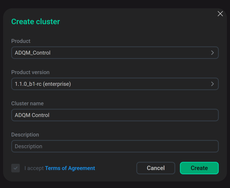
The steps for cluster creation are given below:

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



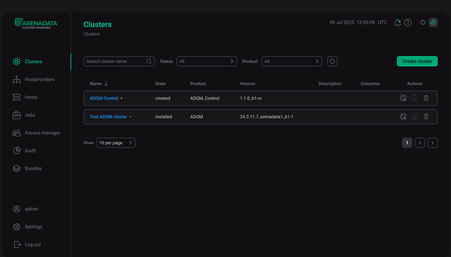
Switch to cluster creation

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.



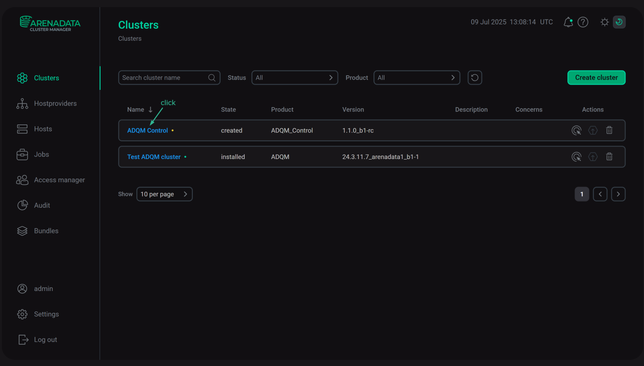
The result of successful cluster creation

Add services

Elena Dvoryadkina, Daria Barysheva

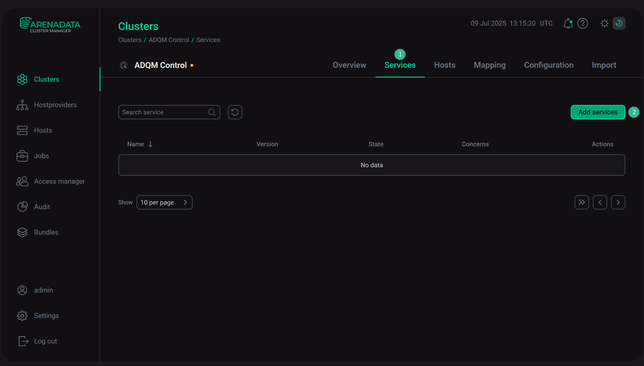
In ADCM a **service** means a software that performs some function. The steps for adding services to a cluster are listed below:

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



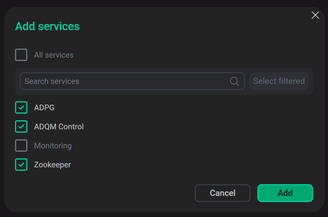
Select a cluster

2. Open the ***Services*** tab on the cluster page and click ***Add service***.



Switch to adding services

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



Select services

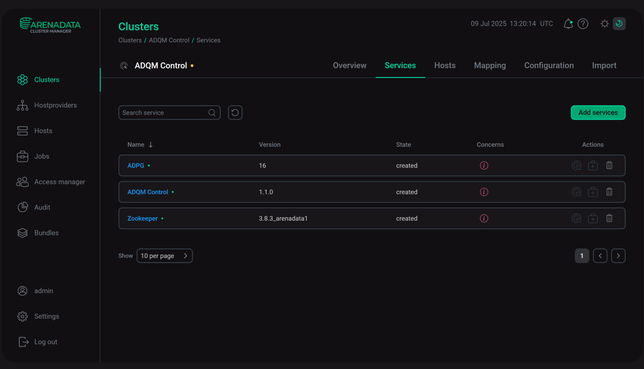
The brief description of available services is listed below.

Services that can be added to the ADQM Control cluster

Service	Mandatory	Purpose
ADQM Control	Yes	Main service that provides the full functionality of the ADQM Control product
ADPG	No	Arenadata Postgres database to store ADQM Control-specific data
Zookeeper	No	Coordination service to store ADQM Control settings specified on the user side
Monitoring	No	Service for collecting and storing ADQM Control cluster monitoring metrics based on the Prometheus monitoring system and Grafana service

For an ADQM Control cluster, it is required to select a PostgreSQL database and a ZooKeeper coordination service to store ADQM Control settings. To do this, you can add the ADPG and Zookeeper services to the ADQM Control cluster or specify an external PostgreSQL server and external ZooKeeper cluster through the corresponding settings of the ADQM Control service.

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

Elena Dvoryadkina, Daria Barysheva

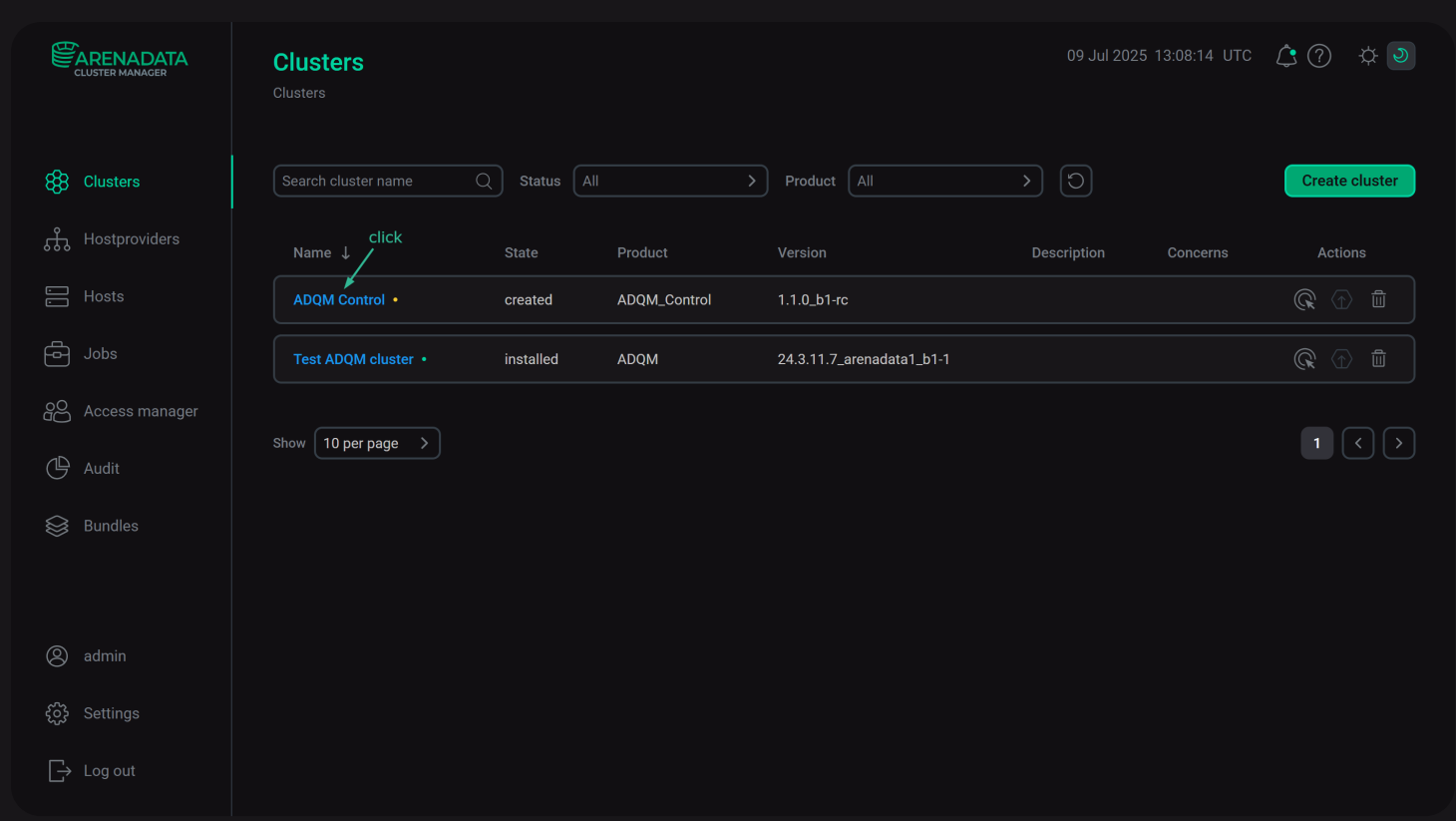
To Table of Contents

NOTE

To set up and install an ADQM Control cluster via ADCM, you should first prepare hosts for the cluster and link these hosts to ADCM. For a detailed description of how to do this, see Prepare hosts in the ADQM documentation.

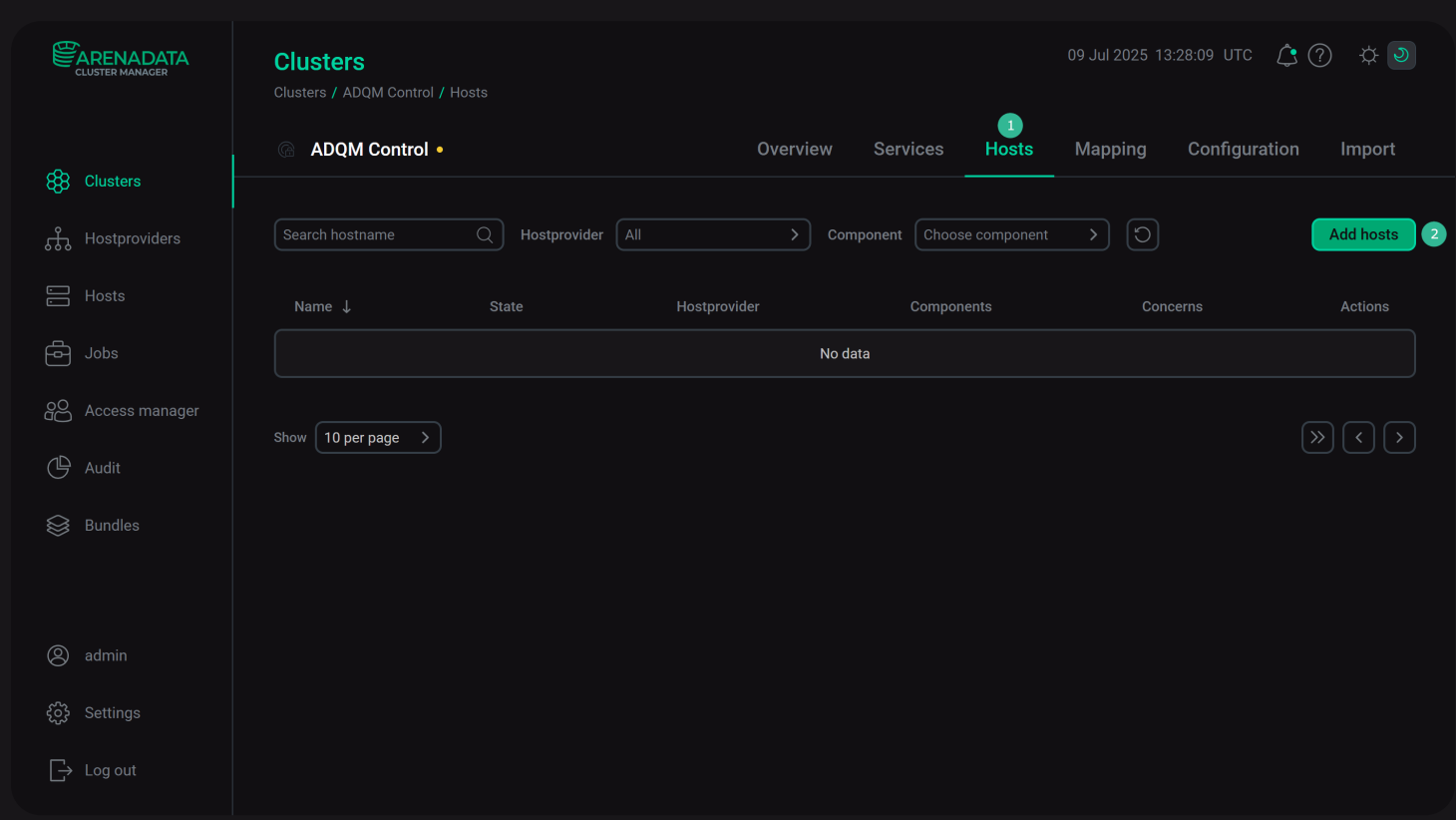
The steps for adding hosts to a cluster are listed below:

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



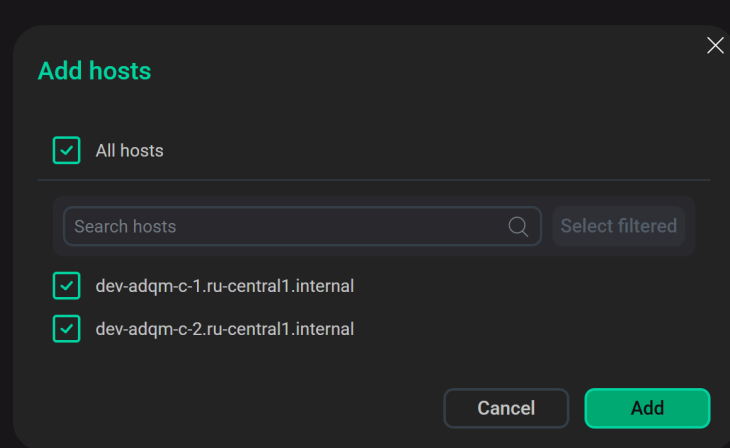
Select a cluster

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



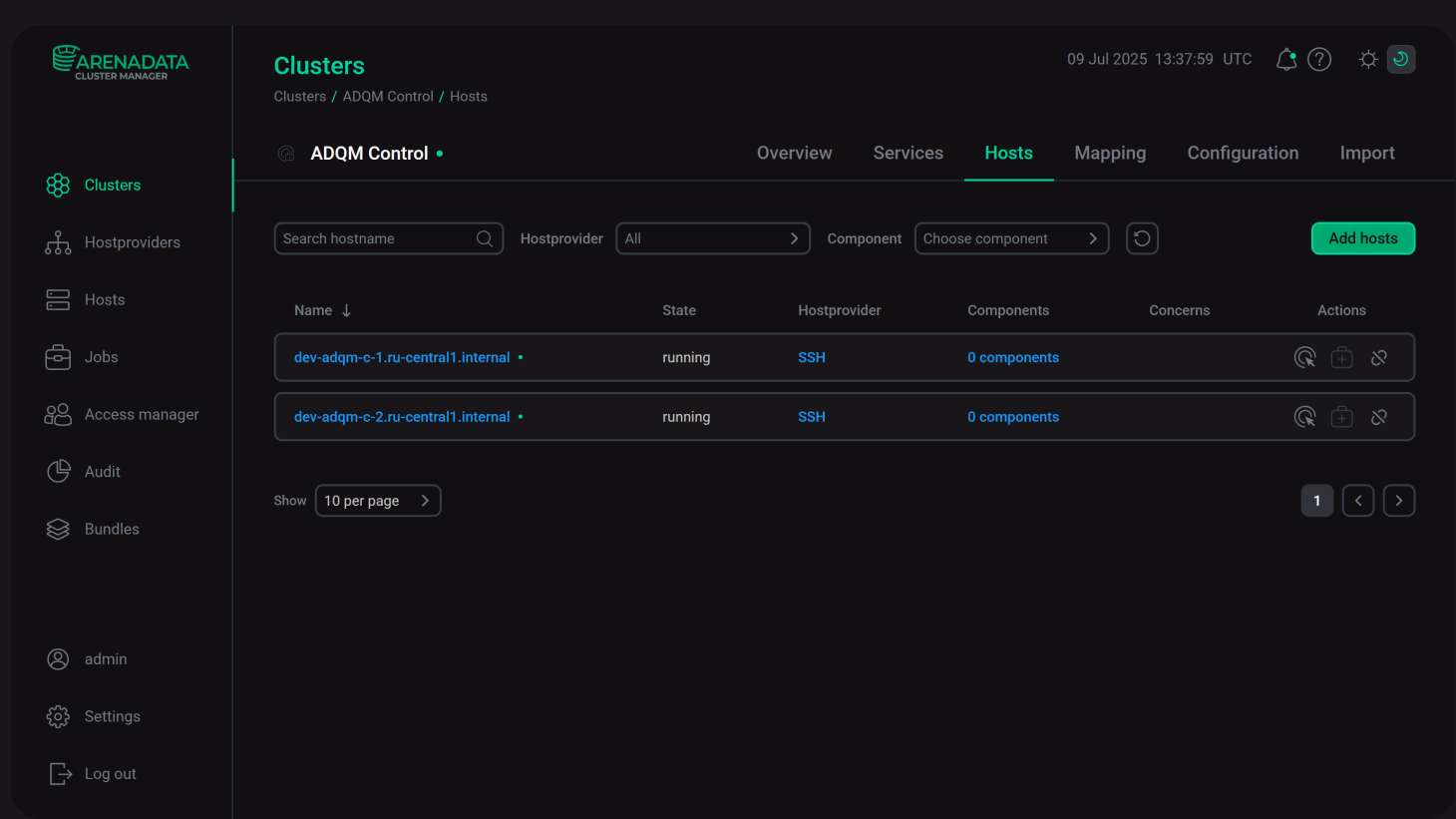
Switch to adding hosts

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

Elena Dvorzokina, Daria Barysheva

ADCM CONTROL

To Table of Contents

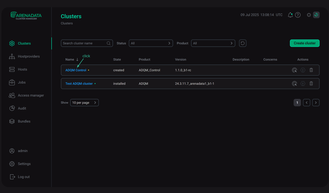
In ADCM a **component** means a part of a service that should be deployed on one or several cluster hosts.

IMPORTANT

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

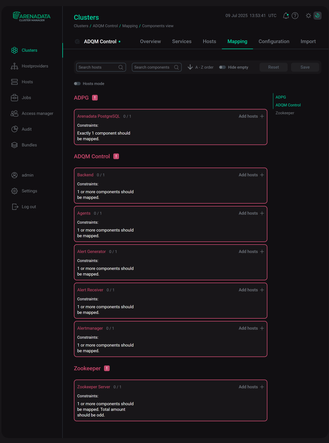
The steps for adding components to hosts are listed below:

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

All components of ADQM Control services are mandatory (highlighted in red). The ADCM UI also displays how many components should be installed.

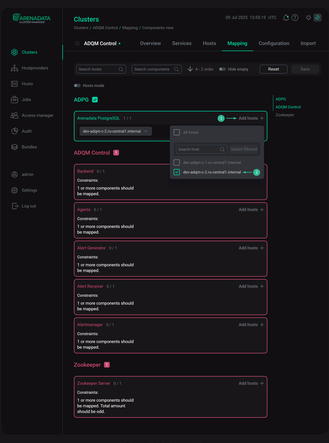
CAUTION

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

Components of ADQM Control services

Service	Component	Notes
ADQM Control	Backend	Implements the product's web interface
	Agents	Collects information on states of hosts and metrics from ADQM clusters
	Alert Generator	Generates alerts and sends them to the alert manager
	Alert Receiver	Receives alerts from the alert manager and writes them to the data storage
	Alertmanager	Handles alerts (mutes alerts of some type, filters alerts, adds additional labels) and resends processed alerts to the alert receiver
ADPG	Arenadata PostgreSQL	Data storage
Zookeeper	Zookeeper Server	Storage of user-defined ADQM Control settings

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.

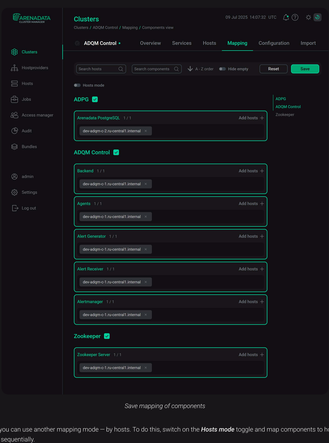


Select hosts for a component

TIP

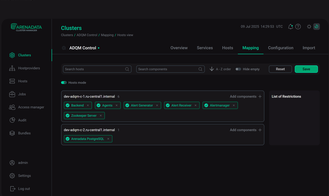
It is recommended to install the Arenadata PostgreSQL component on a dedicated host, separately from the components of the ADQM Control service.

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



Save mapping of components

Starting with ADCM 2.2.6, 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.

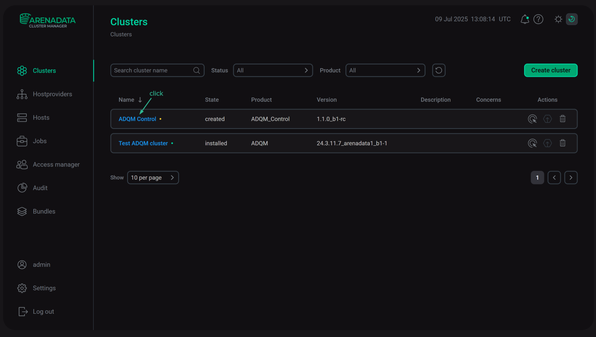


Use mapping by hosts

Configure services

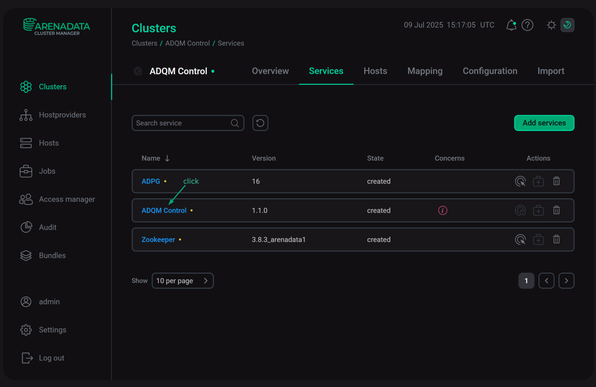
Elena Dvoryadkina, Daria Barysheva

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



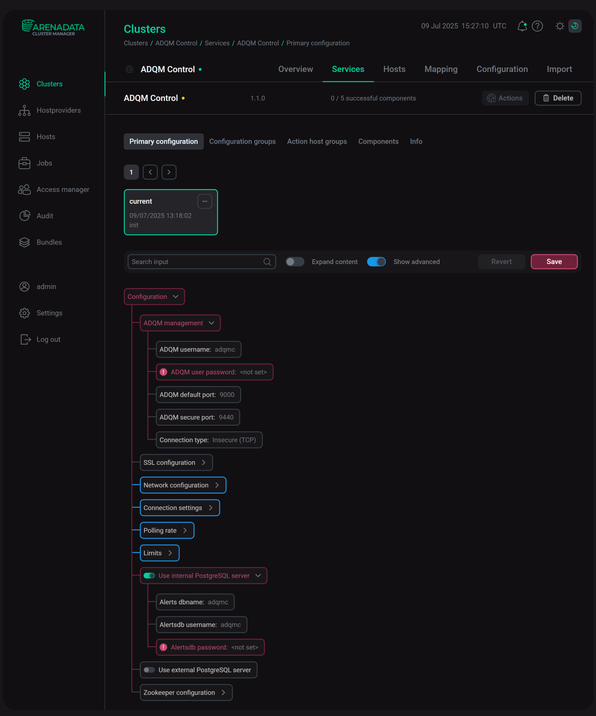
Select a cluster

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

ADQM user credentials

In the ***ADQM username*** and ***ADQM user password*** parameters of the ADQM Control service, specify the name and password of an ADQM user to be used by ADQM Control agents for connecting to ADQM. For this, you can use the `default` ADQM user account with an assigned password or create a separate user account in ADQM (recommended), for example:

```
CREATE USER adqmc ON CLUSTER default_cluster IDENTIFIED WITH sha256_password BY 'qwerty123';
```

```
GRANT ON CLUSTER default_cluster SELECT ON *.* TO adqmc;
```

For more information about all configuration parameters of ADQM Control services, see Configuration parameters.

Configure a cluster

Elena Dvoryadkina, Daria Barysheva

Contents

To Table of Contents

Primary configuration

Configuration groups

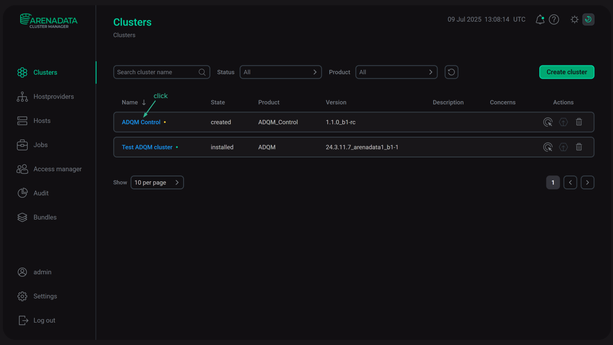
Ansible settings

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

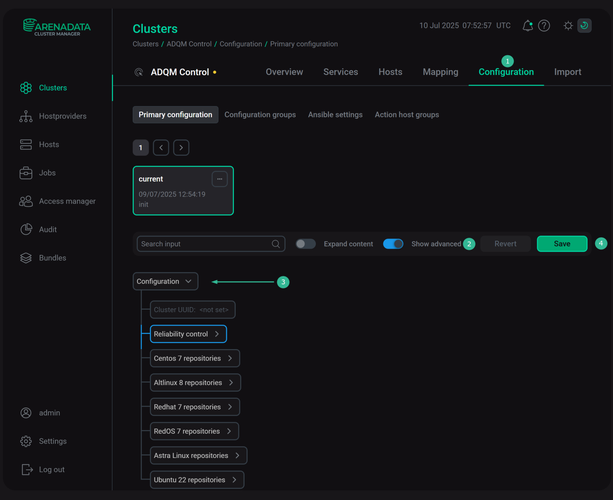
2. Open the ***Configuration*** tab on the cluster page. The ***Configuration*** tab includes the following sections: ***Primary configuration***, ***Configuration groups***, ***Ansible settings***.
3. Fill in all necessary parameters on the selected tab and click ***Save***.

Primary configuration

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

ADQM Control cluster configuration parameters are listed below:

- ***Reliability control*** — the section with the following settings (visible if the ***Show advanced*** option is enabled):
  - ***retries*** — number of retries for cluster operations (`3` by default);
  - ***delay*** — delay between retries in seconds (`10` by default);
  - ***timeout*** — timeout for checks of network ports in seconds (`60` by default).
- Paths to the repositories that are used during the installation process — depending on the cluster operating system (Centos, Alt Linux, Red Hat, RED OS, Astra Linux, or Ubuntu):
  - ***ADQMC_arenadata_community*** — repository required to install the Community edition of ADQM Control;
  - ***arenadata_postgres*** — repository required to install Arenadata PostgreSQL;
  - ***prometheus_repo*** — repository required to install Prometheus;
  - ***ADQMC_arenadata_enterprise*** — repository required to install the Enterprise edition of ADQM Control.



Cluster configuration window

Configuration groups

The ***Configuration → Configuration groups*** tab is designed to set cluster 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

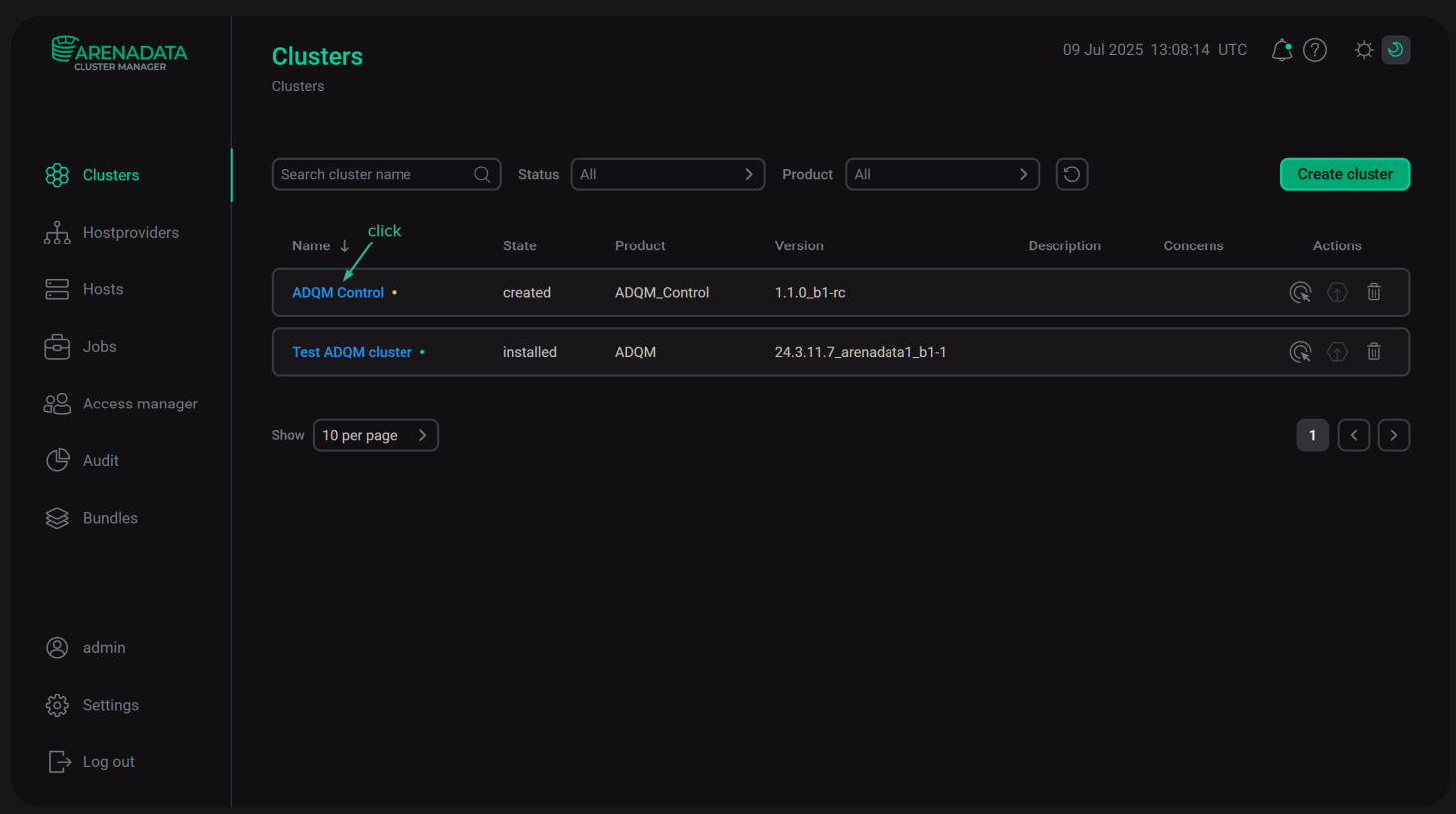
Integrate with an ADQM cluster

Elena Dvoryadkina

To Table of Contents

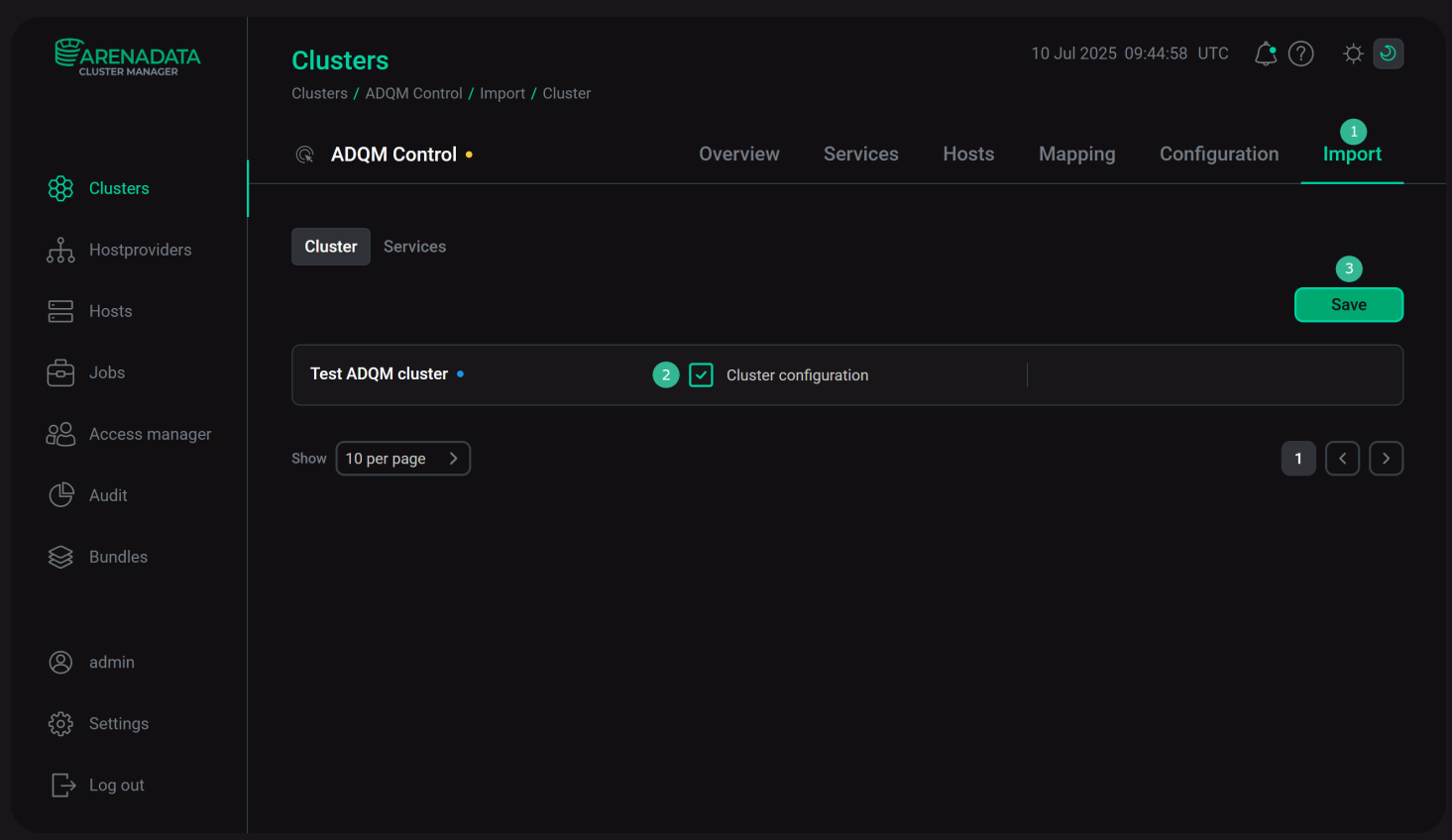
To set up an ADQM Control cluster, you should integrate it with at least one pre-installed ADQM cluster with the Monitoring service configured:

1. On the ***Clusters*** page, click an ADQM Control cluster name in the ***Name*** column.



Select a cluster

2. Open the ***Import*** tab.
3. Select ***Cluster configuration*** for the ADQM cluster to be imported into ADQM Control.
4. Click ***Save***.



Integration with an ADQM cluster

NOTE

A single ADQM Control cluster can also be integrated with multiple ADQM clusters.

Install a cluster

A. Elena Dvorzhedina, Daria Barysheva

ADQM CONTROL

Contents

To Table of Contents

Step 1. Run installation
- Whole cluster
- Single services

Step 2. View the results of installation

Step 3. Check the cluster state after installation

Step 4. Check the ADQM Control web interface

Step 1. Run 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.

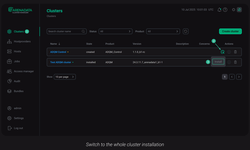
In both cases each service starts automatically after installation.

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

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

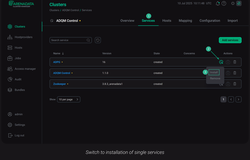
2. Wait until the job is completed.

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.



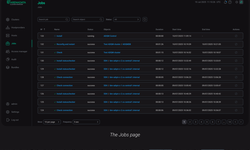
Switch to installation of single services

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

> **IMPORTANT**
>
> When you install ADQM Control services manually, one by one, use the following order:
>
> 1. ADPG, Zookeeper in any order (if connections to external PostgreSQL/ZooKeeper clusters are not configured)
> 2. ADQM Control

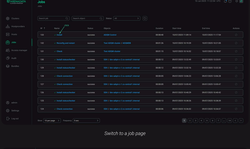
Step 2. View the results of installation

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.

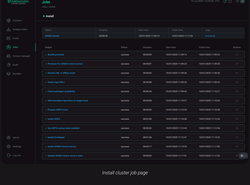


The Jobs page

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.



Switch to a job page

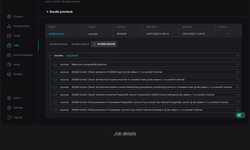


Install cluster job page

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.

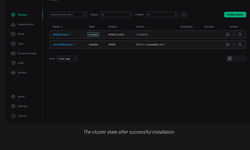


Job details

Step 3. Check the cluster state after installation

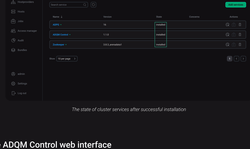
As a result of successful installation, the cluster and its services change their state according to the following rules:

- 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

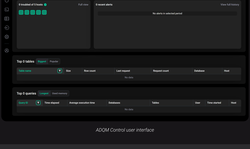
- The state of cluster services, displayed on the **Services** tab of the cluster page, is being changed from `created` to `installed`.



The state of cluster services after successful installation

Step 4. Check the ADQM Control web interface

After the installation of ADQM Control has been completed, its user interface becomes available at *http://<adqmc_backend_host>:5555* (where `<adqmc_backend_host>` is the IP address of the host with the Backend component installed and `5555` is the default value of the **Backend port** parameter in the **Network configuration** section of the ADQM Control service configuration). The `admin` user with the `admin` password and the `Owner` role (system owner) is automatically created – use these credentials to log in to ADQM Control.



ADQM Control user interface

> **IMPORTANT**
>
> - In order for ADQM Control to receive information about data tables and queries of an integrated ADQM cluster, it is required to configure the ADQMDB service parameters for the ADQM cluster as follows:
>
> - enable **query_log** in the Log settings section;
> - in the **Default user and policy settings** section, add a subnetwork or IP address of the host where the ADQM Control's Agents component is installed to the **Default user IP** list.
>
> After you have set the necessary parameters, do not forget to click **Save** and execute the **Reconfig and restart** action for the ADQMDB service to save configuration changes and restart the service.
>
> - If you have imported an ADQM cluster version-up to and including 24.3.1.1.7.1.8.1 into ADQM Control, and the Monitoring service was installed during the installation of the ADQM cluster (i.e. by the **Install** action at the cluster level, not by the service action), ADQM Control may not display the hosts of their cluster. To fix this issue, run the **Reconfigure and restart** action for the Monitoring service of the ADQM cluster. If the Monitoring service was installed into an existing ADQM cluster, this behavior does not occur – states of all ADQM cluster hosts are immediately displayed in ADQM Control.

Install monitoring

Contents

To Table of Contents

Overview
Step 1. Add the service
Step 2. Add components
Step 3. Configure the service
Step 4. Install the service
Step 5. View results
- Metrics in the Prometheus format
- Prometheus web interface
- Grafana web interface

The Monitoring service deploys the Prometheus server inside ADQM CONTROL to collect and store ADQM Control cluster monitoring metrics, and also supports the ability to use the Grafana web application for visualization and analysis of information. This article describes the steps required to add the service to an ADQM Control cluster.

Overview

When adding the Monitoring service to ADQM Control, you can install the node exporter on all or selected hosts of your cluster. It is a monitoring agent that reads host system metrics that Prometheus collects for subsequent analysis and visualization. Prometheus uses ADQM Control services (ADQM Control and Zookeeper). These metrics will be available in the Prometheus format on the ports and endpoints specified in the service configuration. You can use the Prometheus and Grafana web interfaces to view and analyze data that the monitoring service collects.

> **NOTE**
>
> - If you already have a Prometheus-compatible monitoring system set up (for example, your own Prometheus server or VictoriaMetrics), you can use it to collect ADQM Control metrics. In this case, configure access to ADQM Control metrics in your monitoring system using ADQM Control's Prometheus parameters specified on the Monitoring page.
> - You can also use the Telegraf agent for metrics in the Prometheus server deployed in ADQM Control to pass Prometheus.

Step 1. Add the service

1. In the ADQM interface, open the **Clusters** page and click your ADQM Control cluster name. On the cluster page that opens, switch to the **Services** tab and click **Add services**.

Switch to adding services

2. In the opened dialog, select the **Monitoring** service and click **Add**.

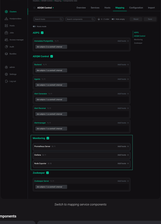
Select the service

As a result, the added service is displayed on the **Services** tab.

The result of successfully adding the service to the cluster

Step 2. Add components

1. On the cluster page, open the **Mapping** tab to proceed to mapping service components to cluster hosts.

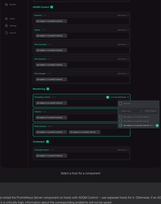


Switch to mapping service components

Monitoring service components

Component	Description
Prometheus Server	Stores and allows requesting all metrics for an ADQM Control cluster
Grafana	Visualizes ADQM Control metrics as graphs and charts organized into dashboards
Node Exporter	Collects system metrics and sends them to Prometheus

2. Assign hosts to components of the Monitoring service – click **Add hosts** and select the desired hosts in the pop-up window.



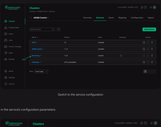
Select a host for a component

> **CAUTION**
>
> It is not recommended to install the Prometheus Server component on hosts with ADQM Control – use separate hosts for it. Otherwise, if an ADQM Control host fails, it will be hard to promptly get information about the corresponding problems or not to be current.

3. After the distribution of components is completed, click **Save**.

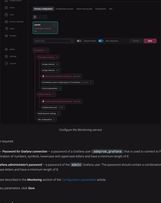
Step 3. Configure the service

1. Open the **Services** tab on the cluster page and click the **Monitoring** service name (i.e. the **Name** column).



Switch to the service configuration

2. On the page that opens, fill in the service's configuration parameters.



Configure the Monitoring service

Fields highlighted in red are required.

- **Prometheus settings** > **Password for Grafana connection** – a password of a Grafana user (`admin_grafana`) that is used to connect to Prometheus. The password must be specified in the Grafana administrator's settings and approximately follows a minimum length of 8.
- **Grafana settings** > **Grafana administrator's password** – a password of the `admin` Grafana user. The password should contain a combination of numbers, symbols, and both lowercase and uppercase letters and approximately follows a minimum length of 8.

All monitoring parameters are described in the **Monitoring** section of the Configuration parameters article.

3. After specifying all necessary parameters, click **Save**.

Step 4. Install the service

1. On the **Services** tab, click the icon ( ) for the Monitoring service in the **Actions** column and run the **Install** action.

Switch to the service installation

2. Wait until the installation is completed. Then, check that the service code has changed from `installed` → `installation`.

Installation is complete

To view the service installation process and analyze errors if they occur, select **Jobs** in the left navigation menu, and click the **Install** job name in the **Jobs** list.

Install service job page

Step 5. View results

The Monitoring service starts automatically after installation. To ensure that monitoring works correctly, check that both system metrics and metrics of ADQM Control's services are collected from the cluster hosts. To do this, view metrics in the Prometheus format in the browser or use the Prometheus or Grafana web interface.

Metrics in the Prometheus format

1. In the address bar of your browser, enter an address of an ADQM Control host with the specified port and endpoint to view the service or system metrics. Port numbers and endpoints are defined by the Prometheus settings.
   - In the **Network configuration** section of the ADQM Control service configuration – ports for access to metrics of the ADQM Control services components (the endpoint is always `/metrics`).
   - In the **Monitoring** section of the Zookeeper service configuration – port for access to metrics of the Zookeeper component (the endpoint is `/metrics`).
   - In the **Prometheus settings** section – port for access to metrics of the Prometheus component and its endpoint (the endpoint is always `/metrics`).

   For example, http://10.92.40.100:9363/metrics is an address to view metrics of the clickhouse-server component on a host with the IP 10.92.40.100.

2. The page that opens will display monitoring metrics from the specified host of the ADQM Control cluster in the Prometheus format.

```
# HELP ClickHouse Version and revision
# TYPE ClickHouse Version and revision gauge
ClickHouse Version and revision{version="22.8.8.3",revision="54460"} 1
# HELP ClickHouse Error*
# TYPE ClickHouse Error* gauge
ClickHouse Error*{name="ACCESS_DENIED"} 0
ClickHouse Error*{name="AIO_READ_FAILED"} 0
ClickHouse Error*{name="AIO_SUBMIT_FAILED"} 0
# HELP ClickHouse Status*
# TYPE ClickHouse Status* gauge
ClickHouse Status*{name="StorageBufferFlushThreadsActiveCopy"} 0
ClickHouse Status*{name="StorageBufferFlushThreadsScheduled"} 0
ClickHouse Status*{name="StorageBufferFlushThreadsWaitingToBackOff"} 0
ClickHouse Status*{name="StorageDistributedThreadsActiveCopy"} 0
ClickHouse Status*{name="StorageDistributedThreadsScheduled"} 0
ClickHouse Status*{name="StorageDistributedThreadsWaitingToBackOff"} 0
ClickHouse Status*{name="TCPConnection"} 1
ClickHouse Status*{name="TCPConnectionForClickHouseDisks"} 0
ClickHouse Status*{name="TCPConnectionForClickHouseDiscs"} 0
ClickHouse Status*{name="TCPConnectionForHDFS"} 0
ClickHouse Status*{name="TCPConnectionForJDBC"} 0
ClickHouse Status*{name="TCPConnectionForKafka"} 0
ClickHouse Status*{name="TCPConnectionForURL"} 0
ClickHouse Status*{name="TemporaryFileForAggregation"} 0
ClickHouse Status*{name="TemporaryFileForJoin"} 0
ClickHouse Status*{name="TemporaryFileForSort"} 0
ClickHouse Status*{name="TemporaryFileInWritePolicy"} 0
ClickHouse Status*{name="TemporaryFileOfSort"} 0
ClickHouse Status*{name="VersionInteger"} 22080803
# HELP ClickHouse Profile*
# TYPE ClickHouse Profile* gauge
ClickHouse Profile*{name="ArenaAllocBytes"} 0
ClickHouse Profile*{name="ArenaAllocChunks"} 0
ClickHouse Profile*{name="CachedReadBufferCacheWriteBytes"} 0
ClickHouse Profile*{name="CachedReadBufferReadFromCacheBytes"} 0
ClickHouse Profile*{name="CachedReadBufferReadFromSourceBytes"} 0
ClickHouse Profile*{name="CachedReadBufferReadFromSourceMicroseconds"} 0
ClickHouse Profile*{name="CachedReadBufferReadFromSourceNum"} 0
ClickHouse Profile*{name="CachedReadBufferCacheWriteMicroseconds"} 0
ClickHouse Profile*{name="CachedReadBufferCacheWriteNum"} 0
ClickHouse Profile*{name="CachedWriteBufferCacheWriteBytes"} 0
ClickHouse Profile*{name="CachedWriteBufferCacheWriteMicroseconds"} 0
ClickHouse Profile*{name="CachedWriteBufferCacheWriteNum"} 0
ClickHouse Profile*{name="CompressedReadBufferBlocks"} 0
ClickHouse Profile*{name="CompressedReadBufferBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferChecksumMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferDecompressedBlocks"} 0
ClickHouse Profile*{name="CompressedReadBufferDecompressedBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferDecompressionMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferDecompressionNum"} 0
ClickHouse Profile*{name="CompressedReadBufferMappedBlocks"} 0
ClickHouse Profile*{name="CompressedReadBufferMappedBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferMappedMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferMappedNum"} 0
ClickHouse Profile*{name="CompressedReadBufferOfDecompressedBlocksBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferOfDecompressedBlocksMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferOfDecompressedBlocksNum"} 0
ClickHouse Profile*{name="CompressedReadBufferOfMappedBlocksBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferOfMappedBlocksMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferOfMappedBlocksNum"} 0
ClickHouse Profile*{name="CompressedReadBufferOfPrefetchedBlocksBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferOfPrefetchedBlocksMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferOfPrefetchedBlocksNum"} 0
ClickHouse Profile*{name="CompressedReadBufferPrefetchedBlocks"} 0
ClickHouse Profile*{name="CompressedReadBufferPrefetchedBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferPrefetchedMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferPrefetchedNum"} 0
ClickHouse Profile*{name="CompressedReadBufferUncompressedBlocks"} 0
ClickHouse Profile*{name="CompressedReadBufferUncompressedBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferUncompressedMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferUncompressedNum"} 0
ClickHouse Profile*{name="CompressedWriteBufferBlocks"} 0
ClickHouse Profile*{name="CompressedWriteBufferBytes"} 0
ClickHouse Profile*{name="CompressedWriteBufferCompressionMicroseconds"} 0
ClickHouse Profile*{name="CompressedWriteBufferCompressionNum"} 0
ClickHouse Profile*{name="CompressedWriteBufferUncompressedBytes"} 0
ClickHouse Profile*{name="ContextLock"} 0
ClickHouse Profile*{name="CreatedHTTPConnections"} 0
ClickHouse Profile*{name="CreatedLogEntryForMerge"} 0
ClickHouse Profile*{name="CreatedLogEntryForMutation"} 0
ClickHouse Profile*{name="CreatedReadBufferAIO"} 0
ClickHouse Profile*{name="CreatedReadBufferAIOFailed"} 0
ClickHouse Profile*{name="CreatedReadBufferDirectAIO"} 0
ClickHouse Profile*{name="CreatedReadBufferDirectAIOFailed"} 0
ClickHouse Profile*{name="CreatedReadBufferMMap"} 0
ClickHouse Profile*{name="CreatedReadBufferMMapFailed"} 0
ClickHouse Profile*{name="CreatedReadBufferOrdinary"} 0
ClickHouse Profile*{name="CreatedWriteBufferAIO"} 0
ClickHouse Profile*{name="CreatedWriteBufferAIOFailed"} 0
ClickHouse Profile*{name="CreatedWriteBufferDirectAIO"} 0
ClickHouse Profile*{name="CreatedWriteBufferDirectAIOFailed"} 0
ClickHouse Profile*{name="CreatedWriteBufferOrdinary"} 0
ClickHouse Profile*{name="DelayedInsertsMilliseconds"} 0
ClickHouse Profile*{name="DelayedInsertsToBackgroundPoolErrors"} 0
ClickHouse Profile*{name="DelayedMutations"} 0
ClickHouse Profile*{name="DelayedMutationsMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrors"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytes"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytes"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytes"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytes"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytes"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytes"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytes
```

Offline installation

To Table of Contents

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

Before installing ADQM Control according to the current guide, you should first install ADCM, deploy an ET cluster, and prepare an ADQM cluster with the monitoring service. If you have not done this already, perform the steps 1-5 listed below (each of these steps is described in detail in the ADQM documentation). Also, upload a special ADQM Control installation shell script to the ET cluster (you can request it from the Arenadata support team, as well as a shell script for offline installation of ADQM) — see Upload packages for offline installation.

Step 1.

Install ADCM

Step 2.

Prepare hosts

Step 3.

Install ET cluster

Step 4.

Install ADQM cluster

Step 5.

Install monitoring in ADQM

Step 6.

Install ADQM Control

Step 7.

Install monitoring in ADQM Control

Create a cluster

Elena Dvoryadkina, 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.

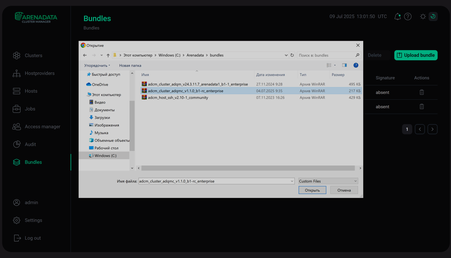
Step 1. Get the Enterprise version of a cluster bundle

Please get a cluster bundle from the Arenadata support team.

Step 2. Upload a cluster bundle to ADCM

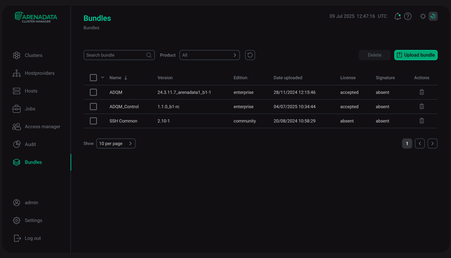
To upload a bundle to ADCM, follow the steps:

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

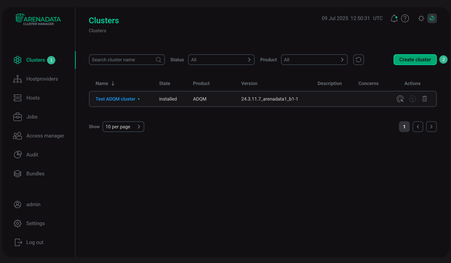
Step 3. Create a cluster based on the uploaded bundle

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.

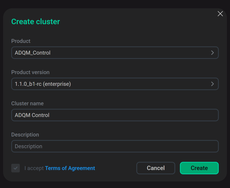
The steps for cluster creation are given below:

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



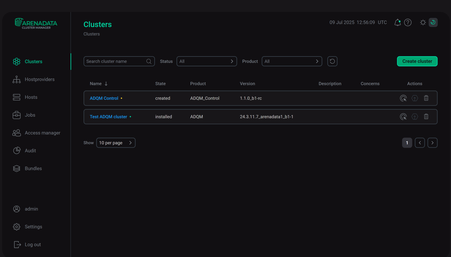
Switch to cluster creation

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.



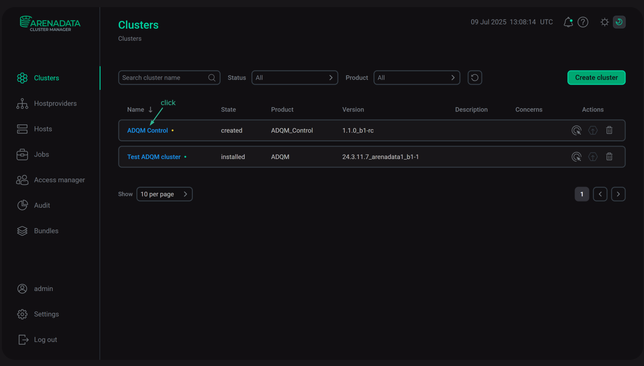
The result of successful cluster creation

Add services

Elena Dvoryadkina, Daria Barysheva

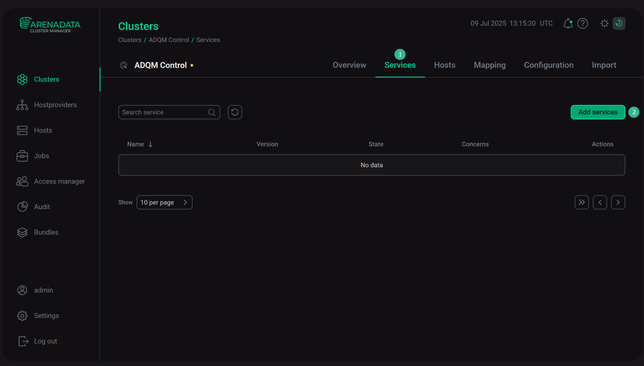
In ADCM a **service** means a software that performs some function. The steps for adding services to a cluster are listed below:

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



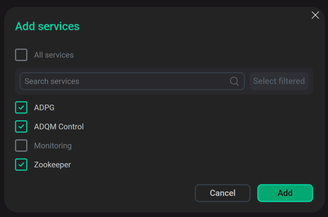
Select a cluster

2. Open the ***Services*** tab on the cluster page and click ***Add service***.



Switch to adding services

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



Select services

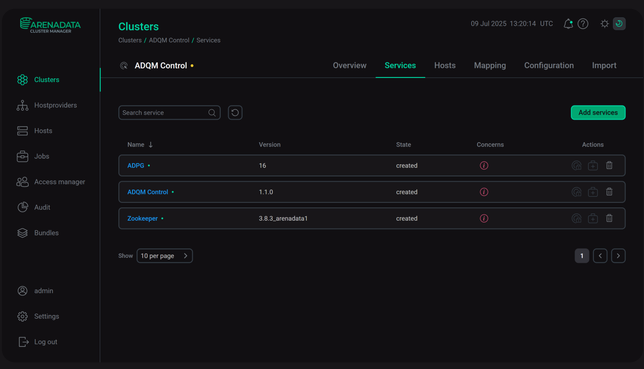
The brief description of available services is listed below.

Services that can be added to the ADQM Control cluster

Service	Mandatory	Purpose
ADQM Control	Yes	Main service that provides the full functionality of the ADQM Control product
ADPG	No	Arenadata Postgres database to store ADQM Control-specific data
Zookeeper	No	Coordination service to store ADQM Control settings specified on the user side
Monitoring	No	Service for collecting and storing ADQM Control cluster monitoring metrics based on the Prometheus monitoring system and Grafana service

For an ADQM Control cluster, it is required to select a PostgreSQL database and a ZooKeeper coordination service to store ADQM Control settings. To do this, you can add the ADPG and Zookeeper services to the ADQM Control cluster or specify an external PostgreSQL server and external ZooKeeper cluster through the corresponding settings of the ADQM Control service.

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

Elena Dvoryadkina, Daria Barysheva

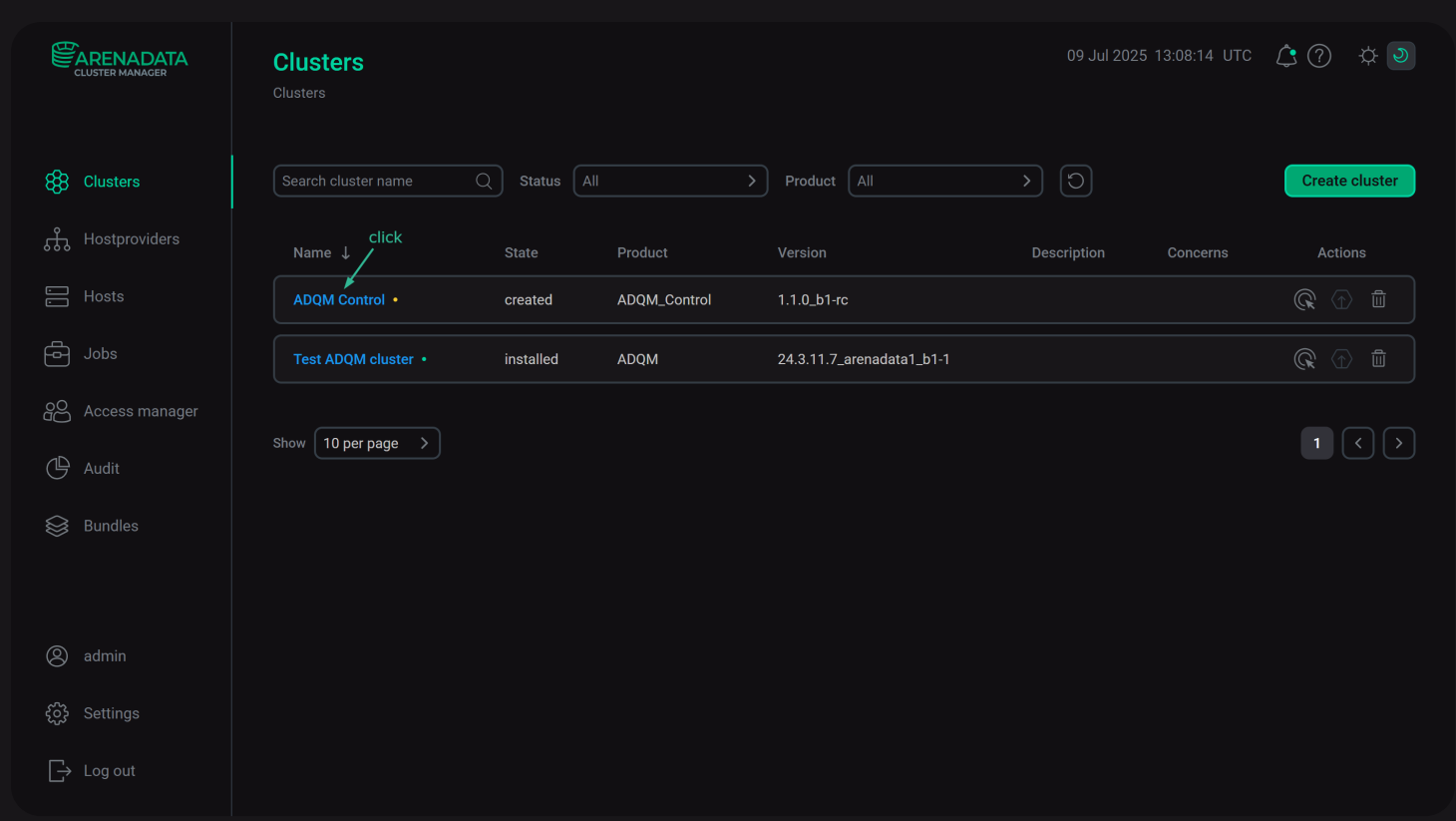
To Table of Contents

NOTE

To set up and install an ADQM Control cluster via ADCM, you should first prepare hosts for the cluster and link these hosts to ADCM. For a detailed description of how to do this, see Prepare hosts in the ADQM documentation.

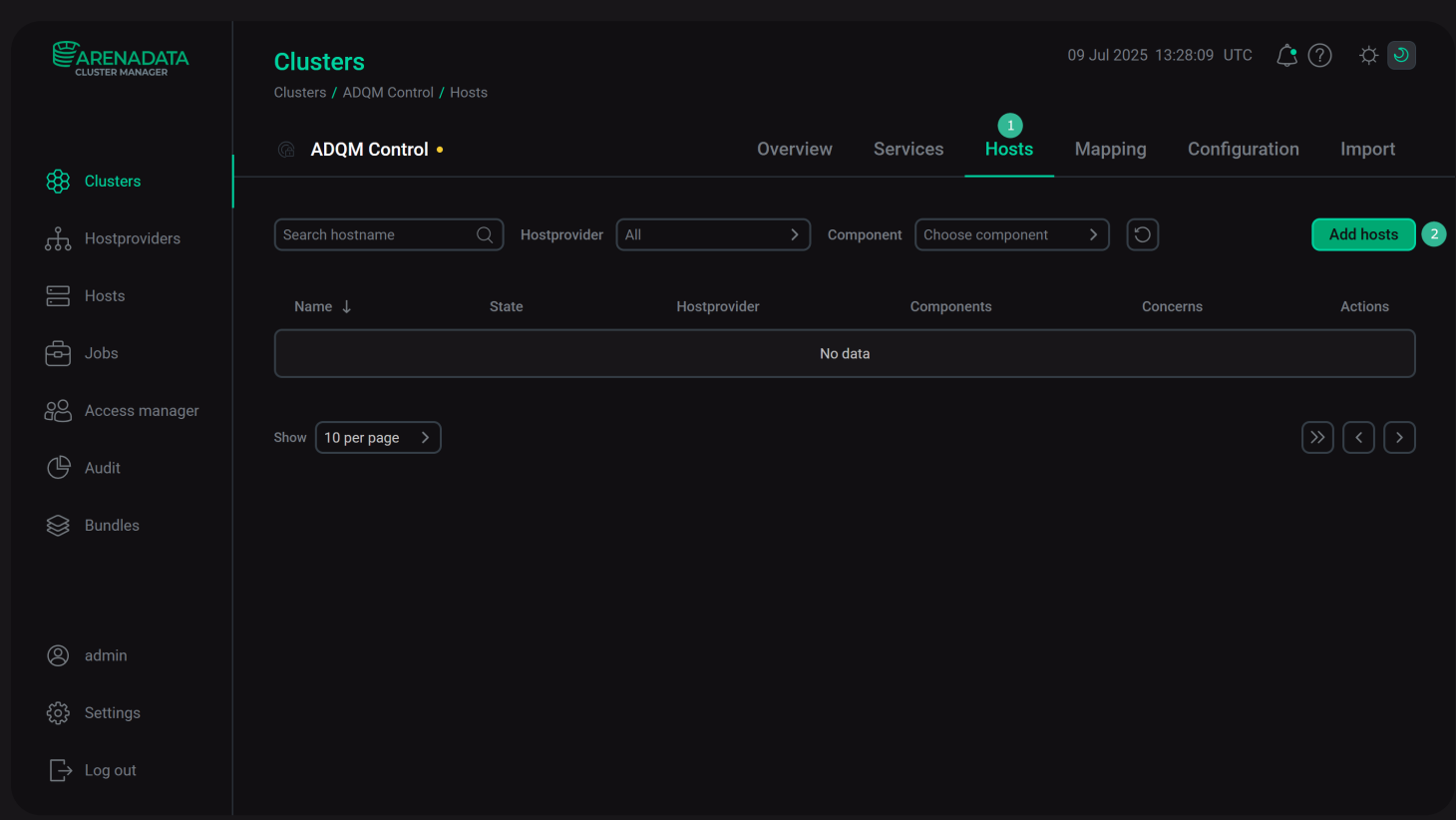
The steps for adding hosts to a cluster are listed below:

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



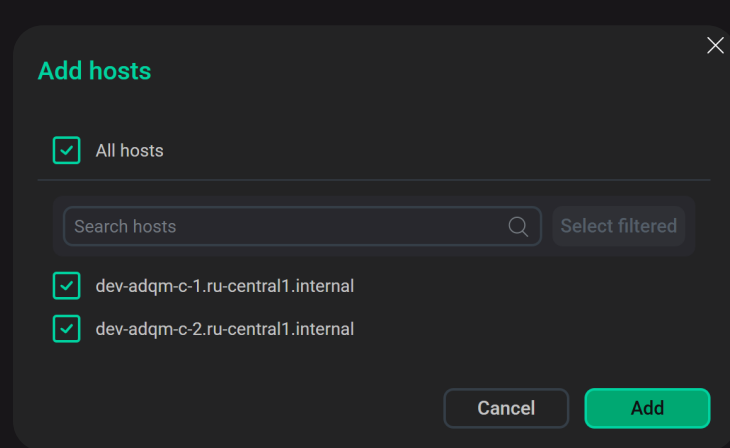
Select a cluster

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



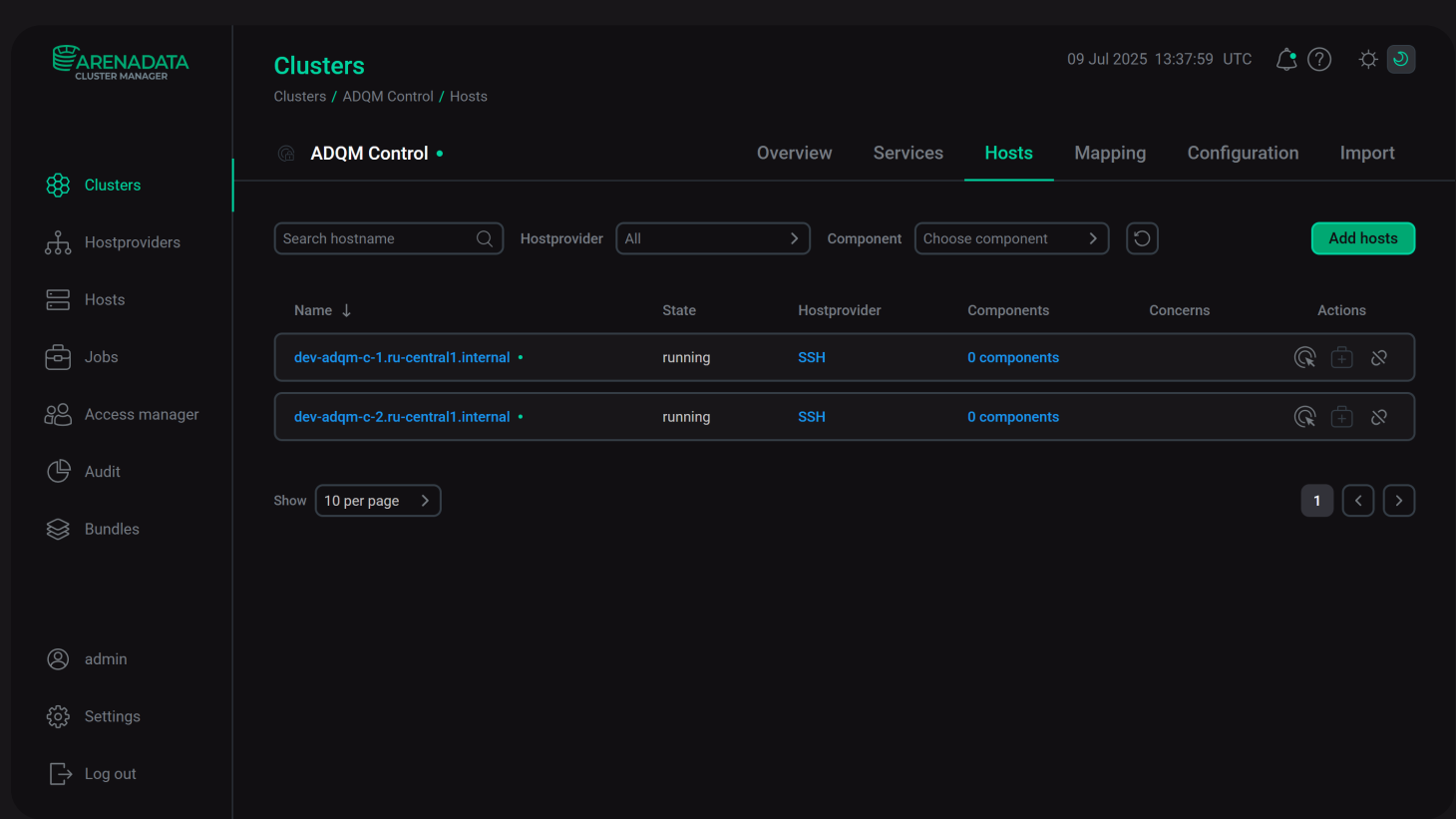
Switch to adding hosts

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

Elena Dvorzokina, Daria Barysheva

ADCM CONTROL

To Table of Contents

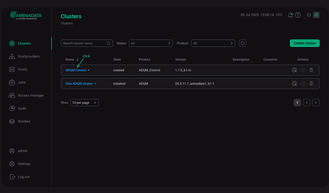
In ADCM a **component** means a part of a service that should be deployed on one or several cluster hosts.

IMPORTANT

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

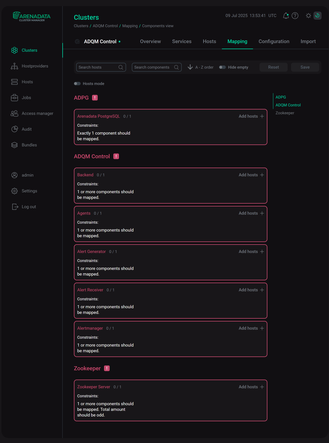
The steps for adding components to hosts are listed below:

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

All components of ADQM Control services are mandatory (highlighted in red). The ADCM UI also displays how many components should be installed.

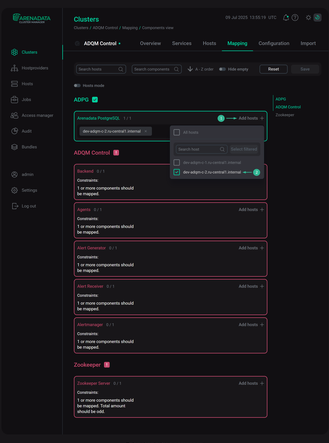
CAUTION

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

Components of ADQM Control services

Service	Component	Notes
ADQM Control	Backend	Implements the product's web interface
	Agents	Collects information on states of hosts and metrics from ADQM clusters
	Alert Generator	Generates alerts and sends them to the alert manager
	Alert Receiver	Receives alerts from the alert manager and writes them to the data storage
	Alertmanager	Handles alerts (mutes alerts of some type, filters alerts, adds additional labels) and resends processed alerts to the alert receiver
ADPG	Arenadata PostgreSQL	Data storage
Zookeeper	Zookeeper Server	Storage of user-defined ADQM Control settings

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.

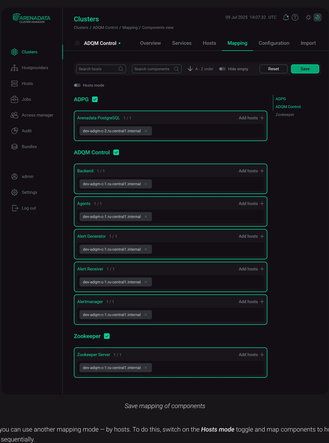


Select hosts for a component

TIP

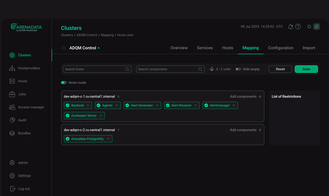
It is recommended to install the Arenadata PostgreSQL component on a dedicated host, separately from the components of the ADQM Control service.

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



Save mapping of components

Starting with ADCM 2.2.6, 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.

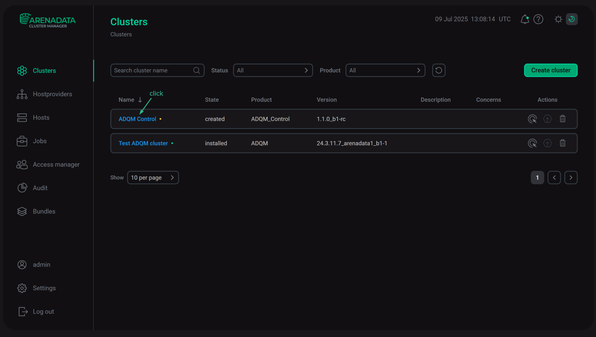


Use mapping by hosts

Configure services

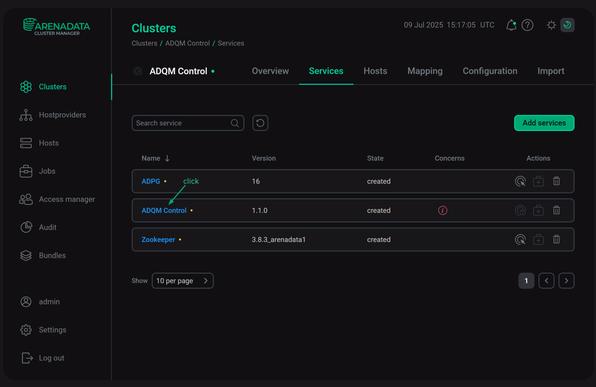
Elena Dvoryadkina, Daria Barysheva

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



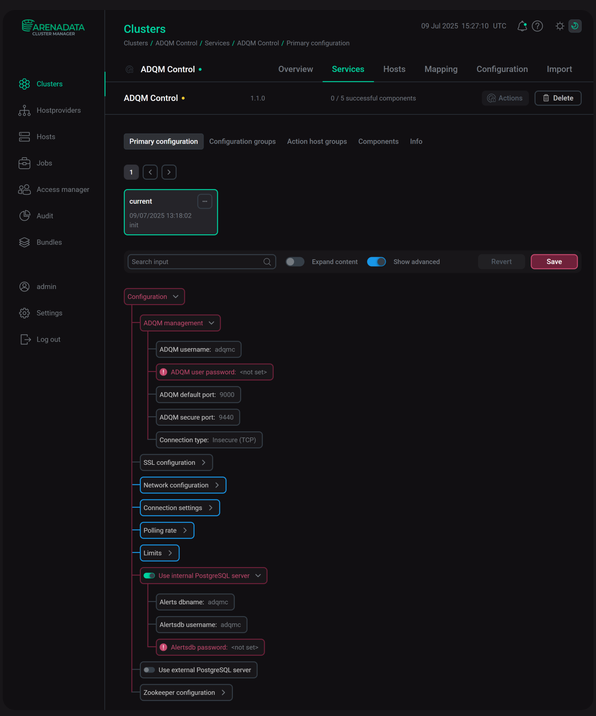
Select a cluster

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

ADQM user credentials

In the ***ADQM username*** and ***ADQM user password*** parameters of the ADQM Control service, specify the name and password of an ADQM user to be used by ADQM Control agents for connecting to ADQM. For this, you can use the `default` ADQM user account with an assigned password or create a separate user account in ADQM (recommended), for example:

```
CREATE USER adqmc ON CLUSTER default_cluster IDENTIFIED WITH sha256_password BY 'qwerty123';
```

```
GRANT ON CLUSTER default_cluster SELECT ON *.* TO adqmc;
```

For more information about all configuration parameters of ADQM Control services, see Configuration parameters.

Configure a cluster

Elena Dvoryadkina, Daria Barysheva

Contents

To Table of Contents

Primary configuration

Configuration groups

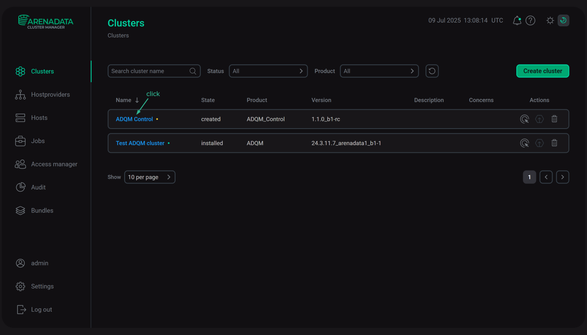
Ansible settings

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

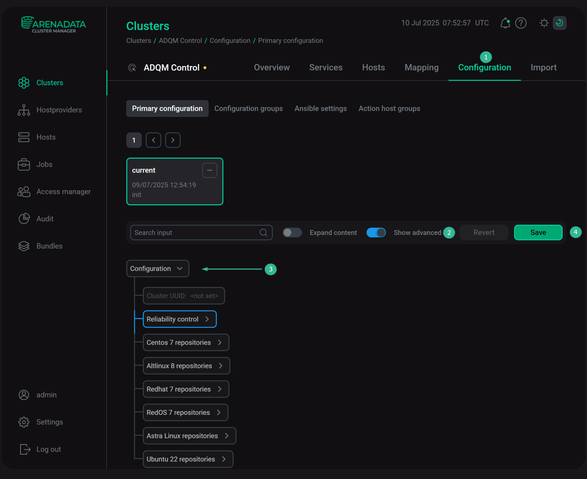
2. Open the ***Configuration*** tab on the cluster page. The ***Configuration*** tab includes the following sections: ***Primary configuration***, ***Configuration groups***, ***Ansible settings***.
3. Fill in all necessary parameters on the selected tab and click ***Save***.

Primary configuration

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

ADQM Control cluster configuration parameters are listed below:

- ***Reliability control*** — the section with the following settings (visible if the ***Show advanced*** option is enabled):
  - ***retries*** — number of retries for cluster operations (`3` by default);
  - ***delay*** — delay between retries in seconds (`10` by default);
  - ***timeout*** — timeout for checks of network ports in seconds (`60` by default).
- Paths to the repositories that are used during the installation process — depending on the cluster operating system (Centos, Alt Linux, Red Hat, RED OS, Astra Linux, or Ubuntu):
  - ***ADQMC_arenadata_community*** — repository required to install the Community edition of ADQM Control;
  - ***arenadata_postgres*** — repository required to install Arenadata PostgreSQL;
  - ***prometheus_repo*** — repository required to install Prometheus;
  - ***ADQMC_arenadata_enterprise*** — repository required to install the Enterprise edition of ADQM Control.



Cluster configuration window

Configuration groups

The ***Configuration*** → ***Configuration groups*** tab is designed to set cluster 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

NOTE

During offline installation, these parameters are ignored. The data stored locally in the Enterprise Tools cluster is used instead.

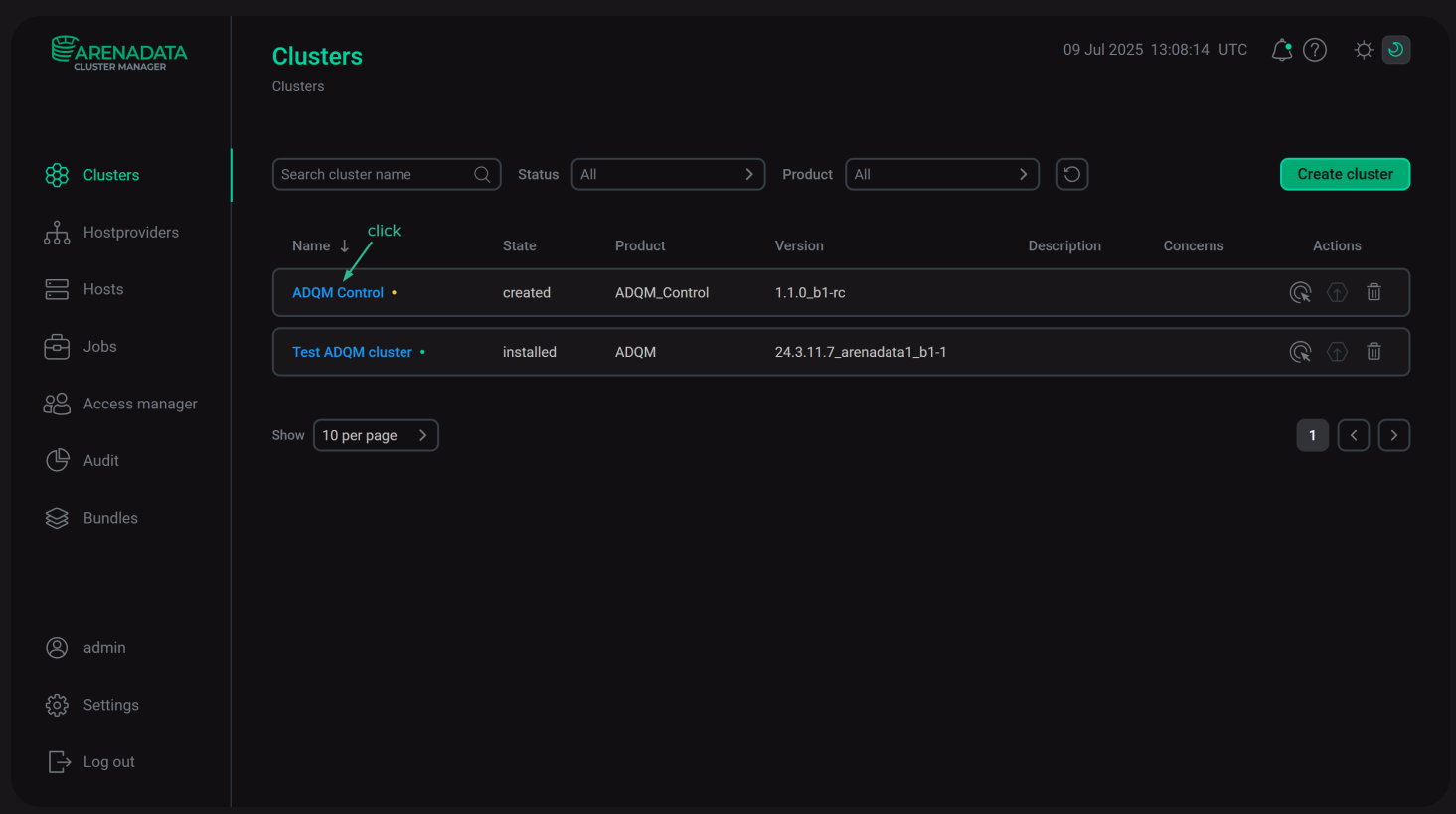
Integrate with an ADQM cluster

Elena Dvoryadkina

To Table of Contents

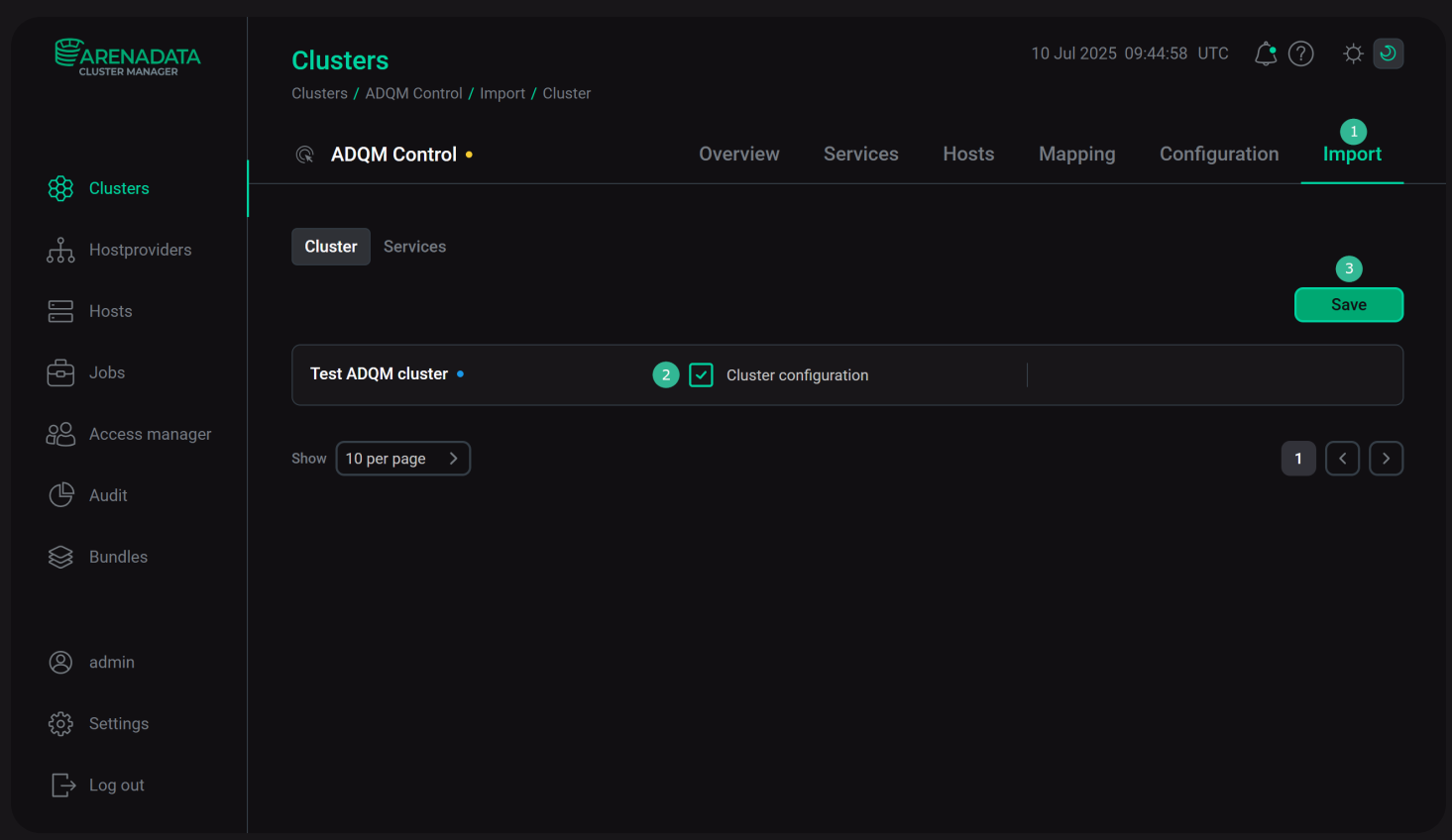
To set up an ADQM Control cluster, you should integrate it with at least one pre-installed ADQM cluster with the Monitoring service configured:

1. On the ***Clusters*** page, click an ADQM Control cluster name in the ***Name*** column.



Select a cluster

2. Open the ***Import*** tab.
3. Select ***Cluster configuration*** for the ADQM cluster to be imported into ADQM Control.
4. Click ***Save***.



Integration with an ADQM cluster

NOTE

A single ADQM Control cluster can also be integrated with multiple ADQM clusters.

Import ET settings

Daria Barysheva, Elena Dvoryadkina

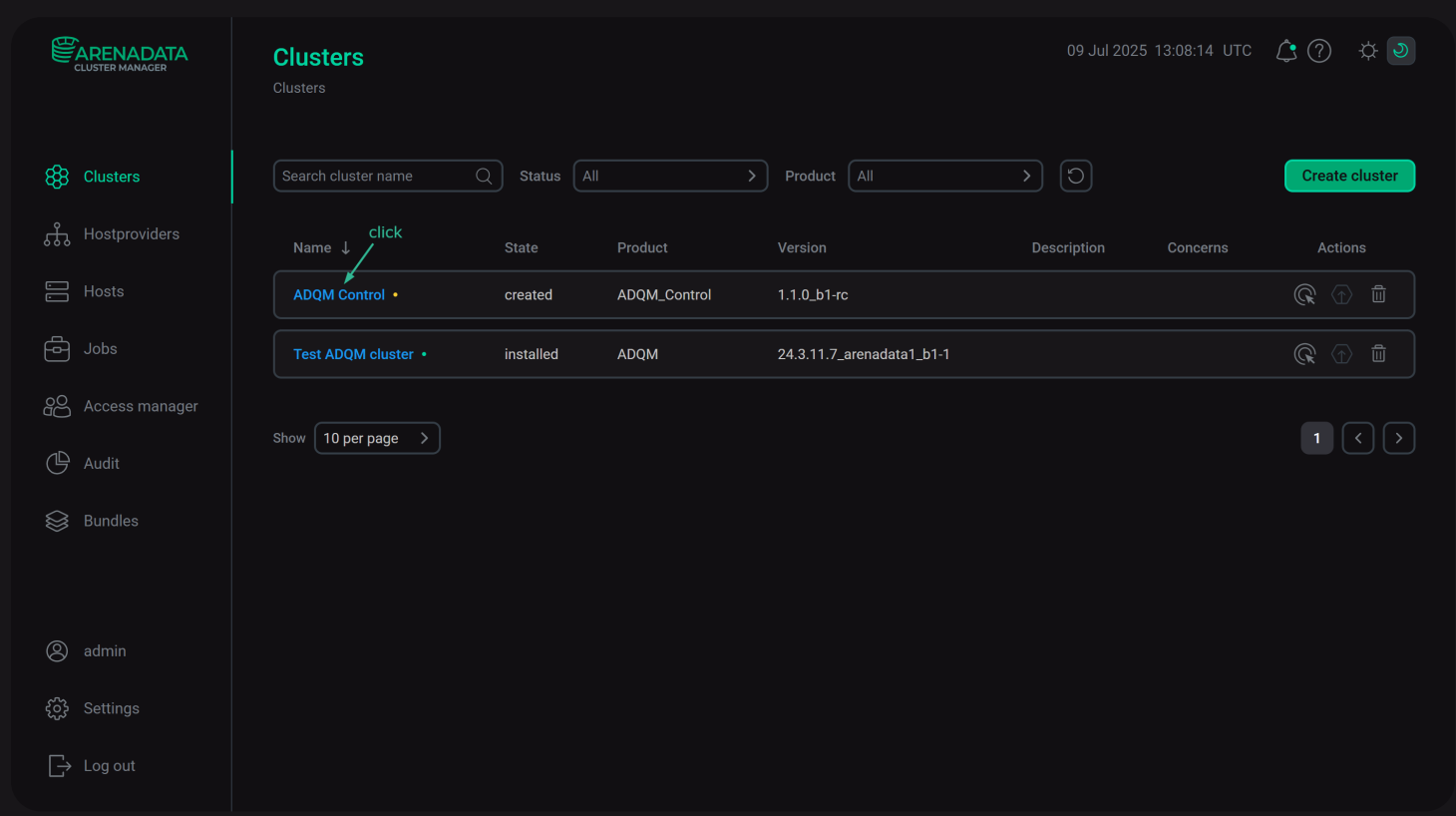
To Table of Contents

IMPORTANT

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

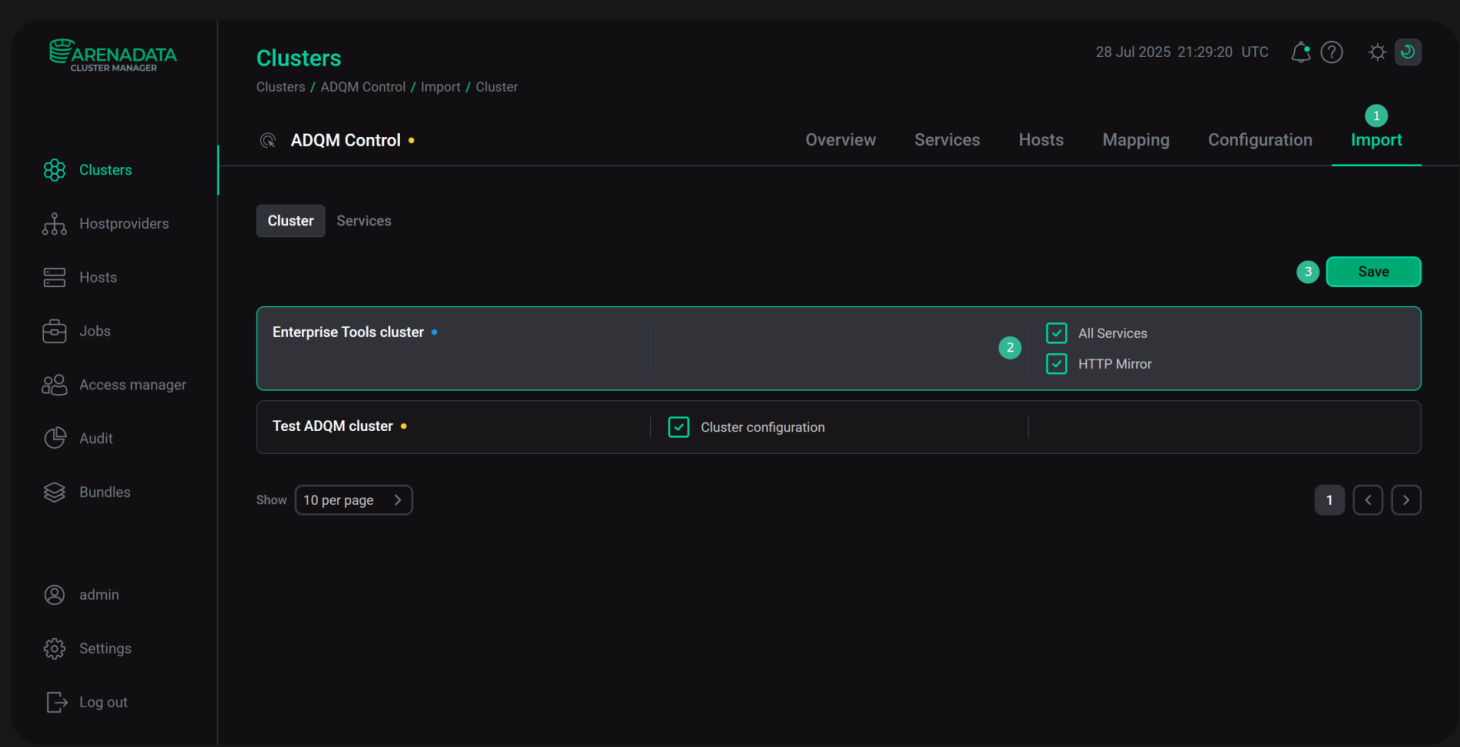
In order to determine that a new cluster should be installed offline, it is necessary to integrate it with the previously created Enterprise Tools (ET) cluster. Follow the steps:

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 on the cluster page.
3. Select the ***HTTP Mirror*** service of the previously created ET cluster.
4. Click ***Save***.



Integration with the 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

A. Elena Dvorzhedina, Daria Barysheva

ADQM CONTROL

Contents

To Table of Contents

Step 1. Run installation
- Whole cluster
- Single services

Step 2. View the results of installation

Step 3. Check the cluster state after installation

Step 4. Check the ADQM Control web interface

Step 1. Run 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.

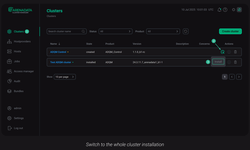
In both cases each service starts automatically after installation.

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

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

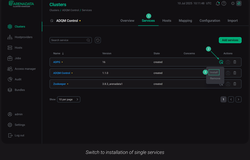
2. Wait until the job is completed.

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.



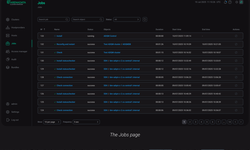
Switch to installation of single services

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

> **IMPORTANT**
>
> When you install ADQM Control services manually, one by one, use the following order:
>
> 1. ADPG, Zookeeper in any order (if connections to external PostgreSQL/ZooKeeper clusters are not configured)
> 2. ADQM Control

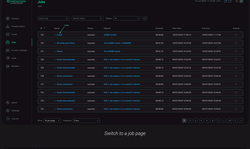
Step 2. View the results of installation

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.

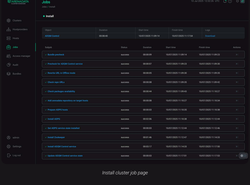


The Jobs page

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.



Switch to a job page

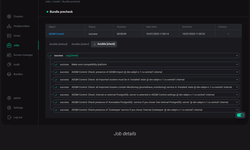


Install cluster job page

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.

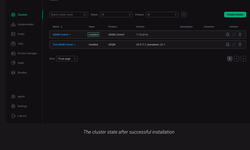


Job details

Step 3. Check the cluster state after installation

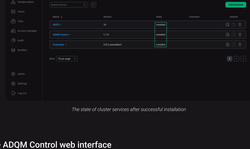
As a result of successful installation, the cluster and its services change their state according to the following rules:

- 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

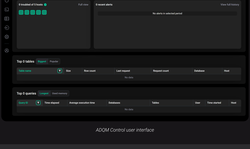
- The state of cluster services, displayed on the **Services** tab of the cluster page, is being changed from `created` to `installed`.



The state of cluster services after successful installation

Step 4. Check the ADQM Control web interface

After the installation of ADQM Control has been completed, its user interface becomes available at *http://<adqmc_backend_host>:5555* (where `<adqmc_backend_host>` is the IP address of the host with the Backend component installed and `5555` is the default value of the **Backend port** parameter in the **Network configuration** section of the ADQM Control service configuration). The `admin` user with the `admin` password and the `Owner` role (system owner) is automatically created – use these credentials to log in to ADQM Control.



ADQM Control user interface

> **IMPORTANT**
>
> - In order for ADQM Control to receive information about data tables and queries of an integrated ADQM cluster, it is required to configure the ADQMDB service parameters for the ADQM cluster as follows:
>
> - enable **query_log** in the Log settings section;
> - in the **Default user and policy settings** section, add a subnetwork or IP address of the host where the ADQM Control's Agents component is installed to the **Default user IP** list.
>
> After you have set the necessary parameters, do not forget to click **Save** and execute the **Reconfig and restart** action for the ADQMDB service to save configuration changes and restart the service.
>
> - If you have imported an ADQM cluster version-up to and including 24.3.1.1.7.1.8.1 into ADQM Control, and the Monitoring service was installed during the installation of the ADQM cluster (i.e. by the **Install** action at the cluster level, not by the service action), ADQM Control may not display the hosts of their cluster. To fix this issue, run the **Reconfigure and restart** action for the Monitoring service of the ADQM cluster. If the Monitoring service was installed into an existing ADQM cluster, this behavior does not occur – states of all ADQM cluster hosts are immediately displayed in ADQM Control.

Install monitoring

Contents

To Table of Contents

Overview
Step 1. Add the service
Step 2. Add components
Step 3. Configure the service
Step 4. Install the service
Step 5. View results
- Metrics in the Prometheus format
- Prometheus web interface
- Grafana web interface

The Monitoring service deploys the Prometheus server inside ADQM CONTROL to collect and store ADQM Control cluster monitoring metrics, and also supports the ability to use the Grafana web application for visualization and analysis of information. This article describes the steps required to add the service to an ADQM Control cluster.

Overview

When adding the Monitoring service to ADQM Control, you can install the node exporter on all or selected hosts of your cluster. It is a monitoring agent that reads host system metrics that Prometheus collects for subsequent analysis and visualization. Prometheus uses ADQM Control services (ADQM Control and Zookeeper). These metrics will be available in the Prometheus format on the ports and endpoints specified in the service configuration. You can use the Prometheus and Grafana web interfaces to view and analyze data that the monitoring service collects.

> **NOTE**
>
> - If you already have a Prometheus-compatible monitoring system set up (for example, your own Prometheus server or VictoriaMetrics), you can use it to collect ADQM Control metrics. In this case, configure access to ADQM Control metrics in your monitoring system using ADQM Control's Prometheus parameters specified on the Monitoring page.
> - You can also use the Telegraf agent for metrics in the Prometheus server deployed in ADQM Control to pass Prometheus.

Step 1. Add the service

1. In the ADQM interface, open the **Clusters** page and click your ADQM Control cluster name. On the cluster page that opens, switch to the **Services** tab and click **Add services**.

Switch to adding services

2. In the opened dialog, select the **Monitoring** service and click **Add**.

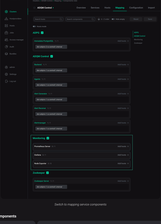
Select the service

As a result, the added service is displayed on the **Services** tab.

The result of successfully adding the service to the cluster

Step 2. Add components

1. On the cluster page, open the **Mapping** tab to proceed to mapping service components to cluster hosts.

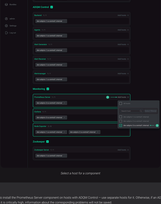


Switch to mapping service components

Monitoring service components

Component	Description
Prometheus Server	Stores and allows requesting all metrics for an ADQM Control cluster
Grafana	Visualizes ADQM Control metrics as graphs and charts organized into dashboards
Node Exporter	Collects system metrics and sends them to Prometheus

2. Assign hosts to components of the Monitoring service – click **Add hosts** and select the desired hosts in the pop-up window.



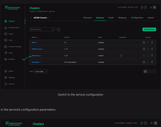
Select a host for a component

> **CAUTION**
>
> It is not recommended to install the Prometheus Server component on hosts with ADQM Control – use separate hosts for it. Otherwise, if an ADQM Control host fails, it will be hard to promptly get information about the corresponding problems or not to be current.

3. After the distribution of components is completed, click **Save**.

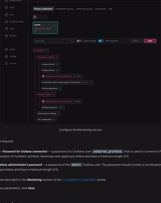
Step 3. Configure the service

1. Open the **Services** tab on the cluster page and click the **Monitoring** service name (i.e. the **Name** column).



Switch to the service configuration

2. On the page that opens, fill in the service's configuration parameters.



Configure the Monitoring service

Fields highlighted in red are required.

- **Prometheus settings** > **Password for Grafana connection** – a password of a Grafana user (`admin_grafana`) that is used to connect to Prometheus. The password must be specified in the Grafana administrator's settings and approximately follows a minimum length of 8.
- **Grafana settings** > **Grafana administrator's password** – a password of the `admin` Grafana user. The password should contain a combination of numbers, symbols, and both lowercase and uppercase letters and approximately follows a minimum length of 8.

All monitoring parameters are described in the **Monitoring** section of the Configuration parameters article.

3. After specifying all necessary parameters, click **Save**.

Step 4. Install the service

1. On the **Services** tab, click the icon ( ) for the Monitoring service in the **Actions** column and run the **Install** action.

Switch to the service installation

2. Wait until the installation is completed. Then, check that the service code has changed from `installed` → `installation`.

Installation is complete

To view the service installation process and analyze errors if they occur, select **Jobs** in the left navigation menu, and click the **Install** job name in the **Jobs** list.

Install service job page

Step 5. View results

The Monitoring service starts automatically after installation. To ensure that monitoring works correctly, check that both system metrics and metrics of ADQM Control's services are collected from the cluster hosts. To do this, view metrics in the Prometheus format in the browser or use the Prometheus or Grafana web interface.

Metrics in the Prometheus format

1. In the address bar of your browser, enter an address of an ADQM Control host with the specified port and endpoint to view the service or system metrics. Port numbers and endpoints are defined by the Prometheus settings.
   - In the **Network configuration** section of the ADQM Control service configuration – ports for access to metrics of the ADQM Control services components (the endpoint is always `/metrics`).
   - In the **Monitoring** section of the Zookeeper service configuration – port for access to metrics of the Zookeeper component (the endpoint is `/metrics`).
   - In the **Prometheus settings** section – port for access to metrics of the Prometheus component and its endpoint (the endpoint is always `/metrics`).

   For example, http://10.92.40.100:9363/metrics is an address to view metrics of the clickhouse-server component on a host with the IP 10.92.40.100.

2. The page that opens will display monitoring metrics from the specified host of the ADQM Control cluster in the Prometheus format.

```
# HELP ClickHouse Version and revision
# TYPE ClickHouse Version and revision gauge
ClickHouse Version and revision{version="22.8.8.3",revision="54460"} 1
# HELP ClickHouse Error*
# TYPE ClickHouse Error* gauge
ClickHouse Error*{name="ACCESS_DENIED"} 0
ClickHouse Error*{name="AIO_READ_FAILED"} 0
ClickHouse Error*{name="AIO_SUBMIT_FAILED"} 0
# HELP ClickHouse Status*
# TYPE ClickHouse Status* gauge
ClickHouse Status*{name="StorageBufferFlushThreadsActiveCopy"} 0
ClickHouse Status*{name="StorageBufferFlushThreadsScheduled"} 0
ClickHouse Status*{name="StorageBufferFlushThreadsWaitingToBackOff"} 0
ClickHouse Status*{name="StorageDistributedThreadsActiveCopy"} 0
ClickHouse Status*{name="StorageDistributedThreadsScheduled"} 0
ClickHouse Status*{name="StorageDistributedThreadsWaitingToBackOff"} 0
ClickHouse Status*{name="TCPConnection"} 1
ClickHouse Status*{name="TCPConnectionForClickHouseDisks"} 0
ClickHouse Status*{name="TCPConnectionForClickHouseDiscs"} 0
ClickHouse Status*{name="TCPConnectionForHDFS"} 0
ClickHouse Status*{name="TCPConnectionForJDBC"} 0
ClickHouse Status*{name="TCPConnectionForKafka"} 0
ClickHouse Status*{name="TCPConnectionForURL"} 0
ClickHouse Status*{name="TemporaryFileForAggregation"} 0
ClickHouse Status*{name="TemporaryFileForJoin"} 0
ClickHouse Status*{name="TemporaryFileForSort"} 0
ClickHouse Status*{name="TemporaryFileInWritePolicy"} 0
ClickHouse Status*{name="TemporaryFileOfSort"} 0
ClickHouse Status*{name="VersionInteger"} 22080803
# HELP ClickHouse Profile*
# TYPE ClickHouse Profile* gauge
ClickHouse Profile*{name="ArenaAllocBytes"} 0
ClickHouse Profile*{name="ArenaAllocChunks"} 0
ClickHouse Profile*{name="CachedReadBufferCacheWriteBytes"} 0
ClickHouse Profile*{name="CachedReadBufferReadFromCacheBytes"} 0
ClickHouse Profile*{name="CachedReadBufferReadFromSourceBytes"} 0
ClickHouse Profile*{name="CachedReadBufferReadFromSourceMicroseconds"} 0
ClickHouse Profile*{name="CachedReadBufferReadFromSourceNum"} 0
ClickHouse Profile*{name="CachedReadBufferCacheWriteMicroseconds"} 0
ClickHouse Profile*{name="CachedReadBufferCacheWriteNum"} 0
ClickHouse Profile*{name="CachedWriteBufferCacheWriteBytes"} 0
ClickHouse Profile*{name="CachedWriteBufferCacheWriteMicroseconds"} 0
ClickHouse Profile*{name="CachedWriteBufferCacheWriteNum"} 0
ClickHouse Profile*{name="CompressedReadBufferBlocks"} 0
ClickHouse Profile*{name="CompressedReadBufferBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferChecksumMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferDecompressedBlocks"} 0
ClickHouse Profile*{name="CompressedReadBufferDecompressedBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferDecompressionMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferDecompressionNum"} 0
ClickHouse Profile*{name="CompressedReadBufferMappedBlocks"} 0
ClickHouse Profile*{name="CompressedReadBufferMappedBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferMappedMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferMappedNum"} 0
ClickHouse Profile*{name="CompressedReadBufferOfDecompressedBlocksBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferOfDecompressedBlocksMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferOfDecompressedBlocksNum"} 0
ClickHouse Profile*{name="CompressedReadBufferOfMappedBlocksBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferOfMappedBlocksMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferOfMappedBlocksNum"} 0
ClickHouse Profile*{name="CompressedReadBufferOfPrefetchedBlocksBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferOfPrefetchedBlocksMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferOfPrefetchedBlocksNum"} 0
ClickHouse Profile*{name="CompressedReadBufferPrefetchedBlocks"} 0
ClickHouse Profile*{name="CompressedReadBufferPrefetchedBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferPrefetchedMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferPrefetchedNum"} 0
ClickHouse Profile*{name="CompressedReadBufferUncompressedBlocks"} 0
ClickHouse Profile*{name="CompressedReadBufferUncompressedBytes"} 0
ClickHouse Profile*{name="CompressedReadBufferUncompressedMicroseconds"} 0
ClickHouse Profile*{name="CompressedReadBufferUncompressedNum"} 0
ClickHouse Profile*{name="CompressedWriteBufferBlocks"} 0
ClickHouse Profile*{name="CompressedWriteBufferBytes"} 0
ClickHouse Profile*{name="CompressedWriteBufferCompressionMicroseconds"} 0
ClickHouse Profile*{name="CompressedWriteBufferCompressionNum"} 0
ClickHouse Profile*{name="CompressedWriteBufferUncompressedBytes"} 0
ClickHouse Profile*{name="ContextLock"} 0
ClickHouse Profile*{name="CreatedHTTPConnections"} 0
ClickHouse Profile*{name="CreatedLogEntryForMerge"} 0
ClickHouse Profile*{name="CreatedLogEntryForMutation"} 0
ClickHouse Profile*{name="CreatedReadBufferAIO"} 0
ClickHouse Profile*{name="CreatedReadBufferAIOFailed"} 0
ClickHouse Profile*{name="CreatedReadBufferDirectAIO"} 0
ClickHouse Profile*{name="CreatedReadBufferDirectAIOFailed"} 0
ClickHouse Profile*{name="CreatedReadBufferMMap"} 0
ClickHouse Profile*{name="CreatedReadBufferMMapFailed"} 0
ClickHouse Profile*{name="CreatedReadBufferOrdinary"} 0
ClickHouse Profile*{name="CreatedWriteBufferAIO"} 0
ClickHouse Profile*{name="CreatedWriteBufferAIOFailed"} 0
ClickHouse Profile*{name="CreatedWriteBufferDirectAIO"} 0
ClickHouse Profile*{name="CreatedWriteBufferDirectAIOFailed"} 0
ClickHouse Profile*{name="CreatedWriteBufferOrdinary"} 0
ClickHouse Profile*{name="DelayedInsertsMilliseconds"} 0
ClickHouse Profile*{name="DelayedInsertsToBackgroundPoolErrors"} 0
ClickHouse Profile*{name="DelayedMutations"} 0
ClickHouse Profile*{name="DelayedMutationsMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrors"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytes"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytes"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytes"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytes"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytes"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedMilliseconds"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytes"} 0
ClickHouse Profile*{name="DelayedPartsToBackgroundPoolErrorsRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytesRejectedRowsAndBytes
```

Cluster actions

Elena Dvoryadkina

Contents

To Table of Contents

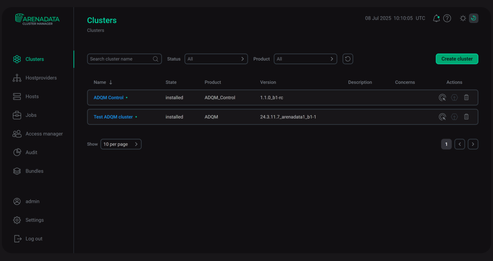
Overview

Manage a cluster

This article describes actions available for working with an ADQM Control cluster in the ADCM interface.

Overview

You can find cluster actions on the ***Clusters*** page.

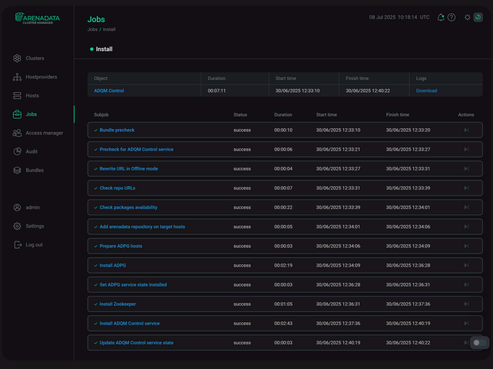


The Clusters page

The ***Clusters*** page displays a table with the following columns:

- ***Name*** — name specified during the cluster creation.
- ***State*** — the current state of the cluster (`created`, `installed`).
- ***Product*** — product name.
- ***Version*** — version of a bundle used to install the cluster.
- ***Description*** — description specified during the cluster creation.
- ***Concerns***. The icon ⓘ indicates if there are critical errors in the cluster configuration or some job is running. When you hover over it, a pop-up window appears showing a description with a link that you can follow to perform the required configuration or get details. If the cluster is configured correctly and there are no currently running actions, the column is empty.
- ***Actions***. The column shows icons for managing a cluster:
  - — opens a drop-down list that offers actions to manage the cluster.
  - — indicates whether a new version of a bundle is available and allows you to upgrade the cluster version.
  - — deletes information about the cluster from ADCM (it does not remove ADQM Control or make any changes to hosts that belong to the cluster).

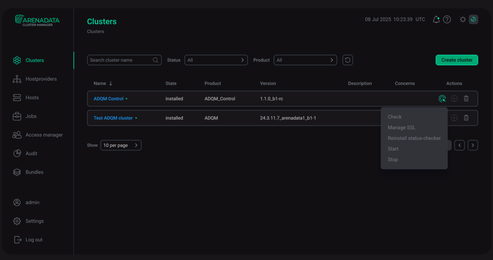
You can click the icons described above to perform the corresponding cluster actions. When an action starts, ADCM displays the process and result of its execution on the ***Jobs*** page. From this page, you can navigate to a page with details on an individual job (by clicking a job name) to see inner steps of that job execution and analyze errors if they occur.



The Jobs page

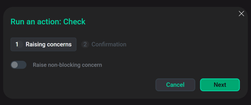
Manage a cluster

Click the icon in the ***Actions*** column to open a drop-down list with available actions and select an action to run it.

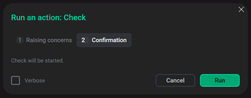


ADQM Control cluster actions

When you choose an action, ADCM displays a dialog window where you can run an action using the non-blocking mode (the ***Raising concerns*** step) and then confirm the launch of the action (the ***Confirmation*** step). On the ***Confirmation*** step, you can select the ***Verbose*** checkbox to see additional execution details on the ***Jobs*** page.



Option to run an action in the non-blocking mode



Action confirmation

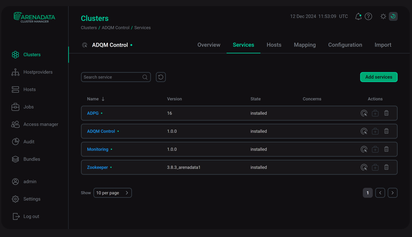
A set of available actions depends on the cluster state:

- For a new cluster (the cluster state is `created`), an available action is:
  - ***Install*** — installs all cluster services.
- After a cluster is installed (the cluster state is `installed`), the following actions are available:
  - ***Check*** — checks if all components of the installed cluster work correctly.
  - ***Manage SSL*** — allows you to configure and enable/disable SSL encryption for services of the ADQM Control cluster.
  - ***Reinstall status-checker*** — reconfigures and restarts the status checker for all cluster services. Use this action when migrating a cluster to a new ADCM server.
  - ***Start*** — starts all services of the cluster.
  - ***Stop*** — stops all services of the cluster.

Service actions

Elena Dvoryadkina

ADCM UI provides actions to manage services of ADQM Control. These actions are available on the ***Services*** page. To open this page, click an ADQM Control cluster name on the ***Clusters*** page and switch to the ***Services*** tab.



ADQM Control cluster services in the ADCM interface

The ***Services*** page contains a table with the following columns:

- ***Name*** — service name.
- ***Version*** — service version.
- ***State*** — the current state of the service (for example, `created` or `installed`).
- ***Concerns***. The icon indicates if there are critical errors in the service configuration (for example, mandatory components are not deployed on cluster hosts) or some job is running. When you hover over it, a pop-up window appears showing a description with a link that you can follow to perform the required configuration or get details. If the service is configured correctly and there are no currently running actions, the column is empty.
- ***Actions***. When you click the icon , a drop-down list offers actions to manage the service. A set of actions depends on the service type.

  Actions available for all services:

  - ***Install*** (if a service state is `created`) — installs a service.
  - ***Check*** — checks the health of a service.
  - ***Start*** — starts a service.
  - ***Stop*** — stops a service.

  Service-specific actions:

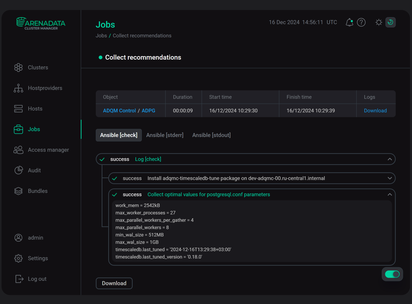
ADQM Control

- ***Reconfigure and restart*** — updates the service configuration files according to settings specified on the service configuration page, and restarts the service.
- ***Reinstall*** — reinstalls the service.

ADPG

- ***Reconfig and restart*** — updates the service configuration and restarts the service.
- ***Collect recommendations*** — collects optimal values for *postgresql.conf* configuration parameters to improve ADPG performance.

  After the action is performed, open the corresponding job on the ***Jobs*** page (by clicking ***Collect recommendations*** in the list of jobs) and in the ***Ansible [check]*** section, make sure that the ***Collect optimal values for postgresql.conf parameters*** subtask has been completed successfully. The result of this subtask contains a list of optimal parameter values — assign these values to the corresponding parameters in the ADPG configurations section of the ADPG service configuration page and run the ***Reconfig and restart*** action for the service.



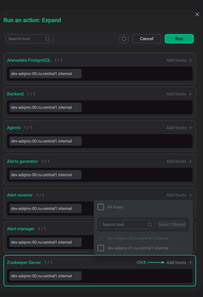
The result of the "Collect recommendations" action

- ***Remove*** — removes the service from the cluster. This action can be used to delete an already installed service, unlike the icon that allows you to remove a non-mapped service (a service whose components have not yet been distributed among cluster hosts).

Zookeeper

- ***Reconfig and restart*** — updates the service configuration and restarts the service.
- ***Expand*** — installs the Zookeeper Server component on more hosts.

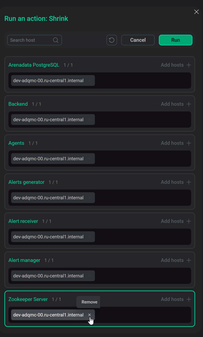
  When you select this action, ADCM opens the ***Run an action*** dialog where you can add new hosts on which you want to install the Zookeeper Server component.



Add Zookeeper Server to hosts

- ***Shrink*** — removes the Zookeeper Server component from hosts.

  In the ***Run an action*** dialog that appears when you select the ***Shrink*** action, remove hosts on which the Zookeeper Server component should not be installed.



Remove Zookeeper Server from hosts

Monitoring

- ***Add/Remove components*** — opens the component-host mapping interface where you can add, remove, or redistribute service components.
- ***Reconfigure and restart*** — applies configuration to the service according to settings specified on the service configuration page, and restarts the service.
- ***Reinstall*** — reinstalls the service.

When an action starts, ADCM displays its execution process and result on the ***Jobs*** page. On this page, you can also analyze errors if they occur.

The ***Actions*** column also displays separate icons for the following actions:

- — sets up the maintenance mode for a service. This functionality is currently not implemented in the ADQM Control bundle.
- — removes a service from the cluster.

Configure alerts in ADQM Control via web UI

Elena Dvoryadkina

Contents

To Table of Contents

System alerts
- Configure system alerts
- Types of system alerts

Internal alerts

The ***Alerts*** tab on the ***Settings*** page in the ADQM Control web interface allows you to configure parameters to be criteria for generating alerts.

To change default settings, edit the required fields and click ***Save***. Click ***Revert all*** if you need to cancel changes that have not been yet saved by clicking ***Save***.

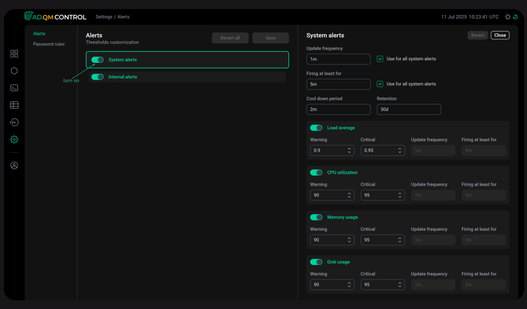
Alerts in ADQM Control are grouped into modules. Currently, ADQM Control can send and allows you to configure alerts of the System alerts and Internal alerts modules.

System alerts

The ***System alerts*** module unites system alerts — alerts that are generated based on values of system metrics that indicate general characteristics of ADQM cluster hosts, usually related to resource consumption (see the System alerts table). For each system metric, you can set thresholds against which ADQM Control compares the current value of the metric and determines whether to generate an alert and its severity level.

Configure system alerts

Use the ***System alerts*** switch on the ***Settings/Alerts*** page to enable/disable the generation of alerts based on system metrics. When the switch is enabled, the ***System alerts*** form on the right allows you to configure settings of system alerts.



Configure system alerts

System alert settings

Parameter	Description	Default value
Update frequency	Specifies how often to compare a metric value with thresholds. Once the metric exceeds a threshold, an alert is generated in ADQM Control (but not sent for processing). The parameter value should be in the range of 1-5 minutes	1m (m — minutes)
Firing at least for	Time during which a metric value should exceed a threshold for the corresponding alert to be sent for processing, after which it will appear in the ADQM Control interface. The parameter value should be in the range of 1-15 minutes	5m (m — minutes)
Cool down period	Period (after sending an alert) during which ADQM Control ignores an update when a metric value no longer exceeds a threshold. If the metric after this period still does not exceed the threshold, then the alert is considered to be no longer valid. The period starts again after each update when the metric exceeds the threshold (during the previous period) — in other words, it is extended. The parameter value should be in the range of 2-15 minutes, greater than the value of the ***Update frequency*** parameter	2m (m — minutes)
Warning	Metric value at which a medium-significance alert is generated. This level of alert importance indicates that there is a potential problem on an ADQM cluster host due to an increase in the value of the corresponding system metric (but this problem has not reached the critical level yet)	See default thresholds in the System alerts table
Critical	Metric value at which a high-significance alert is generated, indicating that a critical problem has been detected on an ADQM cluster host	See default thresholds in the System alerts table

You can set up the ***Update frequency*** and ***Firing at least for*** parameters in two ways:

- enter required values at the top of the ***System alerts*** form and enable the ☑ ***Use for all system metrics*** option to apply the specified values to all system alerts;
- specify parameter values separately for each alert type.

Types of system alerts

The table below describes the types of alerts that ADQM Control can generate while monitoring the corresponding system metrics on ADQM cluster hosts and comparing their values to specified thresholds.

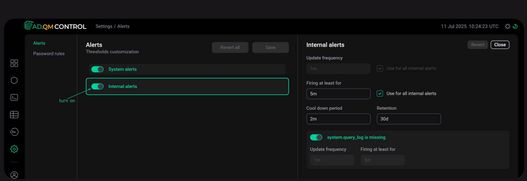
System alerts

Alert type	Condition for generating an alert	Default thresholds
Load average	The system load average (a value is in the range `[0, 1]` for a single CPU and can be higher for multi-core systems) exceeds a threshold — `(LA15 + LA5)/2 > threshold`. This means either high CPU usage or disk IO activity take too long	Warning — 0.9, Critical — 0.95
CPU utilization	The CPU usage (as a percentage) exceeds a threshold	Warning — 90, Critical — 95
Memory usage	The RAM usage (as a percentage) exceeds a threshold	Warning — 90, Critical — 95
Disk usage	The disk capacity (as a percentage) exceeds a threshold	Warning — 90, Critical — 95

For each alert type, there is a switch in the ***System alerts*** form that you can use to disable the generation of alerts based on the corresponding metric.

Internal alerts

In the ***Internal alerts*** module settings, you can enable/disable the generation of alerts if the `system.query_log` log table with information about executed queries does not exist in an ADQM cluster.



Configure internal alerts

As with system alerts, you can control the following parameters for internal alerts:

- ***Update frequency*** — frequency of checking for existence of the `system.query_log` table;
- ***Firing at least for*** — time during which the log table should be missing for the corresponding alert to be added to ADQM Control;
- ***Cool down period*** — period after the alert is sent, during which ADQM Control ignores an update when logging to the `system.query_log` table has started.

Manage access

Elena Dvoryadkina

AD.QM CONTROL

Contents

To Table of Contents

View a list of users

Add a user

Edit a user

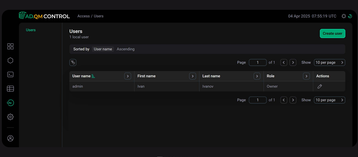
Set password requirements

Users log in to ADQM Control under the accounts previously created for them in the system. To manage users, you can use the ***Users*** tab on the ***Access*** page in the ADQM Control web interface.

After an ADQM Control cluster has been installed, the `admin` user with the `admin` password and the `Owner` role (system owner) is automatically created.

View a list of users

The list of ADQM Control users is available on the ***Access*** → ***Users*** page. For each user, it displays the login, first name, last name, and role.



The Users page

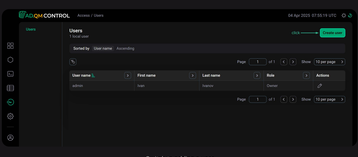
In column headers of the table with a list of users, there are filters that you can use to select specific data. To open a filter, click the > icon.

The icon means that a filter is defined for a column. To reset all filters, click .

Add a user

To add a new user, follow the steps:

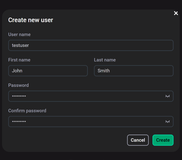
1. On the ***Access*** → ***Users*** page, click ***Create user***.



Switch to adding a user

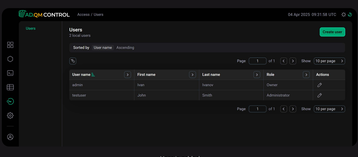
2. In the window that opens, fill in the following fields:
   - ***User name*** — user's login;
   - ***First name*** — user's name;
   - ***Last name*** — user's last name;
   - ***Password*** — user's password.

> **TIP**
>
> By default, the ***Password*** field shows the hint with password requirements (configured on the ***Settings*** → ***Password rules*** page) when you start filling it. To get this information, you can also hover the mouse over the icon, which appears to the right of the ***Password*** field name on hover.



Enter user data

3. Click ***Create***. As a result, a new user is displayed on the ***Access*** → ***Users*** page.



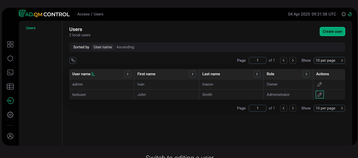
User is added

> **NOTE**
>
> New users are currently created with the `Administrator` role (system administrator) that provides the same set of access rights as `Owner`.

Edit a user

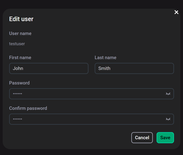
To edit user data, follow the steps:

1. Click the icon in the ***Actions*** column on the ***Access*** → ***Users*** page.



Switch to editing a user

2. In the window that opens, edit necessary fields. For example, you can specify another password for the user (taking into account the specified password complexity requirements). The user's login in the ***User name*** field cannot be changed.

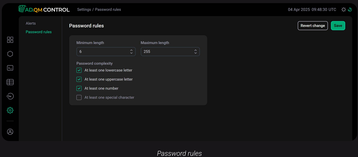


Edit user data

3. Click ***Save*** to apply changes.

Set password requirements

On the ***Settings*** → ***Password rules*** page, you can set requirements for the complexity of user passwords.



Password rules

If you need to edit the default settings, change the necessary parameters and click ***Save***. Click ***Revert change*** if you need to undo the changes that have not been yet saved by clicking ***Save***.

Parameters to set up password rules

Parameter	Description	Default value
Minimum length	Minimum length of a password	1
Maximum length	Maximum length of a password	255
At least one lowercase letter	Indicates whether a password should contain at least one lowercase character	Disabled
At least one uppercase letter	Indicates whether a password should contain at least one uppercase character	Disabled
At least one number	Indicates whether a password should contain at least one numeric character	Disabled
At least one special character	Indicates whether a password should contain at least one special character (excluding characters from the extended ASCII table)	Disabled

View a dashboard

ADQMCONTROL

Version: 1.0 | Updated: 2022-03-31

Contents

To Table of Contents

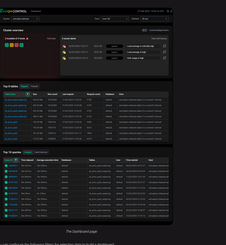
Cluster overview
- Heat map
- Recent alerts
- Acknowledged alerts
- Open alert details
- Go to the Cluster metrics page

Top 10 tables

Top 10 queries

The **Dashboard** page of the ADQM Control web interface displays general information about an ADQM cluster state as:

- Heat map – graphically represents the states of the ADQM cluster hosts.
- Recent alerts – lists the latest alerts about potential and critical issues detected on the ADQM cluster hosts.
- Top 10 tables – 10 largest tables by data amount in the ADQM cluster and 10 tables for which the most queries were sent.
- Top 10 queries – 10 queries that took the longest to execute and 10 queries that used the most memory.



The Dashboard page

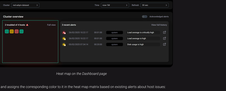
In the fields at the top of the screen, you can configure the following filters for selecting data to build a dashboard:

- **Cluster** – ADQM cluster for which information should be displayed.
- **Time** – time period for which information is required. In the entries that opens when you click the field, you can select an interval from the offered options or the **Range** tab or set a custom time range (at least 1 hour) on the **Calendar** tab.
- **Refresh** – frequency of data updates.

Cluster overview

Heat map

A heat map visualizes information about the states of all hosts in the ADQM cluster – each host is shown as a square, and its color indicates the system state at the selected time interval.



Heat map on the Dashboard page

ADQM Control determines a host state and assigns the corresponding color to it in the heat map hosts based on existing alerts about host issues:

- there are no alerts about any issues with a host in ADQM Control (a healthy host);
- only potential issues have been detected on a host (for example, the increase in the values of some system metrics), but they are not yet critical. Appropriate warning alerts about those issues have been generated in ADQM Control;
- at least one critical issue has been detected on a host, and the corresponding alert of the high importance level about it exists in ADQM Control.

Recent alerts

The **Dashboard** page also displays the most recent alerts (up to 5) about problems on the ADQM cluster hosts, sorted in descending order by the time the alerts were generated, not by that importance.

Recent alerts on the Dashboard page

Alert severity levels

Alert severity level	Description	Condition for generating an alert
warning	A potential problem has been detected on a host	A system metric value exceeds a threshold set via the **Warning** parameter in the System alerts configuration
critical level of importance	A critical problem has been detected on a host	A system metric value exceeds a threshold set via the **Critical** parameter in the System alerts configuration

You can select one or more hosts in **Heat map** by clicking+ then the list of recent alerts will include the alerts only for the selected hosts.

Recent alerts about problems on the selected host

To get a full list (history) of alerts for all hosts of the cluster generated during the specified period of time, click **View full history** in the top right corner within the **Recent alerts** section – it will open the Alert history tab on the **Cluster metrics** page.

Acknowledged alerts

The **Acknowledged alerts** (alert allows you to specify which alerts about problems on ADQM cluster hosts Control should take into account when generating the heat map and the list of recent alerts:

- **Acknowledged alerts** – all alerts generated during the selected time period;
- **Acknowledged alerts** – alerts, on which you have set an acknowledged. ADQM Control highlights relevant alerts (requiring your attention to be resolved) with the bell sign in their icons.

For example, no issues were detected for the specified period of time on the second host of the cluster (it is selected in the image below) or all alerts were set as acknowledged – in this case, with the **All** / **Acknowledged alerts** option enabled, this host is filled with green in **Heat map** and the **Recent alerts** list is empty.

A selected host has no issues that require analysis

Then, the new **Disk usage is high** acknowledgement appeared on the host, and ADQM Control generated a warning alert. This alert is now displayed in the **Recent alerts** list and the color of the host is yellow in **Heat map** as shown in the following image.

Example of an unresolved issue on the host

If the disk usage issue has been analyzed and resolved, it no longer requires attention and you can add the corresponding alert to the list of acknowledged alerts (on the page with alert details – see the next section). After that, ADQM Control will not take into account this alert when monitoring the cluster health – it will display the host in green in **Heat map** again and exclude the alert from the **Recent alerts** list for the host.

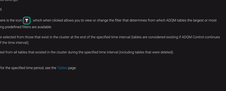
If the **Acknowledged alerts** option is enabled, the **Recent alerts** list shows the latest alerts among all alerts (active and "closed") generated during the specified time period, and all alerts affect the heat map. For example, in the image below, no issues were detected in **Heat map** on host 1 – this means that at least one critical issue was detected on each of these hosts during the selected time period (the last month in this example). The **Recent alerts** list shows that there are no relevant alerts among the most recent alerts about problems on the selected host (alert icons are without the bell sign).

The Acknowledged alerts option is enabled

Open alert details

From the **Dashboard** page, you can access detailed information about a problem related to a host in one of the following ways:

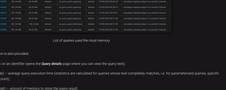
- In **Heat map**, hover the mouse over a host – a pop-up window will show alerts about the last two problems found on that host. Click **Open details** below an alert corresponding to an issue you want to receive more information about.



Summary of host issues

- In the **Recent alerts** list, click the icon in an alert row.

Each of these actions opens a separate page with a detailed description of the corresponding issue. For example, the details of an alert **Disk usage is high**, show which device is exceeding its usage limit. The **Threshold** field contains the metric threshold value set in Settings → Alerts, based on which the alert was generated.



Host issue details

Use the **Acknowledged alerts** option on this page to change an alert status and mark the corresponding issue as acknowledged (in other words, known or resolved).

Go to the Cluster metrics page

From **Heat map** on the **Dashboard** page, you can go to the Cluster metrics tab on the **Cluster metrics** page. It also shows the heat map matrix for an ADQM cluster and additionally displays the hosts' uptime and the number of active alerts for each host. In the left pane, you can select one or more hosts from the searched hosts or the left-hand tree, and then you can get more more information about a specific alert by clicking the corresponding row in the table. Further details will appear below the tree.

Heat map on the "Cluster metrics" page

To go to the **Cluster metrics** → **Heat map** page, do one of the following in **Heat map** of the **Dashboard** page:

- Click **Show all** in the pop-up window that appears when you hover the mouse over the selected host by clicking+ tool.
- Click **Full view** in the top right corner of **Heat map**.

A set of selected hosts and the **Acknowledged alerts** setting are synchronized between the **Dashboard** and **Cluster metrics** pages – changing these settings on one page automatically updates the same settings on the other page.

Top 10 tables

The **Top 10 tables** section contains two tabs with information about data tables in an ADQM cluster for a specified time interval:

- *Biggest* – 10 tables by the amount of data they store (sorted in descending order by **Size** – size of compressed data in a table at the end of the selected time interval);
- *Popular* – 10 tables to which the most queries were sent (sorted in descending order by **Request count** – number of queries sent to a table during the selected time interval).

List of the largest tables

List of tables received the most queries during the specified time interval

This section also provides the following information about tables:

- **Table name** – table name (click on a table name opens a page with detailed information on table columns);
- **Row count** (for **Biggest** tab) – number of rows over a table at the end of the selected time interval;
- **Last request** – time that any query to a table was run during the selected time interval;
- **Increase** (on the **Popular** tab) – change in a table size over the selected time interval (a negative value means a decrease in the amount of data in the table);
- **Database** – database to which a table belongs;
- **Host** – host where a table is located.

In the header of the **Table name** field, there is the icon, which when clicked allows you to view or change the filter that determines from which ADQM tables the largest or most popular tables are selected. The following parameters filters are available:

- *Existing* (default) – top 10 tables are selected from those that exist in the cluster at the end of the specified time interval (tables are considered existing if ADQM Control continues to receive their metrics, that are sent every minute);
- *Historical* – top 10 tables are selected from all tables that existed in the cluster during the specified time interval (including tables that were deleted).

To get information on all cluster tables for the specified time period, see the Tables page.

Top 10 queries

The **Top 10 queries** section contains two tabs with information about queries sent to ADQM cluster databases during a specified time interval:

- *Longest* – top 10 queries with the longest execution time (sorted in descending order by **Duration** – duration of a query execution);
- *Used memory* – top 10 queries required the most memory (sorted in descending order by **Used memory** – amount of memory consumed by a query).

List of the longest running queries

List of queries used the most memory

For each query, the following information is also provided:

- *Query ID* – the query identifier (click on an identifier opens the **Query details** page where you can view the query text);
- *Average duration* (for **Longest** tab) – average query execution time (statistics are calculated for queries whose text completely matches, i.e. for parameterized queries, specific parameter values are taken into account);
- *Peak memory* (for **Used memory** tab) – amount of memory to store the query result;
- *Databases* – names of databases to which the query is sent;
- *Tables* – names of tables to which the query is sent;
- *User* – name of an ADQM user who initiated the query execution;
- *Time started* – the query execution start time;
- *Host* – host where the query was run.

In the header of the **Query ID** field, there is the icon, by clicking on which, you can view or change the filter that determines from which queries the longest or most memory intensive ones are selected. The following predefined filters are available:

- *Completed* (on the **Longest** tab) – top 10 longest queries are selected from all the completed queries (i.e. from all queries except active ones);
- *All* (on the **Used memory** tab) – top 10 most memory-intensive queries are selected from all queries;
- *Successful* (default filter) – top 10 queries are selected from those that were completed successfully.

To view the full history of queries for the required period of time, switch to the Queries page.

Cluster metrics

Monitoring

ADQM CONTROL

Contents

To Table of Contents

Heat Map
- Heat state
- Alerts related to a host

Incidents
- How incidents are generated
- Settings

Alerts history
- Alert filters
- Alert details

The *Cluster metrics* page of the ADQM Control web interface is designed to display information about an ADQM cluster state based on values of metrics collected from the cluster hosts. The page includes three tabs: Heat map, Incidents, and Alerts history.

To select data to be shown on the *Cluster metrics* page, you can use the fields at the top of the screen:

- *Cluster* – ADQM cluster for which the information is to be displayed;
- *Time* – time period for which information is required. In the window that opens when you click the field, you can select an interval from the offered options or the *Range* tab or set a custom time range in the *Calendar* tab;
- *Refresh* – frequency of data updates.

Data filters on the "Cluster metrics" page

Heat Map

A heat map visualizes information about states of all hosts in an ADQM cluster – each host is shown as a square and its color indicates the system state.

Heat map on the "Cluster metrics" page

Host state

ADQM Control determines a host state and assigns the corresponding color to it in the *Heat map*: hosts marked in existing alerts about host issues

- Green are no alerts about issues with a host or all alerts are resolved;
- Light yellow issues have been detected in a host (for example, the increase in the values of some system metrics), but they are not yet critical. Appropriate warning alerts on these issues have been raised and remain relevant in ADQM Control;
- Dark yellow critical issues have been detected in a host. Appropriate alerts of the high importance level about it exists in ADQM Control.

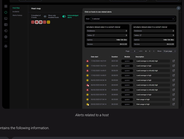
The *Show only troubled hosts* switch at the top of the *Heat map* tab allows you to configure the heat map so that it displays only troubled hosts.

Heat map shows only troubled hosts

The *Acknowledged alerts* option specifies which alerts about problems on hosts ADQM Control should take into account when creating the cluster heat map – all alerts generated during the selected time period or only those that are not marked as acknowledged. See the example in the Acknowledged alerts section.

Alerts related to a host

Hovering a square on the heat map during there in the heat map displays a table with a list of host-related alerts generated during the specified period of time. Depending on the *Acknowledged alerts* setting, the list includes all alerts or only alerts requiring analysis – see Acknowledged alerts for details. In the *Host* field above the table, you can change a set of hosts for which a list of alerts is displayed. There is also general information on each selected host where Clickhouse is installed: number of databases, number of tables, server uptime, and ClickHouse version.



Alerts listed in a host

The table with a list of alerts contains the following information.

Field	Description
Date start	Timestamp when an alert was generated (in the `DD/MM/YYYY hh:mm:ss` format)
Duration	Total time that a host issue alert remained active
Module	Module that includes an alert. Currently, ADQM Control only supports these modules: `system` (alerts generated based on system metrics of hosts) and `internal` (alerts about missing the `system.query_log` system table)
Description	Description of an issue detected on a host

The *Severity* column of the table with a list of alerts contains an icon that indicates the severity level of an alert:

- warning – alert that informs about potential issues on an ADQM cluster host the issue has not reached the critical level yet. Generated when a system metric value exceeds a threshold set via the *Warning* parameter in the system.alerts configuration;
- high importance – alert that informs about critical issues when a system metric value exceeds a threshold set via the *Critical* parameter in the system.alerts configuration.

Severe issues and the last-stage indicate alerts about issues that are considered acknowledged (in other words, known/resolved). You can set this status on the alert details page. The icons

- are assigned to new and attention-requiring alerts that are not yet acknowledged.

To get more details about an alert, do one of the following:

- Click an alert row in the list – alert details will appear below the row. The *threshold* field contains the metric threshold value based on which the alert was generated.

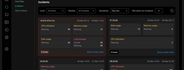
Alert details within the list

- Click the icon within an alert row – this will take you to a separate page with alert details (the page has a unique address that you can share). The *Acknowledged alerts* option allows you to change the alert status – mark the corresponding alert as acknowledged or back to show it affect the cluster heat map.

Alert details on a separate page

Incidents

On the *Incidents* tab, ADQM Control combines alerts about issues occurred on cluster hosts during the selected time period into *incidents* – groups of alerts that are closest to each other in time of creation.



Incidents on the "Cluster metrics" page

An incident is displayed as a block that includes:

- incident duration;
- incident timeline – from the time the first alert in the incident was generated until the time the last alert in the incident was deactivated (the metric related issue was resolved);
- metrics for which thresholds were exceeded on cluster hosts, with information about how many warnings and high-importance alerts were generated;
- number of hosts where problems were detected.

- *Show cluster view* tick – refers to the Overview page with the Acknowledged alerts option enabled and the *Time* period equal to the incident interval. If an incident interval is less than an hour, ADQM Control automatically increases it to 1 hour by adding equal parts to each boundary (where 1 hour is the minimum allowed time interval that can be set in the *Time* field). On the *Dashboard* page you can analyze your cluster state during the time period corresponding to the incident – for example, you can see which hosts had problems, which queries were too long or memory-consuming during this period, which tables were the largest in size and which ones were the most queried.

A block highlighted in red indicates an active incident – if a new alert arrives, it will be added to this incident.

How incidents are generated

To generate incidents, ADQM Control calculates time intervals between alerts and determines the interquartile range (IQR) based on this data. Then, it calculates a threshold value that defines the maximum allowed time interval, after which the next alert will be added to a new incident: $t_{max_diff_threshold} = Q3 + wins_coeff \cdot IQR$, where $wins_coeff$ is the winsorization coefficient and $Q3$ is the third quartile.

The following image illustrates the sequence of incident generation and their arrangement on the page.

Arrangement of incidents on the page

Alerts are grouped into incidents dynamically depending on the specified parameters (see the section below) and the time interval selected in the *Time* field:

- For a fixed time interval in the past, the structure of incidents is generated when the *Incidents* page is opened and remains fixed because the set of alerts for this time interval does not change.
- If you select a time interval relative to the current moment (for example, the interval `now-12h, now-02, now-10`), the structure of incidents on the page may change, since the actual boundaries of the time interval shift over time, which means the set of alerts may change (new alerts may occur or the time of previously valid alerts may become outside the current time frame).

Settings

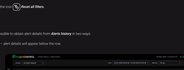
The *Incidents* tab contains the following parameters that affect which alerts are used to build incidents and how alerts are combined:

- *Level* – alert importance level. Possible values: `warning`, `critical`;
- *Module* – module to which alerts belong. Possible values: `system`, `internal`;
- *Sensitivity* – coefficient that affects how much time should pass between an alert and the previous one so that they fall into different incidents. The lower the sensitivity, the longer the time should pass between alerts to place them into different incidents. Possible values: `Very low`, `Low`, `Medium`, `High`, `Very high`.

Example

The images below show examples of the alert distribution for the same time interval across incidents depending on the specified sensitivity.

- With the `Very low` sensitivity, all alerts for the selected time period are combined into one incident:



Sensitivity is very low

- With the `Very high` sensitivity, alerts for the same time period are grouped into two incidents:

Sensitivity is very high

- *Min alerts for incidents* – minimum number of alerts to create an incident.

Alerts history

The *Alerts history* tab displays a full list of alerts that were generated based on metrics of ADQM cluster hosts for the period of time specified in the *Time* field at the top of the screen.

Alerts history on the "Cluster metrics" page

Most of the fields in the *Alerts history* table match the fields that are described above for the *Heat map* tab. Additionally, the following fields are available:

- *Date finished* – timestamp (in the `DD/MM/YYYY hh:mm:ss` format) when an alert became no longer valid, i.e. the corresponding issue was resolved on the host. If an alert is still active, the field shows a dash in the future.
- *Host* – host on which an issue was detected.

Hovering ... indicates alerts that an issue requires attention or is set as acknowledged.

Alert filters

You can use filters located at the top of the *Alerts history* tab to specify which alerts to display in the table:

- *Level* – alert importance level. Possible values:
  - `warning` – potential issues on hosts;
  - `critical` – critical issues on hosts.
- *Module* – module into which alerts are grouped. Possible values: `system`, `internal`;
- *Host* – host for which you want to display alerts on cluster hosts in the system. You can select multiple or all hosts at once.

To reset all applied filters, click the icon.

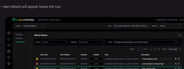
Alert details

As for the *Heat map* tab, it is possible to obtain alert details from *Alerts history* in two ways:

- Click an alert row in the list – alert details will appear below the row.

Alert details within the list

- Click the icon within an alert row – a separate page with alert details will open.



Alert details on a separate page

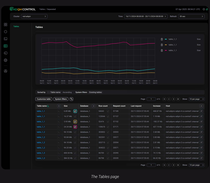
Work with tables in ADQM Control

← back to overview

Contents

To Table of Contents

List of tables
Graphs
Filter and sort data
Table details
Related queries



The Tables page

At the top of the screen, you can set up the following filters to select which data the **Tables** page should display:

- **Cluster** – ADQM cluster where data tables are located.
- **Time** – time period for which information about tables should be displayed. In the window that opens when you click the field, you can select an interval from the offered options on the **Range** tab or set a custom time range (at least 1 hour) on the **Calendar** tab.
- **Refresh** – frequency of data updates.

ADQM Control supports the following ADQM Clickhouse table engines (i.e. can show tables based on these engines on the **Tables** page):

- engines of the MergeTree family (except versions of the AlterMergeTree engine and replicated table engines)
- engines of the Log family – Log and StripeLog
- the Memory, Merge, Join, Set, null engines.

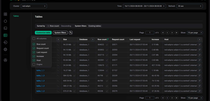
For integration table engines (used to access external data storage systems – for example, MySQL, MongoDB, HDFS, Hive, S3, Kafka, EmbeddedRocksDB, RabbitMQ, PostgreSQL), the correctness and completeness of the information displayed in ADQM Control is currently not guaranteed.

List of tables

The **Tables** page lists data tables of an ADQM cluster as a table with the following fields.

Field	Description
Table name	Name of an ADQM table
Size	Size of compressed data in a table at the end of the selected time interval
Database	Database to which a table belongs

It is also possible to add more fields with additional information on ADQM tables – to do this, click **Customize table** and select the required fields in the drop-down list.



Additional fields

Additional fields with information about ADQM tables

Field	Description
Row count	Number of data rows in a table at the end of the selected time interval
Request count	Number of queries sent to a table during the selected time interval
Last request	Time the last query to a table was run during the selected time interval
Increase	Change in a table size over the selected time interval (a negative value means a decrease in the amount of data in the table)
Host	ADQM cluster host where a table is located. Clicking on the icon opens Clickhouse web interface at http://<adqm_host_name>:8123/play, where <adqm_host_name> is the ADQM host FQDN (the system should be configured so that it can resolve the host FQDN into the IP address).
Engine	Table engine

Graphs

On the **Tables** page, you can get graphs to track the dynamics of changes in the **Size** and **Row count** metrics of ADQM tables (size of compressed data on disk and number of rows, respectively) over a selected period of time. Click the icon in the column of the corresponding metric – above the list of tables, ADQM Control will display a section with a graph built based on this metric of the ADQM table. You can add up to 8 graphs (inclusive) simultaneously.

To see values of metrics at some point in time, hover the mouse over the graph area.

Sample graphs by size of data in tables

To delete a graph, click the icon corresponding to the graph color in the table (for example, or in the image above) or the icon in the legend to the right of the graphs.

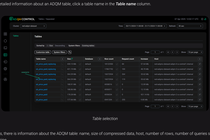
Filter and sort data

In the **Table name**, **Database**, **Host**, and **Engine** column headers, there are filters that you can use to select the necessary data on the **Tables** page. To set a filter, click the icon and enter a value you are looking for (a table/database/engine name or its part, or full names of hosts). The icon means that a filter is defined for the column. To reset all filters by columns, click .

Clicking on **System filters** opens a list of predefined filters that you can activate with the switch:

- **Show system tables** – display ADQM system tables from the **system** database.
- **Only existing tables** – display only tables that exist in the cluster at the last point of the specified time interval (tables for which ADQM Control received metrics at the end of this time period).
- **Only tables with graph** – display only the tables based on which graphs are currently plotted.

It is also possible to sort data on the **Tables** page by values in any column of the table with the list of ADQM tables. To apply sorting by a column, click the icon , or in its header. Above the list of tables, there is a panel that shows the sorting (**Sorted by**) and filtering (**System filters**, **Filtered by**) conditions currently applied to data.



Current conditions for sorting and filtering data on the Tables page

You can also view the list of the largest tables by data amount or the list of the most frequently queried tables in the ADQM cluster in the **Top 10 tables** section on the Dashboard page.

Table details

To open a separate page with detailed information about an ADQM table, click a table name in the **Table name** column.

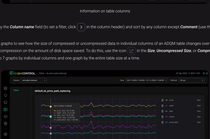
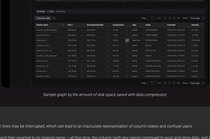


Table selection

At the top of the page that opens, there is information about the ADQM table: name, size of compressed data, host, number of rows, number of queries sent to the table, time of the last query, data size change, and engine.

This page also displays a table with the following fields that contain detailed information about each column of the selected ADQM table (to control the visibility of some fields, use the **Customize table** option).

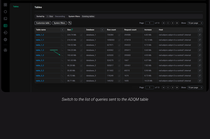
Field	Description
Column name	Column name at the end of the selected time interval
Size	Size of column data in compressed form
Uncompressed Size	Size of column data in uncompressed form
Compression	Percent of disk space saved by data compression
Type	Column data type
Codec	Data compression codecs applied to a column
Comment	Comment for a column
Increase	Change in size of column data over the selected time interval



Information on table columns

You can filter data in this table by the **Column name** field (to set a filter, click in the column header) and sort by any column except **Comment** (use the icons , and in column headers).

On the page, you can also build graphs to see how the size of compressed or uncompressed data in individual columns of an ADQM table changes over the selected time interval, so to analyze effect of applying data compression on the amount of disk space saved. To do this, use the icon in the **Size**, **Uncompressed Size** or **Compression** column, respectively. The graph area can display up to 7 graphs by individual columns and one graph by the entire table size at a time.



Sample graphs by the size of compressed data in individual columns of an ADQM table

Sample graph by the amount of disk space saved with data compression

CAUTION

In the following cases, graph lines may be interrupted, which can lead to an inaccurate representation of column states and confuse users:

- a column was renamed and then reverted to its original name – at this time, the column (with any name) continued to exist and store data, and its metrics were collected;
- a column was dropped and then re-created with the same name – in this case, there are two different columns existing at different time intervals.

The rendering of graphs for such cases will be fixed in future releases of ADQM Control.

Related queries

From the **Tables** page (as well as from the page with ADQM table details), you can access the list of queries sent to a specific ADQM table for a selected time interval. To do this, click the icon in the **Table name** (all corresponding to the desired data table – this opens the Queries page with automatically configured filters by table name, database, and host.

Switch to the list of queries sent to the ADQM table

Work with queries in ADQM Control

Anton Kolotovich

ADQM CONTROL

Contents

To Table of Contents

Fields of the Queries table

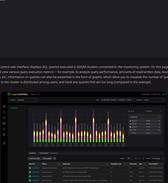
Sort and filter data

Query details

Graphs

- Duration
- User
- Kind

The **Queries** page in the ADQM Control web interface displays SQL queries executed in ClickHouse cluster(s) monitored by the monitoring system. On this page, you can get an overview of queries in the cluster, analyze their execution and see query execution metrics. For example, to analyze query performance, amount of resources that takes, resource usage, errors if they have occurred during query execution, etc. The information on queries can be presented in the form of graphs, which allow you to investigate the number of queries of different types, see how the total number of queries changes in time, find a dedicated among queries, and have a query that are too long (compared to the average).



The Queries page

At the top of the screen, you can see all the following filters to select which data the **Queries** page should display:

- **Cluster** - ADQM cluster name to analyze queries on;
- **Time** - time period when queries were launched, in the window that opens when you click the field, you can select an interval from the offered options on the **Range** tab or set a custom time range on the **Custom** tab;
- **Refresh** - frequency of data updates.

Query history is always displayed in the **Queries** page according to these filters - it contains all queries run on the selected ADQM cluster during the specified time period. For each DDL query (`ON CLUSTER`), it also includes the queries executed on each host of the cluster and displays the `JV_WC_entire-query-use_hl` entry (where the host text of such queries.

You can also get a list of the longest-running queries or queries that use system tables the most interfacing to the **Top 10 queries** section on the Overview page.

Fields of the Queries table

The table with the list of queries on the **Queries** page displays the following fields:

Field	Description
Query ID	Unique query identifier. The column also shows an icon indicating the query status: - is a query was successfully executed (the `Query finished` status); - is a query is being executed at the current time (the `Query started` status); - is an initial query of a query execution chain (the `Exception before start` status); - is an error occurred during a query execution (the `Exception while processing` status); - is a query executed on a cluster initiated by some requestor (the `Distributed` status).
Query text	Query text. If a query ends by `;` (semicolon) and what you think the mouse cursor over the column and, it is displayed as `Query start: hh:mm:ss` and `Query end: hh:mm:ss` - You can see the full text of a query on a separate page - To open it, click the query ID in the **Query ID** column.
Status	Current status of a query. Possible values: - `Query started` - a query execution is in progress; - `Query finished` - a query was successfully executed; - `Exception before start` - an error occurred before a query execution started; - `Exception while processing` - an error occurred during a query execution; - `Unexpected` - a query execution was not completed for some reason, i.e. there is no information in the log about the successful completion or error of the query execution (for example, if a host was restarted during the query execution).
Host	Host where a query was run.
Exception	Text of an error occurred before or during a query execution.
Exception code	Code of an error occurred before or during a query execution.
Frequency	How many times a query was executed in the cluster during the selected time period in the context of all filters currently applied on the page. Queries with the same query text completely matches (i.e. if parameterized queries, specific parameter values are taken into account).
Databases	Names of databases accessed by a query (there may be multiple - for example, if a query uses `JOIN`).
Tables	Names of tables accessed by a query (there may be multiple - for example, if a query uses `JOIN`).
Used memory	Amount of memory consumed by a query.
User	Name of an ADQM user who initiated a query execution. For queries executed on cluster hosts when running a distributed query, a user is not defined - the **User** column shows an empty string (`''`).
Time started	Query execution start time.
Duration	Duration of a query execution on the current execution host (`ORDER BY time_started`). If a query that has not yet been completed is selected, then the current time is used in the **Duration** field (which is replaced with the real query execution time since ADQM Control receives information about queries from ADQM the next time).
Average duration	Average query execution time. Statistics are calculated for queries whose text completely matches (i.e. for parameterized queries, specific parameter values are taken into account).
Read bytes	Total amount of data read from all tables and table functions participated in a query. For distributed queries, this is the number of bytes read on all nodes - each node reports its `Read bytes` value and the server initiator of the query sums all received and local values.
Result bytes	Amount of memory used by a query result.
Result rows	Number of rows in the result of a `SELECT` query or number of rows in an `INSERT` query.
Read rows	Total number of rows read from all tables and table functions participated in a query. For distributed queries, this is the total number of rows read on all nodes - each node reports its `Read rows` value and the server initiator of the query sums all received and local values.
Written rows	Number of rows written by an `INSERT` query (for other queries, the field value is `0`).
Written bytes	Amount of data written by an `INSERT` query (for other queries, the field value is `0`).
Type	Query type. Possible values: - `ALTER`, `TRUN CATE`, `INSERT`, `ALTER TABLE`, `DROP`, `UNDROP`, `RELOATE`, `RENAME`, `BRANCH`, `OPTIMIZE`, `GRANT`, `REVOKE`, `ATTACH`, `DETACH`, `SET`, `SYSTEM`, `KILL QUERY`, `KILLUSER`, `CREATE`, `DELETE`, `SET TRANSACTION`, `EXPLAIN`, `EXTRC`, `CHECK`, `FLUSH`, `EXTERNAL`, `SQL`, `UPDATE`; - `Unk ` - for `SELECT`, `EXPLAIN`, `DESCRIBE`, `SHOW`, `EXISTS`, `USE`, `EXECUTE` queries and queries that failed with the `SYNTAX_ERROR` or `INVALID_SYNTAX` errors.

To display the fields of the **Queries** table, click **Customize table** and select the desired fields. In the drop-down list The **Query ID** and **Query text** fields are always visible, they cannot be hidden.

Managing fields of the Queries table

Sort and filter data

In the column headers of the table with the list of queries, there are filters that you can use to select specific data. To open a filter, click the icon. Filters are available in the following columns:

- **Query text** - enter a truncated query text (as it is displayed in a tooltip that appears when you hover the mouse cursor over a column cell) or its fragment;
- **Databases**, **Tables** - select a value or a combination of values in the drop-down lists with complete databases or tables respectively (or enter a substring in the **Databases** / **Tables** column). Filtering works only if the search is performed by one item from a list of databases or tables (e.g. with);
- **Exception code** - enter a specific error code;
- **Host name** - you can enter names of multiple hosts or users, respectively. For queries initiated by a distributed query, filtering by an empty value in the **User** column is not supported;
- **Duration** - you can enter the minimum and maximum query execution time (in seconds);
- **Status**, **Type** - you can select a value in query status or query type from the drop-down list.

Through the icon, you can also enter a filtering expression in the opened window.

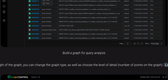
Data sorting is available in the following columns: **Exception** (actually sorting is performed by error code), **Exception code**, **Used memory**, **Time started**, **Duration**, **Read bytes**, **Result bytes**, **Read rows**, **Result rows**, **Written rows**, and **Written bytes**. To sort data by a column, click the icon in the column header.

Above the table with queries, there is a panel that shows the sorting (**Sorted by**) and filtering (**Filtered by**) conditions currently applied to data.

Current conditions for sorting and filtering data

Query details

To view detailed information on a query, click its identifier in the **Query ID** field of the **Queries** table.



Selecting a query

The **Query details** page that opens contains the full text of the selected query as well as the description of an error if it has occurred during the query processing. The page also displays query metrics in the columns selected on the **Queries** page via **Customize table**.

The Query details page

Graphs

On the **Queries** page, you can build graphs to analyze queries sent to your ADQM cluster tables during a specified period of time, based on the following metrics:

- **Duration** - query execution time;
- **User** - number of queries sent by particular users;
- **Kind** - number of queries of certain types.

To build a graph, click **Show graph** and select the desired graph type from the drop-down list.

Building a graph for query analysis

In the drop-down menu to the right of the graph, you can change the graph type, as well as choose the level of detail (number of points on the graph): `Low`, `High`.

Change the graph type and select the level of detail

Duration

The **Duration** graph shows the overall trend of query execution duration allowing you to find queries that have taken a long time to complete (relative to other queries in the same time interval and need to be analyzed).

Graph of queries by execution duration

This graph displays the following information on the execution time of queries sent to an ADQM cluster during a specified period of time:

- **95th Quantile** - 95th quantile of query execution time
- **Average** - average query execution time
- **Extreme ones** (one calculated in red if the **Average** line is above the **95th Quantile** line) - time interval when there were one or more long queries relative to most others. For example, in the graph above, such a zone is observed in the time range from `12:00` to `12:05`. In this zone, queries, one can see that on `06:00:00` there was a query with a duration of `20` (`s`), while all other queries were completed from 0 to 10 seconds (the value at the scale are sorted by the **Duration** column in descending order).

Calm zone (only between the **Average** and **95th Quantile** lines) - time interval when duration of all queries was comparable. A peak in this zone means that all queries in a given time interval have become longer to complete.

- **Query** - number of queries
- **No data** (was shaded in gray) - time interval for which there is no data about queries according to the applied filters. The **No data** item appears in the legend only if the graph contains one or more time intervals without any information.

Example of a graph with no data on queries

Graph including "No data" intervals

For the **Duration** graph, you can select the scale type: `Normal`, or `Log` (the logarithmic scale can be more convenient for large time intervals - for example, a month or a week).

User

The **User** graph allows you to see the number of queries sent to the cluster by each user and analyzes the number relative to the number of queries from other users and to the total of all queries. The graph is built using data on queries from users whose names are listed in the filter by the **User** column. You can also specify user names in the fields to the right of the graph the list of selected users is displayed in a tooltip with the filter by the **User** column.

Graph showing the number of queries from different users

Kind

On the **Kind** graph, you can see how many queries of different types were executed - select the required query type using the drop-down list to the right of the graph.

Graph showing the number of queries of different types

The graph is built according to the filters set in the table of queries. For example, the graph above shows the number of queries performed by the `SAMPLE` user for the selected time period.

SSL encryption

Contents

Create SSL certificates
Configure AQOM Control to enable SSL encryption
Manage SSL cluster wizard
Manage SSL cluster wizard
Configure AQOM Control to enable SSL encryption
Test a connection
Disable SSL

To hide table of contents click here

Create SSL certificates

To create a self-signed SSL certificate for your Kafka cluster, use the **openssl** command-line utility.

To create a root CA and SSL server certificate, run the following commands:

```
$ openssl req -new -x509 -keyout ca-key -out ca-cert -days 365
Generating a 4096 bit RSA private key
.....++++++
writing new private key to 'ca-key'
Enter PEM pass phrase:Verify - Enter PEM pass phrase:
Enter PEM pass phrase:Verify - Enter PEM pass phrase:
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:California
Locality Name (eg, city) []:Palo Alto
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Confluent
Organizational Unit Name (eg, section) []:Confluent
Common Name (e.g. server FQDN or YOUR name) []:ca1.test.confluent.io
Email Address []:
$ openssl genrsa -des3 -out server.key 4096
Generating RSA private key, 4096 bit long modulus
.....++++++
.....++++++
e is 65537 (0x10001)
Enter pass phrase for server.key:Verify - Enter pass phrase for server.key:
```

The first command creates a private key and certificate, and prompts for the passphrase and some Distinguished Name details that are used for the certificate's subject, issuer, and other attributes.

The second command creates a private key and certificate signing request (CSR) for the broker, which is used to request a certificate from the CA.

To create a keystore for the broker certificate, it is recommended to use the PKCS12 format, which is the default in recent versions of Java. Import the broker certificate into the keystore using the following command:

```
$ openssl pkcs12 -export -in server.crt -inkeystore server.keystore.p12 -name broker
Enter Export Password:Verify - Enter Export Password:
```

To create a truststore for the broker certificate, it is recommended to use the PKCS12 format. Import the CA certificate into the truststore using the following command:

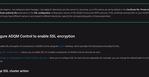
```
$ keytool -import -keystore server.truststore.p12 -alias CARoot -storetype PKCS12 -file ca-cert
Enter keystore password:Verify - Enter keystore password:
Owner: CN=ca1.test.confluent.io, O=Confluent, OU=Confluent, L=Palo Alto, ST=California, C=US
Issuer: CN=ca1.test.confluent.io, O=Confluent, OU=Confluent, L=Palo Alto, ST=California, C=US
Serial number: 8d0a0ad0ad0a0ad0
Valid from: Tue Jan 19 00:00:00 PST 2024 until: Wed Jan 19 00:00:00 PST 2026
Certificate fingerprints:
	SHA1: 1A:2B:3C:4D:5E:6F:7A:8B:9C:0D:1E:2F:3A:4B:5C:6D:7E:8F:9A:0B
	SHA256: 1A:2B:3C:4D:5E:6F:7A:8B:9C:0D:1E:2F:3A:4B:5C:6D:7E:8F:9A:0B:1C:2D:3E:4F:5A:6B:7C:8D:9E:0F:1A:2B
Extensions: #0: ObjectId: 2.5.29.35 ClassA: false
AuthorityKeyIidentifier [
KeyIdentifier [
0000: 1A 2B 3C 4D 5E 6F 7A 8B 9C 0D 1E 2F 3A 4B 5C 6D  .+<M^o z. ..  :K\m
0010: 7E 8F 9A 0B                                       ~... 
]
]
#1: ObjectId: 2.5.29.19 ClassA: false
BasicConstraints: [
CA:true
PathLen:2147483647
]
#2: ObjectId: 2.5.29.14 ClassA: false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 1A 2B 3C 4D 5E 6F 7A 8B 9C 0D 1E 2F 3A 4B 5C 6D  .+<M^o z. ..  :K\m
0010: 7E 8F 9A 0B                                       ~... 
]
]
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

Configure AQOM Control to enable SSL encryption

To enable SSL encryption, you must configure AQOM Control to use the Kafka cluster's SSL port and provide the required SSL keystore and truststore.

Manage SSL cluster wizard

Manage SSL cluster


Kafka cluster Manage SSL
Clusters
Cluster name
Kafka cluster
Kafka cluster
Search
Add
Configuration
Kafka cluster
Kafka cluster
Kafka cluster

Configuration parameters

Contents

AQMS Control
AQMS
Monitoring
Sensorbox

To Table of contents

The list of **Configuration parameters** contains all the parameters that can be configured in the AQMS platform using the **Configuration** section, the **Monitoring** section and the **Sensorbox** section.

Some of the parameters are only available in the AQMS platform for **Pro and enterprise** licensing level.

AQMS Control

Time configuration

Parameter	Description	Default value
ntpserver	Hostname of the NTP server	pool.ntp.org
ntpdate_period_seconds	Interval between two NTP requests, in seconds, to update the system clock	3600
ntpdate_debug_level	Verbosity level of the ntpdate program (from 0 to 9)	0
ntpdate_timeout_seconds	Timeout, in seconds, for the response from the NTP server	5
timezone_name	Name of the timezone used by the system, in the format "Continent/City" (e.g. "Europe/Rome", "America/New_York", "Asia/Tokyo")	Europe/UTC

Note:

- Both parameters are stored in the database table `system_properties` in the `aqmd` database under the `core` namespace.
- Changes to these parameters require a restart of the `aqmd-ntpdate` service to take effect. The `aqmd-ntpdate` service reads these values at startup and uses them to configure the NTP synchronization and system timezone.

```
INSERT INTO aqmd.system_properties (namespace, key, value, value_type, is_runtime_editable, is_required, description)
VALUES
('core', 'ntpdate_period_seconds', '3600', 'int', 1, 1, 1, 'Interval between two NTP requests, in seconds, to update the system clock'),
('core', 'ntpdate_debug_level', '0', 'int', 1, 1, 1, 'Verbosity level of the ntpdate program (from 0 to 9)');
```

Mqtt configuration

Note:
The broker parameters are configured under the `Pro` or `enterprise` licensing level in the `Sensorbox` section.

Parameter	Description	Default value
[broker] [name] = [hostname]	Defines a broker, identified by the configured name. Multiple brokers can be defined.	broker = localhost
[broker] [port] = [port]	Port for the connection to the broker	broker = 1883
[broker] [keepalive] = [seconds]	Keepalive interval, in seconds, for the connection to the broker	broker = 60
[broker] [tls] = [true/false]	Enables TLS/SSL encryption for the connection to the broker	broker = false
[broker] [tls_insecure] = [true/false]	Disables certificate verification when TLS is enabled (insecure mode)	broker = false
[broker] [cafile] = [path]	Path to the CA certificate file for TLS/SSL connections	broker =
[broker] [certfile] = [path]	Path to the client certificate file for TLS/SSL connections	broker =
[broker] [keyfile] = [path]	Path to the client private key file for TLS/SSL connections	broker =
[broker] [keepalive_period_seconds] = [seconds]	Interval, in seconds, between keepalive messages sent to the broker	broker = 60
[broker] [reconnect_delay_seconds] = [seconds]	Delay, in seconds, before attempting to reconnect after a connection loss	broker = 5
[broker] [max_inflight_messages] = [count]	Maximum number of messages in flight (sent but not yet acknowledged)	broker = 20
[broker] [max_queued_messages] = [count]	Maximum number of messages queued for delivery	broker = 1000
[broker] [protocol] = [3.1/3.1.1/5]	MQTT protocol version to use for the connection	broker = 3.1.1
[broker] [transport] = [tcp/websocket]	Transport layer to use for the connection	broker = tcp
[broker] [websocket_path] = [path]	HTTP path for the WebSocket connection (only applicable when transport = websocket)	broker = /mqtt
[broker] [publish_topic_prefix] = [prefix]	Prefix prepended to all topics published to this broker	broker =
[broker] [subscribe_topic] = [topic]	Topic to subscribe to on this broker	broker =
[broker] [publish_qos] = [0/1/2]	Default QoS level for publishing messages to this broker	broker = 0
[broker] [subscribe_qos] = [0/1/2]	Default QoS level for subscribing to this broker	broker = 0
[broker] [publish_retain] = [true/false]	Default retain flag for publishing messages to this broker	broker = false
[broker] [enabled] = [true/false]	Enables or disables this broker connection	broker = true
[broker] [priority] = [number]	Priority of this broker connection (used for ordering)	broker = 0
[broker] [description] = [text]	Optional human-readable description of this broker	broker =
[broker] [tls_ca_path] = [path]	Path to the directory containing CA certificates for TLS/SSL	broker =
[broker] [tls_cert_reqs] = [none/optional/required]	Level of certificate verification required from the broker	broker = required
[broker] [tls_ciphers] = [cipher_list]	Cipher suite to use for TLS/SSL connections	broker =
[broker] [tls_version] = [tlsv1.2/tlsv1.3]	Minimum TLS/SSL version to use	broker = tlsv1.2
[broker] [tls_psk] = [psk_identity:hexkey]	Pre-shared key for TLS/SSL in the format identity:hexkey	broker =
[broker] [tls_psk_ciphers] = [cipher_list]	Cipher suite to use for TLS-PSK connections	broker =
[broker] [tls_alpn] = [protocol]	ALPN protocol to negotiate with the broker	broker =
[broker] [tls_keylog] = [true/false]	Enables TLS/SSL key logging for debugging (writes session keys to a file)	broker = false
[broker] [tls_keylog_file] = [path]	Path to the file where TLS/SSL key material is logged	broker =

Network configuration

Parameter	Description	Default value
[network] [name] = [interface]	Defines a network interface, identified by the configured name. Multiple interfaces can be defined.	network = eth0
[network] [type] = [ethernet/wifi]	Interface type	network = ethernet
[network] [enabled] = [true/false]	Enables or disables this interface	network = true
[network] [priority] = [number]	Priority of this interface (used for ordering)	network = 0
[network] [description] = [text]	Optional human-readable description of this interface	network =
[network] [ipv4_mode] = [dhcp/static/disabled]	IPv4 addressing mode	network = dhcp
[network] [ipv4_address] = [cidr]	IPv4 address in CIDR notation (e.g. 192.168.1.10/24)	network =
[network] [ipv4_gateway] = [address]	IPv4 default gateway	network =
[network] [ipv4_dns] = [address1,address2]	IPv4 DNS servers (comma-separated)	network =
[network] [ipv6_mode] = [dhcp/static/disabled]	IPv6 addressing mode	network = disabled
[network] [ipv6_address] = [cidr]	IPv6 address in CIDR notation (e.g. 2001:db8::1/64)	network =
[network] [ipv6_gateway] = [address]	IPv6 default gateway	network =
[network] [ipv6_dns] = [address1,address2]	IPv6 DNS servers (comma-separated)	network =
[network] [wifi_ssid] = [ssid]	Wi-Fi network SSID	network =
[network] [wifi_psk] = [passphrase]	Wi-Fi pre-shared key (passphrase)	network =
[network] [wifi_security] = [none/wpa-psk/wpa-eap]	Wi-Fi security mode	network = wpa-psk
[network] [wifi_hidden] = [true/false]	Connect to a hidden Wi-Fi network	network = false
[network] [wifi_country] = [code]	Wi-Fi regulatory country code (ISO 3166-1 alpha-2)	network =
[network] [wifi_band] = [2.4ghz/5ghz/6ghz]	Wi-Fi frequency band	network =
[network] [wifi_channel] = [number]	Wi-Fi channel	network =
[network] [wifi_bssid] = [mac]	Wi-Fi BSSID (MAC address of the access point)	network =
[network] [wifi_mode] = [client/ap]	Wi-Fi operating mode	network = client
[network] [wifi_ap_ssid] = [ssid]	Wi-Fi AP mode SSID	network =
[network] [wifi_ap_psk] = [passphrase]	Wi-Fi AP mode pre-shared key (passphrase)	network =
[network] [wifi_ap_channel] = [number]	Wi-Fi AP mode channel	network =
[network] [wifi_ap_hidden] = [true/false]	Wi-Fi AP mode hidden SSID	network = false
[network] [wifi_ap_country] = [code]	Wi-Fi AP mode regulatory country code (ISO 3166-1 alpha-2)	network =
[network] [wifi_ap_band] = [2.4ghz/5ghz/6ghz]	Wi-Fi AP mode frequency band	network =
[network] [wireless_ap] = [name]	Associates this network with a wireless access point defined in the wireless section	network =

Wireless configuration

Parameter	Description	Default value
[wireless] [name] = [interface]	Defines a wireless access point, identified by the configured name. Multiple access points can be defined.	wireless = wlan0
[wireless] [driver] = [driver]	Wireless driver to use	wireless = nl80211
[wireless] [hw_mode] = [a/b/g/n/ac/ax]	Hardware mode	wireless = g
[wireless] [channel] = [number]	Wireless channel	wireless = 6
[wireless] [country_code] = [code]	Regulatory country code (ISO 3166-1 alpha-2)	wireless = US
[wireless] [ssid] = [ssid]	Network SSID	wireless = aqmd-ap
[wireless] [wpa_passphrase] = [passphrase]	Pre-shared key (passphrase)	wireless =
[wireless] [wpa_key_mgmt] = [wpa-psk/wpa-eap/sae]	Key management mode	wireless = wpa-psk
[wireless] [rsn_pairwise] = [ccmp/tkip/ccmp-256/gcmp/gcmp-256]	Pairwise cipher suite	wireless = ccmp
[wireless] [ieee80211n] = [0/1]	Enable 802.11n (HT)	wireless = 1
[wireless] [ieee80211ac] = [0/1]	Enable 802.11ac (VHT)	wireless = 1
[wireless] [ieee80211ax] = [0/1]	Enable 802.11ax (HE)	wireless = 1
[wireless] [ieee80211w] = [0/1/2]	Management frame protection (PMF/802.11w)	wireless = 1
[wireless] [ht_capab] = [capabilities]	HT capabilities	wireless = [HT40+][SHORT-GI-20][SHORT-GI-40]
[wireless] [vht_capab] = [capabilities]	VHT capabilities	wireless = [SHORT-GI-80]
[wireless] [bridge] = [interface]	Bridge interface to add the access point to	wireless = br-lan
[wireless] [disabled] = [0/1]	Disable this access point	wireless = 0

Ethernet configuration

Parameter	Description	Default value
[ethernet] [name] = [interface]	Defines an Ethernet interface, identified by the configured name. Multiple interfaces can be defined.	ethernet = eth0
[ethernet] [enabled] = [true/false]	Enables or disables this interface	ethernet = true
[ethernet] [priority] = [number]	Priority of this interface (used for ordering)	ethernet = 0
[ethernet] [description] = [text]	Optional human-readable description of this interface	ethernet =
[ethernet] [ipv4_mode] = [dhcp/static/disabled]	IPv4 addressing mode	ethernet = dhcp
[ethernet] [ipv4_address] = [cidr]	IPv4 address in CIDR notation (e.g. 192.168.1.10/24)	ethernet =
[ethernet] [ipv4_gateway] = [address]	IPv4 default gateway	ethernet =
[ethernet] [ipv4_dns] = [address1,address2]	IPv4 DNS servers (comma-separated)	ethernet =
[ethernet] [ipv6_mode] = [dhcp/static/disabled]	IPv6 addressing mode	ethernet = disabled
[ethernet] [ipv6_address] = [cidr]	IPv6 address in CIDR notation (e.g. 2001:db8::1/64)	ethernet =
[ethernet] [ipv6_gateway] = [address]	IPv6 default gateway	ethernet =
[ethernet] [ipv6_dns] = [address1,address2]	IPv6 DNS servers (comma-separated)	ethernet =
[ethernet] [wireless_ap] = [name]	Associates this network with a wireless access point defined in the wireless section	ethernet =

Modem configuration

Parameter	Description	Default value
[modem] [name] = [interface]	Defines a modem, identified by the configured name. Multiple modems can be defined.	modem = wwan0
[modem] [type] = [qmi/mbim/ppp/at]	Modem type	modem = qmi
[modem] [enabled] = [true/false]	Enables or disables this modem	modem = true
[modem] [priority] = [number]	Priority of this modem (used for ordering)	modem = 0
[modem] [description] = [text]	Optional human-readable description of this modem	modem =
[modem] [device] = [path]	Path to the modem device	modem = /dev/cdc-wdm0
[modem] [apn] = [apn]	Access Point Name (APN)	modem =
[modem] [pdp_type] = [ipv4/ipv6/ipv4v6]	PDP type	modem = ipv4v6
[modem] [pin] = [pin]	SIM PIN	modem =
[modem] [roaming] = [true/false]	Allow data roaming	modem = false
[modem] [ipv4_mode] = [dhcp/static/disabled]	IPv4 addressing mode	modem = dhcp
[modem] [ipv4_address] = [cidr]	IPv4 address in CIDR notation (e.g. 192.168.1.10/24)	modem =
[modem] [ipv4_gateway] = [address]	IPv4 default gateway	modem =
[modem] [ipv4_dns] = [address1,address2]	IPv4 DNS servers (comma-separated)	modem =
[modem] [ipv6_mode] = [dhcp/static/disabled]	IPv6 addressing mode	modem = disabled
[modem] [ipv6_address] = [cidr]	IPv6 address in CIDR notation (e.g. 2001:db8::1/64)	modem =
[modem] [ipv6_gateway] = [address]	IPv6 default gateway	modem =
[modem] [ipv6_dns] = [address1,address2]	IPv6 DNS servers (comma-separated)	modem =
[modem] [wireless_ap] = [name]	Associates this network with a wireless access point defined in the wireless section	modem =

Firewall configuration

Parameter	Description	Default value
[firewall] [name] = [rule]	Defines a firewall rule, identified by the configured name. Multiple rules can be defined.	firewall = allow-ssh
[firewall] [enabled] = [true/false]	Enables or disables this rule	firewall = true
[firewall] [priority] = [number]	Priority of this rule (used for ordering)	firewall = 0
[firewall] [description] = [text]	Optional human-readable description of this rule	firewall =
[firewall] [chain] = [input/output/forward]	Netfilter chain	firewall = input
[firewall] [protocol] = [tcp/udp/icmp/all]	Protocol	firewall = tcp
[firewall] [source] = [cidr]	Source address	firewall = 0.0.0.0/0
[firewall] [source_port] = [port]	Source port	firewall =
[firewall] [destination] = [cidr]	Destination address	firewall = 0.0.0.0/0
[firewall] [destination_port] = [port]	Destination port	firewall = 22
[firewall] [action] = [accept/drop/reject]	Action	firewall = accept
[firewall] [log] = [true/false]	Log matching packets	firewall = false
[firewall] [log_prefix] = [text]	Prefix for log messages	firewall =
[firewall] [rate_limit] = [rate]	Rate limit (e.g. 10/minute)	firewall =
[firewall] [rate_limit_burst] = [count]	Rate limit burst	firewall =
[firewall] [connlimit] = [count]	Connection limit	firewall =
[firewall] [icmp_type] = [type]	ICMP type (when protocol = icmp)	firewall =
[firewall] [ipset] = [name]	Match against an ipset	firewall =
[firewall] [mac] = [mac]	Source MAC address	firewall =
[firewall] [state] = [new/established/related/invalid]	Connection state	firewall =
[firewall] [time_start] = [hh:mm:ss]	Start time	firewall =
[firewall] [time_end] = [hh:mm:ss]	End time	firewall =
[firewall] [days] = [mon,tue,wed,thu,fri,sat,sun]	Days of the week	firewall =
[firewall] [date_start] = [yyyy-mm-dd]	Start date	firewall =
[firewall] [date_end] = [yyyy-mm-dd]	End date	firewall =
[firewall] [enabled_when] = [condition]	Condition for the rule to be active	firewall =
[firewall] [disabled_when] = [condition]	Condition for the rule to be inactive	firewall =

Logging configuration

Parameter	Description	Default value
[logrotate] [name] = [rule]	Defines a logrotate rule, identified by the configured name. Multiple rules can be defined.	logrotate = aqmd
[logrotate] [enabled] = [true/false]	Enables or disables this rule	logrotate = true
[logrotate] [priority] = [number]	Priority of this rule (used for ordering)	logrotate = 0
[logrotate] [description] = [text]	Optional human-readable description of this rule	logrotate =
[logrotate] [path] = [path]	Log file path or glob pattern	logrotate = /var/log/aqmd/*.log
[logrotate] [rotate] = [count]	Number of rotated files to keep	logrotate = 7
[logrotate] [size] = [size]	Rotate when file exceeds this size	logrotate = 100M
[logrotate] [compress] = [true/false]	Compress rotated files	logrotate = true
[logrotate] [delaycompress] = [true/false]	Delay compression by one rotation	logrotate = false
[logrotate] [copytruncate] = [true/false]	Copy and truncate instead of rotating	logrotate = false
[logrotate] [missingok] = [true/false]	Do not error if the file is missing	logrotate = true
[logrotate] [notifempty] = [true/false]	Do not rotate empty files	logrotate = false
[logrotate] [create] = [mode owner group]	Create new files after rotation	logrotate = 0640 root root
[logrotate] [dateext] = [true/false]	Use date extension in rotated file names	logrotate = false
[logrotate] [dateformat] = [format]	Date format for rotated file names	logrotate =
[logrotate] [prerotate] = [command]	Command to run before rotation	logrotate =
[logrotate] [postrotate] = [command]	Command to run after rotation	logrotate =
[logrotate] [firstaction] = [command]	Command to run once before all matching files	logrotate =
[logrotate] [lastaction] = [command]	Command to run once after all matching files	logrotate =

AQMS

Mqtt configuration

Parameter	Description	Default value
[broker] [name] = [hostname]	Defines a broker, identified by the configured name. Multiple brokers can be defined.	broker = localhost
[broker] [port] = [port]	Port for the connection to the broker	broker = 1883
[broker] [keepalive] = [seconds]	Keepalive interval, in seconds, for the connection to the broker	broker = 60
[broker] [tls] = [true/false]	Enables TLS/SSL encryption for the connection to the broker	broker = false
[broker] [tls_insecure] = [true/false]	Disables certificate verification when TLS is enabled (insecure mode)	broker = false
[broker] [cafile] = [path]	Path to the CA certificate file for TLS/SSL connections	broker =
[broker] [certfile] = [path]	Path to the client certificate file for TLS/SSL connections	broker =
[broker] [keyfile] = [path]	Path to the client private key file for TLS/SSL connections	broker =
[broker] [keepalive_period_seconds] = [seconds]	Interval, in seconds, between keepalive messages sent to the broker	broker = 60
[broker] [reconnect_delay_seconds] = [seconds]	Delay, in seconds, before attempting to reconnect after a connection loss	broker = 5
[broker] [max_inflight_messages] = [count]	Maximum number of messages in flight (sent but not yet acknowledged)	broker = 20
[broker] [max_queued_messages] = [count]	Maximum number of messages queued for delivery	broker = 1000
[broker] [protocol] = [3.1/3.1.1/5]	MQTT protocol version to use for the connection	broker = 3.1.1
[broker] [transport] = [tcp/websocket]	Transport layer to use for the connection	broker = tcp
[broker] [websocket_path] = [path]	HTTP path for the WebSocket connection (only applicable when transport = websocket)	broker = /mqtt
[broker] [publish_topic_prefix] = [prefix]	Prefix prepended to all topics published to this broker	broker =
[broker] [subscribe_topic] = [topic]	Topic to subscribe to on this broker	broker =
[broker] [publish_qos] = [0/1/2]	Default QoS level for publishing messages to this broker	broker = 0
[broker] [subscribe_qos] = [0/1/2]	Default QoS level for subscribing to this broker	broker = 0
[broker] [publish_retain] = [true/false]	Default retain flag for publishing messages to this broker	broker = false
[broker] [enabled] = [true/false]	Enables or disables this broker connection	broker = true
[broker] [priority] = [number]	Priority of this broker connection (used for ordering)	broker = 0
[broker] [description] = [text]	Optional human-readable description of this broker	broker =
[broker] [tls_ca_path] = [path]	Path to the directory containing CA certificates for TLS/SSL	broker =
[broker] [tls_cert_reqs] = [none/optional/required]	Level of certificate verification required from the broker	broker = required
[broker] [tls_ciphers] = [cipher_list]	Cipher suite to use for TLS/SSL connections	broker =
[broker] [tls_version] = [tlsv1.2/tlsv1.3]	Minimum TLS/SSL version to use	broker = tlsv1.2
[broker] [tls_psk] = [psk_identity:hexkey]	Pre-shared key for TLS/SSL in the format identity:hexkey	broker =
[broker] [tls_psk_ciphers] = [cipher_list]	Cipher suite to use for TLS-PSK connections	broker =
[broker] [tls_alpn] = [protocol]	ALPN protocol to negotiate with the broker	broker =
[broker] [tls_keylog] = [true/false]	Enables TLS/SSL key logging for debugging (writes session keys to a file)	broker = false
[broker] [tls_keylog_file] = [path]	Path to the file where TLS/SSL key material is logged	broker =

Network configuration

Parameter	Description	Default value
[network] [name] = [interface]	Defines a network interface, identified by the configured name. Multiple interfaces can be defined.	network = eth0
[network] [type] = [ethernet/wifi]	Interface type	network = ethernet
[network] [enabled] = [true/false]	Enables or disables this interface	network = true
[network] [priority] = [number]	Priority of this interface (used for ordering)	network = 0
[network] [description] = [text]	Optional human-readable description of this interface	network =
[network] [ipv4_mode] = [dhcp/static/disabled]	IPv4 addressing mode	network = dhcp
[network] [ipv4_address] = [cidr]	IPv4 address in CIDR notation (e.g. 192.168.1.10/24)	network =
[network] [ipv4_gateway] = [address]	IPv4 default gateway	network =
[network] [ipv4_dns] = [address1,address2]	IPv4 DNS servers (comma-separated)	network =
[network] [ipv6_mode] = [dhcp/static/disabled]	IPv6 addressing mode	network = disabled
[network] [ipv6_address] = [cidr]	IPv6 address in CIDR notation (e.g. 2001:db8::1/64)	network =
[network] [ipv6_gateway] = [address]	IPv6 default gateway	network =
[network] [ipv6_dns] = [address1,address2]	IPv6 DNS servers (comma-separated)	network =
[network] [wireless_ap] = [name]	Associates this network with a wireless access point defined in the wireless section	network =

Wireless configuration

Parameter	Description	Default value
[wireless] [name] = [interface]	Defines a wireless access point, identified by the configured name. Multiple access points can be defined.	wireless = wlan0
[wireless] [driver] = [driver]	Wireless driver to use	wireless = nl80211
[wireless] [hw_mode] = [a/b/g/n/ac/ax]	Hardware mode	wireless = g
[wireless] [channel] = [number]	Wireless channel	wireless = 6
[wireless] [country_code] = [code]	Regulatory country code (ISO 3166-1 alpha-2)	wireless = US
[wireless] [ssid] = [ssid]	Network SSID	wireless = aqmd-ap
[wireless] [wpa_passphrase] = [passphrase]	Pre-shared key (passphrase)	wireless =
[wireless] [wpa_key_mgmt] = [wpa-psk/wpa-eap/sae]	Key management mode	wireless = wpa-psk
[wireless] [rsn_pairwise] = [ccmp/tkip/ccmp-256/gcmp/gcmp-256]	Pairwise cipher suite	wireless = ccmp
[wireless] [ieee80211n] = [0/1]	Enable 802.11n (HT)	wireless = 1
[wireless] [ieee80211ac] = [0/1]	Enable 802.11ac (VHT)	wireless = 1
[wireless] [ieee80211ax] = [0/1]	Enable 802.11ax (HE)	wireless = 1
[wireless] [ieee80211w] = [0/1/2]	Management frame protection (PMF/802.11w)	wireless = 1
[wireless] [ht_capab] = [capabilities]	HT capabilities	wireless = [HT40+][SHORT-GI-20][SHORT-GI-40]
[wireless] [vht_capab] = [capabilities]	VHT capabilities	wireless = [SHORT-GI-80]
[wireless] [bridge] = [interface]	Bridge interface to add the access point to	wireless = br-lan
[wireless] [disabled] = [0/1]	Disable this access point	wireless = 0

Ethernet configuration

Parameter	Description	Default value
[ethernet] [name] = [interface]	Defines an Ethernet interface, identified by the configured name. Multiple interfaces can be defined.	ethernet = eth0
[ethernet] [enabled] = [true/false]	Enables or disables this interface	ethernet = true
[ethernet] [priority] = [number]	Priority of this interface (used for ordering)	ethernet = 0
[ethernet] [description] = [text]	Optional human-readable description of this interface	ethernet =
[ethernet] [ipv4_mode] = [dhcp/static/disabled]	IPv4 addressing mode	ethernet = dhcp
[ethernet] [ipv4_address] = [cidr]	IPv4 address in CIDR notation (e.g. 192.168.1.10/24)	ethernet =
[ethernet] [ipv4_gateway] = [address]	IPv4 default gateway	ethernet =
[ethernet] [ipv4_dns] = [address1,address2]	IPv4 DNS servers (comma-separated)	ethernet =
[ethernet] [ipv6_mode] = [dhcp/static/disabled]	IPv6 addressing mode	ethernet = disabled
[ethernet] [ipv6_address] = [cidr]	IPv6 address in CIDR notation (e.g. 2001:db8::1/64)	ethernet =
[ethernet] [ipv6_gateway] = [address]	IPv6 default gateway	ethernet =
[ethernet] [ipv6_dns] = [address1,address2]	IPv6 DNS servers (comma-separated)	ethernet =
[ethernet] [wireless_ap] = [name]	Associates this network with a wireless access point defined in the wireless section	ethernet =

Modem configuration

Parameter	Description	Default value
[modem] [name] = [interface]	Defines a modem, identified by the configured name. Multiple modems can be defined.	modem = wwan0
[modem] [type] = [qmi/mbim/ppp/at]	Modem type	modem = qmi
[modem] [enabled] = [true/false]	Enables or disables this modem	modem = true
[modem] [priority] = [number]	Priority of this modem (used for ordering)	modem = 0
[modem] [description] = [text]	Optional human-readable description of this modem	modem =
[modem] [device] = [path]	Path to the modem device	modem = /dev/cdc-wdm0
[modem] [apn] = [apn]	Access Point Name (APN)	modem =
[modem] [pdp_type] = [ipv4/ipv6/ipv4v6]	PDP type	modem = ipv4v6
[modem] [pin] = [pin]	SIM PIN	modem =
[modem] [roaming] = [true/false]	Allow data roaming	modem = false
[modem] [ipv4_mode] = [dhcp/static/disabled]	IPv4 addressing mode	modem = dhcp
[modem] [ipv4_address] = [cidr]	IPv4 address in CIDR notation (e.g. 192.168.1.10/24)	modem =
[modem] [ipv4_gateway] = [address]	IPv4 default gateway	modem =
[modem] [ipv4_dns] = [address1,address2]	IPv4 DNS servers (comma-separated)	modem =
[modem] [ipv6_mode] = [dhcp/static/disabled]	IPv6 addressing mode	modem = disabled
[modem] [ipv6_address] = [cidr]	IPv6 address in CIDR notation (e.g. 2001:db8::1/64)	modem =
[modem] [ipv6_gateway] = [address]	IPv6 default gateway	modem =
[modem] [ipv6_dns] = [address1,address2]	IPv6 DNS servers (comma-separated)	modem =
[modem] [wireless_ap] = [name]	Associates this network with a wireless access point defined in the wireless section	modem =

Firewall configuration

Parameter	Description	Default value
[firewall] [name] = [rule]	Defines a firewall rule, identified by the configured name. Multiple rules can be defined.	firewall = allow-ssh
[firewall] [enabled] = [true/false]	Enables or disables this rule	firewall = true
[firewall] [priority] = [number]	Priority of this rule (used for ordering)	firewall = 0
[firewall] [description] = [text]	Optional human-readable description of this rule	firewall =
[firewall] [chain] = [input/output/forward]	Netfilter chain	firewall = input
[firewall] [protocol] = [tcp/udp/icmp/all]	Protocol	firewall = tcp
[firewall] [source] = [cidr]	Source address	firewall = 0.0.0.0/0
[firewall] [source_port] = [port]	Source port	firewall =
[firewall] [destination] = [cidr]	Destination address	firewall = 0.0.0.0/0
[firewall] [destination_port] = [port]	Destination port	firewall = 22
[firewall] [action] = [accept/drop/reject]	Action	firewall = accept
[firewall] [log] = [true/false]	Log matching packets	firewall = false
[firewall] [log_prefix] = [text]	Prefix for log messages	firewall =
[firewall] [rate_limit] = [rate]	Rate limit (e.g. 10/minute)	firewall =
[firewall] [rate_limit_burst] = [count]	Rate limit burst	firewall =
[firewall] [connlimit] = [count]	Connection limit	firewall =
[firewall] [icmp_type] = [type]	ICMP type (when protocol = icmp)	firewall =
[firewall] [ipset] = [name]	Match against an ipset	firewall =
[firewall] [mac] = [mac]	Source MAC address	firewall =
[firewall] [state] = [new/established/related/invalid]	Connection state	firewall =
[firewall] [time_start] = [hh:mm:ss]	Start time	firewall =
[firewall] [time_end] = [hh:mm:ss]	End time	firewall =
[firewall] [days] = [mon,tue,wed,thu,fri,sat,sun]	Days of the week	firewall =
[firewall] [date_start] = [yyyy-mm-dd]	Start date	firewall =
[firewall] [date_end] = [yyyy-mm-dd]	End date	firewall =
[firewall] [enabled_when] = [condition]	Condition for the rule to be active	firewall =
[firewall] [disabled_when] = [condition]	Condition for the rule to be inactive	firewall =

Logging configuration

Parameter	Description	Default value
[logrotate] [name] = [rule]	Defines a logrotate rule, identified by the configured name. Multiple rules can be defined.	logrotate = aqmd
[logrotate] [enabled] = [true/false]	Enables or disables this rule	logrotate = true
[logrotate] [priority] = [number]	Priority of this rule (used for ordering)	logrotate = 0
[logrotate] [description] = [text]	Optional human-readable description of this rule	logrotate =
[logrotate] [path] = [path]	Log file path or glob pattern	logrotate = /var/log/aqmd/*.log
[logrotate] [rotate] = [count]	Number of rotated files to keep	logrotate = 7
[logrotate] [size] = [size]	Rotate when file exceeds this size	logrotate = 100M
[logrotate] [compress] = [true/false]	Compress rotated files	logrotate = true
[logrotate] [delaycompress] = [true/false]	Delay compression by one rotation	logrotate = false
[logrotate] [copytruncate] = [true/false]	Copy and truncate instead of rotating	logrotate = false
[logrotate] [missingok] = [true/false]	Do not error if the file is missing	logrotate = true
[logrotate] [notifempty] = [true/false]	Do not rotate empty files	logrotate = false
[logrotate] [create] = [mode owner group]	Create new files after rotation	logrotate = 0640 root root
[logrotate] [dateext] = [true/false]	Use date extension in rotated file names	logrotate = false
[logrotate] [dateformat] = [format]	Date format for rotated file names	logrotate =
[logrotate] [prerotate] = [command]	Command to run before rotation	logrotate =
[logrotate] [postrotate] = [command]	Command to run after rotation	logrotate =
[logrotate] [firstaction] = [command]	Command to run once before all matching files	logrotate =
[logrotate] [lastaction] = [command]	Command to run once after all matching files	logrotate =

Monitoring

Processor Monitoring

Parameter	Description	Default value
[cpu] [enabled] = [true/false]	Enable/disable the frequency for cpu usage	true
[cpu] [history] = [seconds]	Specifies the history value, with a unit (seconds or minutes)	15s
[cpu] [alarm] = [true/false]	Enable/disable the alarm for cpu usage	true
[cpu] [alarm_threshold] = [percentage]	Specifies the threshold for the alarm for cpu usage	90
[cpu] [alarm_duration] = [seconds]	Specifies the duration for the alarm for cpu usage	5
[cpu] [alarm_cooldown] = [seconds]	Specifies the cooldown for the alarm for cpu usage	60
[cpu] [alarm_command] = [path]	Specifies the path to the command to be executed when the alarm for cpu usage is triggered	
[cpu] [alarm_command_parameters] = [parameters]	Specifies the parameters for the command to be executed when the alarm for cpu usage is triggered	

Memory Monitoring

Parameter	Description	Default value
[memory] [enabled] = [true/false]	Enable/disable the frequency for memory usage	true
[memory] [history] = [seconds]	Specifies the history value, with a unit (seconds or minutes)	15s
[memory] [alarm] = [true/false]	Enable/disable the alarm for memory usage	true
[memory] [alarm_threshold] = [percentage]	Specifies the threshold for the alarm for memory usage	90
[memory] [alarm_duration] = [seconds]	Specifies the duration for the alarm for memory usage	5
[memory] [alarm_cooldown] = [seconds]	Specifies the cooldown for the alarm for memory usage	60
[memory] [alarm_command] = [path]	Specifies the path to the command to be executed when the alarm for memory usage is triggered	
[memory] [alarm_command_parameters] = [parameters]	Specifies the parameters for the command to be executed when the alarm for memory usage is triggered	

Disk Monitoring

Parameter	Description	Default value
[disk] [enabled] = [true/false]	Enable/disable the frequency for disk usage	true
[disk] [history] = [seconds]	Specifies the history value, with a unit (seconds or minutes)	15s
[disk] [alarm] = [true/false]	Enable/disable the alarm for disk usage	true
[disk] [alarm_threshold] = [percentage]	Specifies the threshold for the alarm for disk usage	90
[disk] [alarm_duration] = [seconds]	Specifies the duration for the alarm for disk usage	5
[disk] [alarm_cooldown] = [seconds]	Specifies the cooldown for the alarm for disk usage	60
[disk] [alarm_command] = [path]	Specifies the path to the command to be executed when the alarm for disk usage is triggered	
[disk] [alarm_command_parameters] = [parameters]	Specifies the parameters for the command to be executed when the alarm for disk usage is triggered	

ADQM Control overview

Elena Dvoryadkina

Contents

To Table of Contents

Features

Architecture

Features

Arenadata QuickMarts Control (ADQM Control) is an observability platform for Arenadata QuickMarts (ADQM) clusters that allows you to significantly simplify the administration and optimize the operation of ADQM databases.

Key objectives of ADQM Control are the following:

- Monitor the health of an ADQM cluster. ADQM Control can work with multiple ADQM clusters simultaneously.
- Collect and analyze system metrics and metrics of ADQM services.
- Visualize the states of ADQM cluster hosts as a **heat map**.
- Generate **alerts**. An alert is a notification message that describes a problem detected in an ADQM cluster and provides recommendations how to fix it. ADQM Control groups alerts by triggers that cause them:
  - threshold-based alerts that are generated when some metric in the system has reached the specified threshold;
  - event-based alerts that are generated when some event has happened in the system.

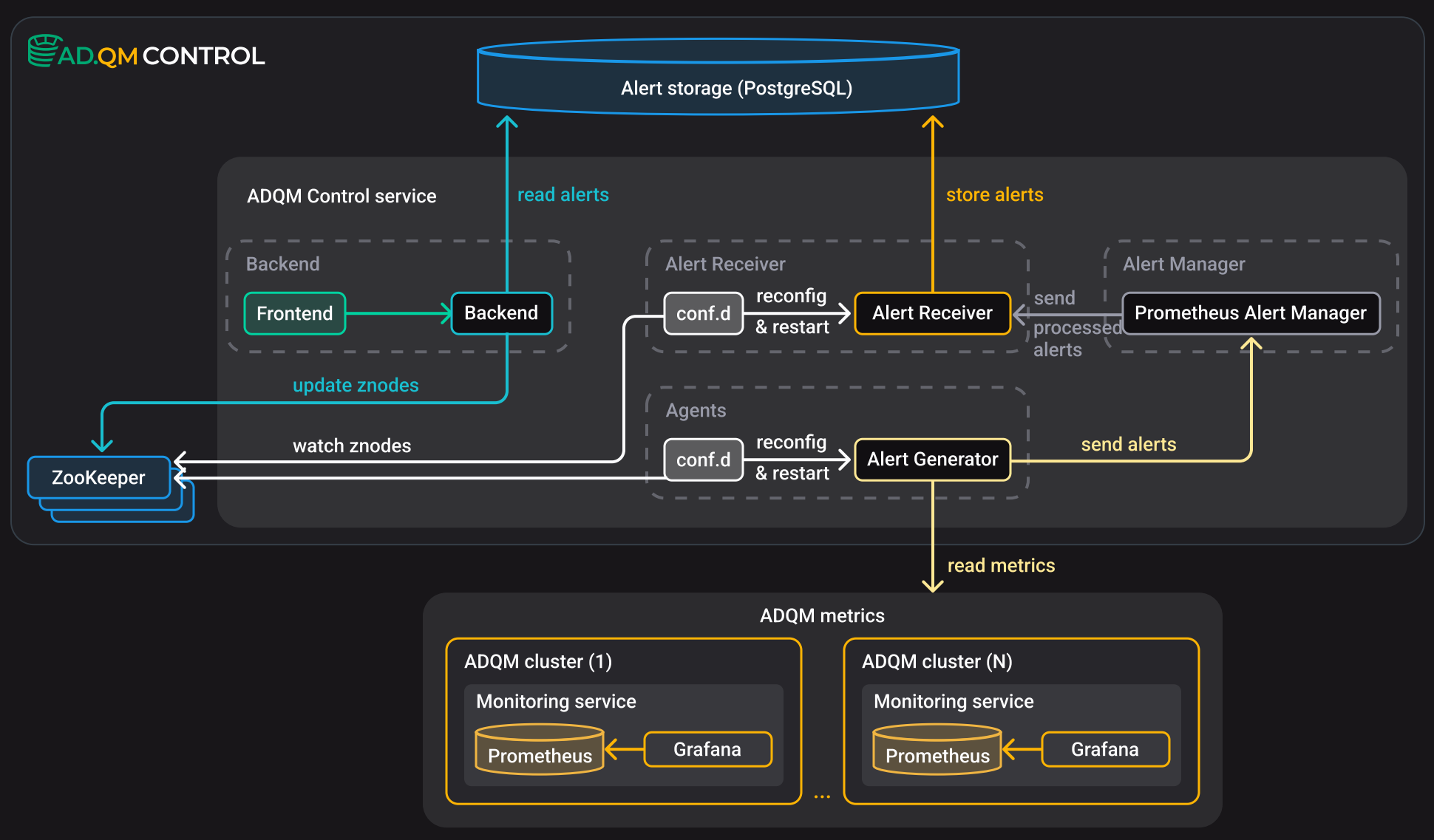
  In the ADQM Control user interface, you can specify criteria for generating alerts of different types (for example, configure thresholds, set timeouts) or disable tracking of some events.

NOTE

ADQM Control requires a fully deployed ADQM cluster with the monitoring service installed.

Architecture

The high-level architecture view of ADQM Control is shown below.



ADQM Control architecture

Main points of the scheme:

- **ADQM Control** is the main service that implements the full functionality of ADQM Control. It includes the following components:
  - **Agents.** The Alert Generator service reads metrics from a Prometheus instance of an ADQM cluster, generates alerts if necessary (taking into account criteria and timeouts for creating alerts specified in ADQM Control settings), sends alerts to Alert Manager.
  - **Alert Manager.** The Prometheus Alert Manager service handles alerts received from Alert Generator (filters alerts, mutes alerts of some type, adds additional labels, etc.) and resends processed alerts to Alert Receiver.
  - **Alert Receiver.** The Alert Receiver service receives processed alerts from Alert Manager and stores them in the alert database.
  - **Backend.** The Backend service communicates with the ADQM Control web interface (Frontend in the scheme) via REST API — handles user requests from the frontend, saves specified ADQM Control settings to ZooKeeper, retrieves alerts from the alert database to transmit them to the web interface.

  The Agents and Alert Receiver components also include the conf.d service, which monitors ADQM Control settings stored in ZooKeeper — if they change, it updates the configuration file of the Alert Generator/Alert Receiver service, and restarts the service.
- **ZooKeeper** is the ZooKeeper coordination service for storing user-defined settings of ADQM Control. To set up ZooKeeper for ADQM Control, you can:
  - install the Zookeeper service in the ADQM Control cluster;
  - use an external ZooKeeper cluster.
- **Alert storage** is a PostgreSQL database for storing alerts. You can set up it in one of the following ways:
  - install the Arenadata PostgreSQL service in the ADQM Control cluster;
  - specify an external PostgreSQL cluster using ADQM Control configuration parameters.

Release notes

Elena Dvoryadkina

ADQM CONTROL

Contents

To Table of Contents

1.1.0

1.0.0

1.1.0

Date: 24.07.2025

New features

- Introduced the ***Duration***, ***Kind***, and ***User*** graphs for queries
- Introduced incidents
- Added support for SSL in all services of ADQM Control
- Added the ***Comment*** field for ADQM table columns on the Table details page
- Implemented a new alert on missing the `system.query_log` table
- Added the threshold label to ADQM Control alerts
- Implemented acknowledgement for alerts
- Added multi-user, query duration, query text, and exception filters on the ***Queries*** page
- Added search filters on the Users page
- Extended ***Heat map*** with statistics information for each host
- Refactored paginator on the ***Queries*** page by removing the page counter

1.0.0

Date: 17.12.2024

New features

- Introduced ADQM Control services, including Agents, Alert generator, Alert manager, Alert receiver, and Backend
- Introduced basic functionality for managing users within ADQM Control
- Introduced the Dashboard page providing an overview of the physical cluster state, alerts, top 10 tables, and top 10 queries
- Introduced the Tables page for displaying table information over a specified time window
- Introduced functionality for viewing detailed information about columns within tables over a specified time window
- Introduced charts for tracking table metrics over a defined time period
- Introduced charts for tracking column-level metrics over a defined time period
- Introduced the Queries page that displays queries providing sorting and filtering options
- Introduced the Alerts history page that displays a full list of alerts providing sorting and filtering options
- Introduced the Heat map view for alerts — a visual representation of alert distribution over time
- Introduced the Settings page for configuring ADQM Control parameters, such as password complexity and rules for generating alerts
- Introduced functionality for using the ADPG service or an external PostgreSQL instance as a database for ADQM Control
- Introduced functionality for using either the built-in Zookeeper service or an external ZooKeeper instance as a coordination service

Contents

To Table of Contents

ADPG

ADPG

Parameter	Description	Default value
Data directory	Directories used to store data on ADPG hosts	/pg_data1

ADPG configurations

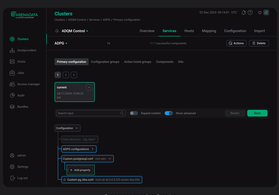
TIP

You can obtain the optimal parameter values to achieve the best ADPG performance using the **Collect recommendations** action of the ADPG service.

Parameter	Description	Default value
listen_addresses	TCP/IP addresses on which the server listens for connections from client applications	*
port	TCP port the server listens on	5432
max_connections	Maximum number of concurrent connections to the database server	100
shared_buffers	Amount of memory the database server uses for shared memory buffer. A reasonable starting value is 25% of the memory in your system	2048 MB
max_worker_processes	Maximum number of background processes that the system can support	23
max_parallel_workers	Maximum number of workers that the system can support for parallel operations	4
max_parallel_workers_per_gather	Maximum number of workers that can be started by a single Gather or Gather Merge node	2
max_parallel_maintenance_workers	Maximum number of parallel workers that can be started by a single utility command	2
effective_cache_size	Sets the ADPG query planner's assumption about the effective size of the disk cache available to a single query. This is factored into estimates of the cost of using an index. The higher this value, the more likely the index scan will be applied. The lower this value, the more likely the sequential scan will be selected	6144 MB
maintenance_work_mem	Memory limit for maintenance operations. Increasing the value for this parameter can improve the performance of operations for cleaning and restoring the database from a copy	1024 MB
work_mem	Non-shared memory limit used for internal operations when the server executes queries (for example, queries to temporary tables) before temporary files on the disk are used. This limit is also a simple resource control, preventing the host from going into swapping due to overallocation. Since the allocated memory is non-shared memory, large and complex requests can exceed the specified limit	5242 kB
min_wal_size	Until WAL disk usage stays below the `min_wal_size` value, old WAL files are recycled for future use at a checkpoint, rather than removed. This ensures that enough WAL space is reserved to handle spikes in WAL usage, for example, when running large batch jobs	1024 MB
max_wal_size	Memory limit to which the log size can grow between automatic checkpoints. Increasing this setting may increase the recovery time after a failure. The specified limit can be exceeded automatically with a high load on ADPG	4096 MB
wal_keep_size	Specifies the minimum size of segments retained in the pg_wal directory, in case a standby server needs to fetch them for streaming replication. If a standby server connected to a sending server falls behind by more than `wal_keep_size` MB, the sending server might remove a WAL segment still needed by the standby. In this case, the replication connection is terminated. Downstream connections also fail as a result. If WAL archiving is enabled, the standby server can fetch the segment from archive and recover. The `wal_keep_size` parameter sets only the minimum size of segments retained in pg_wal. The system might need to retain more segments for WAL archival or to recover from a checkpoint. If `wal_keep_size` is set to `0`, the system does not keep any extra segments for standby purposes, so the number of old WAL segments available to standby servers depends on the location of the previous checkpoint and status of WAL archiving	0 MB
huge_pages	Defines whether huge pages can be requested for the main shared memory area. Possible values: • `try` – the server tries to request huge pages. If this fails, the server falls back to the default page • `on` – if a huge page request fails, the server does not start. • `off` – huge pages are not used.	try
superuser_reserved_connections	Number of connection 'slots' that are reserved for PostgreSQL superuser connections	3
timescaledb.max_background_workers	Maximum number of background worker processes allocated to TimescaleDB	16
wal_buffers	Amount of shared memory used for WAL data that has not yet been written to disk	16 MB
default_statistics_target	Default statistics target for table columns without a column-specific target set via `ALTER TABLE_SET STATISTICS`	100
random_page_cost	Planner's estimate of the cost of a non-sequentially-fetched disk page	1.1
checkpoint_completion_target	Target time of checkpoint completion as a fraction of total time between checkpoints	0.9
max_locks_per_transaction	The shared lock table tracks locks on `max_locks_per_transaction * (max_connections + max_prepared_transactions)` objects (for example, tables). Hence, no more than this number of distinct objects can be locked at any one time. This parameter controls the average number of object locks allocated for each transaction. Individual transactions can lock more objects as long as the locks of all transactions fit in the lock table	128
autovacuum_max_workers	Maximum number of autovacuum processes (excluding the autovacuum launcher) that are allowed to run at the same time	10
autovacuum_naptime(seconds)	Minimum delay between autovacuum runs on any given database. In each round, the autovacuum daemon examines the database and executes the `VACUUM` and `ANALYZE` commands if it is needed for tables in that database	10
default_toast_compression	Default TOAST compression method for values of a compressible column. Possible values: `lz4`, `pglz`	lz4
effective_io_concurrency	Number of concurrent disk I/O operations that can be executed simultaneously. Raising this value will increase the number of I/O operations that a PostgreSQL session attempts to initiate in parallel	256
jit	Specifies whether PostgreSQL may use JIT compilation, if it is available	off
logging_collector	Enables the logging collector — a background process that captures log messages sent to stderr and redirects them into log files	Enabled
log_directory	Directory that contains log files. It can be specified as an absolute path, or relative to the ADPG cluster data directory	log (the absolute path is /pg_data1/adpg14/log)
log_filename	Log file name pattern. A value can include strftime %-escapes to define time-varying file names. If you specify a file name pattern without escapes, use a log rotation utility to save disk space	postgresql-%a.log
log_rotation_age	Maximum period of time to use a log file, after which a new log file is created. If this value is specified without units, it is taken as minutes. Set `log_rotation_age` to `0` to disable time-based log file creation	1d
log_rotation_size	Maximum size of a log file. After a log file reaches the specified size, a new log file is created. If the value is set without units, it is taken as kilobytes. Set `log_rotation_size` to `0` to disable size-based log file creation	0
log_min_messages	Minimum severity level of messages to be written to a log file. Possible values are `debug5`, `debug4`, `debug3`, `debug2`, `debug1`, `info`, `notice`, `warning`, `error`, `log`, `fatal`, and `panic` (see the Severity levels table). Messages with the specified severity or higher are included in the log file. For example, if you set `log_min_messages` to `warning`, the log file will include the `warning`, `error`, `log`, `fatal`, and `panic` messages	warning
log_min_error_statement	Specifies which SQL statements that cause errors are logged. Possible values are `debug5`, `debug4`, `debug3`, `debug2`, `debug1`, `info`, `notice`, `warning`, `error`, `log`, `fatal`, and `panic` (see the Severity levels table). The log file includes SQL statements for messages of the specified severity or higher. To disable error statement logging, set `log_min_error_statement` to `panic`	error

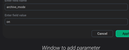
Custom postgresql.conf

In the **Custom postgresql.conf** section, you can set additional parameters to be written to the postgresql.conf file. To add a parameter, click **Add property**



Custom postgresql.conf section

In the window that appears, enter a parameter (its name and value) and click **Apply**

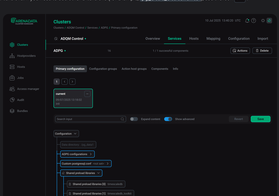


Window to add parameter

You can add multiple parameters.

Shared preload libraries

Section for setting the `shared_preload_libraries` parameter of the postgresql.conf configuration. ADGM Control requires the `timescaledb` and `timescaledb_toolkit` extensions. If you need to add other extensions (for example, `pg_stat_statements`), use the icon < 1 >.



The "Shared preload libraries" section

Custom pg_hba.conf

This field allows you to add records to the pg_hba.conf file that controls client authentication.

SSL configuration

NOTE

For more information on SSL encryption support in ADGM Control, see the SSL encryption article.

Parameter	Description	Default value
Enable SSL	Enables SSL encryption of incoming traffic for the ADPG service	Disabled
Certificate file	Path to the server SSL certificate file	/etc/adpg16/server.crt
Private key file	Path to the file with the private key of the server SSL certificate	/etc/adpg16/server.key
Certificate authority file	Path to the CA certificate file	/etc/adpg16/ca.crt

Contents

To Table of Contents

ADQM Control

ADQM Control

ADQM management

Parameter	Description	Default value
ADQM username	Username to connect to ADQM	adqmc
ADQM user password	Password for the `ADQM username` user to connect to ADQM	—
ADQM default port	Default port if it is not exported from an ADQM cluster	9000
ADQM secure port	Default port for SSL-secured communication with ADQM if it is not exported from an ADQM cluster	9440
Connection type	Type of connection (insecure or secure) between ADQM Control and ADQM. Possible values: `Insecure (TCP)`, `Secure (TCP secure)`	Insecure (TCP)

NOTE

A name and password passed to the `ADQM username` and `ADQM user password` parameters should belong to a user account that exists in ADQM. For a new cluster, ADQM provides the `default` user account without a password by default — you can use this account in ADQM Control, but first assign a password to it. However, it is recommended to create a separate user account in ADQM under which ADQM Control will connect to ADQM, for example:

```
CREATE USER adqmc ON CLUSTER default_cluster IDENTIFIED WITH sha256_password BY 'qwerty123';
```

```
GRANT ON CLUSTER default_cluster SELECT ON *.* TO adqmc;
```

SSL configuration

NOTE

For details on how to configure and enable SSL for components of ADQM Control's services, see the SSL encryption article.

Parameter	Description	Default value
[ADQM Control's Backend] → Enable SSL	Enables SSL for the Backend component	Disabled
[ADQM Control's ADQM Agent] → Enable SSL	Enables SSL for the Agents component	Disabled
[ADQM Control's Alert Generator] → Enable SSL	Enables SSL for the Alert Generator component	Disabled
[ADQM Control's Alert Receiver] → Enable SSL	Enables SSL for the Alert Receiver component	Disabled
[Alertmanager] → Enable SSL	Enables SSL for the Alertmanager component	Disabled
[ADQM Control Services] → Certificate file	Path to the server SSL certificate file for the Backend, Agents, Alert Generator, Alert Receiver components	/etc/adqmc/server.crt
[ADQM Control Services] → Private key file	Path to the file with the private key of the server SSL certificate for the Backend, Agents, Alert Generator, Alert Receiver components	/etc/adqmc/server.key
[ADQM Control Services] → Certificate authority file	Path to the file with the CA certificate for the Backend, Agents, Alert Generator, Alert Receiver components	/etc/adqmc/ca.crt
[Alertmanager] → Certificate file	Path to the server SSL certificate file for the Alertmanager component	/etc/admprom/alert-manager/server.crt
[Alertmanager] → Private key file	Path to the file with the private key of the server SSL certificate for the Alertmanager component	/etc/admprom/alert-manager/server.key
[Alertmanager] → Certificate authority file	Path to the file with the CA certificate for the Alertmanager component	/etc/admprom/alert-manager/ca.crt

Network configuration

Parameter	Description	Default value
Backend port	Port to access the ADQM Control web interface	5555
Alert Generator port	Port the alert generator listens on	5001
ADQM agent port	Port the Agents service listens on	5002
Alertmanager port	Port the alert manager listens on	9093
Alert Receiver port	Port the alert receiver listens on	12322

Connection settings

Parameter	Description	Default value
Backend connection pool size	Maximum number of connections to a PostgreSQL database of the Backend service	8
ADQM agent connection pool size	Maximum number of connections to a PostgreSQL database of the Agents service	8

Polling rate

Parameter	Description	Default value
Confd update interval	Time interval between checking for changes in the configuration files of services (in seconds)	30
Hosts scrape interval	Frequency of updating information about the ADQM cluster topology (in seconds)	80
ADQMDB scrape interval	Frequency of updating information about tables in the ADQM cluster (in seconds)	80
Query update interval	Frequency of collecting queries (in seconds)	105
Query normalize interval	Frequency of updating queries (in seconds)	120

Limits

Parameter	Description	Default value
Normalize query raw max queries per iteration	Maximum number of queries normalized per iteration	50000
Collect queries limit	Number of queries collected from the query_log per iteration	10000
Query normalize threshold	Number of attempts to normalize a query	10
Existing tables tries	Number of attempts to get metrics of an ADQM table. If ADQM Control has failed to retrieve the metrics for a table after the expiration of the time `ADQMDB scrape interval * Existing tables tries`, it considers this table deleted	3

Internal PostgreSQL server

NOTE

To access the configuration parameters, activate the ***Use internal PostgreSQL server*** option.

Parameter	Description	Default value
Alerts dbname	Name of a database to be created for use by ADQM Control services	adqmc
Alertsdb username	Name of a user with access to the `Alerts dbname` database	adqmc
Alertsdb password	Password for the `Alertsdb username` user to access the `Alerts dbname` database	—

External PostgreSQL server

NOTE

To access the configuration parameters, activate the ***Use external PostgreSQL server*** option.

Parameter	Description	Default value
server_ip_port	Click ***Add property*** to add the `host` and `port` properties that define an IP address and port to access an external PostgreSQL server	—
alerts_dbname	Database name to be used by ADQM Control services	adqmc
alertsdb_username	Name of a user with access to the `alerts_dbname` database	adqmc
alertsdb_password	Password for the `alerts db_username` user to access the `alerts_dbname` database	—
SSL is used by an external server	Specifies whether SSL is used by an external server	Disabled

Zookeeper configuration

Parameter	Description	Default value
zk_type	Indicates which ZooKeeper cluster ADQM Control uses — internal (installed as the Zookeeper service of the ADQM Control cluster) or external. Possible values are `Internal Zookeeper`, `External Zokeeper`	Internal Zookeeper
zk_hosts	List of external Zookeeper cluster hosts (for example: `host1.internal:2181,host2.internal:2181,host3.internal:2181`). It is required if `zk_type` is set to `External Zokeeper`	—
zk_prefix	Prefix for all paths to znodes in ZooKeeper related to ADQM Control	/arenadata/adqm_control/<cluster_id>
SSL connection	Turns on secure connection to ZooKeeper	Disabled

Contents

To Table of Contents

Monitoring

Monitoring

Prometheus settings

Parameter	Description	Default value
scrape_interval	Specifies how frequently to scrape targets	1m
scrape_timeout	Specifies how long to wait until a scrape request times out	10s
Password for grafana connection	Password of a Grafana user to connect to Prometheus	—
Prometheus users to login/logout to Prometheus	User credentials for logging into the Prometheus web interface	—
Service parameters	Prometheus parameters	—

Grafana settings

Parameter	Description	Default value
Grafana administrator's password	Password of a Grafana administrator user	—
Grafana listen port	Port to access the Grafana web interface	11210

Node Exporter settings

Parameter	Description	Default value
Listen port	Port to listen for a host's system metrics in the Prometheus format	11203
Metrics endpoint	Endpoint to retrieve system metrics	/metrics

SSL configuration

Parameter	Description	Default value
[Prometheus] → Enable SSL	Enables SSL encryption of incoming traffic for the Prometheus Server component	Disabled
[Prometheus] → Certificate file	Path to the server SSL certificate file for the Prometheus Server component	/etc/admprom/prometheus/server.crt
[Prometheus] → Private key file	Path to the file with the private key of the server SSL certificate for the Prometheus Server component	/etc/admprom/prometheus/server.key
[Prometheus] → Certificate authority file	Path to the file with the CA certificate to verify Prometheus targets	/etc/admprom/prometheus/ca.crt
[Grafana] → Enable SSL	Enables SSL encryption of incoming traffic for the Grafana component	Disabled
[Grafana] → Certificate file	Path to the server SSL certificate file for the Grafana component	/etc/admprom/grafana/server.crt
[Grafana] → Private key file	Path to the file with the private key of the server SSL certificate for the Grafana component	/etc/admprom/grafana/server.key
[Grafana] → Certificate authority file	Path to the file with the CA certificate to verify Prometheus certificate	/etc/admprom/grafana/ca.crt
[Node-exporter] → Enable SSL	Enables SSL encryption of incoming traffic for the Node Exporter component	Disabled
[Node-exporter] → Certificate file	Path to the server SSL certificate file for the Node Exporter component	/etc/ssl/server.crt
[Node-exporter] → Private key file	Path to the file with the private key of the server SSL certificate for the Node Exporter component	/etc/ssl/server.key

Contents

To Table of Contents

Zookeeper

Zookeeper

Main

Parameter	Description	Default value
connect	ZooKeeper connection string used by other services or clusters. It is generated automatically	—
dataDir	Location where ZooKeeper stores the in-memory database snapshots and, unless specified otherwise, the transaction log of updates to the database. It is specified as the `dataDir` parameter in the *zoo.cfg* configuration file	/var/lib/zookeeper

zoo.cfg

Parameter	Description	Default value
clientPort	Port to listen for client connections, that is the port that clients attempt to connect to	2181
admin.serverPort	Port that the embedded Jetty server listens on	5181
admin.enableServer	Enables Admin server — an embedded Jetty server that provides an HTTP interface to the four-letter-word commands	Disabled
tickTime	Length of a single tick (in milliseconds) which is the basic time unit used by ZooKeeper to regulate heartbeats and timeouts	2000
initLimit	Amount of time (in ticks) to allow followers to connect and sync with a leader. Increase this value as needed if the amount of data managed by ZooKeeper is large	5
syncLimit	Amount of time (in ticks) to allow followers to sync with ZooKeeper. If followers fall too far behind a leader, they will be dropped	2
maxClientCnxns	Limits the number of active connections from a single client, identified by IP address, to a single member of the ZooKeeper ensemble	0
autopurge.snapRetainCount	When enabled, ZooKeeper auto-purge feature retains the specified number of the most recent snapshots and the corresponding transaction logs in *dataDir* and *dataLogDir* respectively, and deletes the rest. The minimum value is `3`	3
autopurge.purgeInterval	Time interval (in hours), for which the purge task has to be triggered. Set a positive integer to enable the auto purging	24
Add key,value	In this section, you can define values for custom parameters that are not displayed in ADCM UI, but are allowed in the *zoo.cfg* configuration file	—

SSL configuration

Parameter	Description	Default value
Enable SSL	Enables secure incoming connections in ZooKeeper	Disabled
sslQuorum	Enables encrypted quorum communication	Disabled
serverCnxnFactory	Specifies `ServerCnxnFactory` implementation. To use TLS-based server communication, set it to `NettyServerCnxnFactory`	org.apache.zookeeper.server.NettyServerCnxnFactory
ssl.quorum.keyStore.location	Fully-qualified path to the server keystore file	/etc/zookeeper/server.jks
ssl.quorum.keyStore.password	Password for the keystore specified in `ssl.quorum.keyStore.location`	—
ssl.quorum.trustStore.location	Fully-qualified path to the server truststore file	/etc/zookeeper/ca.jks
ssl.quorum.trustStore.password	Password for the truststore specified in `ssl.quorum.trustStore.location`	—
ssl.keyStore.location	Fully-qualified path to the server keystore file to be used for client TLS connections	/etc/zookeeper/server.jks
ssl.keyStore.password	Password for the keystore specified in `ssl.keyStore.location`	—
ssl.trustStore.location	Fully-qualified path to the server truststore file to be used for client TLS connections	/etc/zookeeper/ca.jks
ssl.trustStore.password	Password for the truststore specified in `ssl.trustStore.location`	—
ssl.protocol	Protocol to be used in client TLS negotiation	TLSv1.2
ssl.quorum.protocol	Protocol to be used in quorum TLS negotiation	TLSv1.2
secureClientPort	Port to be used for secure client connections	2281

Metrics

Parameter	Description	Default value
Listen port	Port to listen for Zookeeper service metrics	9020

zookeeper-env.sh

Parameter	Description	Default value
JAVA	Path to Java	$JAVA_HOME/bin/java
ZOOPIDFILE	Directory to store the ZooKeeper process ID	/var/run/zookeeper/zookeeper_server.pid
ZOO_LOG_DIR	Directory to store logs	/var/log/zookeeper
SERVER_JVMFLAGS	Used to set different JVM parameters (for example, parameters related to garbage collection)	-Xmx1024m