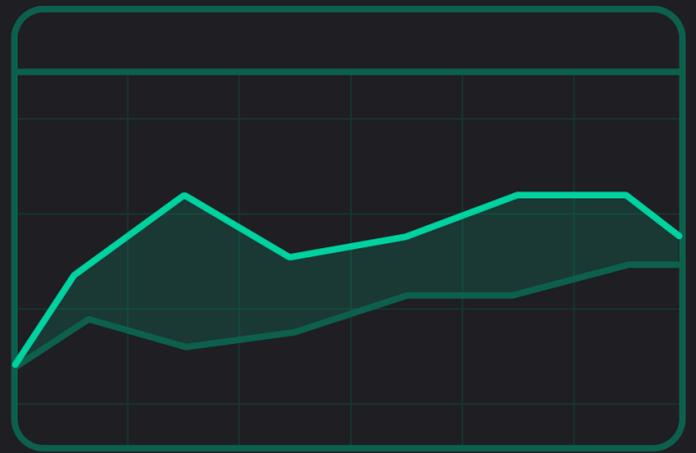


Arenadata QuickMarts Control

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 [1.0.0](#) 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](#)

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](#)

3. [References](#)

- └ [Configuration parameters](#)

4. [Concepts](#)

- └ [ADQM Control overview](#)

5. [Release notes](#)

[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.

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

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.

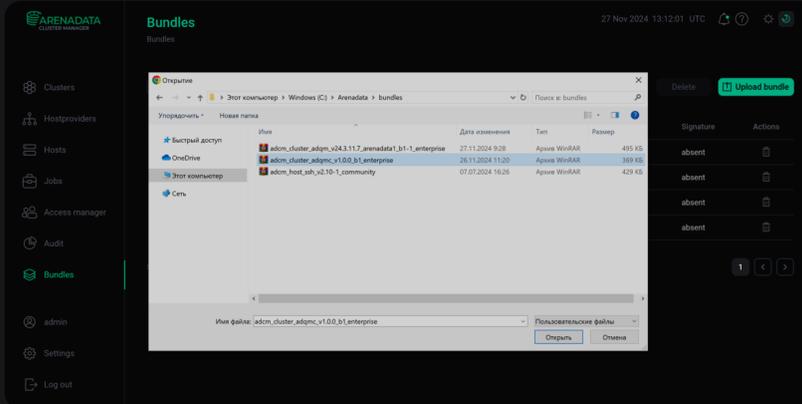
Step 1. Get the Enterprise version of a cluster bundle

Please get a cluster bundle from the Arenadata support team.

Step 2. Upload a cluster bundle to ADCM

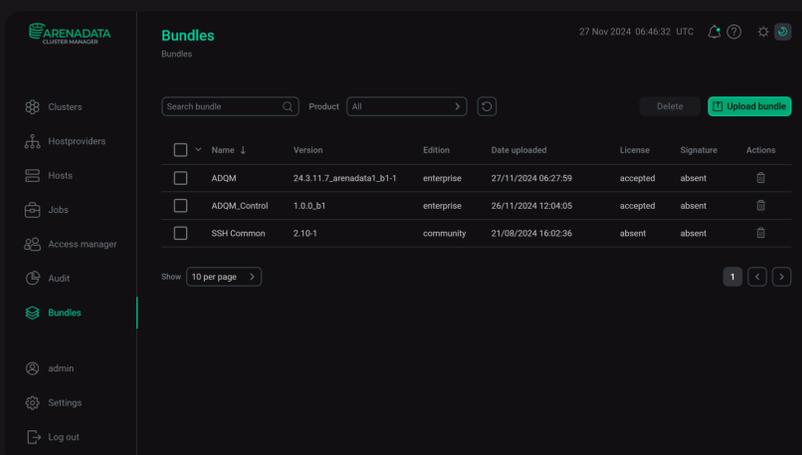
To upload a bundle to ADCM, follow the steps:

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

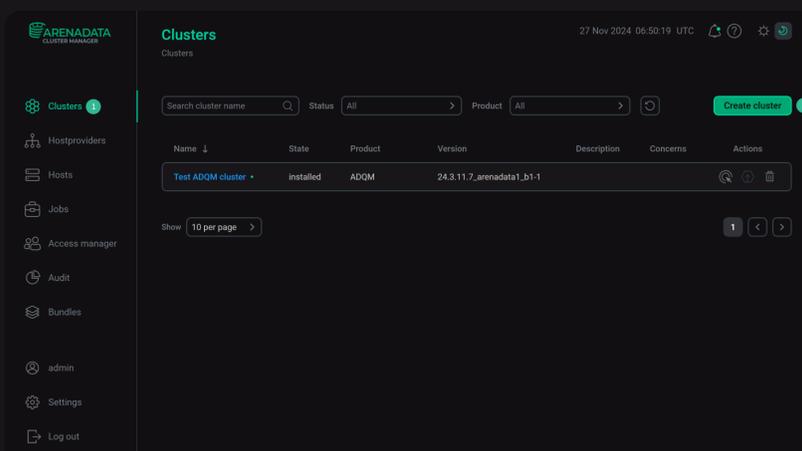
IMPORTANT

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:

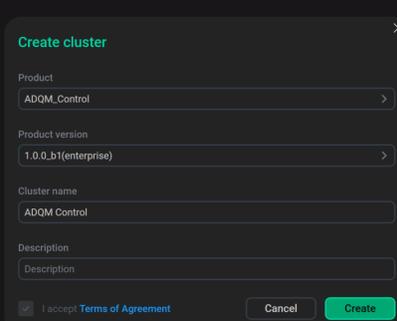
1. Select the **Clusters** item in the left navigation menu and click **Create cluster**.



Switch to cluster creation

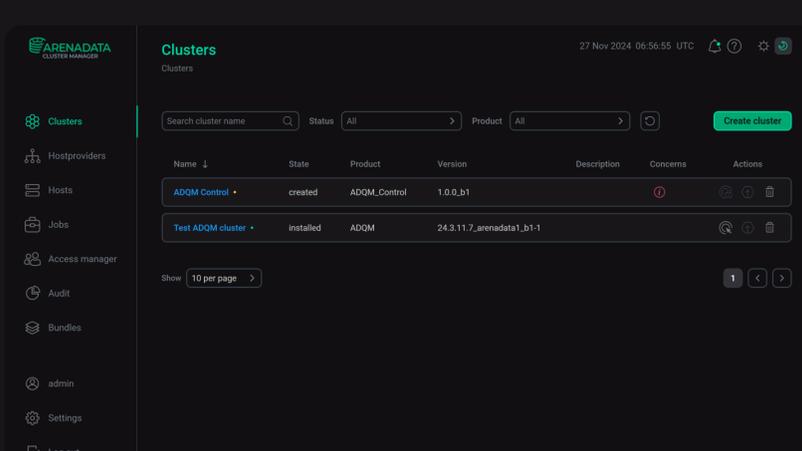
2. In the opened dialog, describe a new cluster instance:

- 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.

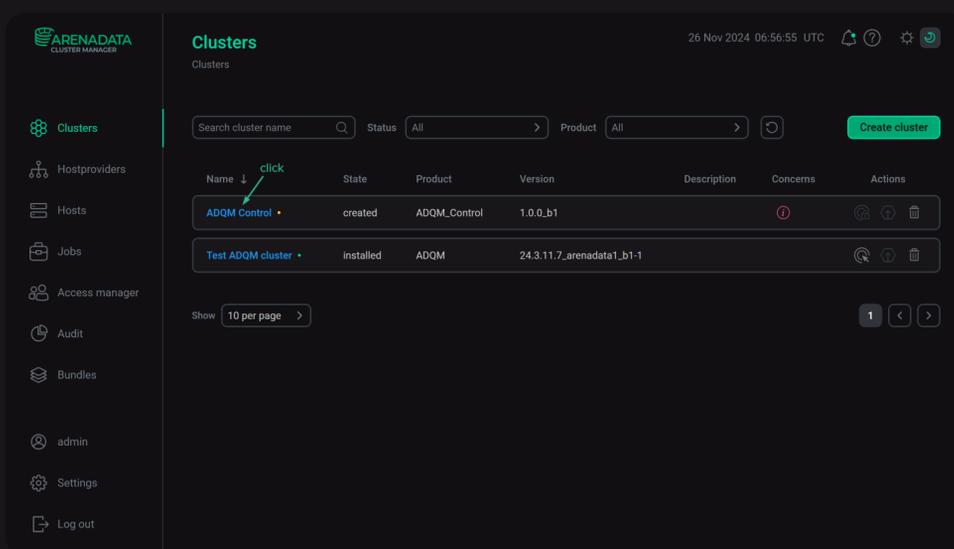


The result of successful cluster creation

[To Table of Contents](#)

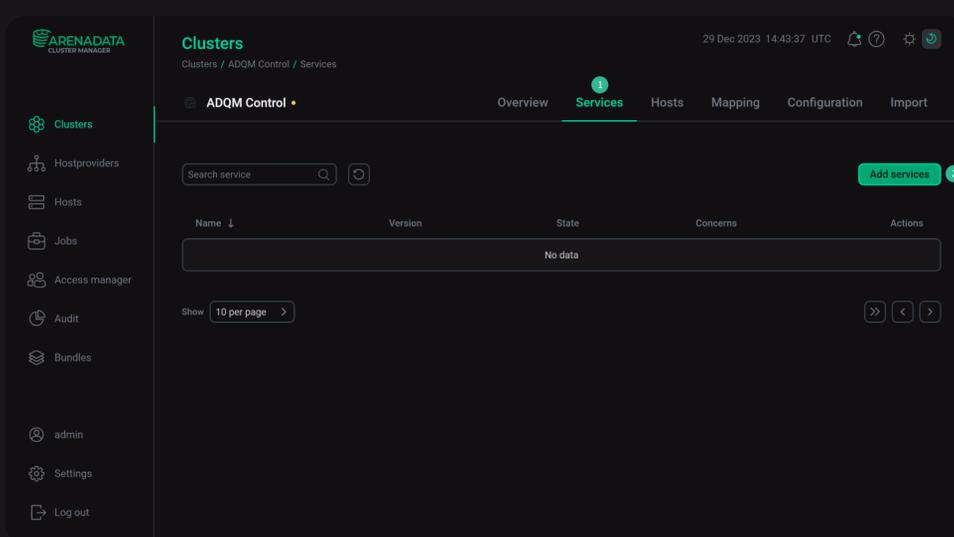
In ADCM a **service** means a software that performs some function. The steps for adding services to a cluster are listed below:

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



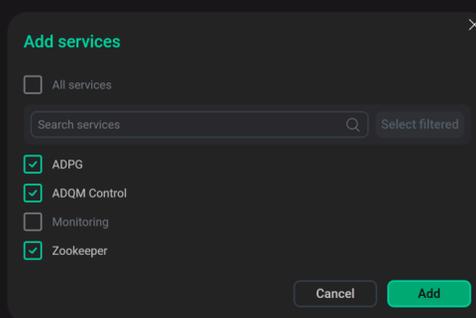
Select a cluster

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



Switch to adding services

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



Select services

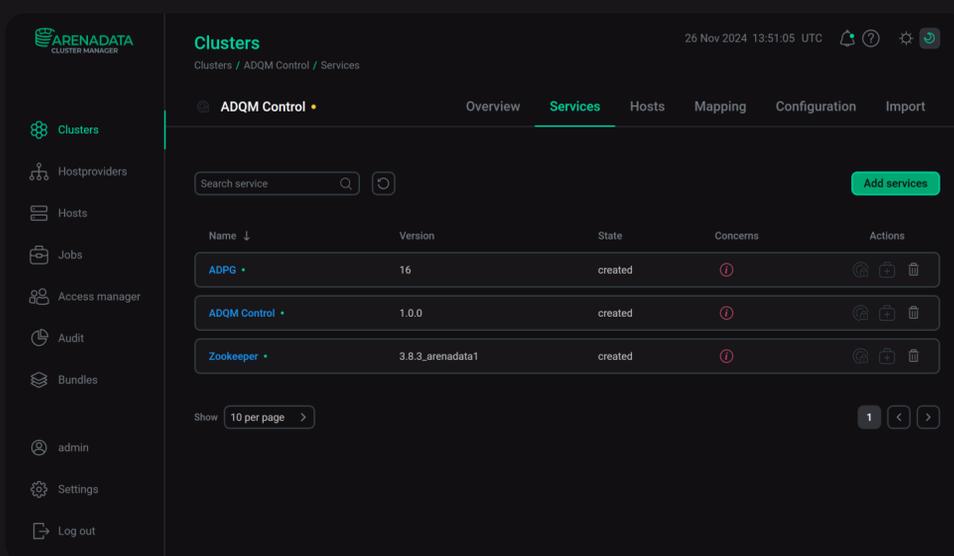
The brief description of available services is listed below.

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

[To Table of Contents](#)

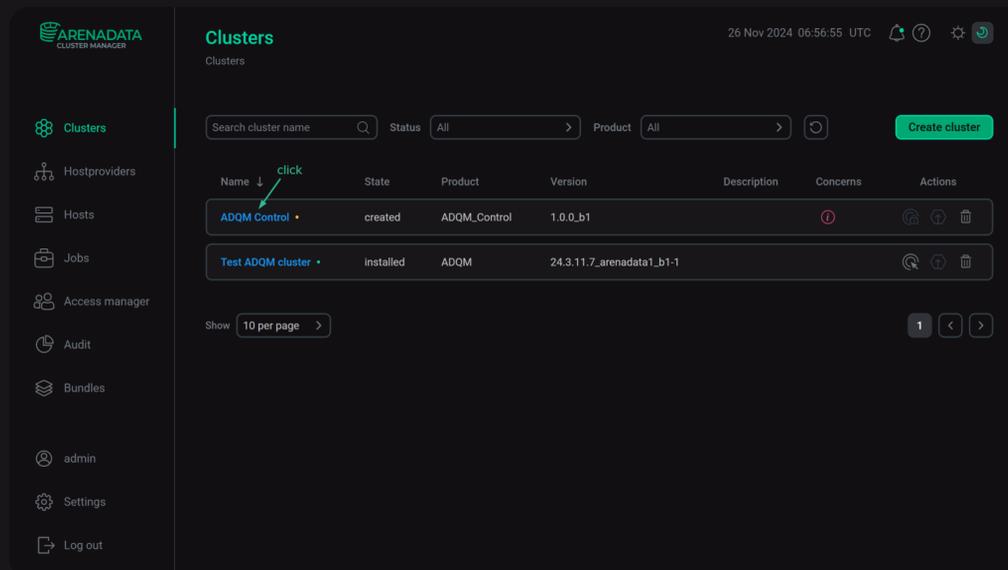
NOTE

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.



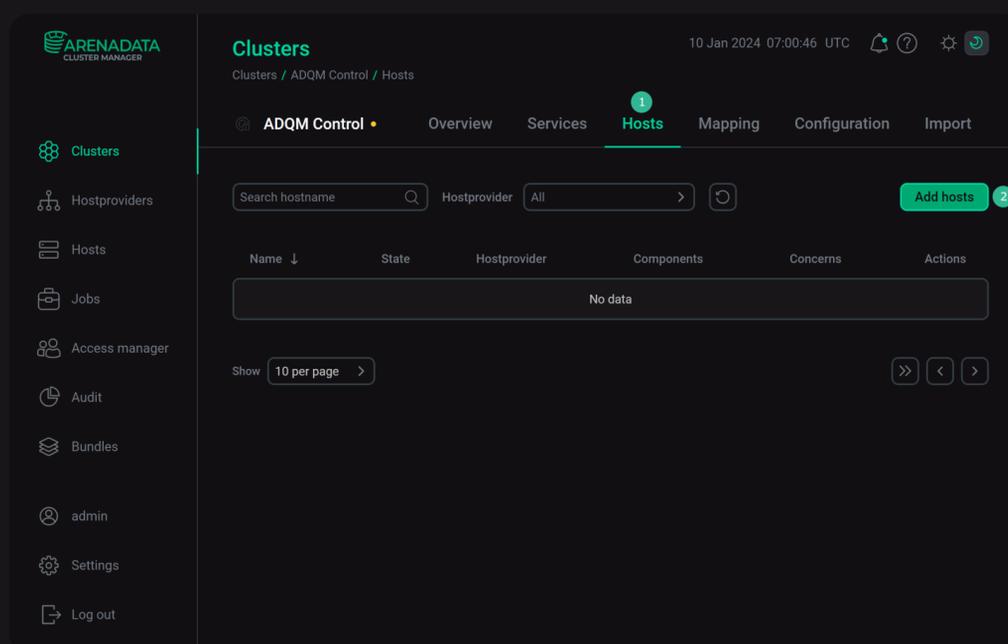
The steps for adding hosts to a cluster are listed below:

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



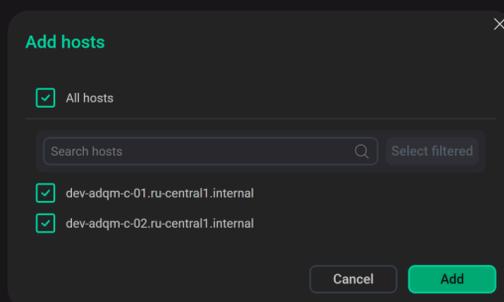
Select a cluster

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



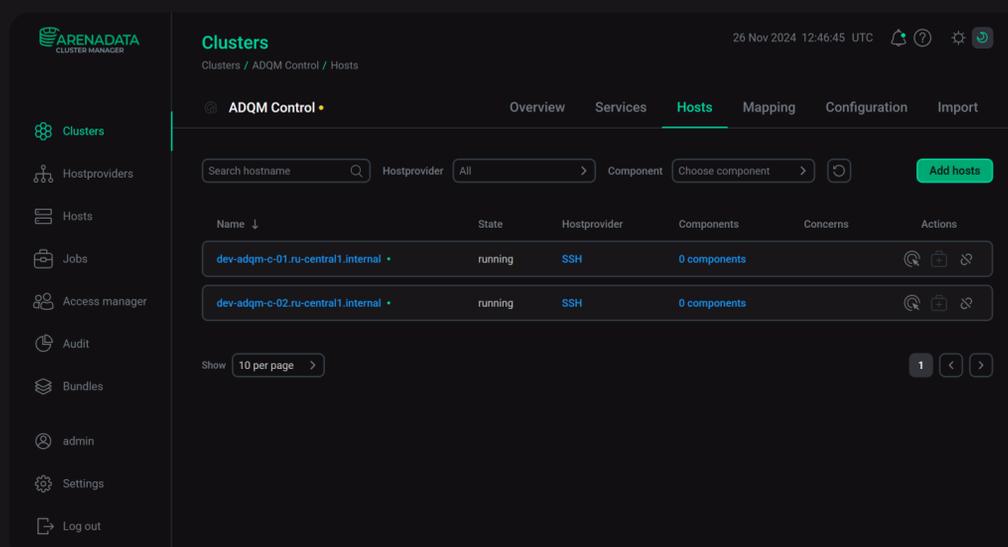
Switch to adding hosts

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



Select hosts

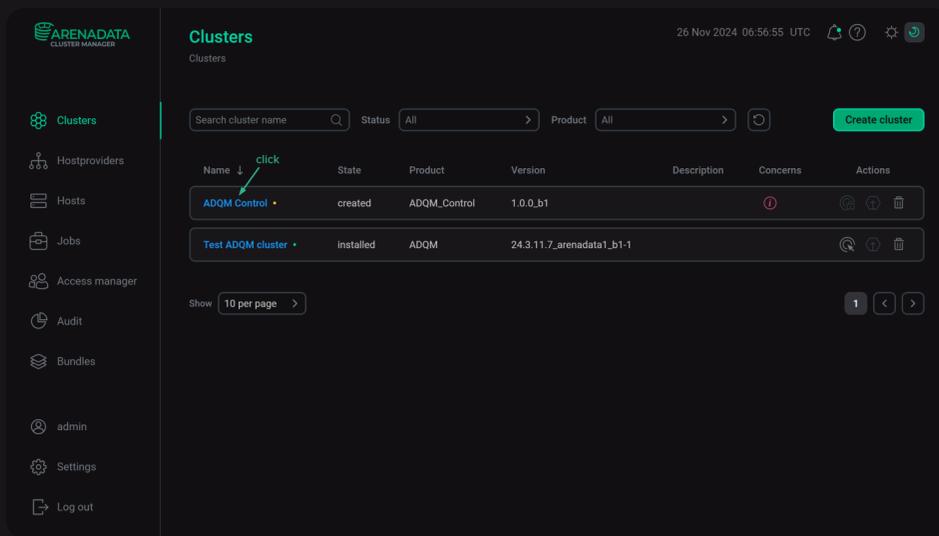
4. As a result, the added hosts are displayed on the *Hosts* tab.



The result of successful adding hosts to a cluster

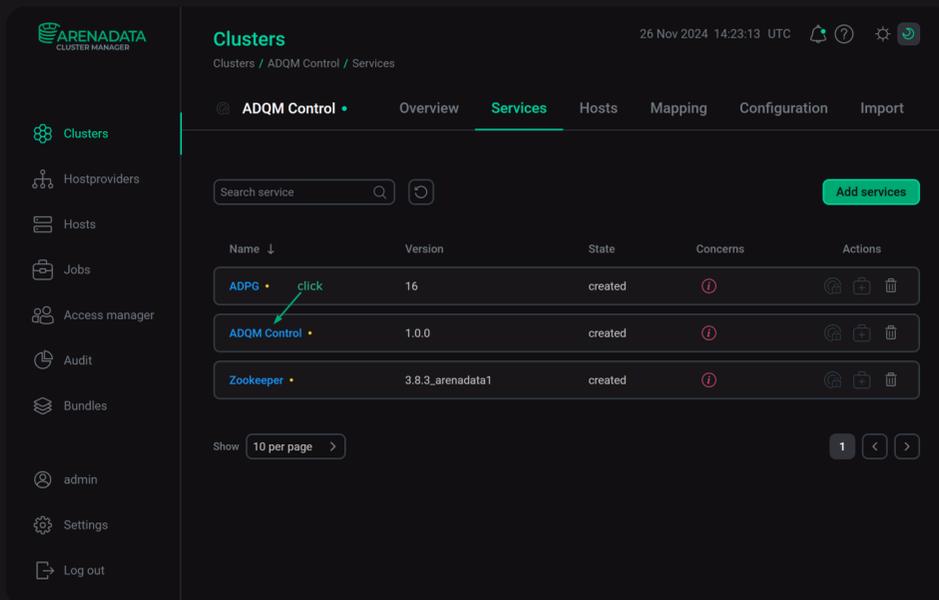
[To Table of Contents](#)

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



Select a cluster

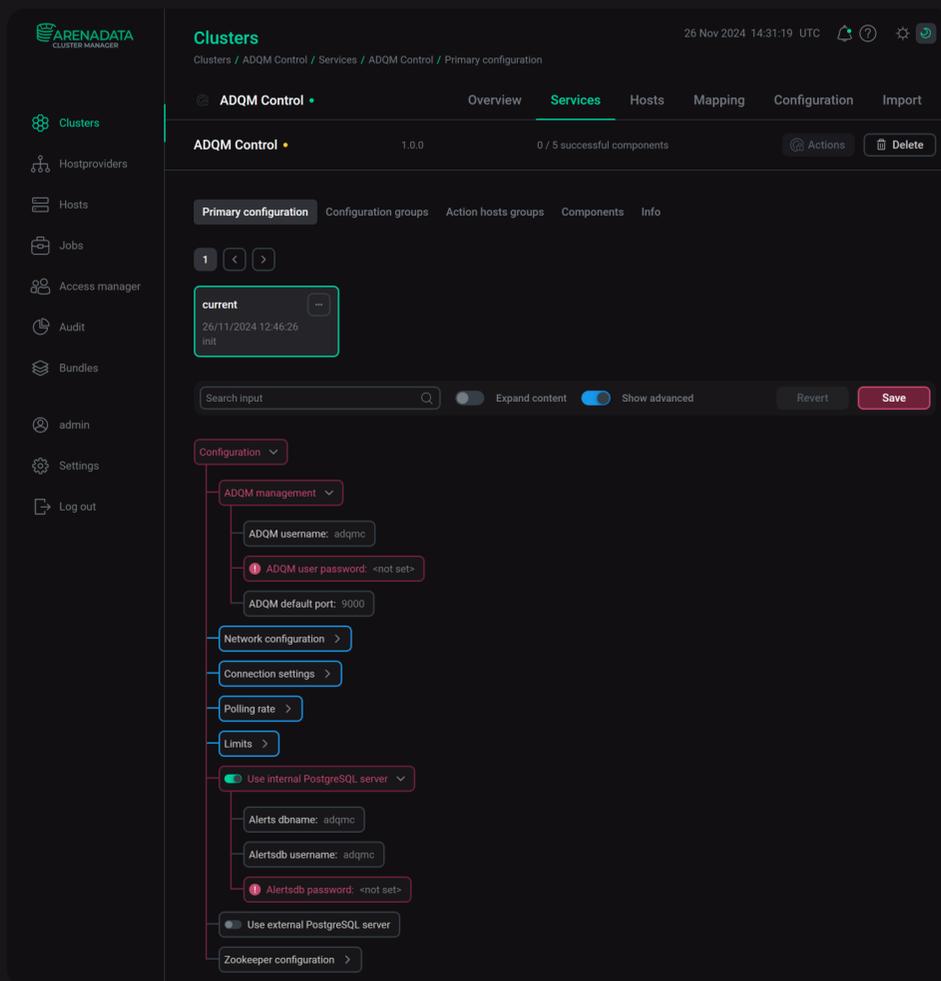
2. Open the **Services** tab on the cluster page, find the row that contains the desired service, and click its name in the **Name** column.



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 'qwerty';
```

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

For more information about all configuration parameters of ADQM Control services, see [Configuration parameters](#).

Contents

[To Table of Contents](#)

[Primary configuration](#)

[Configuration groups](#)

[Ansible settings](#)

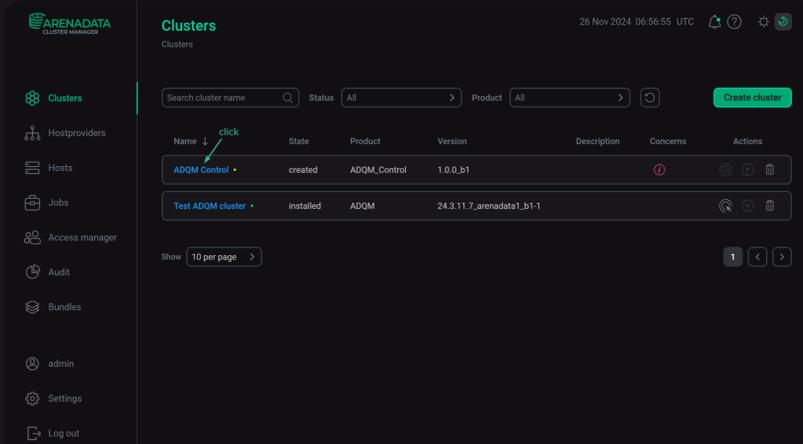
IMPORTANT

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.

Cluster configuration parameters are listed below. Basically, they describe paths to the repositories that are used during the installing process depending on the cluster operation system. Only the marked repositories are registered on the cluster servers. For each parameter, you can change the default URL to the required one.

Centos 7 repositories

Parameter	Description
zookeeper_repo	Repository required to install Arenadata Zookeeper
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

Altlinux 8 repositories

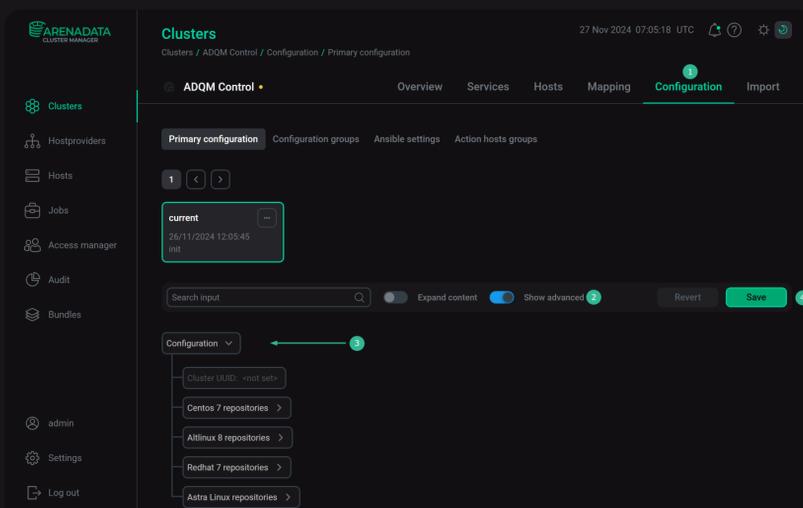
Parameter	Description
zookeeper_repo	Repository required to install Arenadata Zookeeper
arenadata_postgres	Repository required to install Arenadata PostgreSQL
prometheus_repo	Repository required to install Prometheus
ADQMC_arenadata_enterprise	Repository required to install the Enterprise edition of ADQM Control

Redhat 7 repositories

Parameter	Description
zookeeper_repo	Repository required to install Arenadata Zookeeper
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

Astra Linux repositories

Parameter	Description
zookeeper_repo	Repository required to install Arenadata Zookeeper
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.

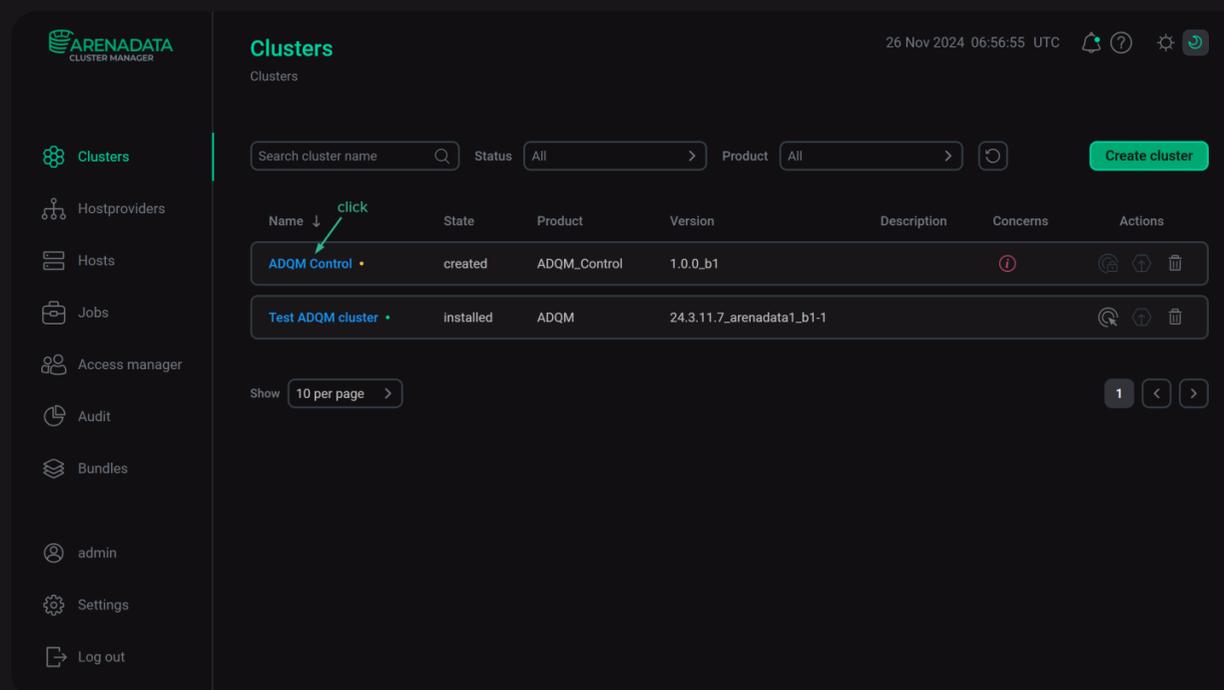
Ansible settings

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

[To Table of Contents](#)

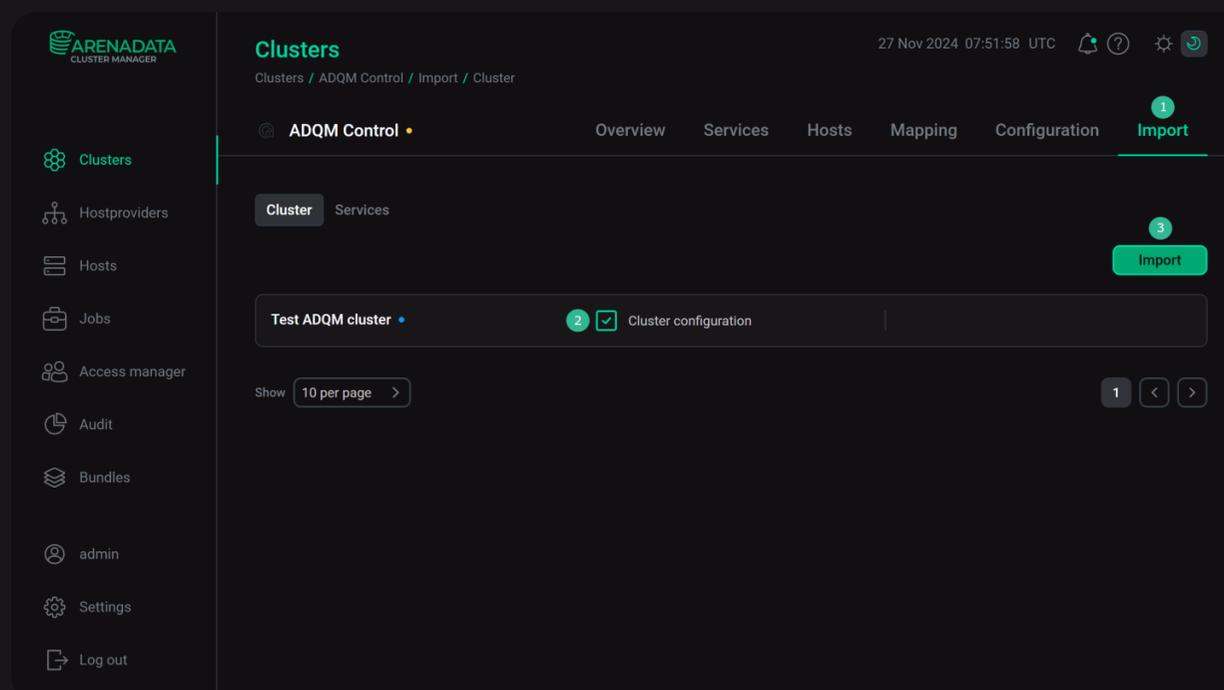
To set up an ADQM Control cluster, you should integrate it with at least one pre-installed ADQM cluster with the [Monitoring service](#) configured:

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 **Import**.



Integration with an ADQM cluster

NOTE

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



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:

- Whole cluster.** In this method all services are installed automatically one by one after applying the **Install** action to a cluster.
- 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.

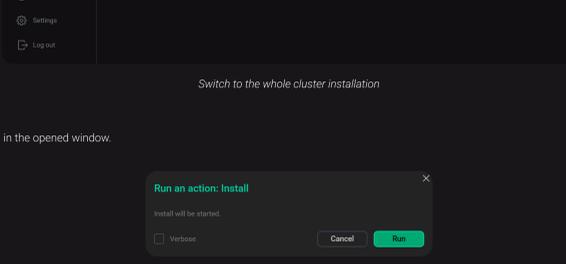
NOTE

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:

- 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

- Verify the requested action in the opened window.



- Wait until the job is completed.

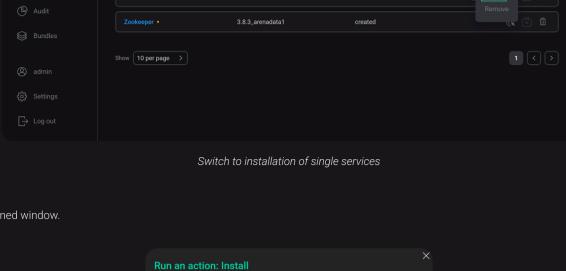
Single services

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

CAUTION

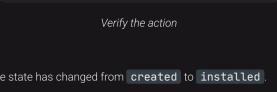
Notice that services are installed one by one. Do not install another service until the installation of the previous one is completed.

- 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

- Verify the action in the opened window.



- Wait until the job is completed. Then check that the service state has changed from **created** to **installed**.

- Repeat the previous actions for other services that should be installed.

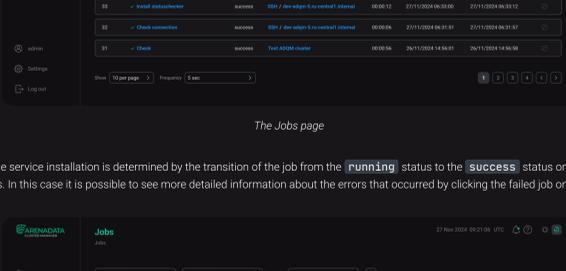
IMPORTANT

When you install ADQM Control services manually, one by one, use the following order:

- ADPG, Zookeeper in any order (if connections to external PostgreSQL/ZooKeeper clusters are not configured)
- ADQM Control

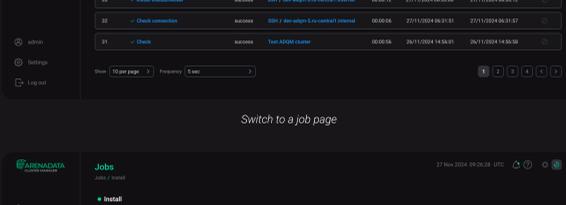
Step 2. View the results of installation

ADQM 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.

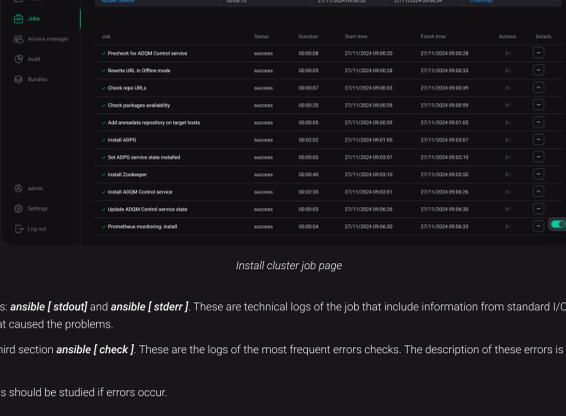


The Jobs page

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

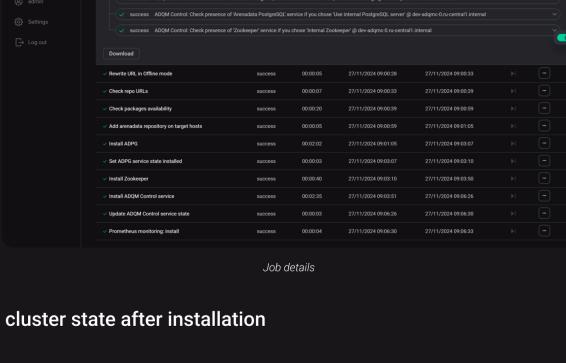


Install cluster job page

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.

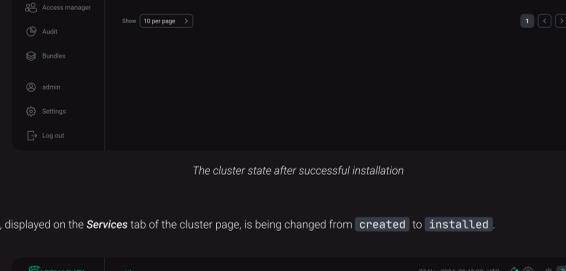


Job details

Step 3. Check the cluster state after installation

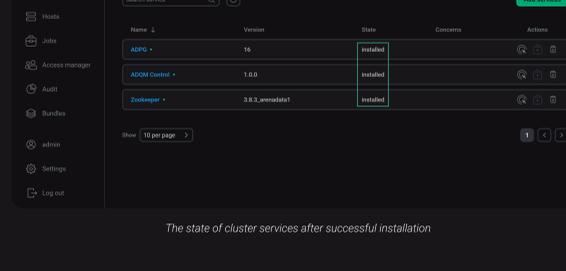
As a result of successful installation, the cluster and its services change their state according to the following rules:

- 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

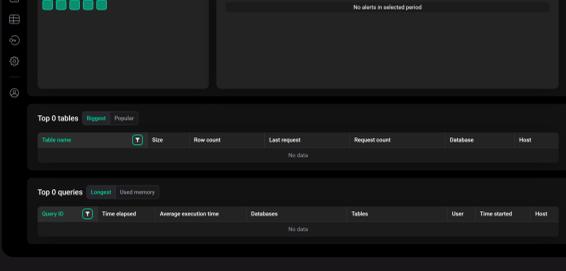
- The state of cluster services, displayed on the **Services** tab of the cluster page, is being changed from **created** to **installed**.



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 24.3.11.7.1 b1 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 that 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.

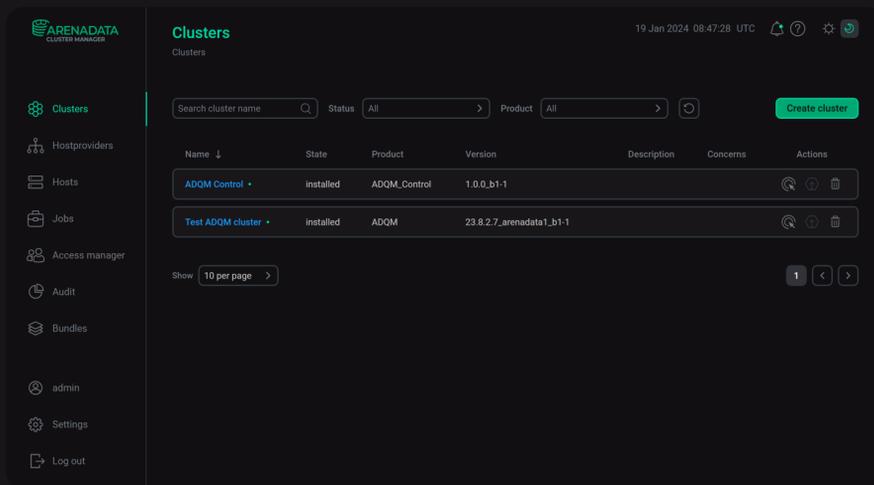
Contents

[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.

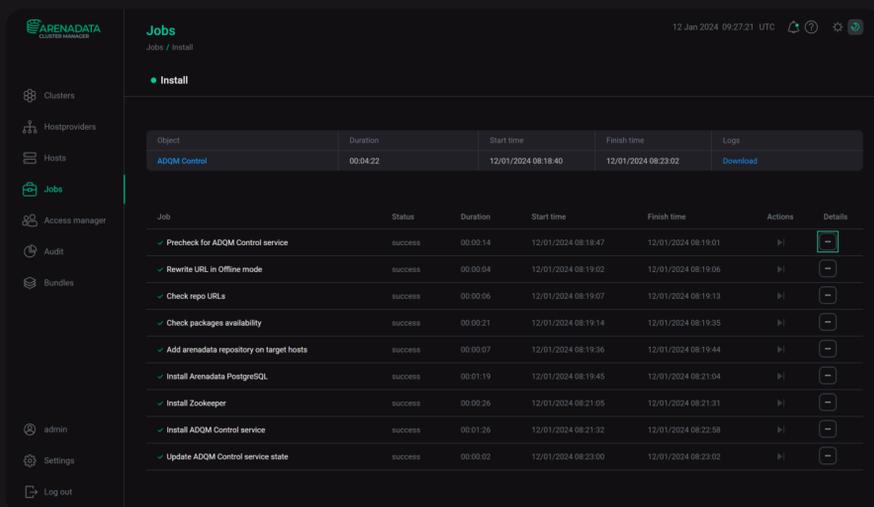


The Clusters page

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).

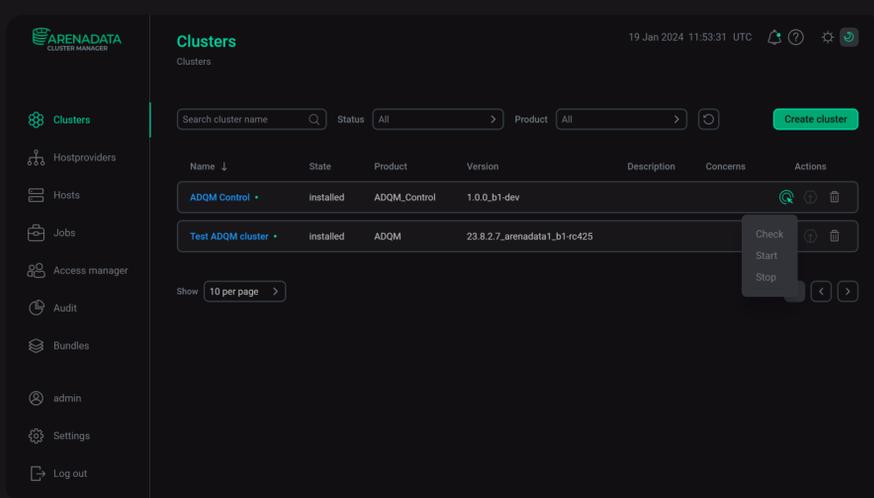
You can click icons described above to perform the corresponding cluster actions. When an action starts, ADCM displays the process and result of its execution on the **Jobs** page. From this page, you can navigate to a page with details on an individual job (by clicking a job name) to see inner steps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



The Jobs page

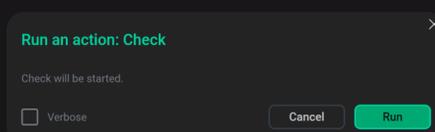
Manage a cluster

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 to confirm the choice. In this dialog window, you can select the **Verbose** checkbox to see additional execution details on the **Jobs** page.



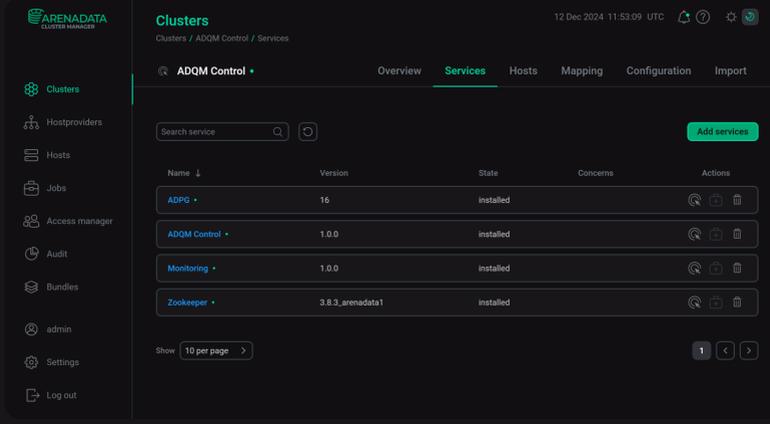
Action confirmation

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.
 - **Start** – starts all services of the cluster.
 - **Stop** – stops all services of the cluster.

[To Table of Contents](#)

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:

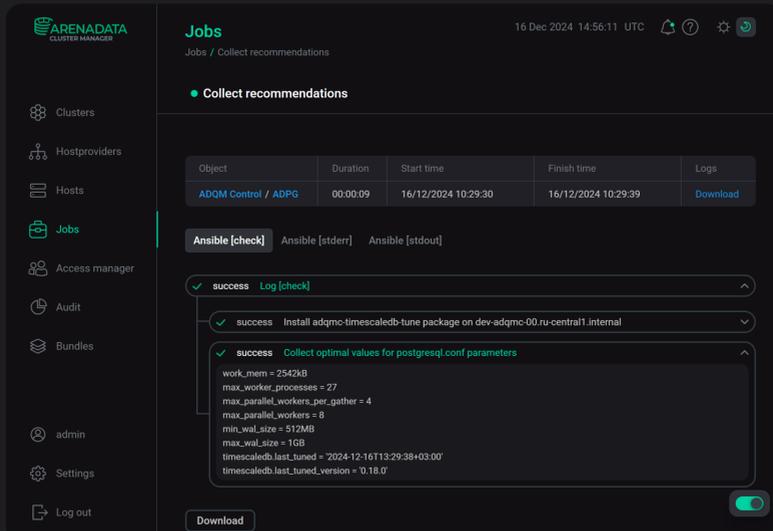
ADQM Control

- **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.



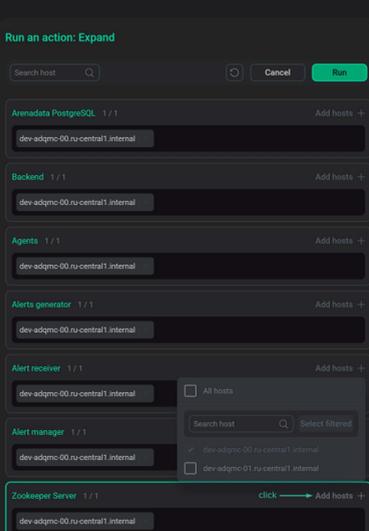
The result of the "Collect recommendations" action

- **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.

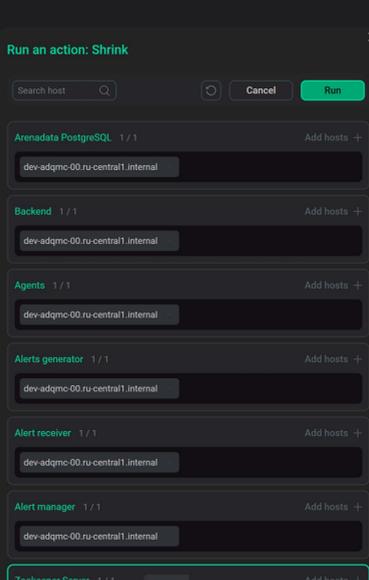
When you select this action, ADCM opens the **Run an action** dialog where you can add new hosts on which you want to install the Zookeeper Server component.



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.

Contents

[To Table of Contents](#)

System alerts

- [Configure system alerts](#)
- [Types of system 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.

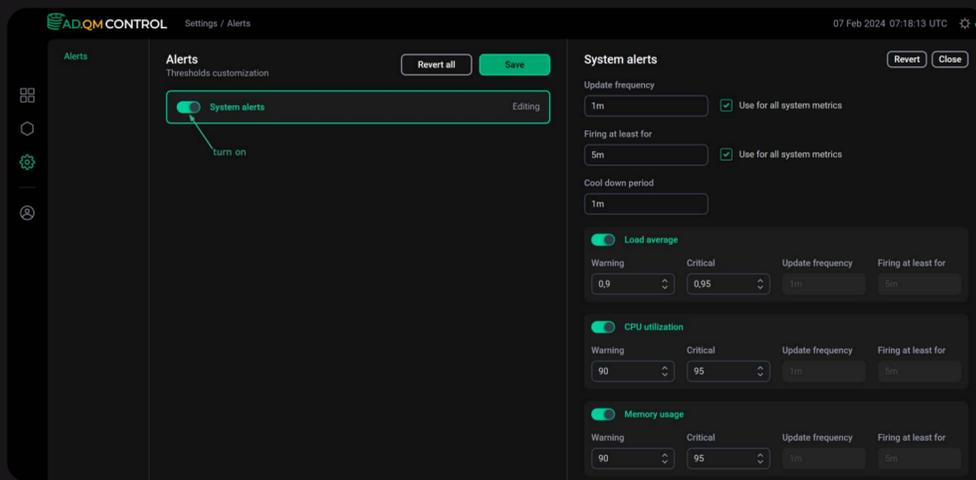
To change default settings, edit the required fields and click **Save**. Click **Revert all** if you need to cancel changes that have not been yet saved by clicking **Save**.

System alerts

Alerts in ADQM Control are grouped into modules. Currently, ADQM Control can send and allows you to configure alerts of the **System alerts** module. This 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 1-15 minutes	1m (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 – $(LA_{15} + LA_5) / 2 > \text{threshold}$. 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.

Contents

[To Table of Contents](#)

[View a list of users](#)

[Add a user](#)

[Edit a user](#)

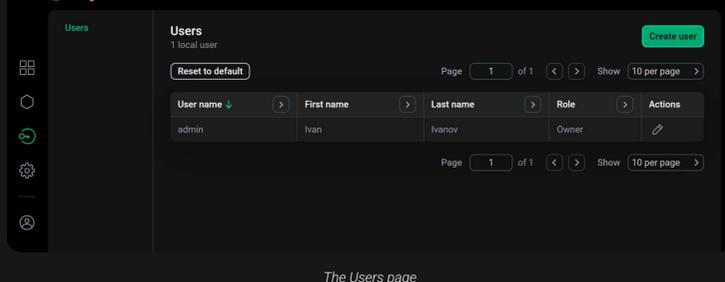
[Set password requirements](#)

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

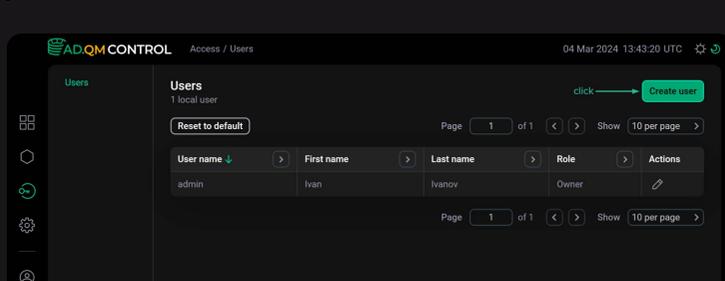
In column headers of the table with a list of users, there are filters that you can use to select specific data. To open a filter, click the icon.

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

Add a user

To add a new user, follow the steps:

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



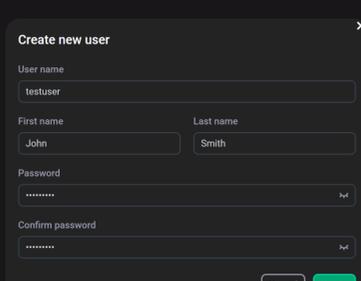
Switch to adding a user

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.

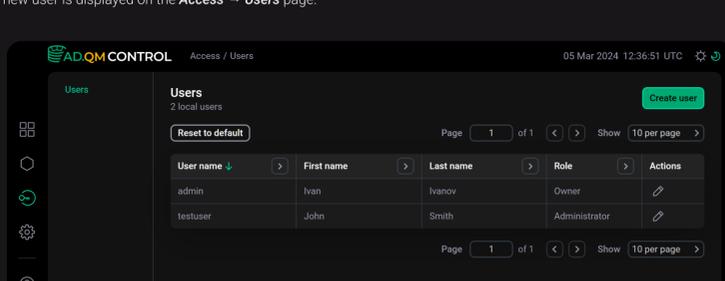
TIP

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

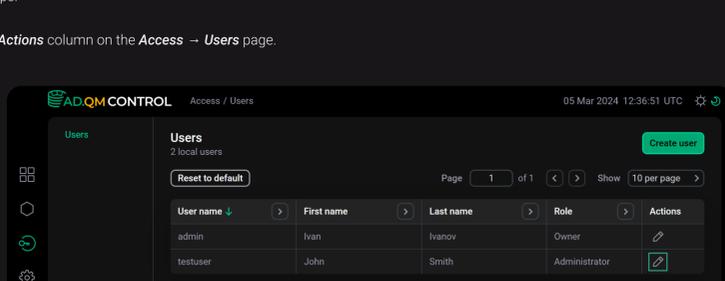
New users are currently created with the **Administrator** role (system administrator) that provides the same set of access rights as **Owner**.



Edit a user

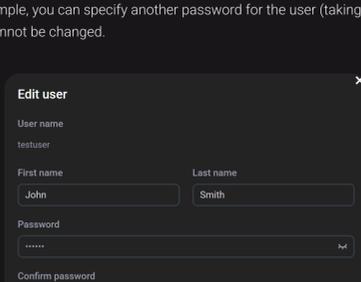
To edit user data, follow the steps:

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.

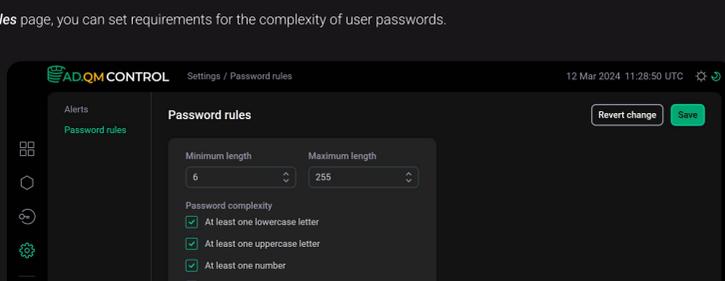


Edit user data

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

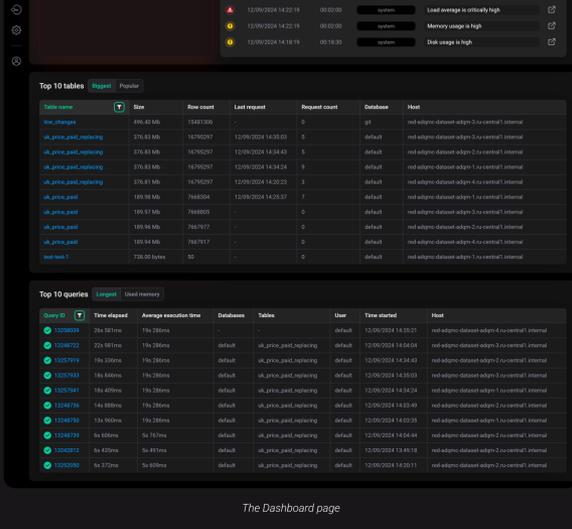
Parameter	Description	Default value
Minimum length	Minimum length of a password	1
Maximum length	Maximum length of a password	255
At least one lowercase letter	Indicates whether a password should contain at least one lowercase character	Disabled
At least one uppercase letter	Indicates whether a password 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

Contents

[To Table of Contents](#)
[Heat map](#)
[Recent alerts](#)
[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 to 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.



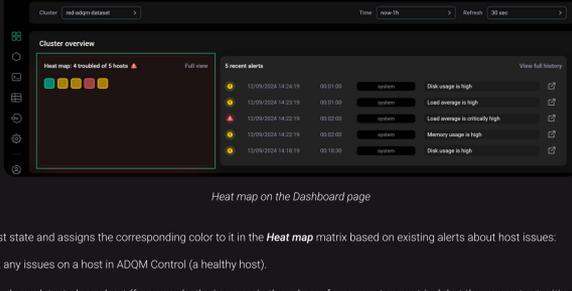
The Dashboard page

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 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.

Heat map

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

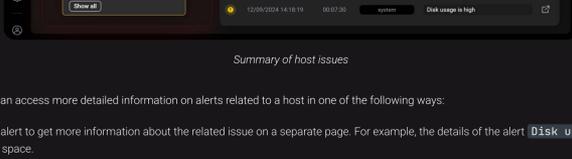


Heat map on the Dashboard page

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

- – there are no alerts about any issues on 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 these issues have been generated and continue to remain relevant 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.

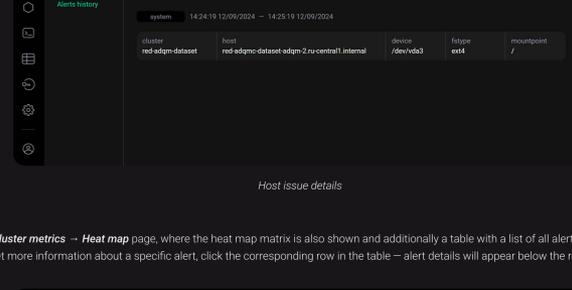
When you hover the mouse over a host in **Heat map**, a pop-up window shows a list of alerts about problems found on this host (for each problem type, only the latest relevant alert is displayed).



Summary of host issues

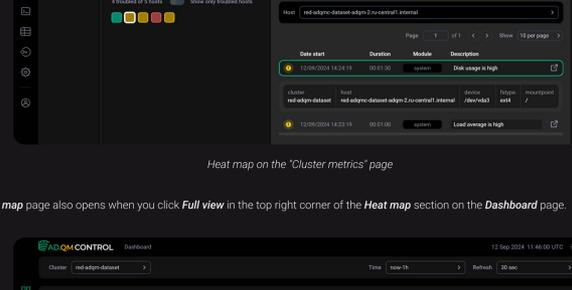
From this pop-up window, you can access more detailed information on alerts related to a host in one of the following ways:

- Click **Open details** below an alert to get more information about the related issue on a separate page. For example, the details of the alert **Disk usage is high** show which device is running out of free space.



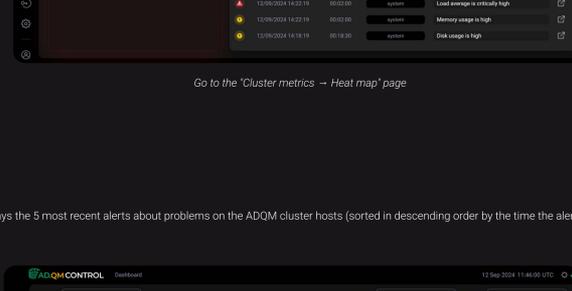
Host issue details

- Click **Show all** to open the **Cluster metrics** – **Heat map** page, where the heat map matrix is also shown and additionally a table with a list of all alerts related to a selected host is displayed on the right. To get more information about a specific alert, click the corresponding row in the table – alert details will appear below the row.



Heat map on the 'Cluster metrics' page

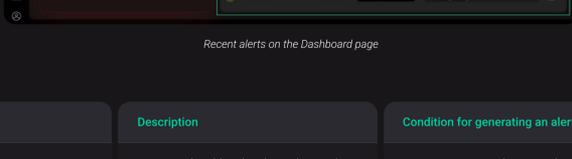
The **Cluster metrics** – **Heat map** page also opens when you click **Full view** in the top right corner of the **Heat map** section on the **Dashboard** page.



Go to the 'Cluster metrics – Heat map' page

Recent alerts

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



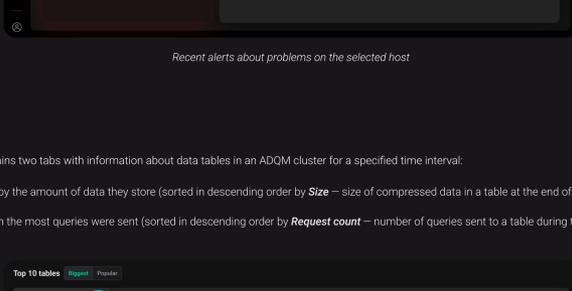
Recent alerts on the Dashboard page

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

Click the icon in an alert row to view the corresponding issue details. To get a full list of alerts for all hosts of the cluster for the specified period of time, click **View full history** in the top right corner within the **Recent alerts** section – it will open the **Alerts history** tab on the **Cluster metrics** page.

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 hosts

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 largest 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).

Table name	Size	Row count	Last request	Request count	Database	Host
uk_price_paid_replacing	121.43 Mb	22184059	21/10/2024 13:02:04	0	default	red-adqm-dataset-adqm-4.ru-central1-internal
uk_price_paid_replacing	109.14 Mb	17681761	21/10/2024 13:01:57	0	default	red-adqm-dataset-adqm-3.ru-central1-internal
uk_price_paid_replacing	109.28 Mb	15481306	0	0	gl	red-adqm-dataset-adqm-3.ru-central1-internal
uk_price_paid_replacing	109.77 Mb	15942121	21/10/2024 13:19:26	12	default	red-adqm-dataset-adqm-3.ru-central1-internal
uk_price_paid	109.52 Mb	6519203	0	0	default	red-adqm-dataset-adqm-3.ru-central1-internal
uk_price_paid	109.22 Mb	6519647	0	0	default	red-adqm-dataset-adqm-4.ru-central1-internal
uk_price_paid	108.06 Mb	6537115	0	0	default	red-adqm-dataset-adqm-3.ru-central1-internal
uk_price_paid	107.01 Mb	6573564	21/10/2024 13:19:24	11	default	red-adqm-dataset-adqm-1.ru-central1-internal
testdb-1	738.00 bytes	0	0	0	default	red-adqm-dataset-adqm-1.ru-central1-internal

List of the largest tables

Table name	Request count	Last request	Size	Increase	Database	Host
uk_price_paid	71	21/10/2024 13:25:01	195.78 Mb	35.13 Mb	default	red-adqm-dataset-adqm-2.ru-central1-internal
uk_price_paid_replacing	19	21/10/2024 13:19:25	372.54 Mb	2.14 Mb	default	red-adqm-dataset-adqm-3.ru-central1-internal
uk_price_paid_replacing	9	21/10/2024 13:01:57	432.91 Mb	7.41 Mb	default	red-adqm-dataset-adqm-3.ru-central1-internal
uk_price_paid_replacing	3	21/10/2024 13:02:04	565.41 Mb	72.11 Mb	default	red-adqm-dataset-adqm-3.ru-central1-internal
uk_price_paid	0	0	186.50 Mb	4.81 Mb	default	red-adqm-dataset-adqm-2.ru-central1-internal
uk_price_paid	0	0	186.64 Mb	4.82 Mb	default	red-adqm-dataset-adqm-4.ru-central1-internal
uk_price_paid	0	0	187.30 Mb	5.87 Mb	default	red-adqm-dataset-adqm-3.ru-central1-internal
uk_price_paid	0	0	388.25 Mb	0 bytes	gl	red-adqm-dataset-adqm-1.ru-central1-internal
testdb-1	0	0 bytes	0 bytes	0 bytes	gl	red-adqm-dataset-adqm-1.ru-central1-internal

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

This section also provides the following information on data tables:

- Table name** – table name (click on a table name opens a page with detailed information on table columns);
- Row count** (on the **Biggest** tab) – number of data rows in a table at the end of the selected time interval;
- Last request** – time the last 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 predefined 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 at the end of the time interval);
- 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 successfully completed queries that took the longest time to process (sorted in descending order by **Time elapsed** – duration of a query execution).
- Used memory** – top 10 queries that required the most memory (sorted in descending order by **Used memory** – amount of memory consumed by a query).

Query ID	Time elapsed	Average execution time	Databases	Tables	User	Time started	Host
13160586	24s 922ms	18s 820ms	default	uk_price_paid_replacing	default	12/09/2024 08:27:21	red-adqm-dataset-adqm-1.ru-central1-internal
13176008	23s 118ms	18s 820ms	default	uk_price_paid_replacing	default	12/09/2024 09:58:11	red-adqm-dataset-adqm-3.ru-central1-internal
13151578	20s 145ms	18s 820ms	default	uk_price_paid_replacing	default	12/09/2024 08:26:22	red-adqm-dataset-adqm-3.ru-central1-internal
13156527	18s 820ms	18s 820ms	default	uk_price_paid_replacing	default	12/09/2024 09:27:02	red-adqm-dataset-adqm-1.ru-central1-internal
13151561	16s 87ms	18s 820ms	default	uk_price_paid_replacing	default	12/09/2024 08:26:07	red-adqm-dataset-adqm-2.ru-central1-internal
13151678	16s 820ms	18s 820ms	default	uk_price_paid_replacing	default	12/09/2024 09:57:57	red-adqm-dataset-adqm-1.ru-central1-internal
13151612	16s 820ms	18s 820ms	default	uk_price_paid_replacing	default	12/09/2024 09:57:41	red-adqm-dataset-adqm-1.ru-central1-internal
13151599	13s 807ms	18s 820ms	default	uk_price_paid_replacing	default	12/09/2024 08:25:02	red-adqm-dataset-adqm-1.ru-central1-internal
13156026	10s 917ms	9s 268ms	default	uk_price_paid_replacing	default	12/09/2024 08:27:50	red-adqm-dataset-adqm-2.ru-central1-internal

List of the longest-running queries

Query ID	Used memory	Result bytes	Databases	Tables	User	Time started	Host
13154594	401.04 Mb	86.39 Mb	default	uk_price_paid_replacing	default	12/09/2024 08:42:31	red-adqm-dataset-adqm-4.ru-central1-internal
13154607	401.04 Mb	86.39 Mb	default	uk_price_paid_replacing	default	12/09/2024 08:42:19	red-adqm-dataset-adqm-2.ru-central1-internal
13154618	401.04 Mb	107.09 Mb	default	uk_price_paid_distributed	default	12/09/2024 08:42:12	red-adqm-dataset-adqm-1.ru-central1-internal
13154619	401.04 Mb	86.39 Mb	default	uk_price_paid_replacing	default	12/09/2024 08:42:12	red-adqm-dataset-adqm-3.ru-central1-internal
13154632	401.04 Mb	86.39 Mb	default	uk_price_paid_replacing	default	12/09/2024 08:42:25	red-adqm-dataset-adqm-4.ru-central1-internal
13160499	401.04 Mb	86.53 Mb	default	uk_price_paid_replacing	default	12/09/2024 08:56:12	red-adqm-dataset-adqm-3.ru-central1-internal
13160541	401.04 Mb	86.53 Mb	default	uk_price_paid_replacing	default	12/09/2024 08:56:10	red-adqm-dataset-adqm-4.ru-central1-internal
13160529	401.04 Mb	86.53 Mb	default	uk_price_paid_replacing	default	12/09/2024 08:57:50	red-adqm-dataset-adqm-3.ru-central1-internal
13160553	401.04 Mb	108.16 Mb	default	uk_price_paid_distributed	default	12/09/2024 08:57:35	red-adqm-dataset-adqm-1.ru-central1-internal
13160546	401.04 Mb	86.53 Mb	default	uk_price_paid_replacing	default	12/09/2024 08:57:44	red-adqm-dataset-adqm-1.ru-central1-internal

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);
- Average execution time** (on the **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);
- Result bytes** (on the **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.

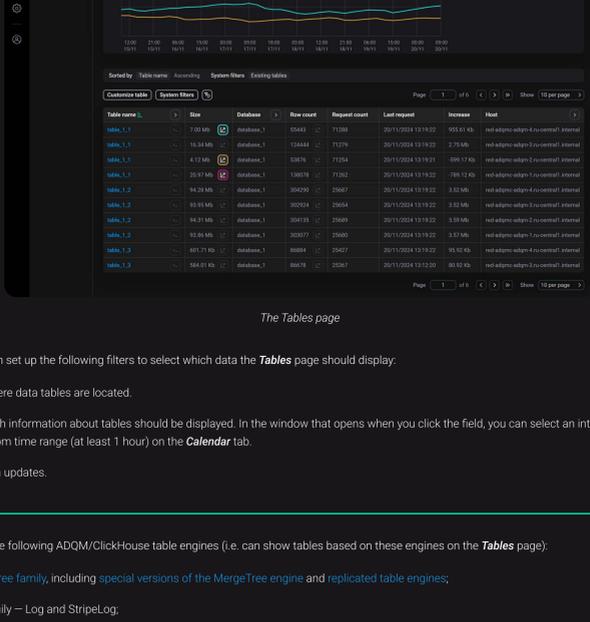
To view the full history of queries for the required period of time, switch to the **Queries** page.

Contents

To Table of Contents

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

The **Tables** page of the ADQM Control web interface displays information about data tables in an ADQM cluster connected to the monitoring system. On this page, you can view a list of all cluster tables with general information about them, switch to details on columns of an individual table, and also visually identify changes in sizes of tables using graphs.



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, including special versions of the MergeTree engine and replicated table engines;
- engines of the Log family – Log and StripeLog;
- the Memory, Merge, Join, Set, Null engines.

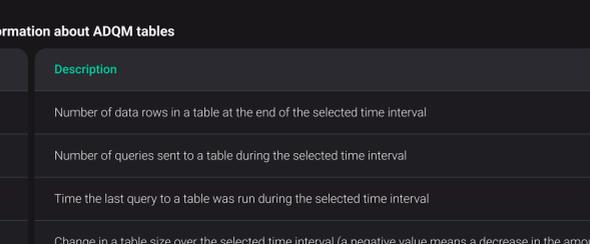
For integration table engines (used to access external data storage systems – for example, MySQL, MongoDB, HDFS, Hive, S3, Kafka, EmbeddedRocksDB, RabbitMQ, PostgreSQL), the correctness and completeness of the information displayed in ADQM Control is currently not guaranteed.

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

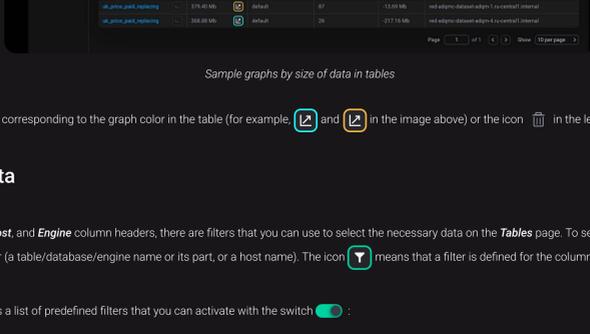
Additional fields with information about ADQM tables

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
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, and in the image above) or the icon in the legend to the right of the graphs.

Filter and sort data

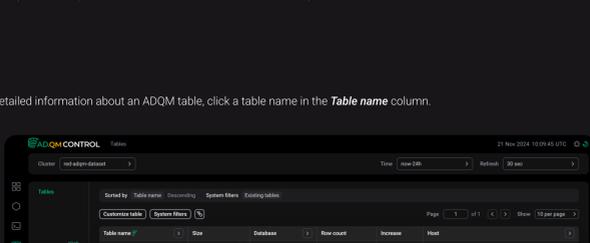
In the **Table name**, **Database**, **Host**, and **Engine** column headers, there are filters that you can use to select the necessary data on the **Tables** page. To set a filter, click the icon and enter a value you are looking for (a table/database/engine name or its part, or a host name). The icon 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 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.

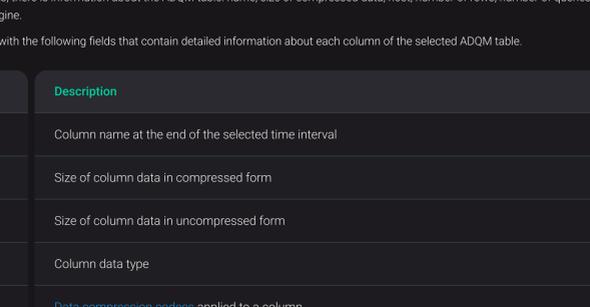
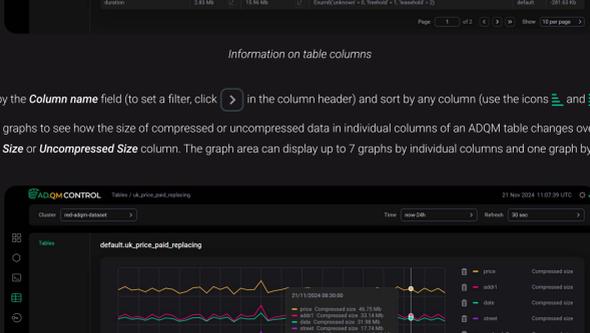


Table selection

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.

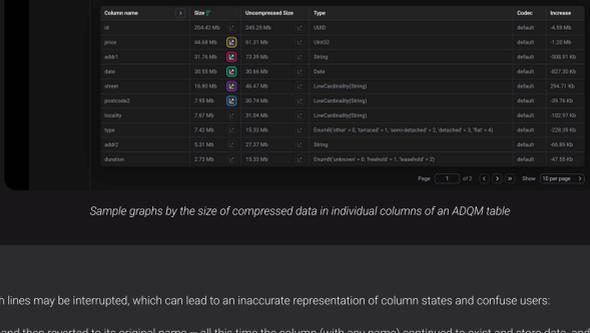
Field	Description
Column name	Column name at the end of the selected time interval
Size	Size of column data in compressed form
Uncompressed Size	Size of column data in uncompressed form
Type	Column data type
Codec	Data compression codecs applied to a column
Increase	Change in size of column data over 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 (use the icons and in column headers).

On this 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. To do this, use the icon in the **Size** or **Uncompressed Size** column. 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

CAUTION

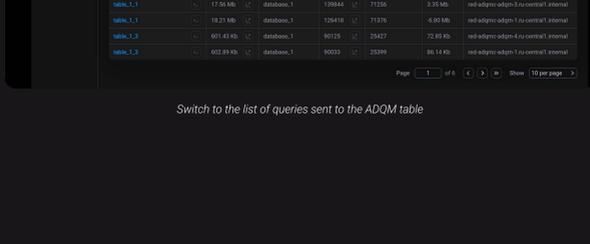
In the following cases, graph lines may be interrupted, which can lead to an inaccurate representation of column states and confuse users:

- a column was renamed and then reverted to its original name – all 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 fact, these 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** cell 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

Contents

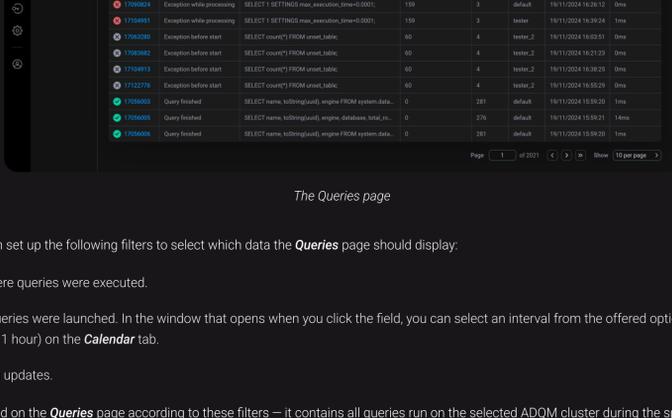
To Table of Contents

Fields of the Queries table

Sort and filter data

Query details

The **Queries** page in the ADQM Control web interface displays SQL queries executed in ADQM clusters connected to the monitoring system. On this page, you can get an overview of queries to cluster databases and view various query execution metrics – for example, to analyze query performance, amounts of read/written data, resource usage, errors if they have occurred during query execution, etc.



The Queries page

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

- **Cluster** – ADQM cluster where queries were executed.
- **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 (at least 1 hour) on the **Calendar** tab.
- **Refresh** – frequency of data updates.

Query history is always displayed on 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 (with **ON CLUSTER**), it also includes the queries executed on each host of the cluster and displays the `/*/ ddL_entry=query-num */` prefix before the main text of such queries.

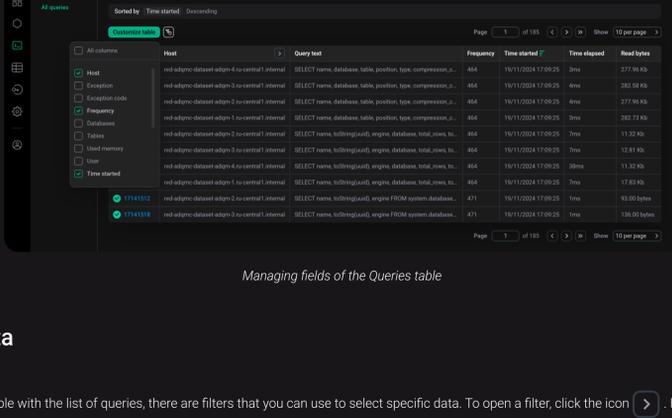
You can also get a list of the longest-running queries or queries used the most memory in the **Top 10 queries** section on the **Dashboard** page.

Fields of the Queries table

The table with the list of queries on the **Queries** page can include the following fields.

Field	Description
Query ID	Unique query identifier. The column also shows an icon indicating the query status: <ul style="list-style-type: none"> ✔ – a query was successfully executed (the Query finished status); 🔄 – a query is being executed at the current time (the Query started status); ⌚ – an error occurred before a query execution started (the Exception before start status); ⚠️ – an error occurred during a query execution (the Exception while processing status); ⏸️ – a query execution was not completed for some reason (the Uncompleted status). Query status can also be separately displayed in the Status column that allows filtering and sorting
Query text	Query text. For a long query, its text is truncated and when you hover the mouse cursor over the column cell, it is displayed as <code>[query start] <...> [query end]</code> . 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: <ul style="list-style-type: none"> ■ 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; ■ Uncompleted – 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. Statistics are calculated for queries whose text completely matches (i.e. for 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
Time started	Query execution start time
Time elapsed	Duration of a query execution or the current execution time (now - time_started) of a query that has not yet been completed (when a query is completed, a value in the Time elapsed column will be replaced with the real query execution time once ADQM Control receives information about queries from ADQM the next time)
Average execution time	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 replicas – each replica sends its Read bytes value and the server-initiator of the query sums all received and local values
Result bytes	Amount of memory to store 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 replicas – each replica sends 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: <ul style="list-style-type: none"> ■ Read – read query; ■ Write – write query.

To display/hide 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

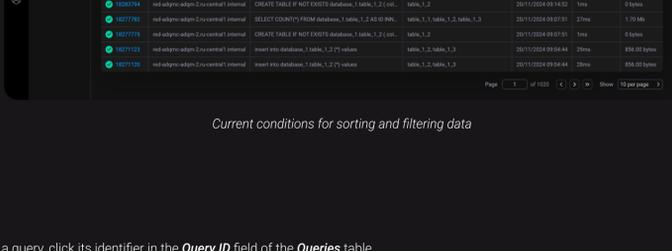
In the column headers of the table with the list of queries, there are filters that you can use to select specific data. To open a filter, click the icon . Filters are available in the following columns:

- **Host, Databases, Tables** – enter a name of a host, database, or table (for rows with multiple databases or tables separated by commas in the **Databases** or **Tables** column, filtering works only if the search is performed by one item from a list of databases or tables in a cell);
- **Exception code** – enter a numerical error code;
- **User** – you can enter a name or name fragment of a user;
- **Status, Type** – you can select a value (query status or type) from the drop-down list.

The icon indicates that a filter by column values is applied. To reset all filter conditions, click .

Data sorting is available in the following columns: **Exception** (actually sorting is performed by error code), **Exception code**, **Used memory**, **Time elapsed**, **Read bytes**, **Result bytes**, **Result rows**, **Read rows**, **Written rows**, and **Written bytes**. To change a sort order by column values, use the icon or in the column header.

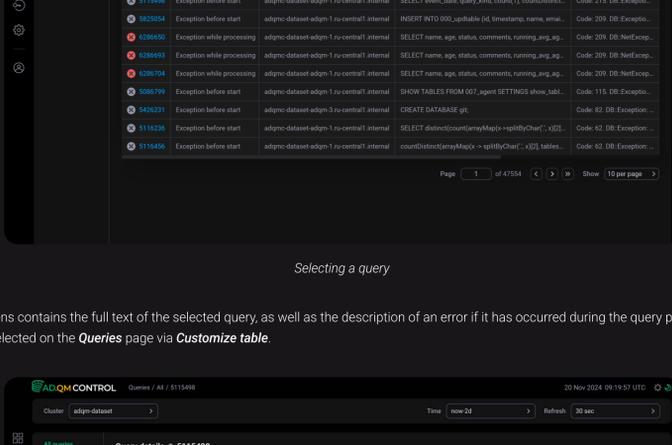
Above the table with queries, there is a panel that shows the sorting (**Sorted by**) and filtering (**Filtered by**) conditions currently applied to data.



Current conditions for sorting and filtering data

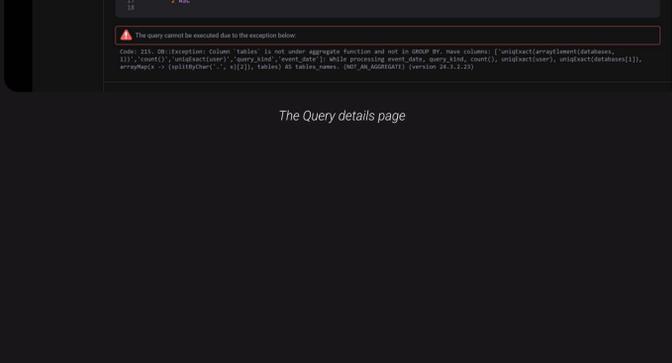
Query details

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

Contents

[To Table of Contents](#)

- ADQM Control
- ADPG
- Zookeeper
- Monitoring

This article describes parameters that can be configured for ADQM Control services via ADQM. To read about the configuration process, refer to the [Configure services](#) article in the section [Online installation of ADQM Control](#).

NOTE

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

ADQM Control

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

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 'qwerty';
```

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

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
Alert manager 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
Confid 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

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

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	—

Zookeeper configuration

Parameter	Description	Default value
zk_type	Indicates which ZooKeeper cluster ADQM Control uses – internal (external as the Zookeeper service of the ADQM Control cluster) or external. Possible values are Internal Zookeeper , External Zookeeper	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 Zookeeper	—
zk_prefix	Prefix for all paths to znodes in ZooKeeper related to ADQM Control	/arenadata/adqm_control/<cluster_id>

ADPG

Parameter	Description	Default value
Data directory	Directories used to store data on ADPG hosts	/pg.data1

ADPG configurations

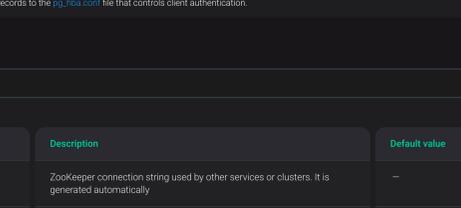
TIP

You can obtain the optimal parameter values to achieve the best ADPG performance using the [Collect recommendations](#) section 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 acts as 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 archiving 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: <ul style="list-style-type: none"> 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 , pg LZ	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**.



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



You can add multiple parameters.

Custom pg_hba.conf

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

Zookeeper

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 be synced 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 ADQM UI, but are allowed in the zoo.cfg configuration file	—

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
ZOO_PIDFILE	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

Monitoring

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

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.

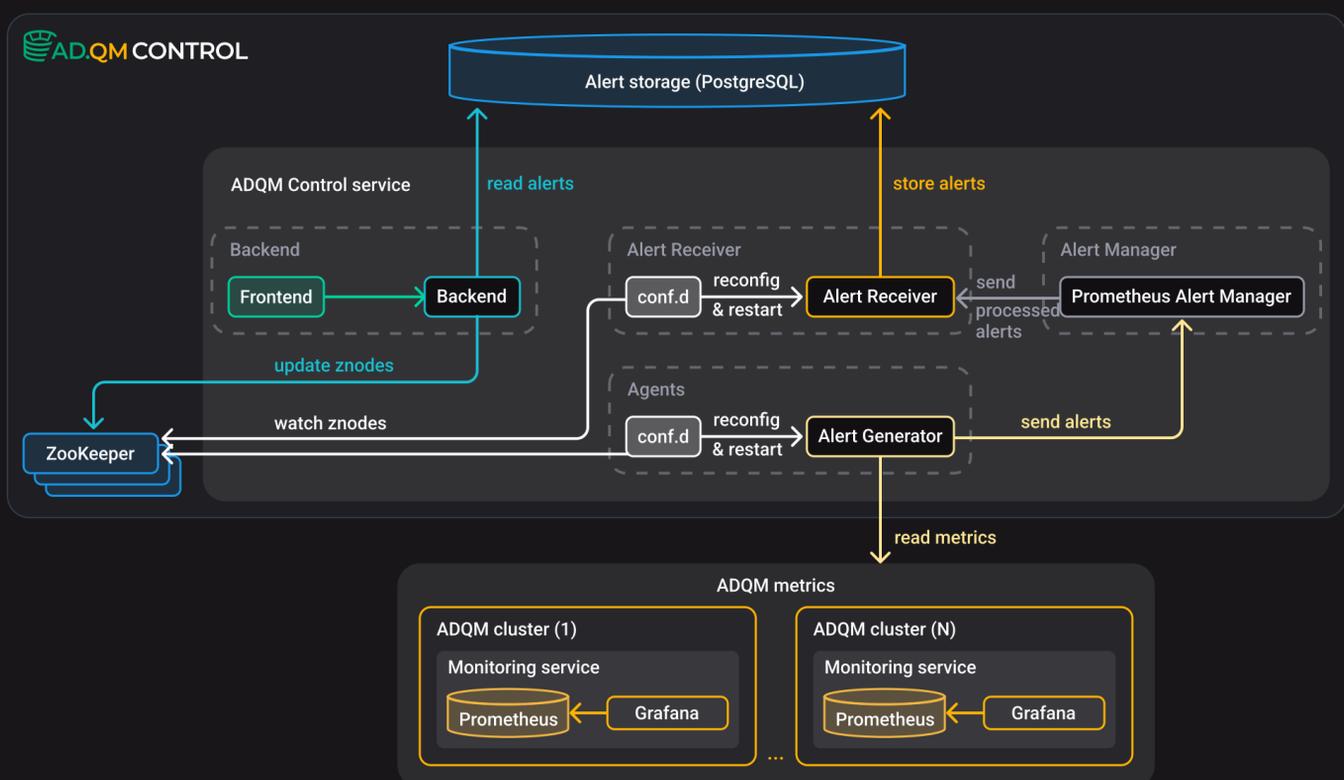
NOTE

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.

Contents

[To Table of Contents](#)

[1.0.0](#)

1.0.0

1.0.0.b1

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

ADPG

ADPG

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

ADPG configurations

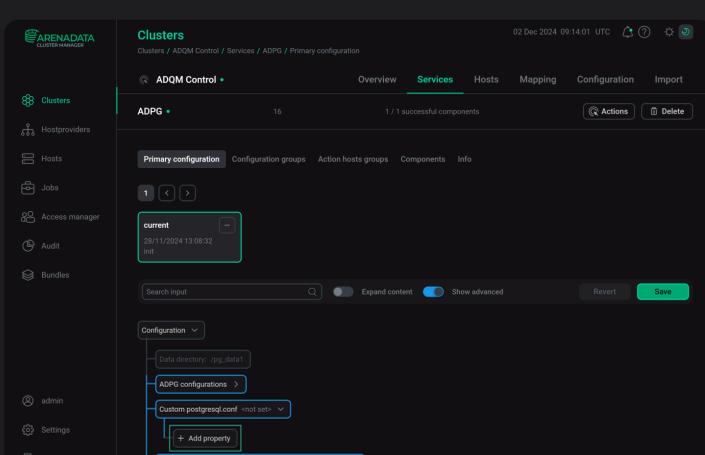
TIP You can obtain the optimal parameter values to achieve the best ADPG performance using the **Collect recommendations** action of the ADPG service.



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 acts as 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 <code>min_wal_size</code> 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 <code>pg_wal</code> 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 <code>wal_keep_size</code> 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 <code>wal_keep_size</code> parameter sets only the minimum size of segments retained in <code>pg_wal</code> . The system might need to retain more segments for WAL archival or to recover from a checkpoint. If <code>wal_keep_size</code> is set to <code>0</code> , 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: <ul style="list-style-type: none"> <code>try</code> — the server tries to request huge pages. If this fails, the server falls back to the default page. <code>on</code> — if a huge page request fails, the server does not start. <code>off</code> — 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 <code>ALTER TABLE...SET STATISTICS</code>	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 <code>max_locks_per_transaction</code> * (<code>max_connections</code> + <code>max_prepared_transactions</code>) 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 <code>VACUUM</code> and <code>ANALYZE</code> 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: <code>lz4</code> , <code>pglz</code>	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 <code>stderr</code> 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 <code>/pg_data1/adpg14/log</code>)
log_filename	Log file name pattern. A value can include <code>strftime</code> %-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 <code>log_rotation_age</code> to <code>0</code> 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 <code>log_rotation_size</code> to <code>0</code> 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 <code>debug5</code> , <code>debug4</code> , <code>debug3</code> , <code>debug2</code> , <code>debug1</code> , <code>info</code> , <code>notice</code> , <code>warning</code> , <code>error</code> , <code>log</code> , <code>fatal</code> , and <code>panic</code> (see the Severity levels table). Messages with the specified severity or higher are included in the log file. For example, if you set <code>log_min_messages</code> to <code>warning</code> , the log file will include the <code>warning</code> , <code>error</code> , <code>log</code> , <code>fatal</code> , and <code>panic</code> messages	warning
log_min_error_statement	Specifies which SQL statements that cause errors are logged. Possible values are <code>debug5</code> , <code>debug4</code> , <code>debug3</code> , <code>debug2</code> , <code>debug1</code> , <code>info</code> , <code>notice</code> , <code>warning</code> , <code>error</code> , <code>log</code> , <code>fatal</code> , and <code>panic</code> (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 <code>log_min_error_statement</code> to <code>panic</code>	error

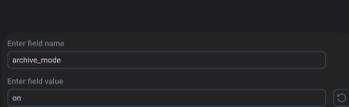
Custom postgresql.conf

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.

Custom pg_hba.conf

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

[ADQM Control](#)

ADQM management

Parameter	Description	Default value
ADQM username	Username to connect to ADQM	adqmc
ADQM user password	Password for the <code>ADQM username</code> user to connect to ADQM	–
ADQM default port	Default port if it is not exported from an ADQM cluster	9000

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 'qwerty';
```

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



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
Alert manager 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 <code>query_log</code> 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 <code>ADQMDB scrape interval * Existing tables tries</code> , 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 <code>Alerts dbname</code> database	adqmc
Alertsdb password	Password for the <code>Alertsdb username</code> user to access the <code>Alerts dbname</code> 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 <code>Add property</code> to add the <code>host</code> and <code>port</code> 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 <code>alerts_dbname</code> database	adqmc
alertsdb_password	Password for the <code>alerts_db_username</code> user to access the <code>alerts_dbname</code> database	–

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 <code>Internal Zookeeper</code> , <code>External Zookeeper</code>	Internal Zookeeper
zk_hosts	List of external Zookeeper cluster hosts (for example: <code>host1.internal:2181,host2.internal:2181,host3.internal:2181</code>). It is required if <code>zk_type</code> is set to <code>External Zookeeper</code>	–
zk_prefix	Prefix for all paths to znodes in ZooKeeper related to ADQM Control	/arenadata/adqm_control/<cluster_id>

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

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 <code>dataDir</code> parameter in the <code>zoo.cfg</code> 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 <code>dataDir</code> and <code>dataLogDir</code> 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 <code>zoo.cfg</code> configuration file	—

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
ZOOIDFILE	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