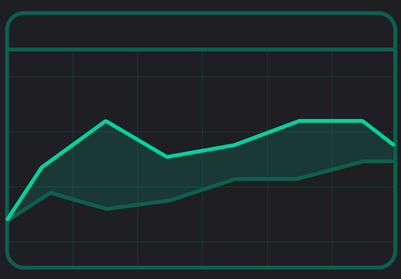
ADQM CONTROL

Arenadata QuickMarts Control



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

Version 1.2.1 Language: EN

Contents

1. Concepts
   - ADQM Control overview
2. Get started
   - Installation
     - Online installation of ADQM Control
       - Create a cluster
       - Add services
       - Add hosts to a cluster
       - Add components
       - Configure services
       - Configure a cluster
       - Integrate with an ADQM cluster
       - Install a cluster
       - Install monitoring
     - Offline installation of ADQM Control
       - Create a cluster
       - Add services
       - Add hosts to a cluster
       - Add components
       - Configure services
       - Configure a cluster
       - Integrate with an ADQM cluster
       - Import ET settings
       - Install a cluster
       - Install monitoring
   - Connect to ADQM Control
3. Access management
   - Manage users
   - SSL encryption
4. Cluster maintenance
   - Cluster actions
   - Service actions
   - Configure an external database
5. Web user interface
   - Alert settings
   - View a dashboard
   - Cluster metrics
   - Tables
   - Queries
6. References
   - Configuration parameters
   - Query exception codes
   - Monitoring metrics
7. Release notes
   - ADQM Control releases

ADQM Control overview

Elena Dvoryadkina

Contents

To Table of Contents

Features

Architecture

Features

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

Key objectives of ADQM Control are the following:

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

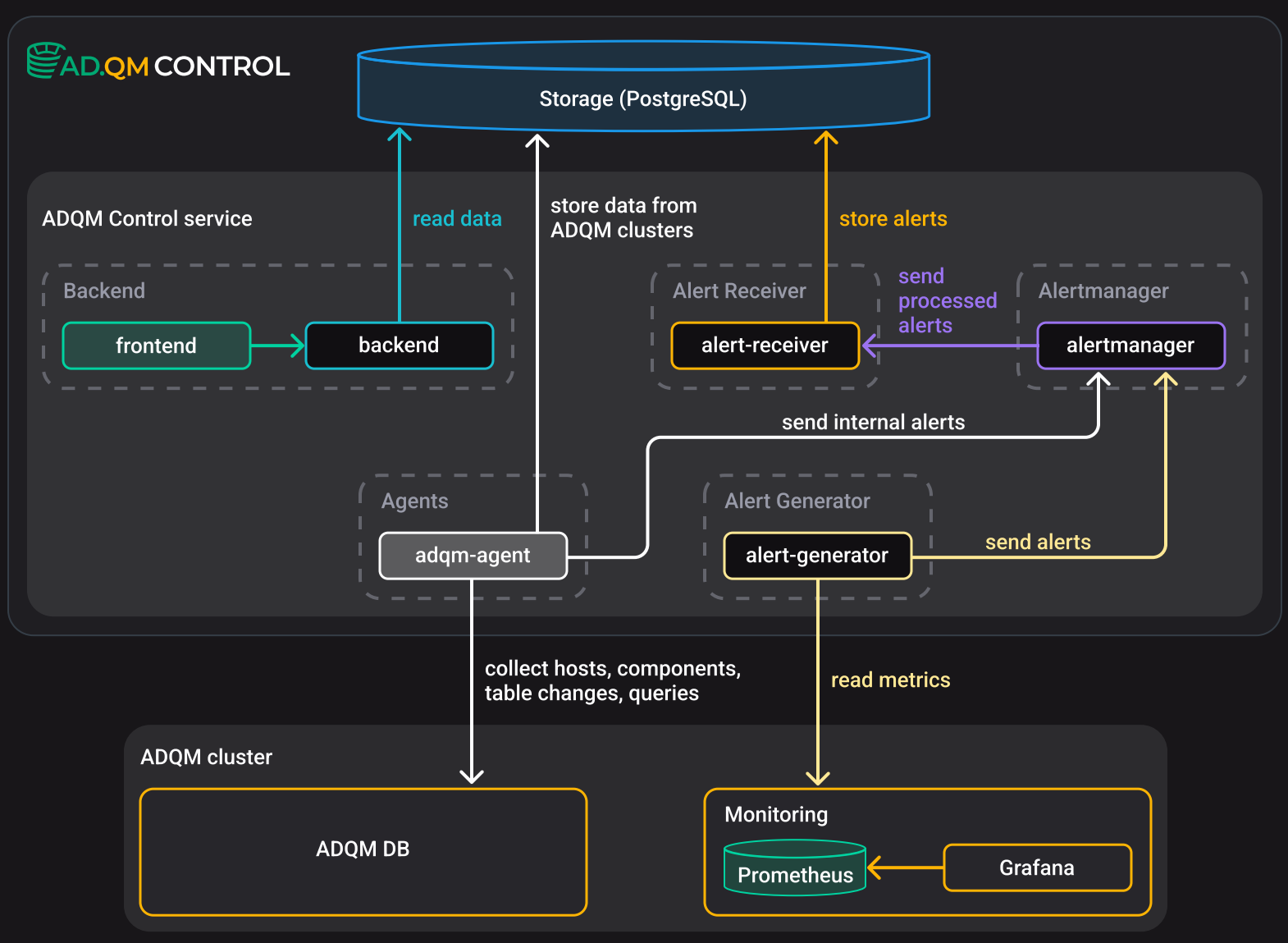
  In the ADQM Control user interface, you can specify criteria for generating alerts of different types (for example, configure thresholds, set timeouts) or disable tracking of some events.

NOTE

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

Architecture

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



ADQM Control architecture

Main points of the scheme:

- **ADQM Control** is the main service that implements the full functionality of ADQM Control. It includes the following components:
  - **Alert Generator.** 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 Alertmanager.
  - **Agents.** The adqm-agent service receives data about hosts, components, queries, and table changes from the ADQM cluster's ADQMDB service and stores this information in the database (Storage), and also generates internal alerts related to ADQM and sends them to Alertmanager.
  - **Alertmanager.** The alertmanager service handles alerts received from Alert Generator and Agents (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 Alertmanager and stores them in the 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, retrieves alerts and data about the ADQM cluster from the storage to transmit them to the web interface.
- **Storage** is a PostgreSQL database for storing alerts, as well as data about users, tables, queries, and settings. 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.

Installation

AD.QM CONTROL

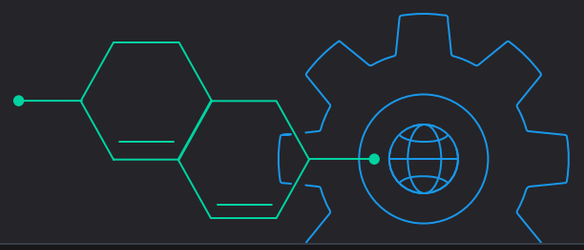
To Table of Contents

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

You can install ADQM Control via ADCM in two ways.

Online installation

on hosts with access to Internet



Offline installation

on hosts without access to Internet

Online installation

To Table of Contents

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

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

Step 1.

Install ADCM

Step 2.

Prepare hosts

Step 3.

Install ADQM cluster

Step 4.

Install monitoring in ADQM

Step 5.

Install ADQM Control

Step 6.

Install monitoring in ADQM Control

Create a cluster

Elena Dvoryadkina, Daria Barysheva

AD.QM 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.

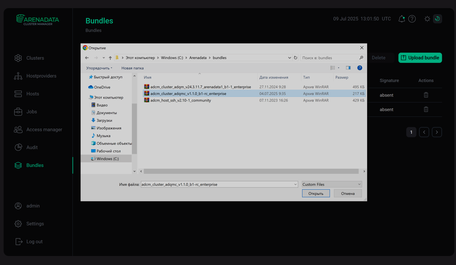
Step 1. Get the Enterprise version of a cluster bundle

Please get a cluster bundle from the Arenadata support team.

Step 2. Upload a cluster bundle to ADCM

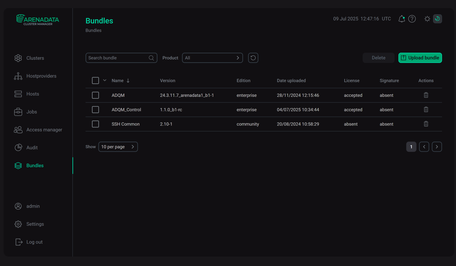
To upload a bundle to ADCM, follow the steps:

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

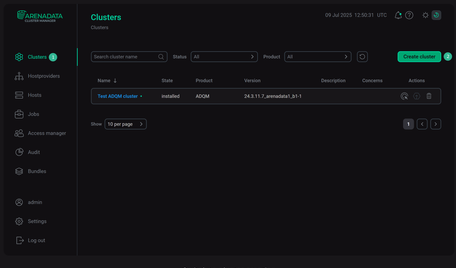
Step 3. Create a cluster based on the uploaded bundle

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.

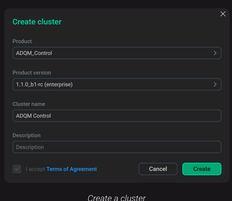
The steps for cluster creation are given below:

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



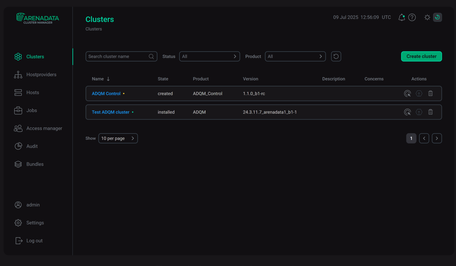
Switch to cluster creation

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

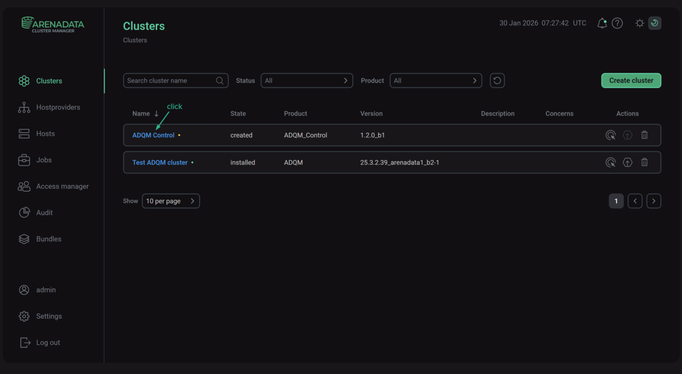
Add services

Elena Dvoryadkina, Daria Barysheva

To Table of Contents

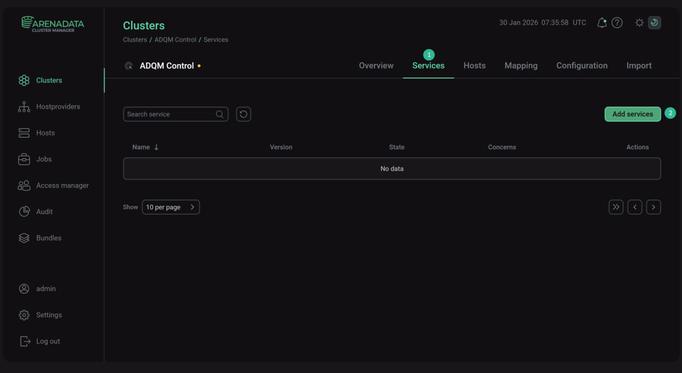
In ADCM, a **service** means a software that performs some function. The steps for adding services to a cluster are listed below:

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



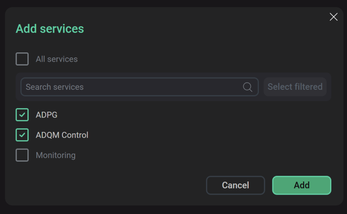
Select a cluster

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



Switch to adding services

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



Select services

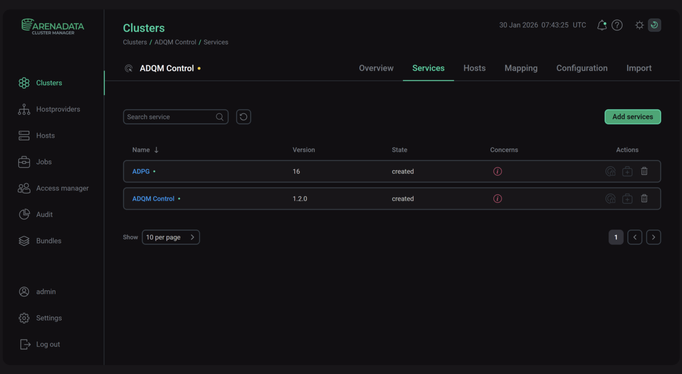
The brief description of available services is listed below.

Services that 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
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 to store ADQM Control data. To do this, you can add the ADPG service to the ADQM Control cluster or specify an external PostgreSQL server through the corresponding settings of the ADQM Control service.

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



The result of successful adding services to a cluster

Add hosts to a cluster

Elena Dvoryadkina, Daria Barysheva

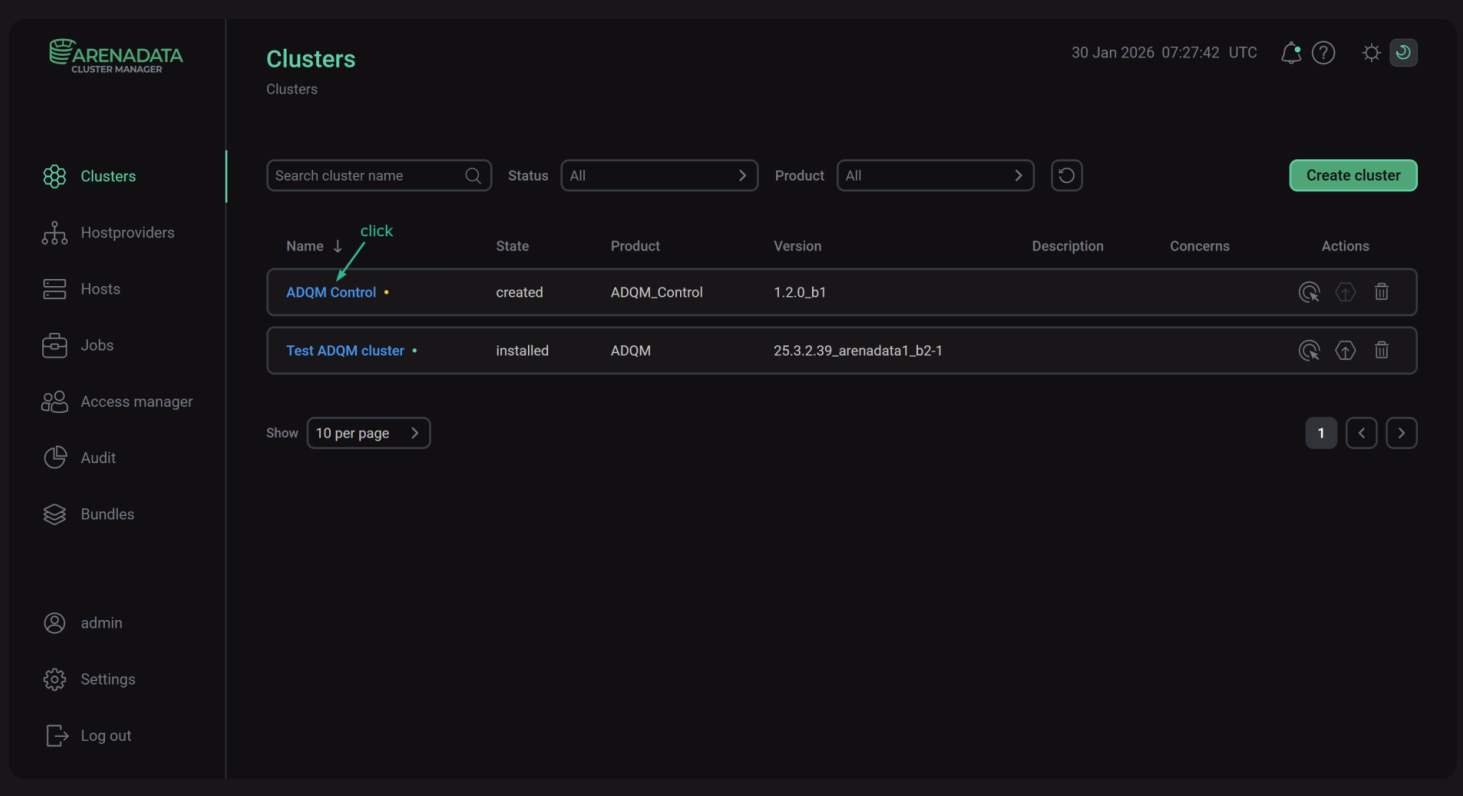
To Table of Contents

NOTE

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

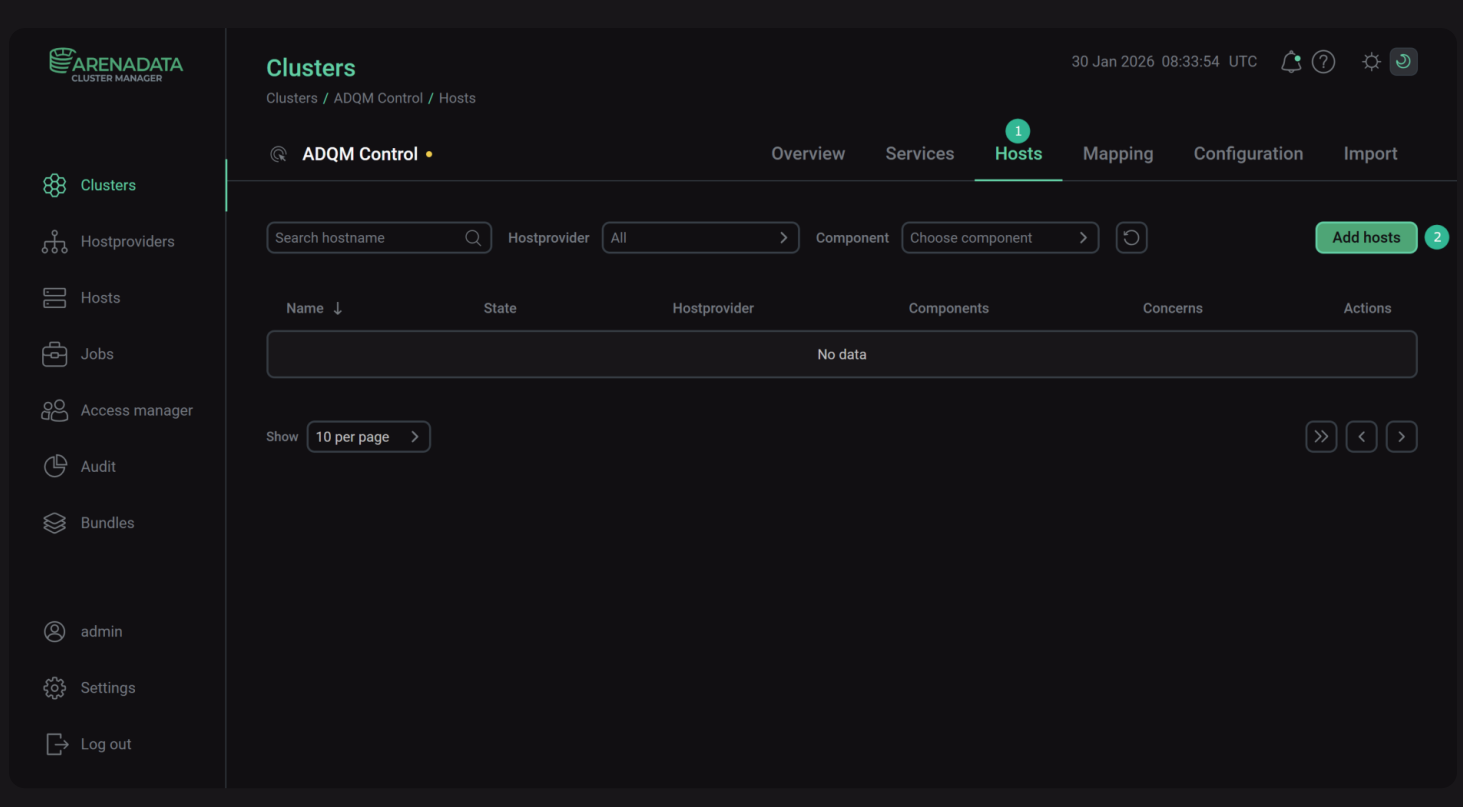
The steps for adding hosts to a cluster are listed below:

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



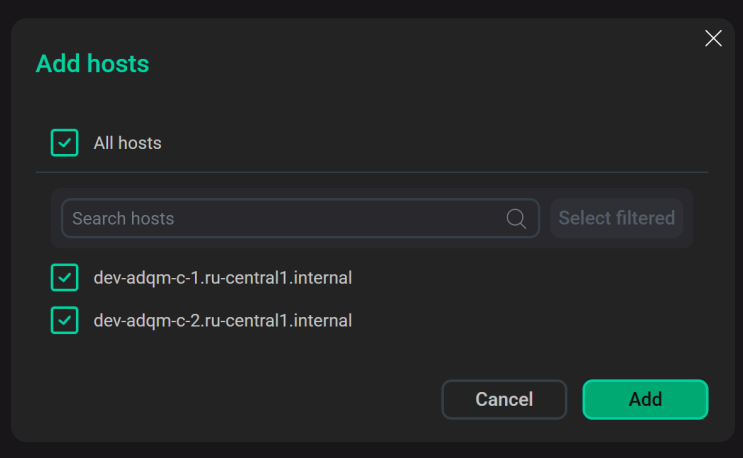
Select a cluster

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



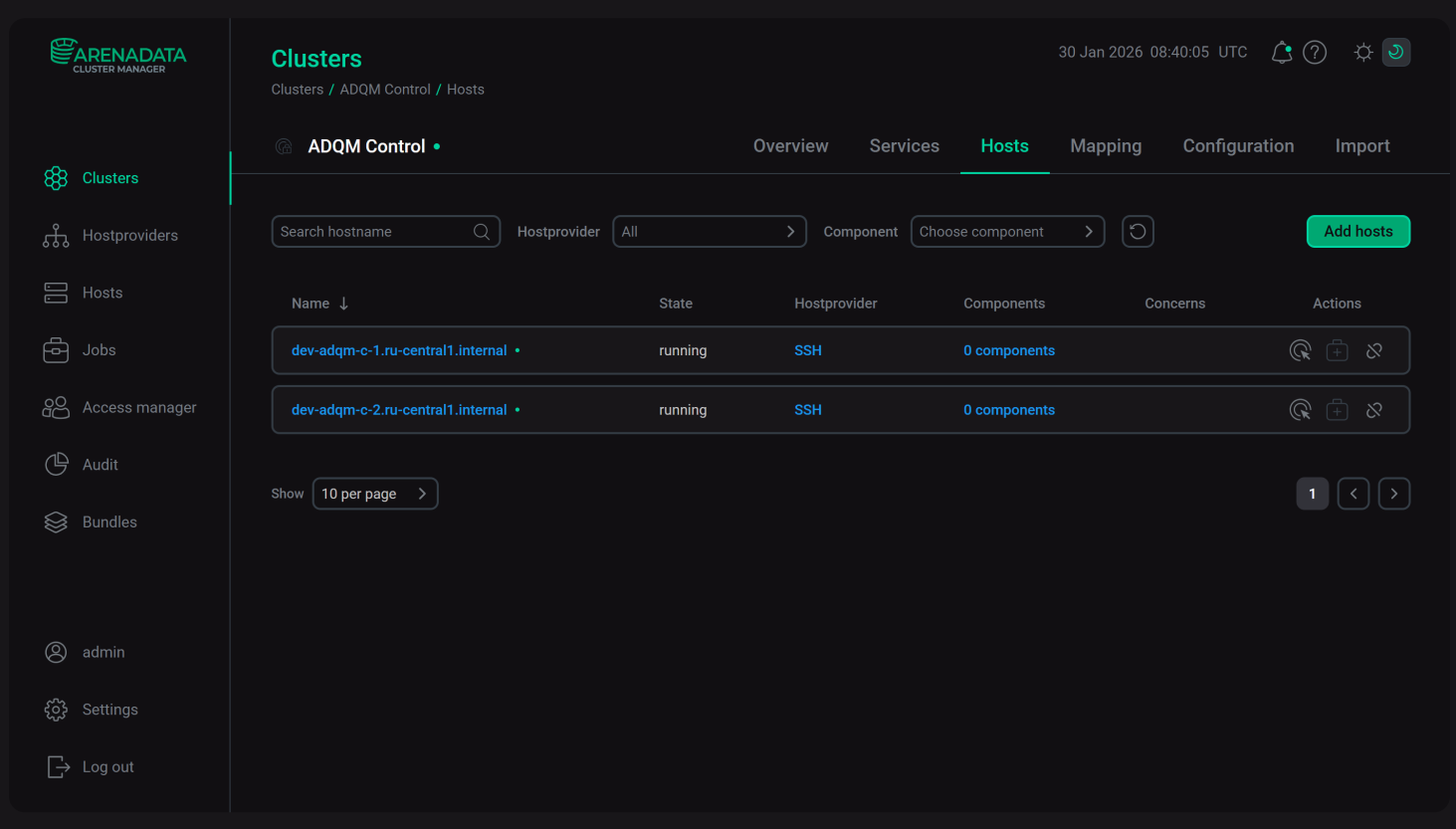
Switch to adding hosts

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



Select hosts

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



The result of successful adding hosts to a cluster

Add components

Elena Dvoryadkina, Daria Barysheva

ADQM CONTROL

To Table of Contents

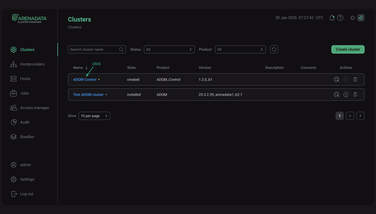
In ADCM, a **component** means a part of a service that should be deployed on one or several cluster hosts.

IMPORTANT

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

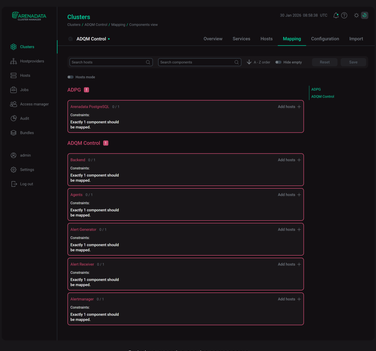
The steps for adding components to hosts are listed below:

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



Select a cluster

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



Switch to mapping service components

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

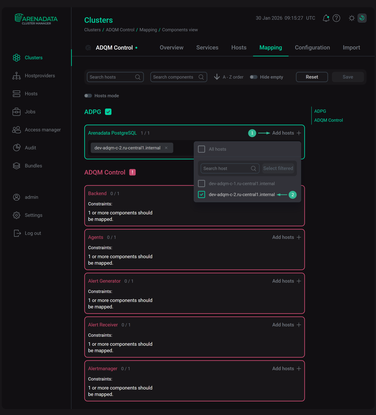
CAUTION

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

Components of ADQM Control services

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

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

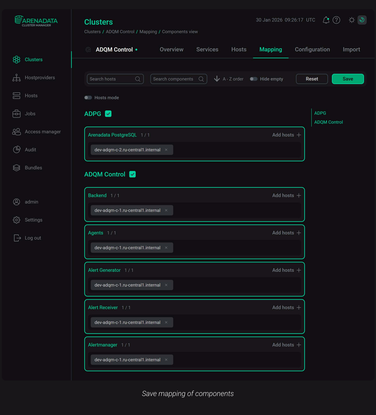


Select hosts for a component

TIP

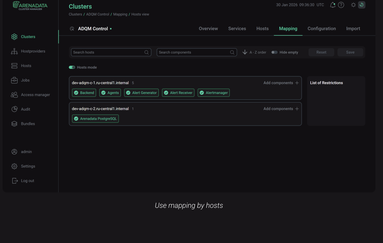
It is recommended to install the Arenadata PostgreSQL component on a dedicated host, separately from the components of the ADQM Control service.

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



Save mapping of components

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



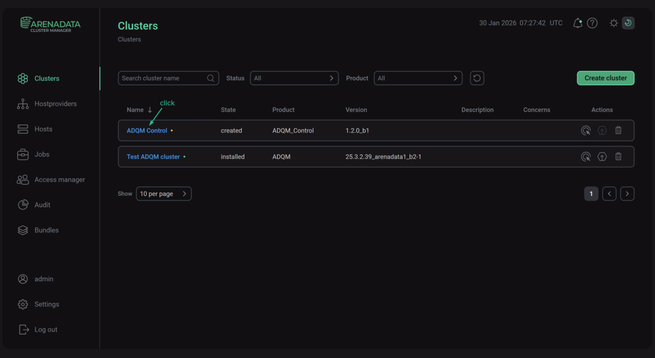
Use mapping by hosts

Configure services

Elena Dvoryadkina, Daria Barysheva

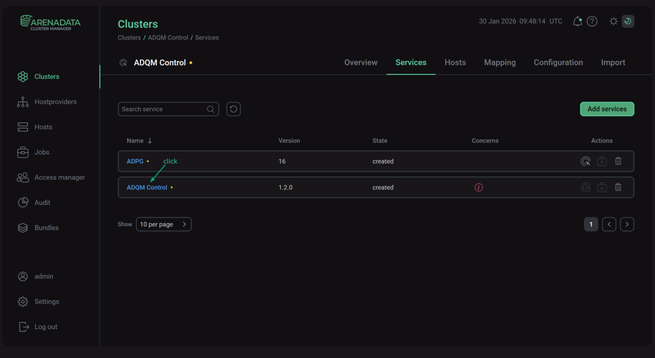
To Table of Contents

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



Select a cluster

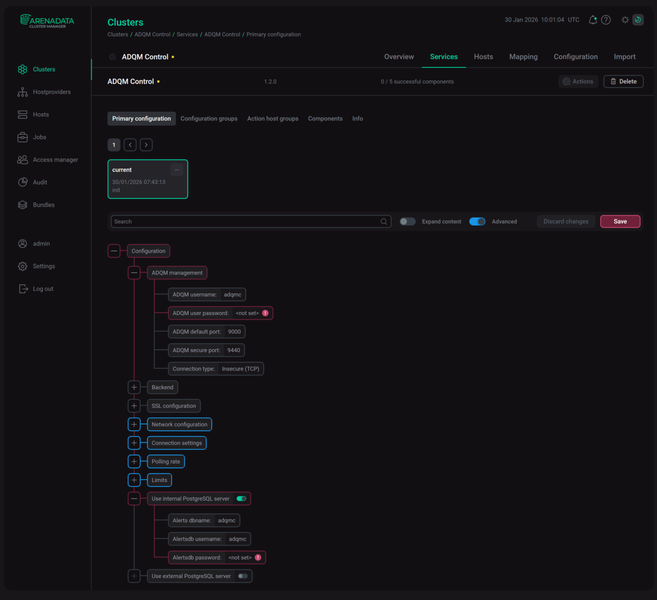
2. Open the ***Services*** tab on the cluster page, find the row that contains the desired service, and click its name in the ***Name*** column.



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 'qwerty123';
```

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

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

Configure a cluster

Elena Dvoryadkina, Daria Barysheva

Contents

To Table of Contents

Primary configuration

Configuration groups

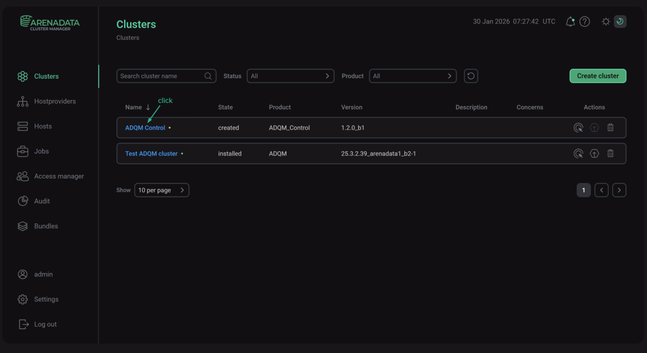
Ansible settings

IMPORTANT

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

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

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



Select a cluster

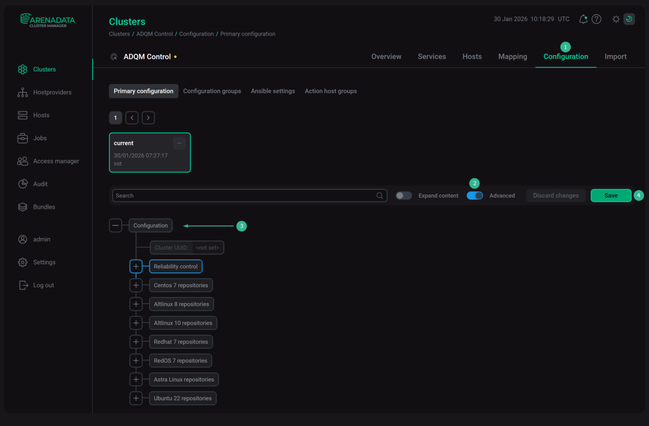
2. Open the ***Configuration*** tab on the cluster page. The ***Configuration*** tab includes the following sections: ***Primary configuration***, ***Configuration groups***, ***Ansible settings***.
3. Fill in all necessary parameters on the selected tab and click ***Save***.

Primary configuration

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

ADQM Control cluster configuration parameters are listed below:

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



Cluster configuration window

Configuration groups

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

Ansible settings

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

Ansible settings

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

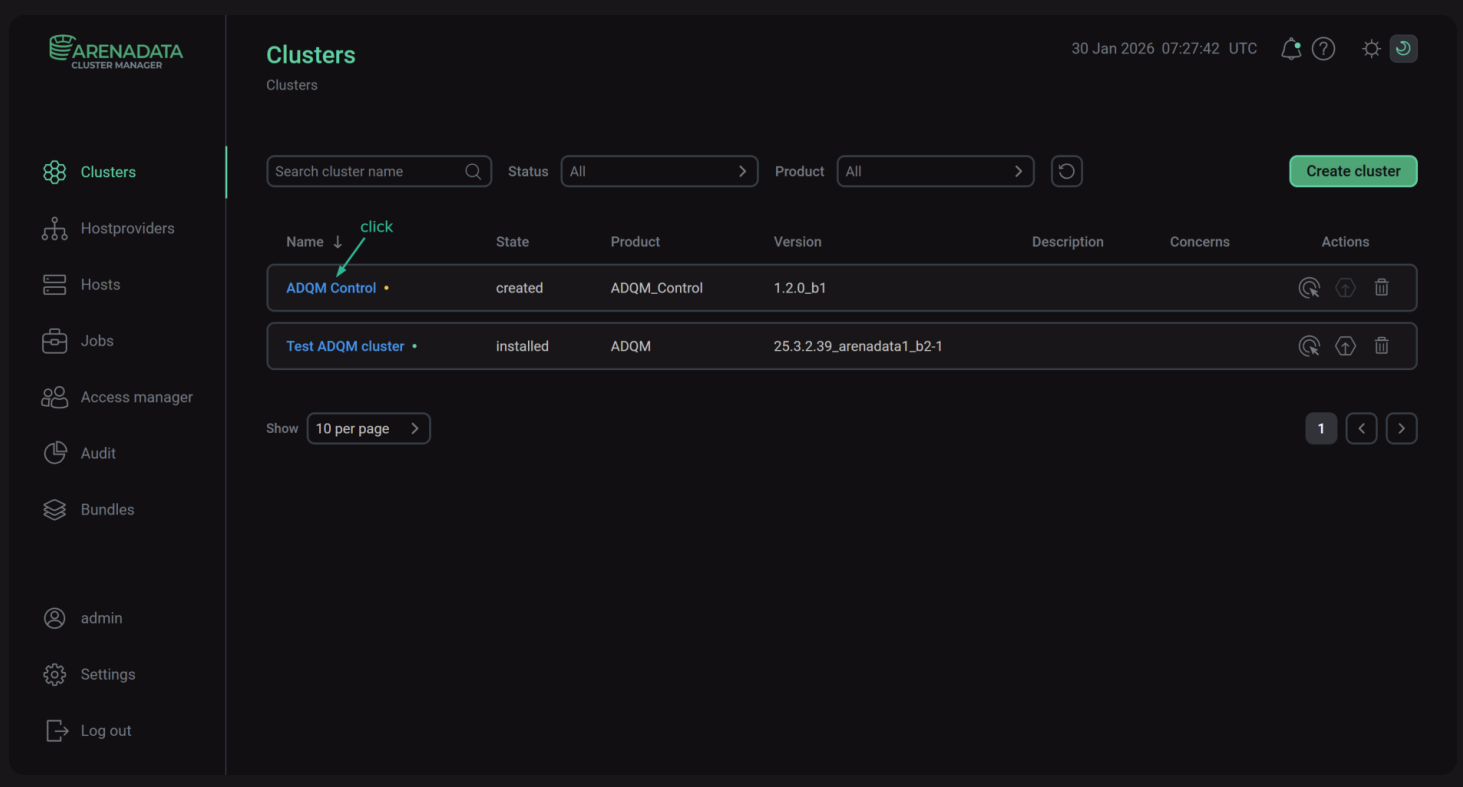
Integrate with an ADQM cluster

Elena Dvoryadkina

To Table of Contents

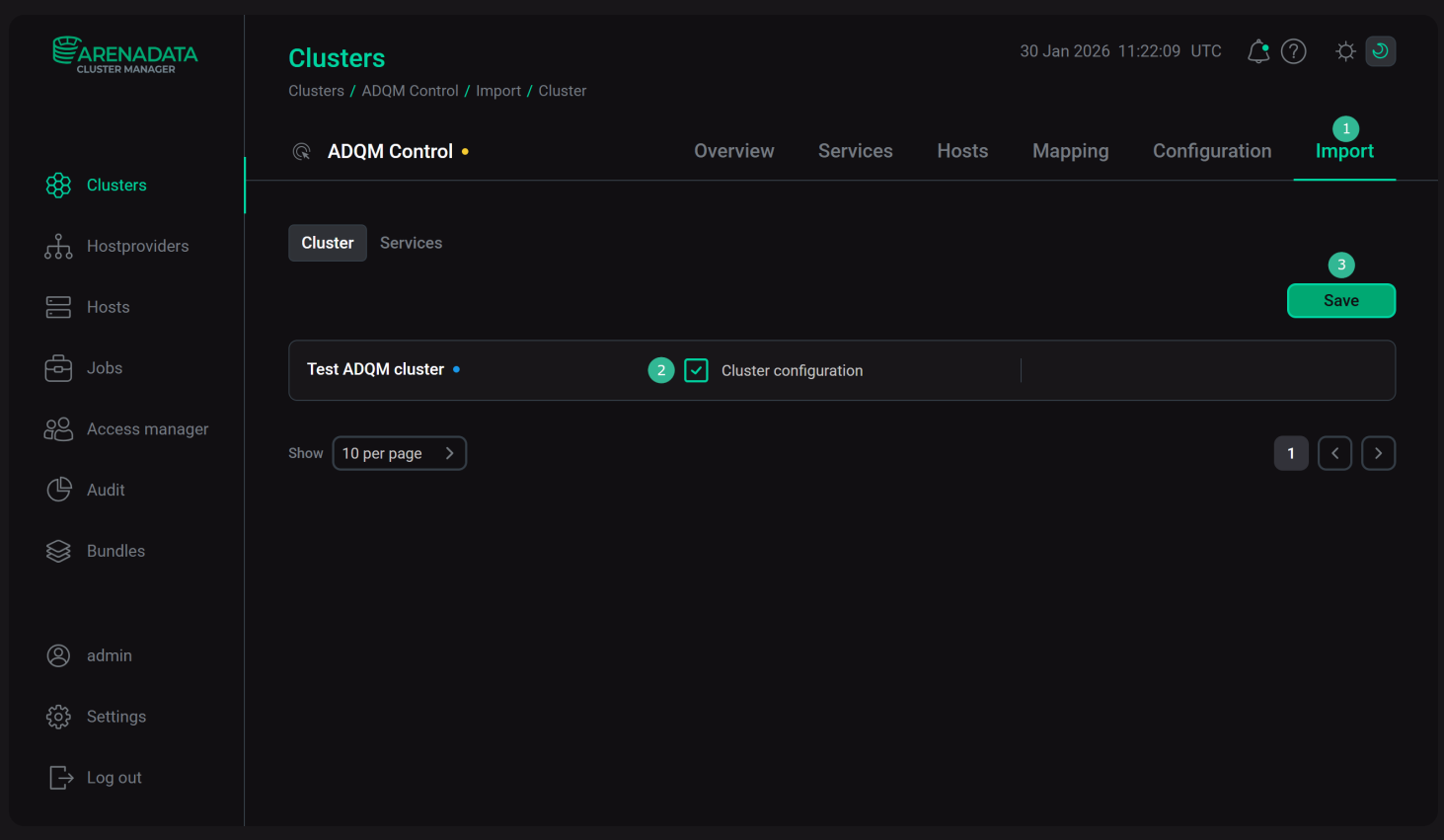
To set up an ADQM Control cluster, you should integrate it with at least one pre-installed ADQM cluster with the Monitoring service configured:

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



Select a cluster

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



Integration with an ADQM cluster

NOTE

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

Install a cluster

Elena Dorosyolkina, Elena Barysheva

ADQM CONTROL

Contents

To Table of Contents

Step 1. Run installation

- Whole cluster
- Single services

Step 2. View the results of installation

Step 3. Check the cluster state after installation

Step 4. Check the ADQM Control web interface

Step 1. Run installation

There are two ways to install cluster services:

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

In both cases each service starts automatically after installation.

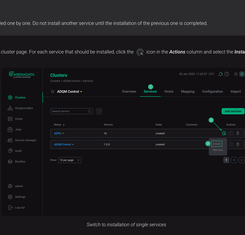
NOTE

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

Whole cluster

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

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



Switch to the whole cluster installation

2. Wait until the job is completed.

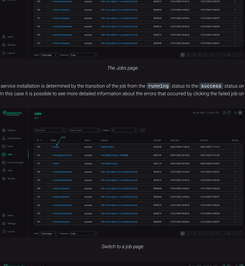
Single services

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

CAUTION

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

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



Switch to installation of single services

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

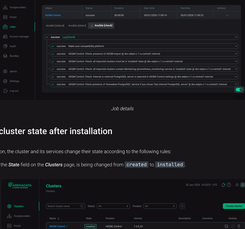
IMPORTANT

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

1. ADPG (if a connection to an external PostgreSQL cluster is not configured)
2. ADQM Control

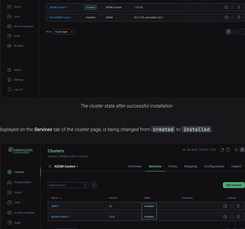
Step 2. View the results of installation

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

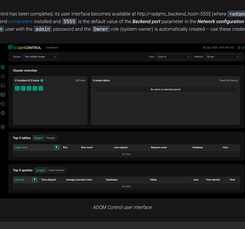


The Jobs page

The successful completion of the service installation is determined by the transition of the job from the `running` status to the `success` status on the **Jobs** page. If the job fails, it switches to the `failed` status. In this case it is possible to see more detailed information about the errors that occurred by clicking the failed job on the **Jobs** page.



Switch to a job page

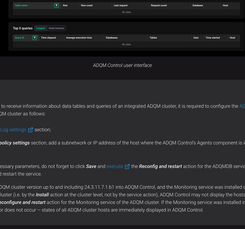


Install cluster job page

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

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

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

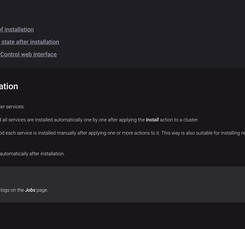


Job details

Step 3. Check the cluster state after installation

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

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



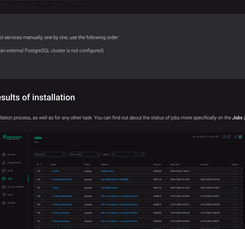
The cluster state after successful installation

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

The state of cluster services after successful installation

Step 4. Check the ADQM Control web interface

After the installation of ADQM Control has been completed, its user interface becomes available at http://<adpqc_backend_host>:5555 (where `<adpqc_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 `Senior` role (system owner) is automatically created - use these credentials to log in to ADQM Control.



ADQM Control user interface

IMPORTANT

- In order for ADQM Control to receive information about data tables and queries of an integrated ADQM cluster, it is required to configure the ADQMDB service parameters for the ADQM cluster as follows:
  - enable *query_log* in the *Log settings* section;
  - in the *Default user and policy settings* section, add a subnetwork or IP address of the host where the ADQM Control's Agents component is installed to the *Default user IP list*.

  After you have set the necessary parameters, do not forget to click *Save* and execute the *Reconfig and restart* action for the ADQMDB service to save configuration changes and restart the service.
- If you have imported an ADQM cluster version up to and including 24.3.1.17.1.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 actions), 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.

Install monitoring

ADQM CONTROL

Table of Contents

Contents

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

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

Overview

When adding the Monitoring service to ADQM Control, you can install the node exporter on all or selected hosts of your cluster. It is a monitoring agent that exports host system metrics. Prometheus server can then collect these metrics from node exporters and store them in its time-series database. You can also configure Prometheus server to collect metrics from other components specified in the clickhouse_exporter section. You can also use the Prometheus web interface to view and analyze data that the monitoring service collects.

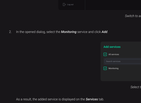
> **NOTE**
>
> - If you already have a Prometheus-compatible monitoring system set up (for example, your own Prometheus server or VictoriaMetrics), you can use it to collect ADQM Control metrics. To do this, configure access to ADQM Control metrics in your monitoring system using ADQM Control's Prometheus parameters specified in the Monitoring service page.
> - You can also use the VictoriaMetrics mechanism to migrate all metrics from the Prometheus server deployed in ADQM Control to VictoriaMetrics.

Step 1. Add the service

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

Switch to adding services

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



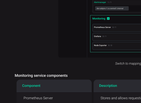
Select the service

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

The result of successfully adding the service to the cluster

Step 2. Add components

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

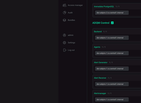


Switch to mapping service components

Monitoring service components

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

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



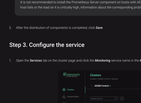
Select a host for a component

> **CAUTION**
>
> It is not recommended to install the Prometheus Server component on hosts with ADQM Control – can separate hosts for it. Otherwise, if an ADQM Control host fails or the load on it is critically high, information about the corresponding problems will not be saved.

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

Step 3. Configure the service

1. Open the **Services** tab on the cluster page and click the **Monitoring** service name in the **Name** column.



Switch to the service configuration

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

Configure the Monitoring service

Fields highlighted in red are required.

- **Prometheus settings** → **Password for ClickHouse connection** – a password of a ClickHouse user (**admin_no_grafana**) that is used to connect to Prometheus. The password should contain a combination of numbers, symbols, lowercase and uppercase letters and have a minimum length of 8.
- **Grafana settings** → **Grafana administrator's password** – a password of the **admin** Grafana user. The password should contain a combination of numbers, symbols, lowercase and uppercase letters and have a minimum length of 8.

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

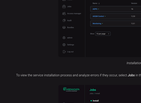
3. After specifying all necessary parameters, click **Save**.

Step 4. Install the service

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

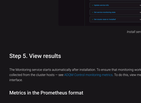
Switch to the service installation

2. Wait until the installation is completed. Then check that the service state has changed from **created** → **installed**.



Installation is complete

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



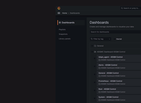
Install service job page

Step 5. View results

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

Metrics in the Prometheus format

1. In the address bar of your browser, enter an IP address of a host on which ClickHouse is deployed and add a port number. The default port is **9363**. For example, http://10.92.40.201:7256. To do this, use the Prometheus interface, use a username and password that are also specified in the **Prometheus settings** section – the **Prometheus users to sign/logout in Prometheus** setting.
2. In the **Expression** field, you can enter a metric and click **Execute** – values of this metric on all hosts of the ADQM Control cluster will be shown in the interface.



Prometheus web interface

Grafana web interface

1. In the address bar of your browser, enter an address of a host on which Grafana is deployed and add a port number. The default port is **3000**. For example, http://10.92.40.201:7256. To log in, use **admin** as a username, and the **Grafana administrator's password** parameter value specified in the **Grafana settings** section of service configuration parameters as a password.
2. In the window that opens, navigate to **Home** → **Dashboards** and expand the **ADQMC Dashboard ADQM_Control_cluster_name** section. In this section, you can select one of the available dashboards to view service's system metrics coming from your ADQM Control cluster.

View ADQM Control metrics in Grafana

Offline installation

To Table of Contents

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

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

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

Step 1.

Install ADCM

Step 2.

Prepare hosts

Step 3.

Install ET cluster

Step 4.

Install ADQM cluster

Step 5.

Install monitoring in ADQM

Step 6.

Install ADQM Control

Step 7.

Install monitoring in ADQM Control

Create a cluster

Elena Dvoryadkina, Daria Barysheva

AD.QM 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.

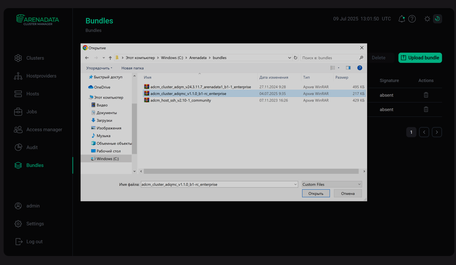
Step 1. Get the Enterprise version of a cluster bundle

Please get a cluster bundle from the Arenadata support team.

Step 2. Upload a cluster bundle to ADCM

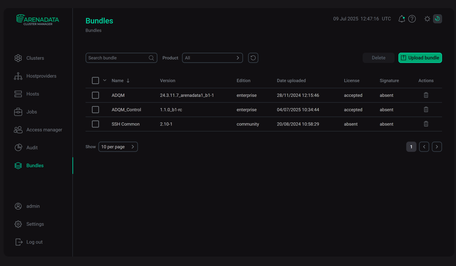
To upload a bundle to ADCM, follow the steps:

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

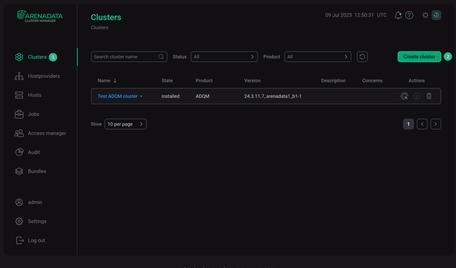
Step 3. Create a cluster based on the uploaded bundle

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.

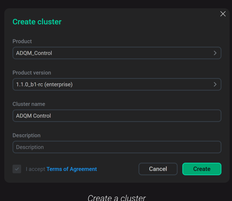
The steps for cluster creation are given below:

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



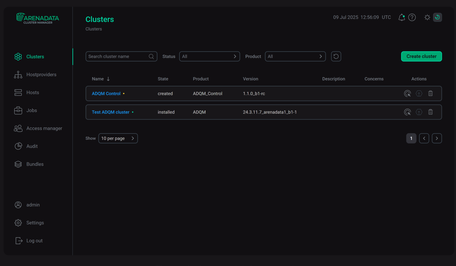
Switch to cluster creation

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

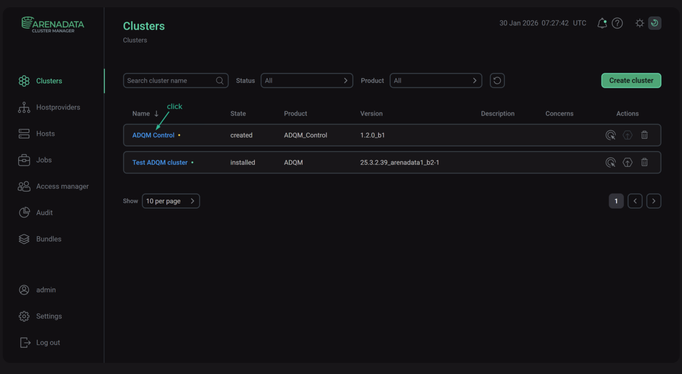
Add services

Elena Dvoryadkina, Daria Barysheva

To Table of Contents

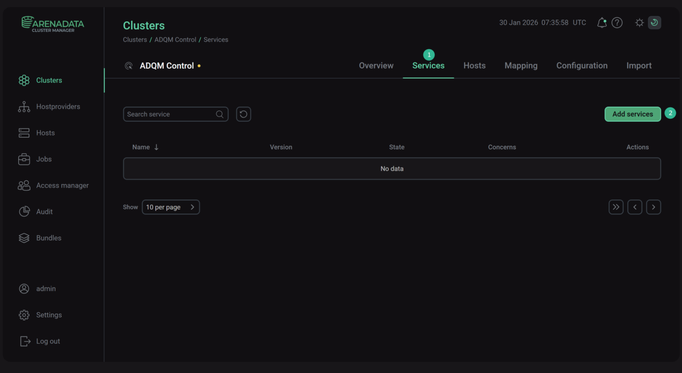
In ADCM, a **service** means a software that performs some function. The steps for adding services to a cluster are listed below:

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



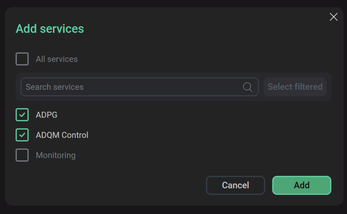
Select a cluster

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



Switch to adding services

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



Select services

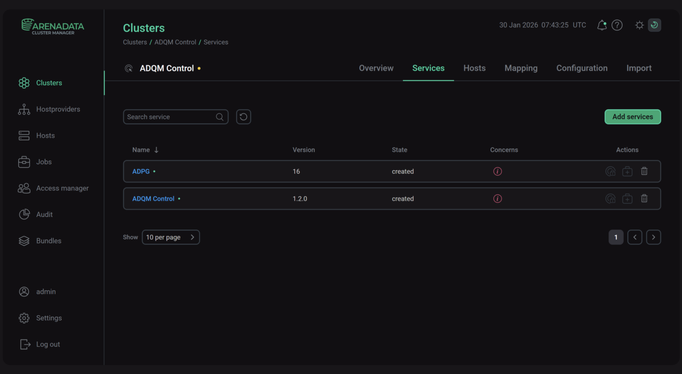
The brief description of available services is listed below.

Services that 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
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 to store ADQM Control data. To do this, you can add the ADPG service to the ADQM Control cluster or specify an external PostgreSQL server through the corresponding settings of the ADQM Control service.

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



The result of successful adding services to a cluster

Add hosts to a cluster

Elena Dvoryadkina, Daria Barysheva

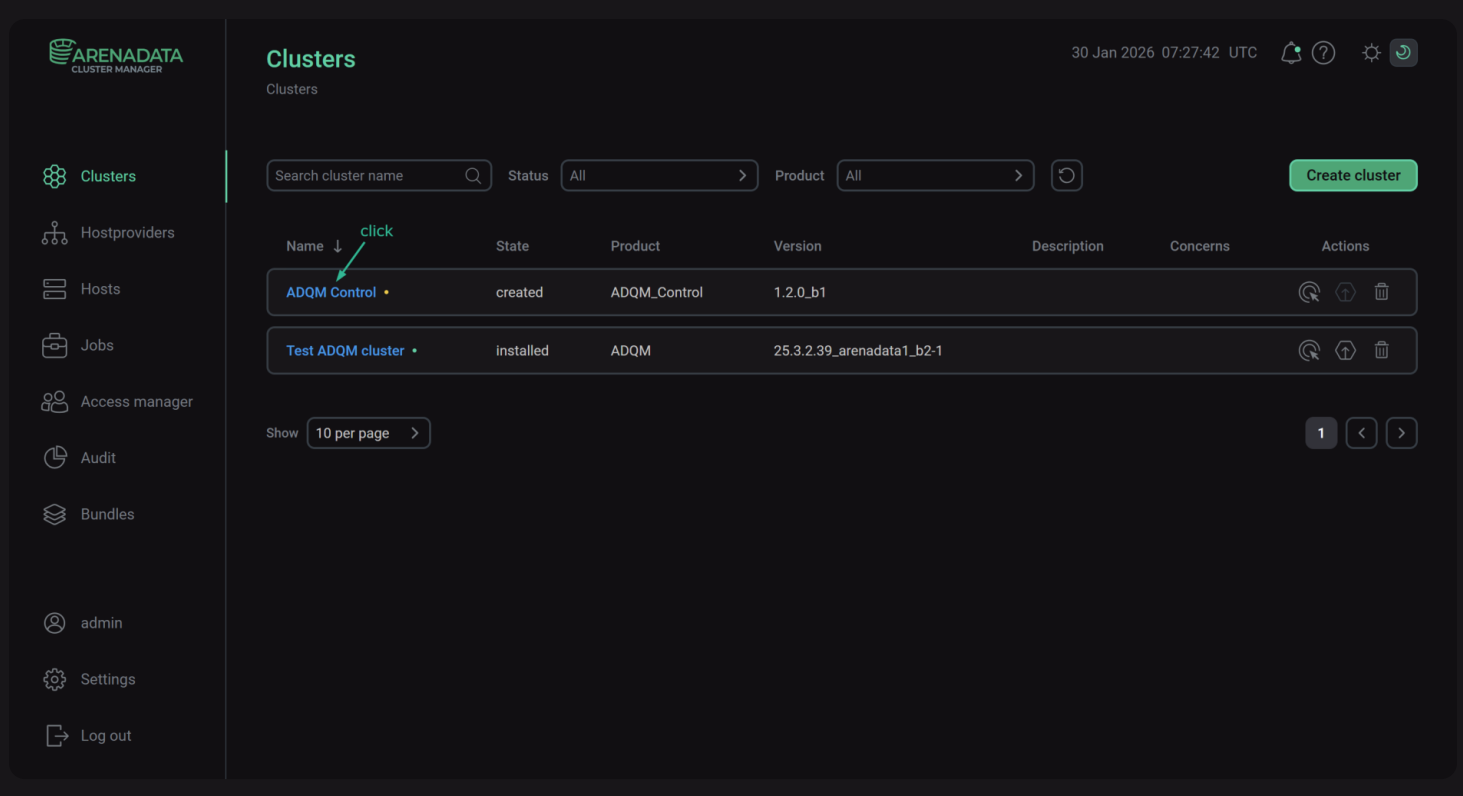
To Table of Contents

NOTE

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

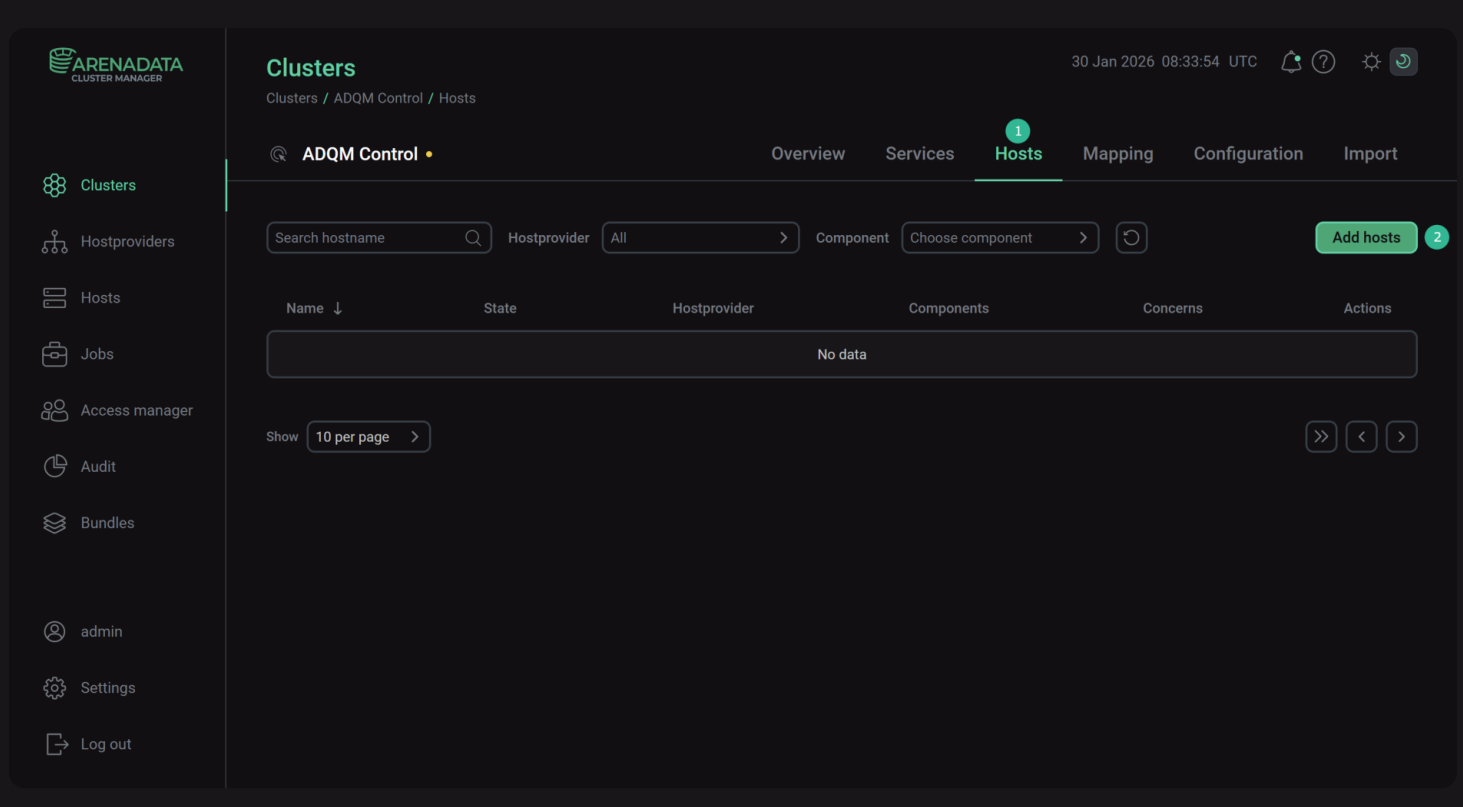
The steps for adding hosts to a cluster are listed below:

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



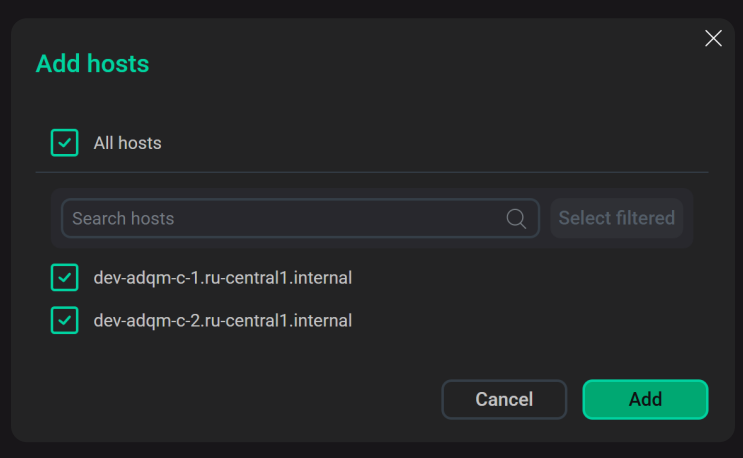
Select a cluster

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



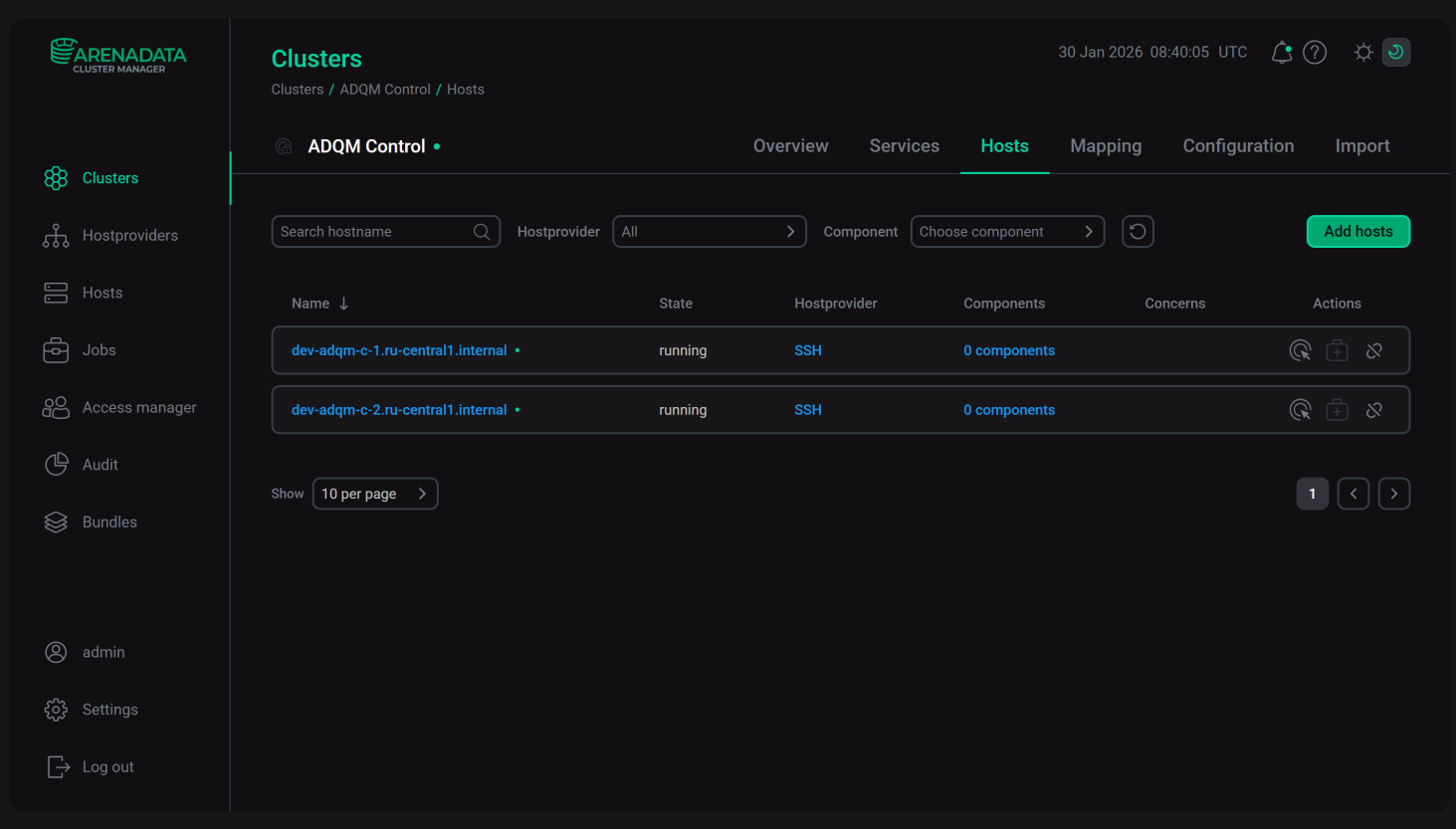
Switch to adding hosts

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



Select hosts

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



The result of successful adding hosts to a cluster

Add components

Elena Dvoryadkina, Daria Barysheva

ADQM CONTROL

To Table of Contents

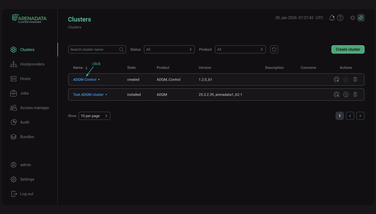
In ADCM, a **component** means a part of a service that should be deployed on one or several cluster hosts.

IMPORTANT

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

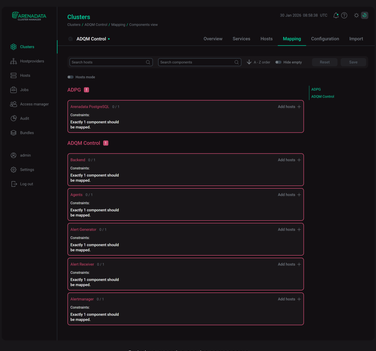
The steps for adding components to hosts are listed below:

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



Select a cluster

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



Switch to mapping service components

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

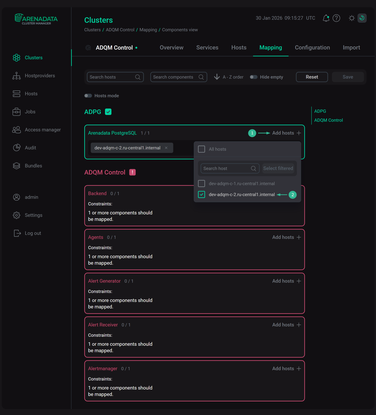
CAUTION

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

Components of ADQM Control services

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

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

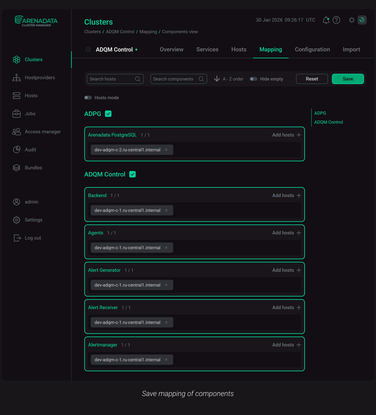


Select hosts for a component

TIP

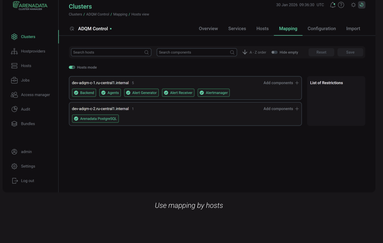
It is recommended to install the Arenadata PostgreSQL component on a dedicated host, separately from the components of the ADQM Control service.

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



Save mapping of components

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



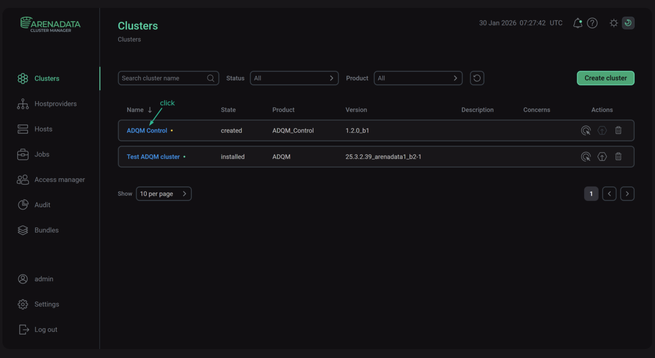
Use mapping by hosts

Configure services

Elena Dvoryadkina, Daria Barysheva

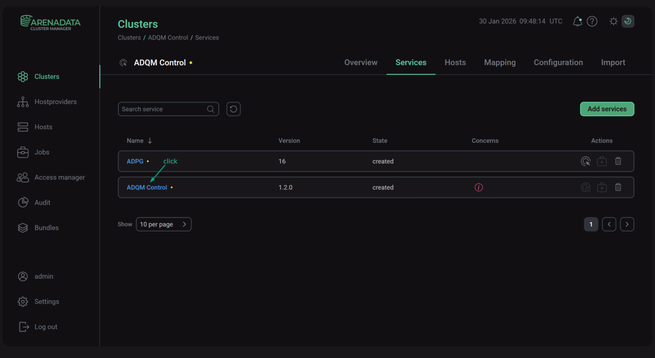
To Table of Contents

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



Select a cluster

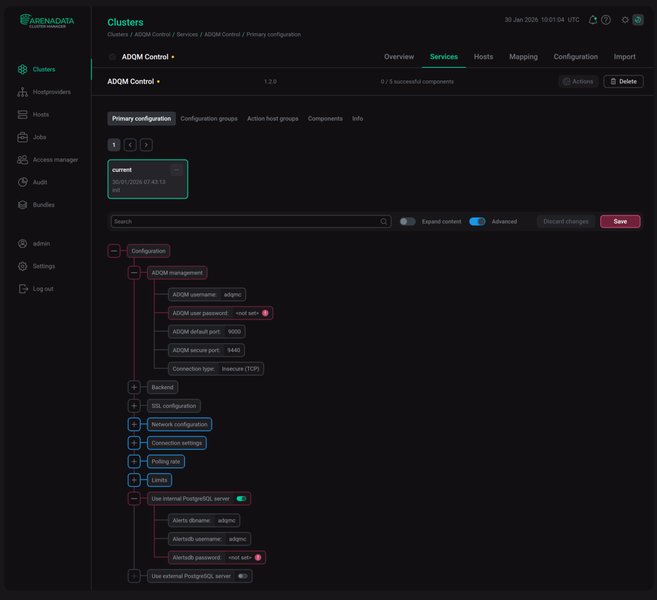
2. Open the ***Services*** tab on the cluster page, find the row that contains the desired service, and click its name in the ***Name*** column.



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 'qwerty123';
```

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

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

Configure a cluster

Elena Dvoryadkina, Daria Barysheva

Contents

To Table of Contents

Primary configuration

Configuration groups

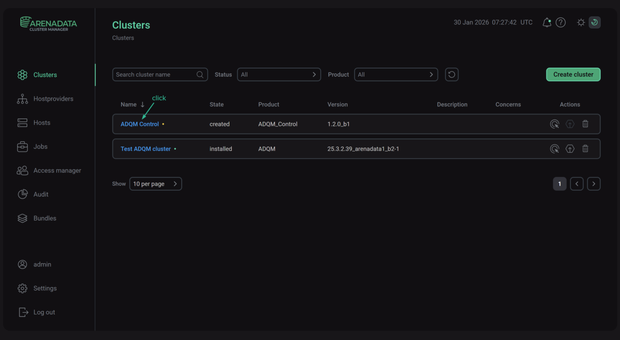
Ansible settings

IMPORTANT

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

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

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



Select a cluster

2. Open the ***Configuration*** tab on the cluster page. The ***Configuration*** tab includes the following sections: ***Primary configuration***, ***Configuration groups***, ***Ansible settings***.

3. Fill in all necessary parameters on the selected tab and click ***Save***.

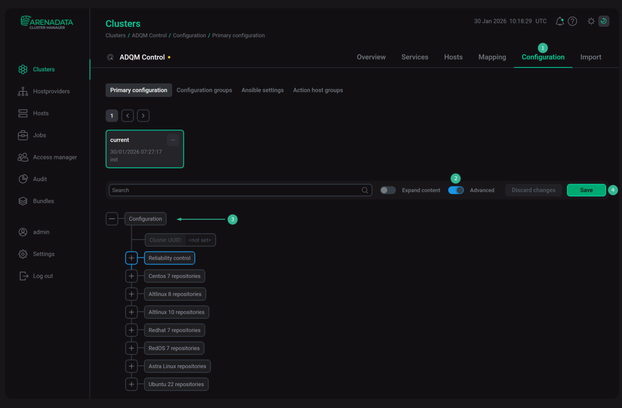
Primary configuration

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

ADQM Control cluster configuration parameters are listed below:

- ***Reliability control*** — the section with the following settings (visible if the ***Show advanced*** option is enabled):
  - ***retries*** — number of retries for cluster operations (`3` by default);
  - ***delay*** — delay between retries in seconds (`10` by default);
  - ***timeout*** — timeout for checks of network ports in seconds (`60` by default).

- Paths to the repositories that are used during the installation process — depending on the cluster operating system (Centos, Alt Linux, Red Hat, RED OS, Astra Linux, or Ubuntu):
  - ***ADQMC_arenadata_community*** — repository required to install the Community edition of ADQM Control;
  - ***arenadata_postgres*** — repository required to install Arenadata PostgreSQL;
  - ***prometheus_repo*** — repository required to install Prometheus;
  - ***ADQMC_arenadata_enterprise*** — repository required to install the Enterprise edition of ADQM Control.



Cluster configuration window

Configuration groups

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

Ansible settings

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

Ansible settings

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

NOTE

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

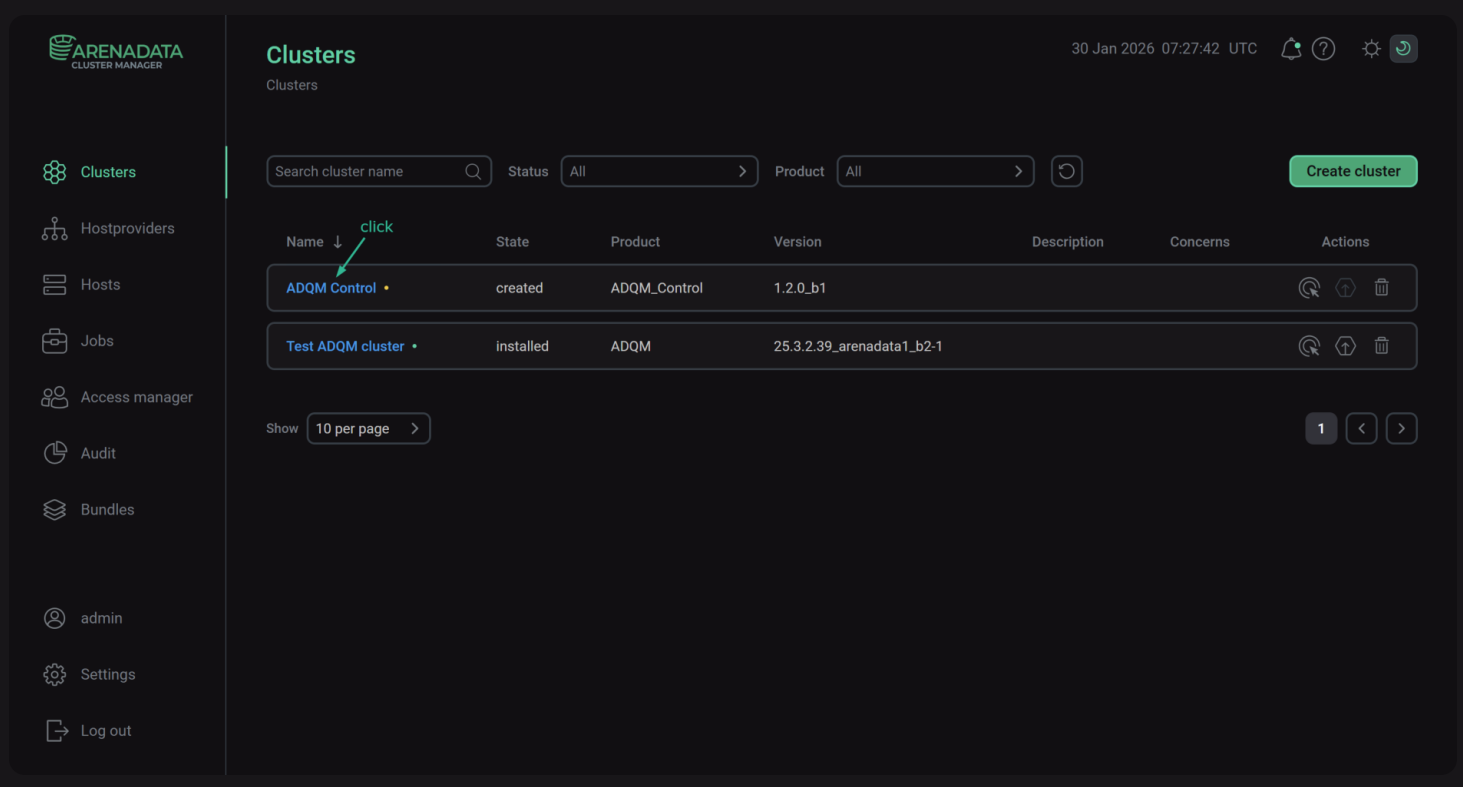
Integrate with an ADQM cluster

Elena Dvoryadkina

To Table of Contents

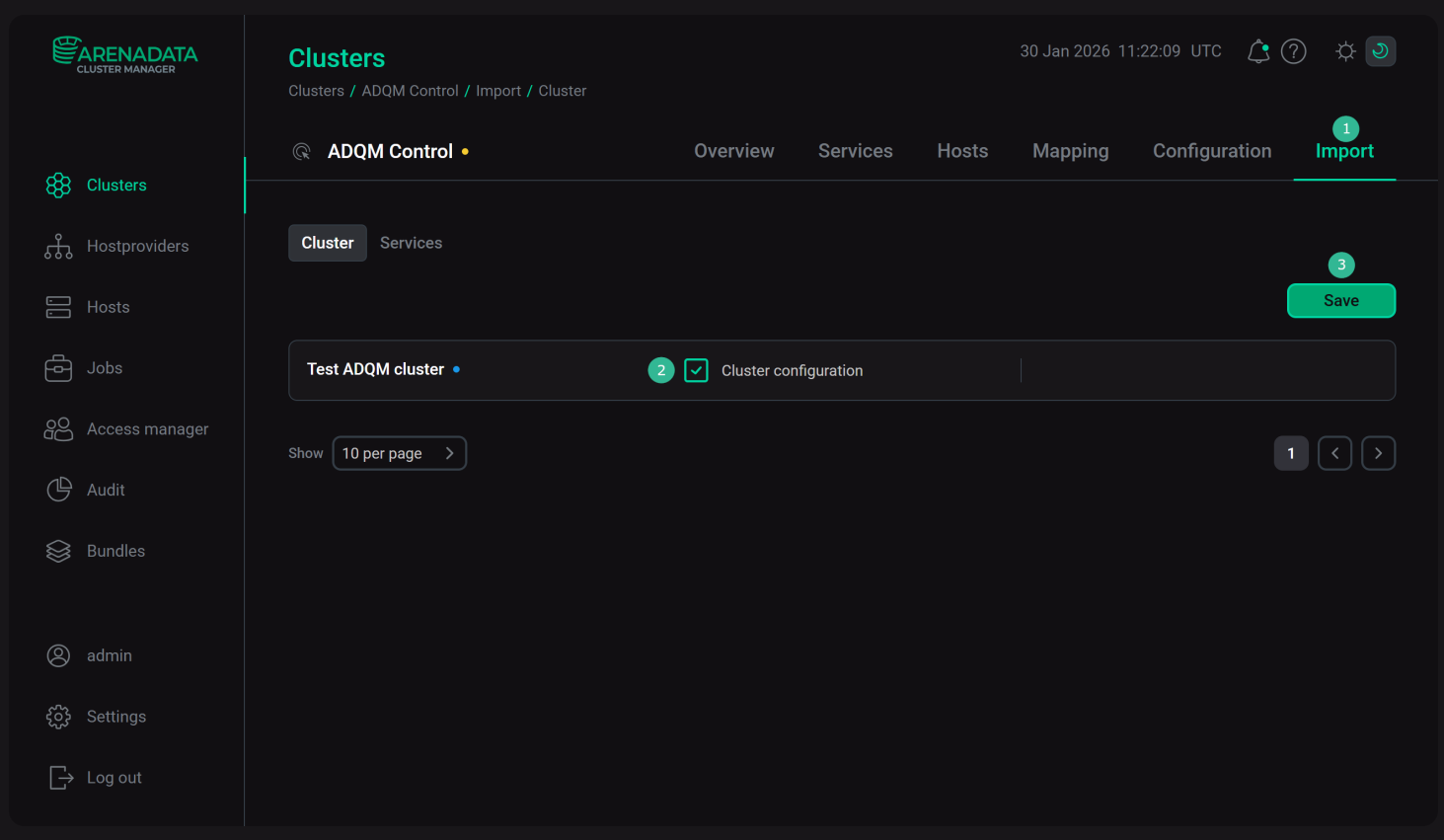
To set up an ADQM Control cluster, you should integrate it with at least one pre-installed ADQM cluster with the Monitoring service configured:

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



Select a cluster

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



Integration with an ADQM cluster

NOTE

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

Import ET settings

Daria Barysheva, Elena Dvoryadkina

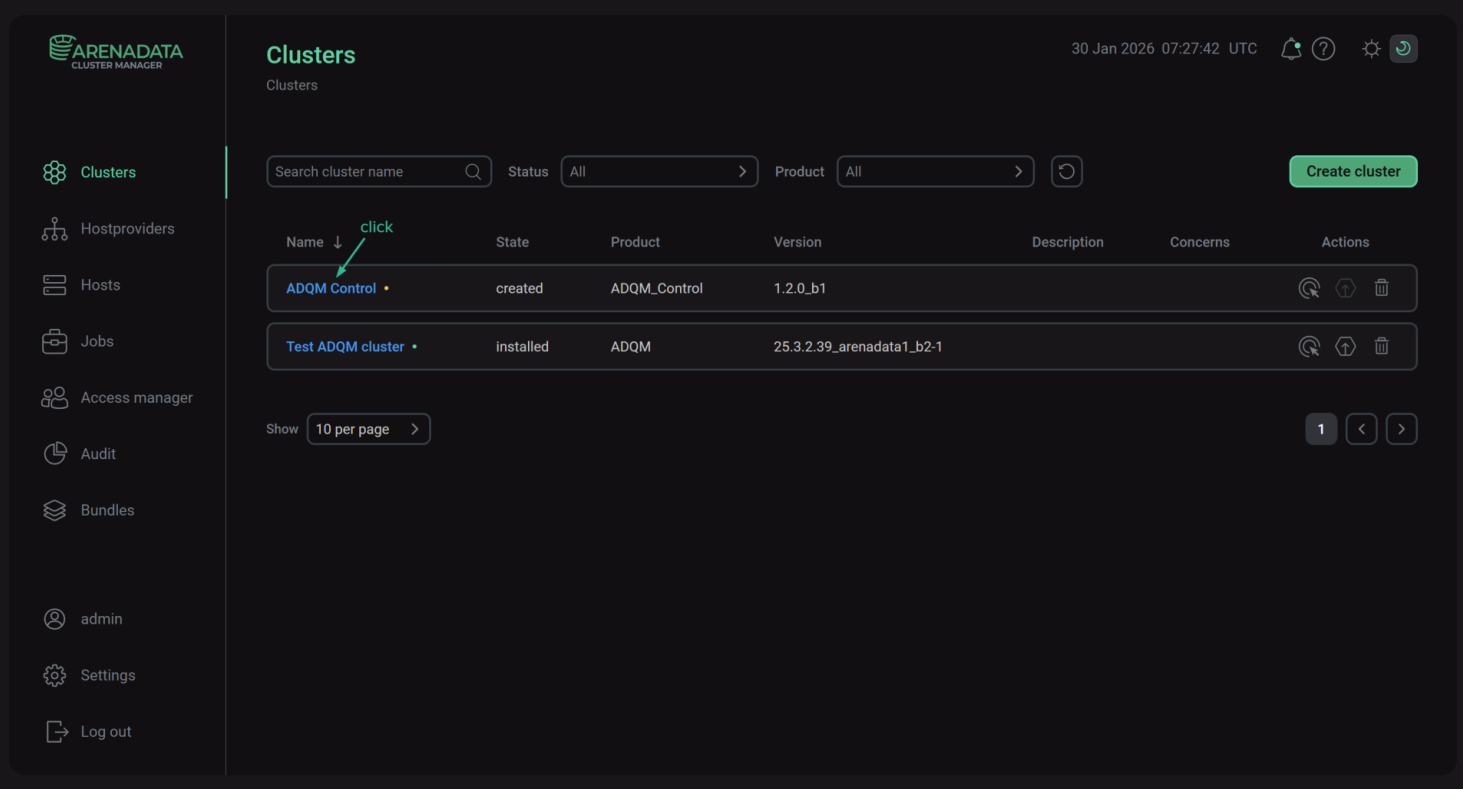
To Table of Contents

IMPORTANT

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

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

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

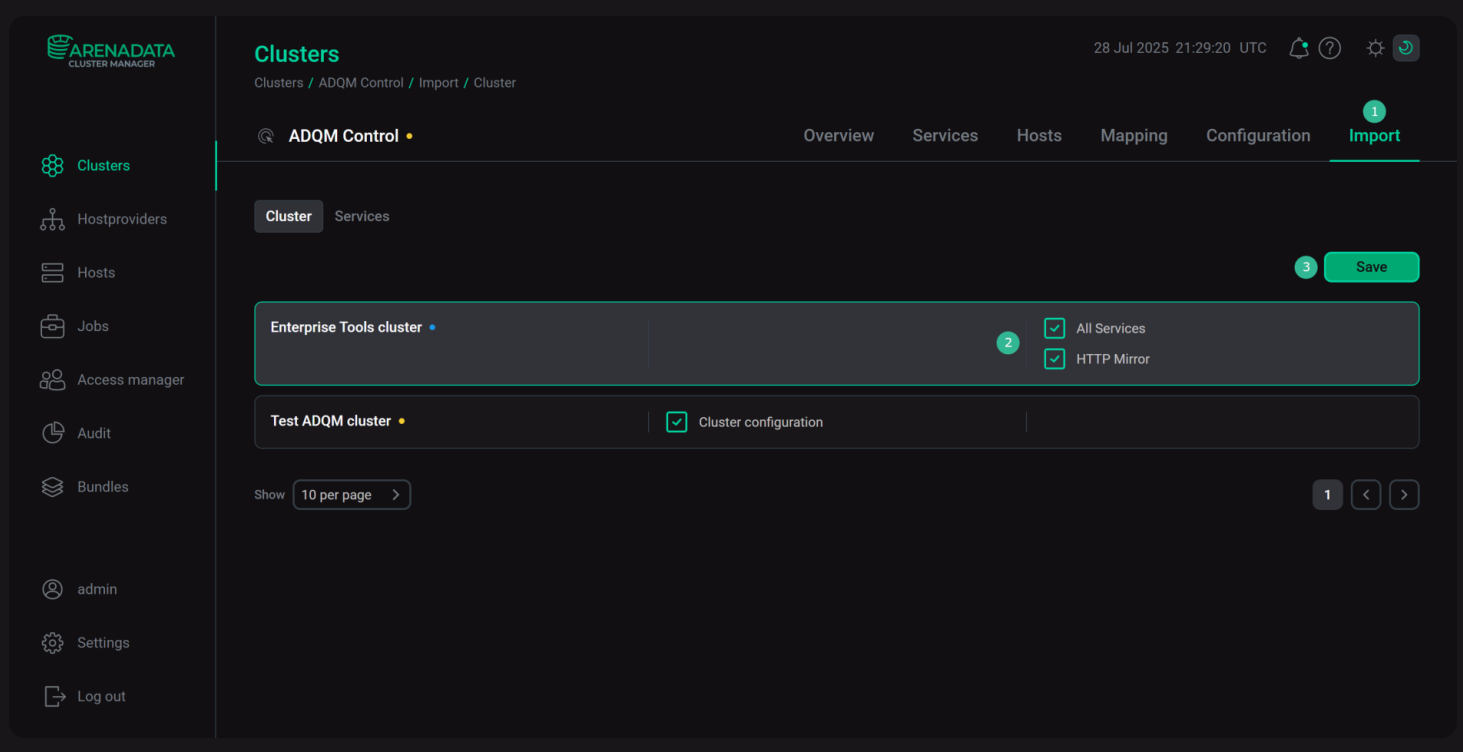


Select a cluster

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

3. Select the ***HTTP Mirror*** service of the previously created ET cluster.

4. Click ***Save***.



Integration with the ET cluster

NOTE

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

Install a cluster

Elena Dorosyolkina, Elena Barysheva

ADQM CONTROL

Contents

To Table of Contents

Step 1. Run installation

- Whole cluster
- Single services

Step 2. View the results of installation

Step 3. Check the cluster state after installation

Step 4. Check the ADQM Control web interface

Step 1. Run installation

There are two ways to install cluster services:

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

In both cases each service starts automatically after installation.

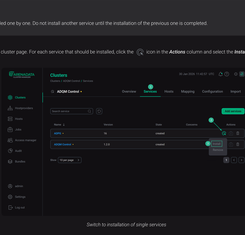
NOTE

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

Whole cluster

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

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



Switch to the whole cluster installation

2. Wait until the job is completed.

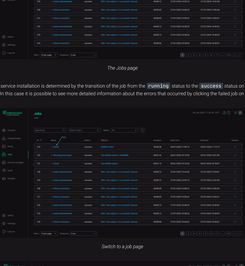
Single services

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

CAUTION

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

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



Switch to installation of single services

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

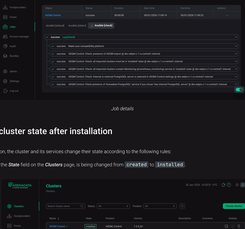
IMPORTANT

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

1. ADPG (if a connection to an external PostgreSQL cluster is not configured)
2. ADQM Control

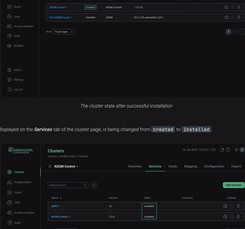
Step 2. View the results of installation

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

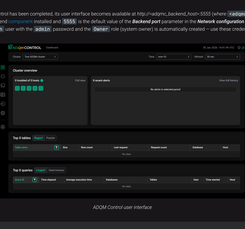


The Jobs page

The successful completion of the service installation is determined by the transition of the job from the `running` status to the `success` status on the **Jobs** page. If the job fails, it switches to the `failed` status. In this case it is possible to see more detailed information about the errors that occurred by clicking the failed job on the **Jobs** page.



Switch to a job page

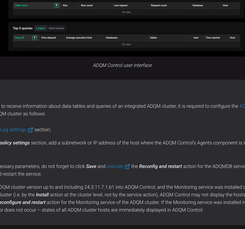


Install cluster job page

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

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

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

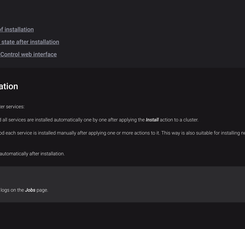


Job details

Step 3. Check the cluster state after installation

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

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



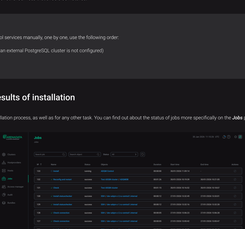
The cluster state after successful installation

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

The state of cluster services after successful installation

Step 4. Check the ADQM Control web interface

After the installation of ADQM Control has been completed, its user interface becomes available at http://<adpqc_backend_host>:5555 (where `<adpqc_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 `Senior` role (system owner) is automatically created - use these credentials to log in to ADQM Control.



ADQM Control user interface

IMPORTANT

- In order for ADQM Control to receive information about data tables and queries of an integrated ADQM cluster, it is required to configure the ADQMDB service parameters for the ADQM cluster as follows:
  - enable *query_log* in the *Log settings* section;
  - in the *Default user and policy settings* section, add a subnetwork or IP address of the host where the ADQM Control's Agents component is installed to the *Default user IP list*.

  After you have set the necessary parameters, do not forget to click *Save* and execute the *Reconfig and restart* action for the ADQMDB service to save configuration changes and restart the service.
- If you have imported an ADQM cluster version up to and including 24.3.1.17.1.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 actions), 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.

Install monitoring

ADQM CONTROL

Table of Contents

Contents

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

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

Overview

When adding the Monitoring service to ADQM Control, you can install the node exporter on all or selected hosts of your cluster. It is a monitoring agent that exports host system metrics. Prometheus server can then collect these metrics from node exporters and store them in its time-series database. You can also configure Prometheus server to collect metrics from other components specified in the clickhouse_exporter section. You can also use the Prometheus web interface to view and analyze data that the monitoring service collects.

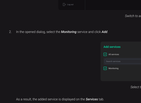
> **NOTE**
>
> - If you already have a Prometheus-compatible monitoring system set up (for example, your own Prometheus server or VictoriaMetrics), you can use it to collect ADQM Control metrics. To do this, configure access to ADQM Control metrics in your monitoring system using ADQM Control's Prometheus parameters specified in the Monitoring service page.
> - You can also use the VictoriaMetrics mechanism to migrate all metrics from the Prometheus server deployed in ADQM Control to VictoriaMetrics.

Step 1. Add the service

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

Switch to adding services

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



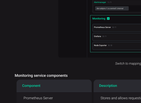
Select the service

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

The result of successfully adding the service to the cluster

Step 2. Add components

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

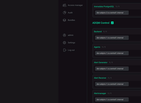


Switch to mapping service components

Monitoring service components

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

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



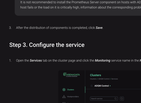
Select a host for a component

> **CAUTION**
>
> It is not recommended to install the Prometheus Server component on hosts with ADQM Control – can separate hosts for it. Otherwise, if an ADQM Control host fails or the load on it is critically high, information about the corresponding problems will not be saved.

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

Step 3. Configure the service

1. Open the **Services** tab on the cluster page and click the **Monitoring** service name in the **Name** column.



Switch to the service configuration

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

Configure the Monitoring service

Fields highlighted in red are required.

- **Prometheus settings** → **Password for ClickHouse connection** – a password of a ClickHouse user (**admin_no_grafana**) that is used to connect to Prometheus. The password should contain a combination of numbers, symbols, lowercase and uppercase letters and have a minimum length of 8.
- **Grafana settings** → **Grafana administrator's password** – a password of the **admin** Grafana user. The password should contain a combination of numbers, symbols, lowercase and uppercase letters and have a minimum length of 8.

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

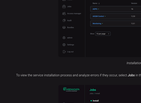
3. After specifying all necessary parameters, click **Save**.

Step 4. Install the service

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

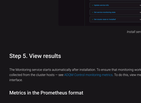
Switch to the service installation

2. Wait until the installation is completed. Then check that the service state has changed from **created** → **installed**.



Installation is complete

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



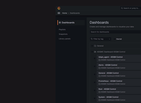
Install service job page

Step 5. View results

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

Metrics in the Prometheus format

1. In the address bar of your browser, enter an IP address of a host on which ClickHouse is deployed and add a port number. The default port is **9363**. For example, http://10.92.40.201:7256. To do this, use the Prometheus interface, use a username and password that are also specified in the **Prometheus settings** section – the **Prometheus users to sign/logout in Prometheus** setting.
2. In the **Expression** field, you can enter a metric and click **Execute** – values of this metric on all hosts of the ADQM Control cluster will be shown in the interface.



Prometheus web interface

Grafana web interface

1. In the address bar of your browser, enter an address of a host on which Grafana is deployed and add a port number. The default port is **3000**. For example, http://10.92.40.201:7256. To log in, use **admin** as a username, and the **Grafana administrator's password** parameter value specified in the **Grafana settings** section of service configuration parameters as a password.
2. In the window that opens, navigate to **Home** → **Dashboards** and expand the **ADQMC Dashboard ADQM_Control_cluster_name** section. In this section, you can select one of the available dashboards to view service's system metrics coming from your ADQM Control cluster.

View ADQM Control metrics in Grafana

Connect to ADQM Control

Elena Dvoryadkina

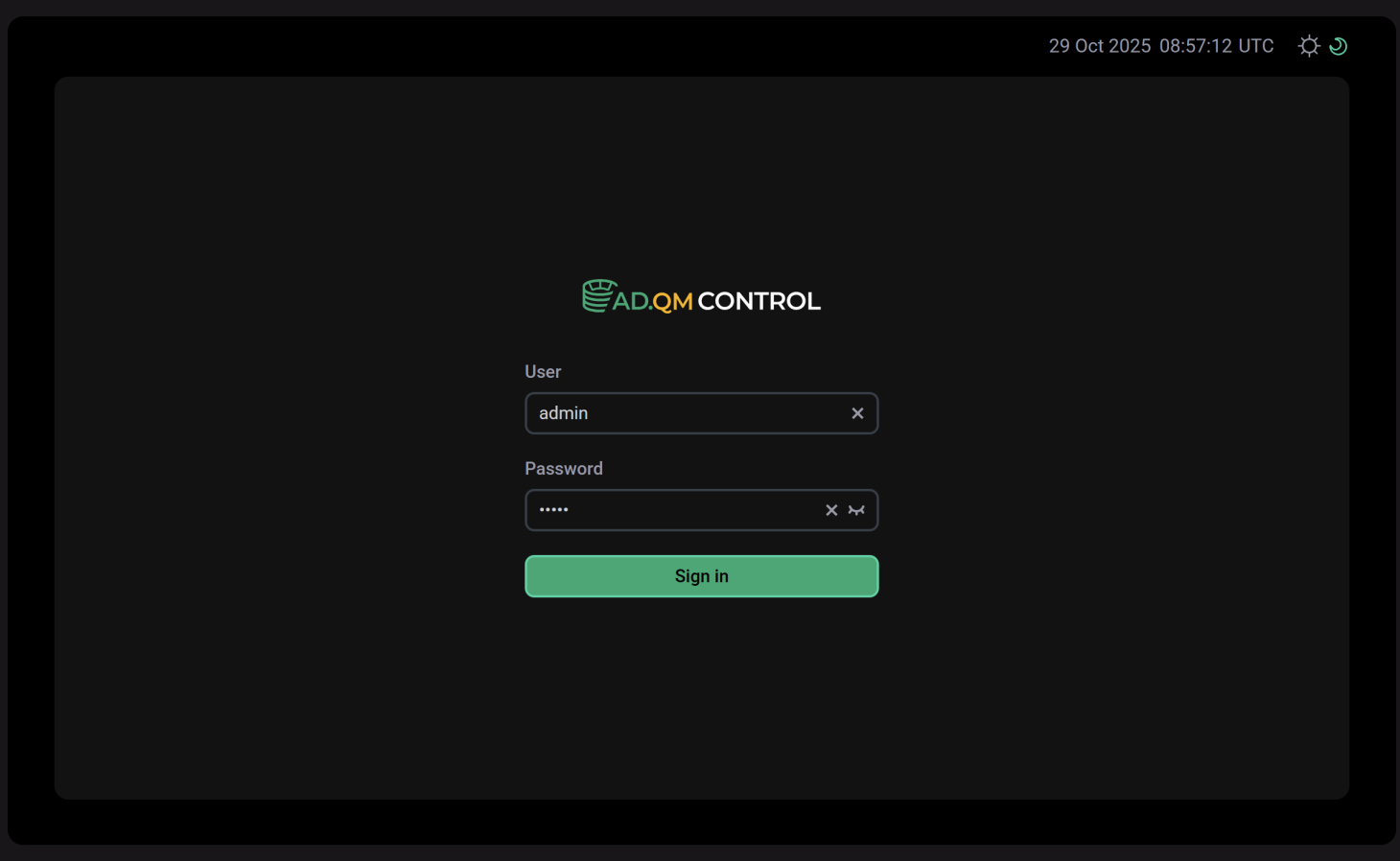
To Table of Contents

To connect to the ADQM Control web interface, follow the steps:

1. In the browser bar, enter the IP address of the host where the Backend ompone is installed. To specify the port number, use the value of the *Backend port* from the *Network configuration* section of the DQM Control service configuratio (`5555` by default).

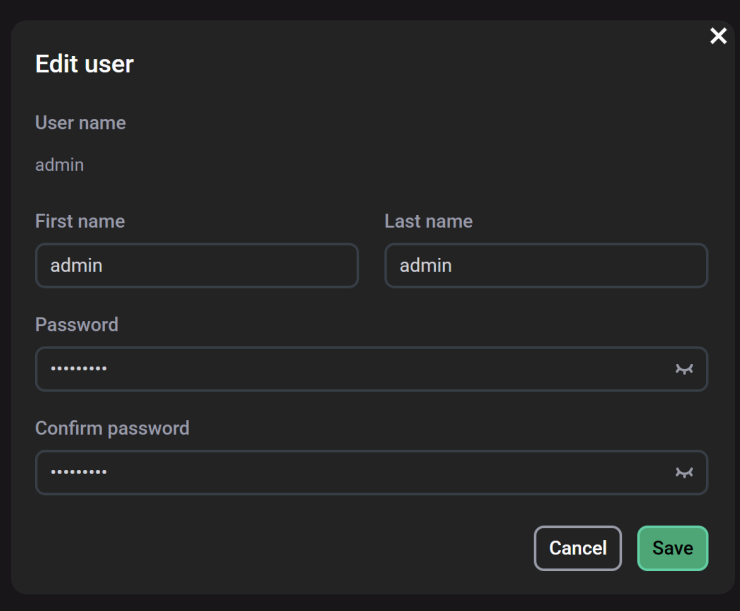
2. To log in for the first time after installation, use the default user credentials:

   - *User:* admin
   - *Password:* admin



Log in to ADQM Control

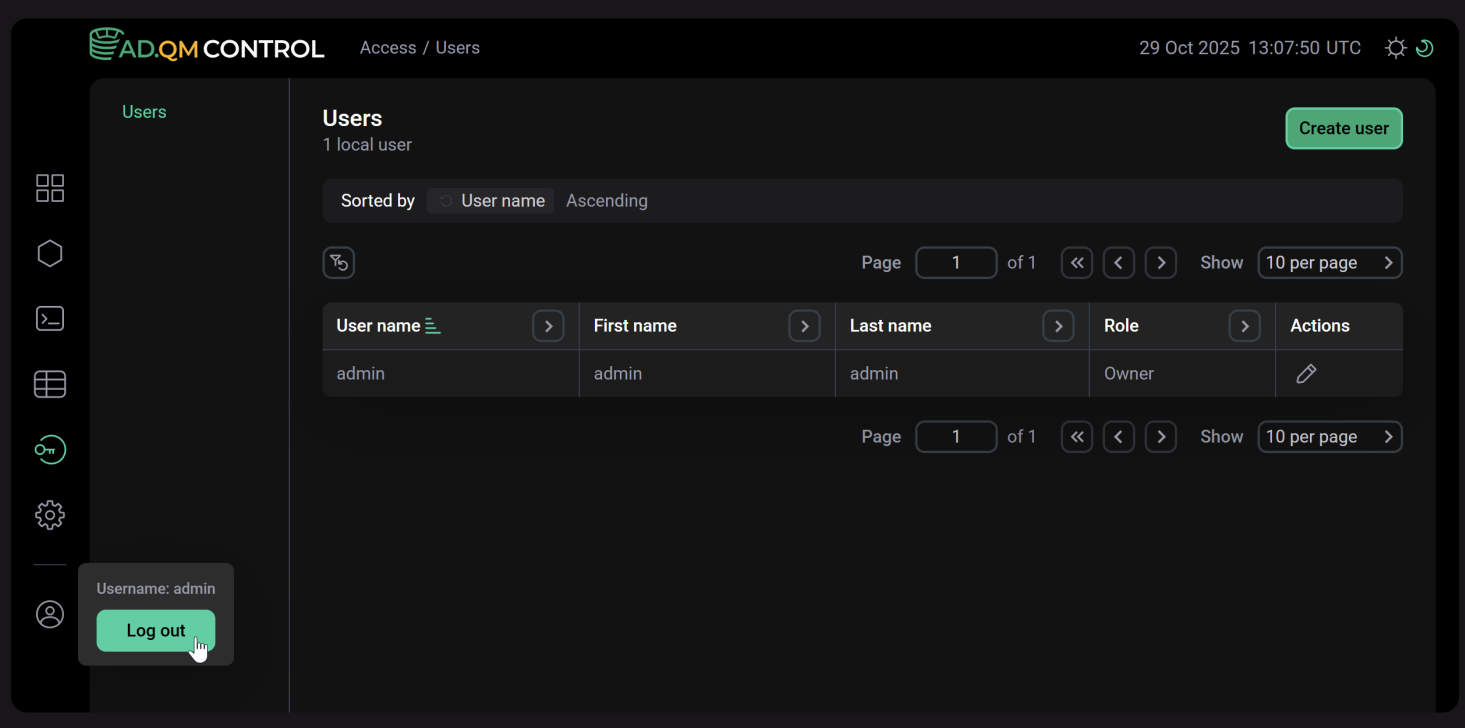
3. It is recommended to assign a new password to the `admin` user instead of the default one. To do this, go to the *Access* → *Users* page, open the dit windo for the `admin` user, set a new password in the *Password* and *Confirm password* fields, and click *Save*.



Change a password

On the *Access* → *Users* page, you can also add new ADQM control users — see the anage user article for details.

4. To log out from ADQM Control, hover the mouse over the bottom icon in the left navigation menu and click *Log out* in the pop-up window (it also displays the current user name).



Log out from ADQM Control

Manage users

Elena Dvoryadkina

Contents

To Table of Contents

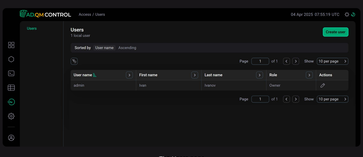
View a list of users
Add a user
Edit a user
Set password requirements

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

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

View a list of users

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



The Users page

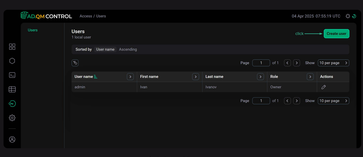
In column headers of the table with a list of users, there are filters that you can use to select specific data. To open a filter, click the › icon.

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

Add a user

To add a new user, follow the steps:

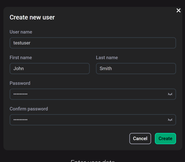
1. On the ***Access*** → ***Users*** page, click ***Create user***.



Switch to adding a user

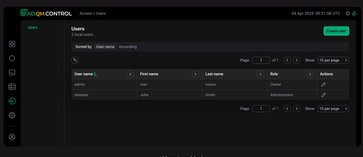
2. In the window that opens, fill in the following fields:
   - ***User name*** – user's login;
   - ***First name*** – user's name;
   - ***Last name*** – user's last name;
   - ***Password*** – user's password.

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



Enter user data

3. Click ***Create***. As a result, a new user (with the `Administrator` role – system administrator) is displayed on the ***Access*** → ***Users*** page.

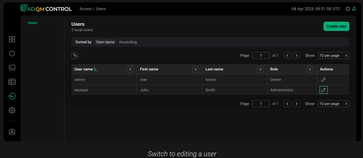


User is added

Edit a user

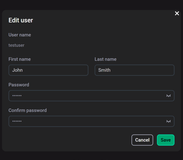
To edit user data, follow the steps:

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.



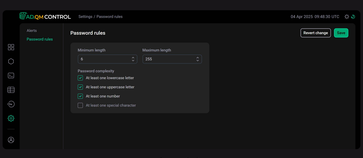
Edit user data

> **NOTE**
>
> Administrators can change only their own passwords. Owners can change passwords for any users.

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

SSL encryption

Contents

Create SSL certificates
Configure ADAM Control to enable SSL encryption
Manage SSL cluster action
SSL parameters in service configurations
Example
Test a connection
Disable SSL

Create SSL certificates

To create and register SSL certificates with the command-line tool, use the `openssl` utility (minimum version 1.1.1).

Go to a path on the computer where you want to create the certificates and run the following commands:

```
$ openssl req -newkey rsa:2048 -nodes -keyout
  adamcontrol.key -x509 -days 365 -out adamcontrol.crt
  -subj "/CN=adamcontrol" -addext "subjectAltName = DNS:adamcontrol,
  DNS:adamcontrol.local, IP:127.0.0.1, IP:::1"
$ openssl rsa -in adamcontrol.key -out adamcontrol.key
$ openssl x509 -in adamcontrol.crt -outform der -out adamcontrol.cer
$ openssl pkcs12 -export -out adamcontrol.pfx -inkey adamcontrol.key
  -in adamcontrol.crt -certfile adamcontrol.crt
```

The first command creates a self-signed certificate and a new private key. The second command removes the passphrase from the key. The third command converts the certificate to DER format. The last command packages the certificate and key into a PFX file.

To run the first command, you will be prompted to enter the Distinguished Name (DN) fields. These fields are optional. You can leave them blank by entering a single dot (`.`) for each field.

```
$ openssl x509 -in adamcontrol.crt -outform der -out adamcontrol.cer
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:.
State or Province Name (full name) [Some-State]:.
Locality Name (eg, city) []:.
Organization Name (eg, company) [Internet Widgits Pty Ltd]:.
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:adamcontrol
Email Address []:.
```

The Common Name field must contain the machine name or fully qualified domain name (FQDN) of your host for which the certificate is being created. Otherwise, the certificate will not be trusted.

The last three commands are used to configure and run the SCRPC service.

```
$ openssl_CERT_FILE=.\
$ openssl_KEY_FILE=.\adamcontrol.key
$ openssl_CA_FILE=.\adamcontrol.crt
$ .\adamcontrol.exe -c screpc.screpc.ini -f screpc -p screpc.pid.ini -l
  screpc.log
$ openssl_CERT_FILE=.\
$ openssl_KEY_FILE=.\adamcontrol.key
$ openssl_CA_FILE=.\adamcontrol.crt
$ .\adamcontrol.exe -c screpc.screpc.ini -f screpc -p screpc.pid.ini -l
  screpc.log
```

- The first command configures the OpenSSL environment variable `openssl_CERT_FILE` to point to the certificate file.
- The second command configures the OpenSSL environment variable `openssl_KEY_FILE` to point to the key file.
- The third command configures the OpenSSL environment variable `openssl_CA_FILE` to point to the certificate file.
- The fourth command runs the SCRPC service.

For the Apache HTTP Server, SSH connections, and Windows Remote Desktop Protocol (RDP) connections, see the corresponding articles in the Certificates: SSL Private Key: SSL certificate articles. For more information about the SCRPC service, see the Configuration: SCRPC service article. For more information about OpenSSL, see the OpenSSL Documentation.

Configure ADAM Control to enable SSL encryption

You can enable SSL encryption of connections to ADAM Control with the `SSL` parameter.

- In the [Configuration: Control service](#) article, the `SSL` enable keyword of the `Control` interface, and enabling the following SSL parameters:
- In the [Configuration: Configuration service](#) article, the `SSL` enable keyword of the `Configuration` interface, and enabling the following SSL parameters:

You can configure SSL encryption of connections to ADAM Control in two ways:

- [Manage SSL cluster action](#) – Configure SSL encryption of incoming and outgoing connections for all the services of ADAM Control.
- [SSL parameters in service configurations](#) – Configure SSL encryption of incoming and outgoing connections for selected services of ADAM Control.

Manage SSL cluster action

1. On the **Cluster** page, click the icon ⚙ of the ADAM Control cluster to open the list of [Cluster actions](#) and select **Manage SSL**.

Launch the "Manage SSL" action

ADAM will display a dialog box window. With the option set to the **Enable** state, click the **Continue** button, observing that by default, without any changes, all ADAM Control services should use secure connections.

Options for enabling SSL in the "Manage SSL" cluster configuration window

2. Fill in the required parameters in the **Manage SSL** action configuration window:

- The parameter for the **Enable** state determines whether ADAM Control will use encrypted connections for various services of ADAM Control.

Parameters to enable SSL for services

Parameter values specified in the **Manage SSL** action window override the values of the corresponding parameters in [service configurations](#) after the action executes in the **Service parameter** column of the table below.

Service SSL parameter	Possible description	Service parameter
ADPTC — Secure ADPTC	Enable SSL encryption of connections to the ADPTC service	**ADPTC** service: **Enable SSL** parameter
ADAM Control — Secure Backend	Enable SSL encryption of connections to the Backend service of ADAM Control	**ADAM Control** service: **[ADAM Control Backend] - Enable SSL** parameter
ADAM Control — Secure ADAM Agent	Enable SSL encryption of connections to the ADAM Agent service of ADAM Control	**ADAM Control** service: **[ADAM Control ADAM Agent] - Enable SSL** parameter
ADAM Control — Secure Alert Generator	Enable SSL encryption of connections to the Alert Generator component of the ADAM Control service	**ADAM Control** service: **[ADAM Control Alert Generator] - Enable SSL** parameter
ADAM Control — Secure Alert Generator	Enable SSL encryption of connections to the Alert Generator component of the ADAM Control service	**ADAM Control** service: **[ADAM Control Alert Generator] - Enable SSL** parameter
ADAM Control — Secure Alert Generator	Enable SSL encryption of connections to the ADAM Agent component of the ADAM Control service	**ADAM Control** service: **[ADAM Control Alert Generator] - Enable SSL** parameter
Monitoring — Secure Alert Generator	Enable SSL encryption of connections to the Alert Generator component of the Monitoring service	**Monitoring** service: **[Alert Generator] - Enable SSL** parameter
Monitoring — Secure Collector	Enable SSL encryption of connections to the Collector component of the Monitoring service	**Monitoring** service: **[Collector] - Enable SSL** parameter
Monitoring — Secure Data Acquirer	Enable SSL encryption of connections to the Data Acquirer component of the Monitoring service	**Monitoring** service: **[Data Acquirer] - Enable SSL** parameter

3. Set the **SSL service identity** parameter to specify whether to check the server host and parameter, after applying the settings.

4. To launch the action, click **Next** and then **Run** in the confirmation window.

Exiting the **Manage SSL** action executes ADAM actions and any tools, reconfigures them to enable or disable SSL encryption, then starts the services again and checks them. Monitor the progress of the action in the **Actions** window. Click the **Log** button after the action executes to view the full sequence of actions, their status, and resulting verbose AtroCore messages.

SSL parameters in service configurations

An alternative way to set up an encrypted connection to ADAM Control is to configure SSL settings for each service separately.

- In the [Configuration: Control service](#) article, the `SSL` enable keyword of the `Control` interface, and enabling the following SSL parameters:
- In the [Configuration: Configuration service](#) article, the `SSL` enable keyword of the `Configuration` interface, and enabling the following SSL parameters:

ADPTC SSL configuration

In the **SSL Configuration** section of the ADPTC service configuration, the following parameters are available:

Parameter	Description	Default value
Enable SSL	Enable SSL encryption of incoming connections to the ADPTC service	Disabled
Certificate File	Path to the server SSL certificate file in the DER format (*.cer) that the ADPTC service should use to identify itself to clients. The file should be located in the specified path on each host of the cluster where these components are installed.	ssl.adptc.cer
Private Key File	Path to the server SSL private key file (*.key) that the ADPTC service should use to identify itself to clients. The file should be located in the specified path on each host of the cluster where these components are installed.	ssl.adptc.key
Certificate Authority File	Path to the trusted certificate authority file (*.cer) that the ADPTC service should use to verify client certificates. The file should be located in the specified path on each host of the cluster where these components are installed.	ssl.adptc.ca.cer

ADPTC service configuration parameters to check SSL certificates of connections

ADAM Control SSL configuration

In the **SSL Configuration** section of the ADAM Control service configuration, the following parameters are available:

Parameter	Description	Default value
[ADAM Control Backend] – Enable SSL	Enable SSL encryption of incoming connections for the Backend component of ADAM Control	Disabled
[ADAM Control ADAM Agent] – Enable SSL	Enable SSL encryption of incoming connections for the Agents component of ADAM Control	Disabled
[ADAM Control Alert Generator] – Enable SSL	Enable SSL encryption of incoming connections for the Alert Generator component of ADAM Control	Disabled
[ADAM Control Alert Generator] – Enable SSL	Enable SSL encryption of incoming connections for the Alert Generator component of ADAM Control	Disabled
[ADAM Control Alert Generator] – Enable SSL	Enable SSL encryption of incoming connections for the ADAM Agent component of ADAM Control	Disabled
[ADAM Control Backend] – Certificate File	Path to the server SSL certificate file for the SSL Backend Agent component (*.cer) that the ADAM Control service should use to identify itself to clients. The file should be located in the specified path on each host of the cluster where these components are installed.	ssl.adamcontrol.backend.cer
[ADAM Control Backend] – Private Key File	Path to the server SSL private key file for the SSL Backend Agent component (*.key) that the ADAM Control service should use to identify itself to clients. The file should be located in the specified path on each host of the cluster where these components are installed.	ssl.adamcontrol.backend.key
[ADAM Control Backend] – Certificate Authority File	Path to the trusted certificate authority file for the SSL Backend Agent component (*.cer) that the ADAM Control service should use to verify client certificates. The file should be located in the specified path on each host of the cluster where these components are installed.	ssl.adamcontrol.backend.ca.cer

Configuration parameters of the ADAM Control service to check SSL certificates of connections

ADAM Control: ADAM management

The **Connection type** parameter in the **ADAM management** section of the ADAM Control configuration sets the type of SSL security for TCP connections to ADAM Agents. The default value is **Standard**. **Point-to-point** connections to ADAM Agents, or **Secure** and **Secure (TCP standard)** connections to ADAM Agents should be set to **Secure (TCP standard)**.

Enable the use of only secure TCP connections to ADAM

Monitoring: SSL configuration

The following parameters in the **SSL configuration** section of the Monitoring service configuration allow you to configure SSL encryption of connections to the Monitoring service. The default value of the **Enable SSL** parameter is **Disabled**. Set it to **Enabled** to enable SSL encryption.

Parameter	Description	Default value
[Frontend] — Enable SSL	Enable SSL certificate files in the DER format (*.cer) and private key files (*.key)	Disabled
[Frontend] — Certificate File	Path to the server SSL certificate file in the DER format (*.cer) that the Monitoring service should use to identify itself to clients	ssl.monitoringservice.server.cer
[Frontend] — Private Key File	Path to the SSL private key file (*.key) that the Monitoring service should use to identify itself to clients	ssl.monitoringservice.server.key
[Frontend] — Certificate Authority File	Path to the SSL certificate authority file (*.cer) that the Monitoring service should use to verify client certificates	ssl.monitoringservice.ca.cer
[Gateway] — Enable SSL	Enable SSL certificate files in the DER format (*.cer) and private key files (*.key)	Disabled
[Gateway] — Certificate File	Path to the server SSL certificate file in the DER format (*.cer) that the Monitoring service should use to identify itself to clients	ssl.monitoringgateway.server.cer
[Gateway] — Private Key File	Path to the SSL private key file (*.key) that the Monitoring service should use to identify itself to clients	ssl.monitoringgateway.server.key
[Gateway] — Certificate Authority File	Path to the SSL certificate authority file (*.cer) that the Monitoring service should use to verify client certificates	ssl.monitoringgateway.ca.cer
[Node exporter] — Enable SSL	Enable SSL certificate files in the DER format (*.cer) and private key files (*.key)	Disabled
[Node exporter] — Certificate File	Path to the server SSL certificate file in the DER format (*.cer) that the Monitoring service should use to identify itself to clients	ssl.nodeexp.server.cer
[Node exporter] — Private Key File	Path to the SSL private key file (*.key) that the Monitoring service should use to identify itself to clients	ssl.nodeexp.server.key

Configuration parameters of the Monitoring services for SSL connections

3. After selecting all the necessary parameters for configuring SSL connections, click **Save** and then **Reconfigure and restart** for services to apply the changes.

Example

In the **ADAM management** section of the ADAM Control configuration, set the **Connection type** parameter to **Secure (TCP standard)** so that ADAM Control can connect to ADAM only through the TCP port that has the **ADAM secure port** value.

Options to enable the use of only secure TCP connections to ADAM

Click **Save**.

3. Click **Manage SSL** action for the cluster. In the configuration window that appears, activate **Enable SSL** and also turn on the following options to encrypt traffic between ADAM Control components:

- **[ADPTC] – Secure ADPTC**
- **[ADAM Control] – Secure Backend**
- **[ADAM Control] – Secure ADAM Agent**
- **[ADAM Control] – Secure Alert Generator**
- **[ADAM Control] – Secure Alert Generator**
- **[ADAM Control] – Secure Alert Generator**

Enable SSL and ADAM Control via the "Manage SSL" cluster action

Test a connection

To test, run the command `openssl s_client -connect host:port` (for example, `openssl s_client -connect 10.10.10.10:5029`) in the command line to the IP address or the host name for the desired component's connection. Note that you cannot use secure ADAM Control via HTTP.

Disable SSL

To disable SSL, click the **Manage SSL** action and click the **Disable SSL** switch, leaving it turned off. Click **Continue**. Wait for the configuration of all services.

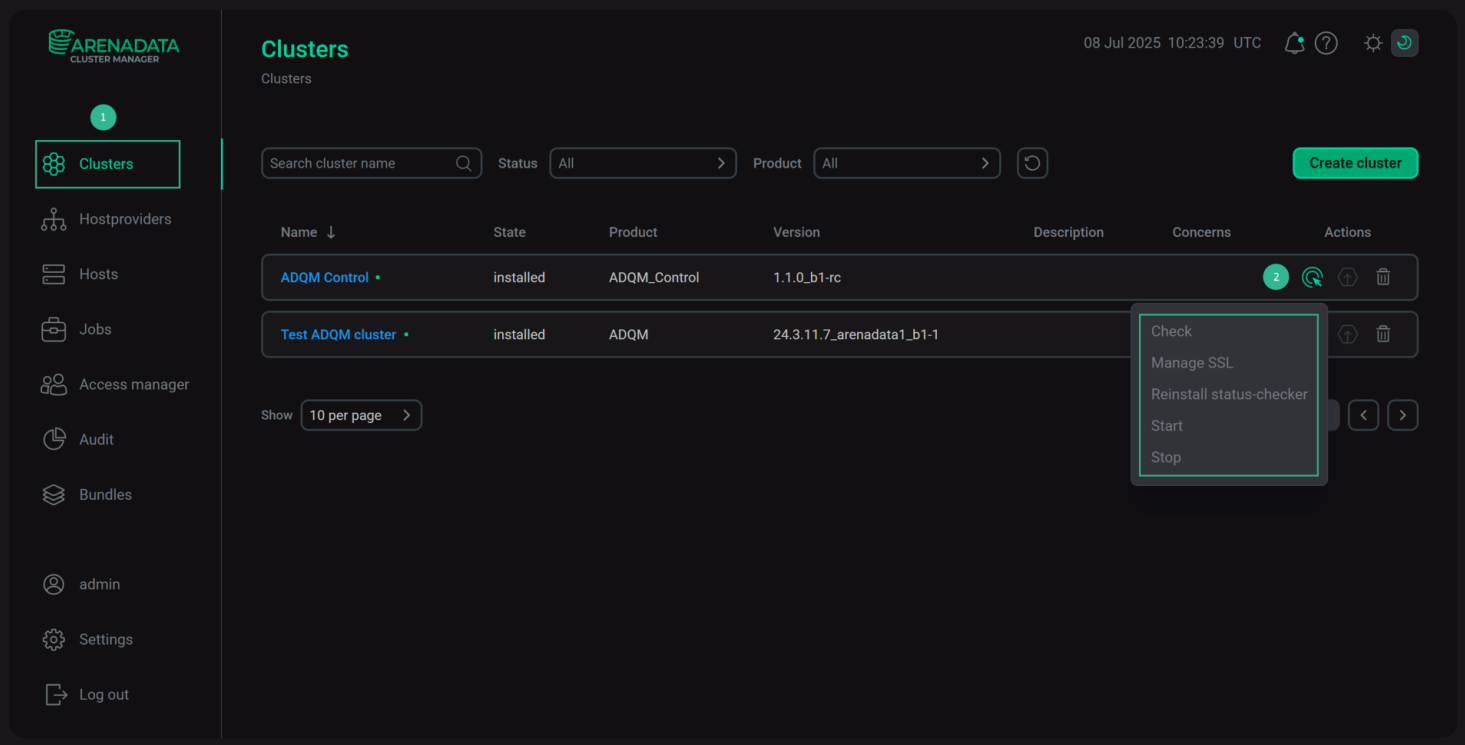
Disable SSL with the "Manage SSL" action

Cluster actions

Elena Dvoryadkina

To Table of Contents

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



ADQM Control cluster in the ADCM UI

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

A set of actions available for an ADQM Control cluster depends on the cluster state:

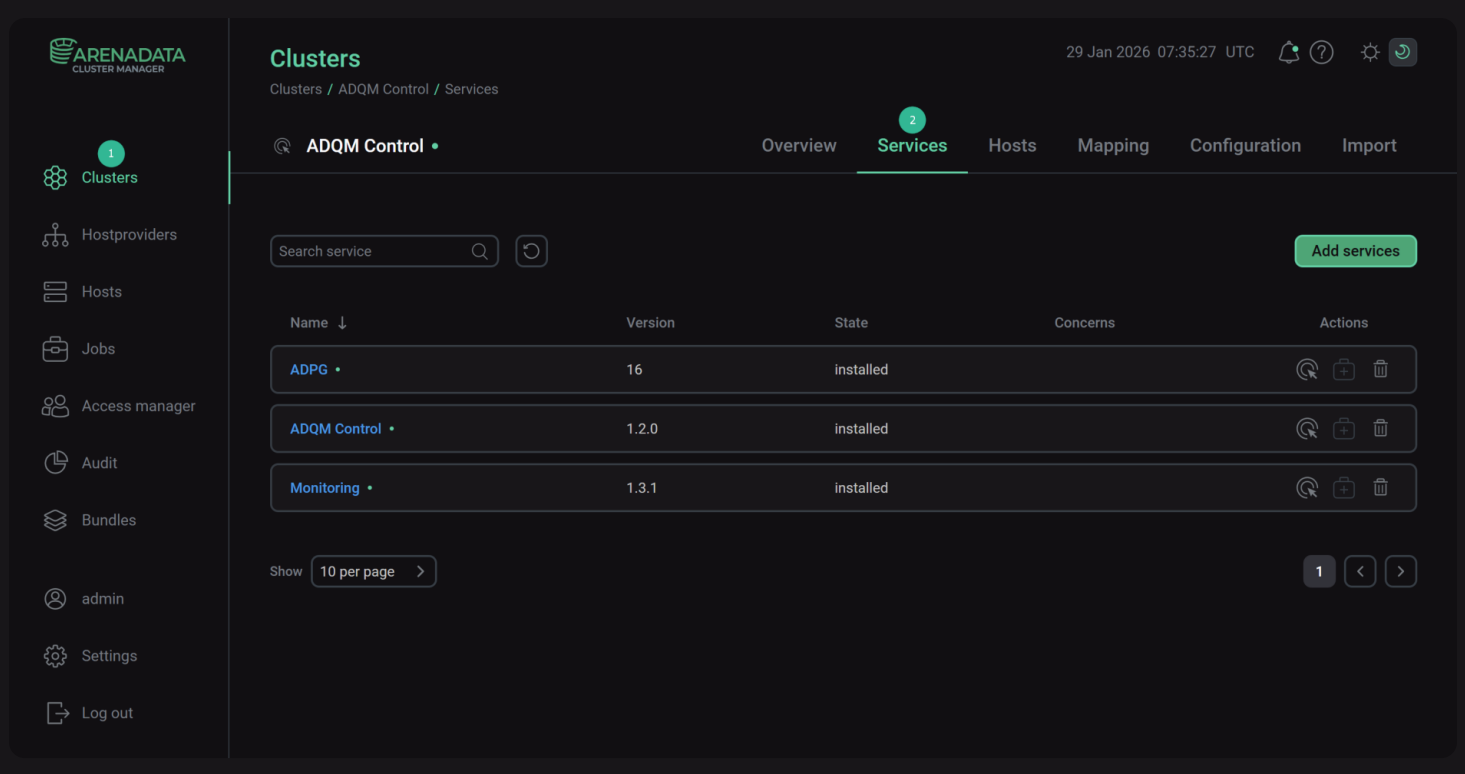
- For a new cluster (the cluster state is `created`), an available action is:
  - ***Install*** — installs all cluster services.
- After a cluster is installed (the cluster state is `installed`), the following actions are available:
  - ***Check*** — checks if all components of the installed cluster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

Service actions

Elena Dvoryadkina

To Table of Contents

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



Services of an ADQM Control cluster in the ADCM interface

Refer to ADCM documentation for an overview of the ***Services*** tab and common actions. A set of actions available for working with a service of an ADQM Control cluster 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:

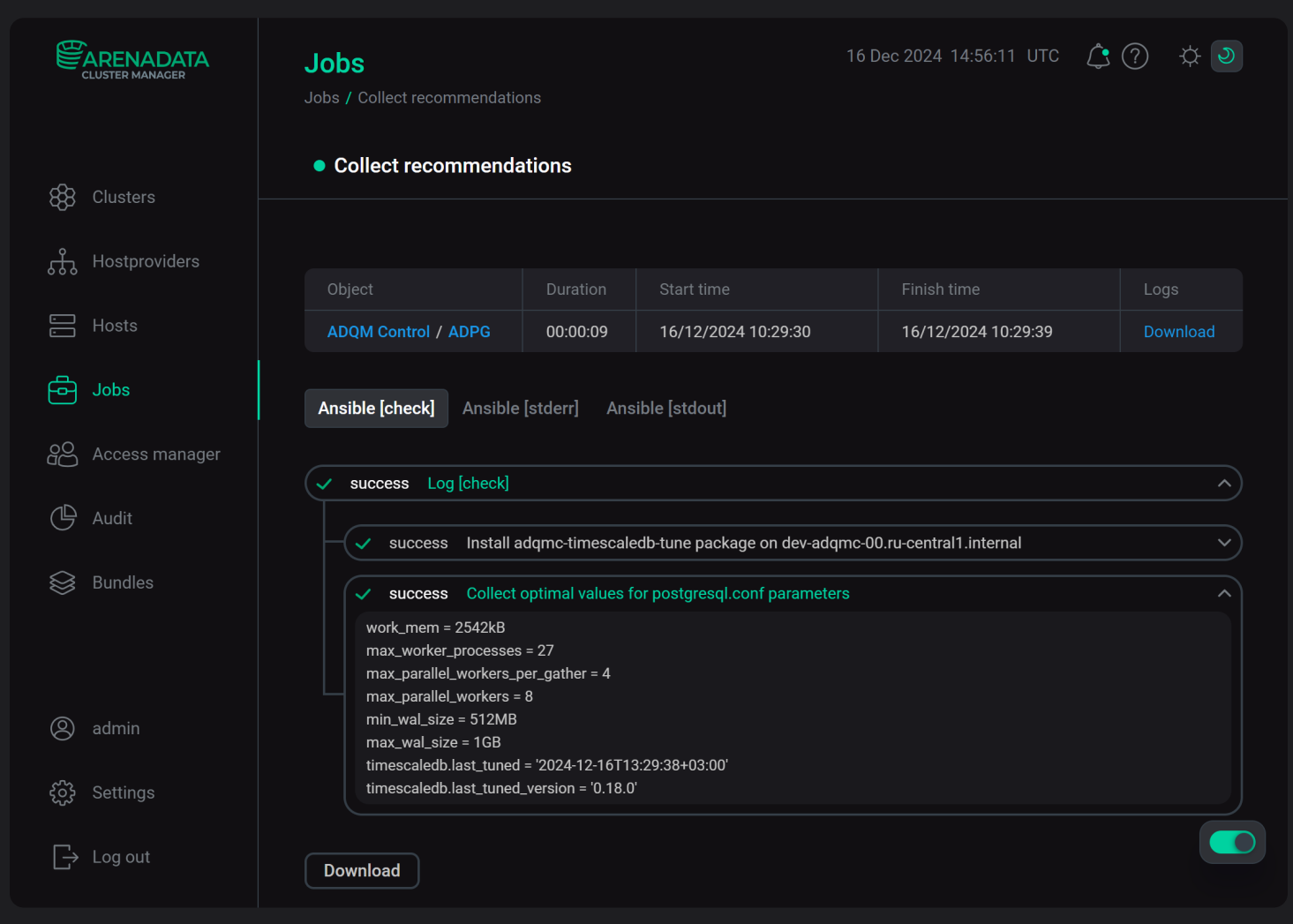
ADQM Control

- ***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).

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.

Configure an external database

Elena Dvoryadkina

To Table of Contents

For production environments, it is recommended to use an external PostgreSQL database as ADQM Control data storage for high performance and fault tolerance.

TIP

The recommended external service for ADQM Control is Arenadata Postgres (ADPG).

To configure an external database, follow the steps described below:

1. In PostgreSQL, run the following queries to create a user for ADQM Control to connect to an external database and a database owned by that user for use by ADQM Control services (the queries should be run under the default `postgres` user with `superuser` privileges):

```
CREATE ROLE <user_name> LOGIN PASSWORD '<user_password>';
CREATE DATABASE <database_name> OWNER <user_name>;
```

2. Connect to the created database under a superuser and load the following extensions:

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

NOTE

Recommended extension versions:

- `timescaledb` — 2.15.0
- `timescaledb_toolkit` — 1.19.0
- `pgcrypto` — 1.3
- `pg_trgm` — 1.6

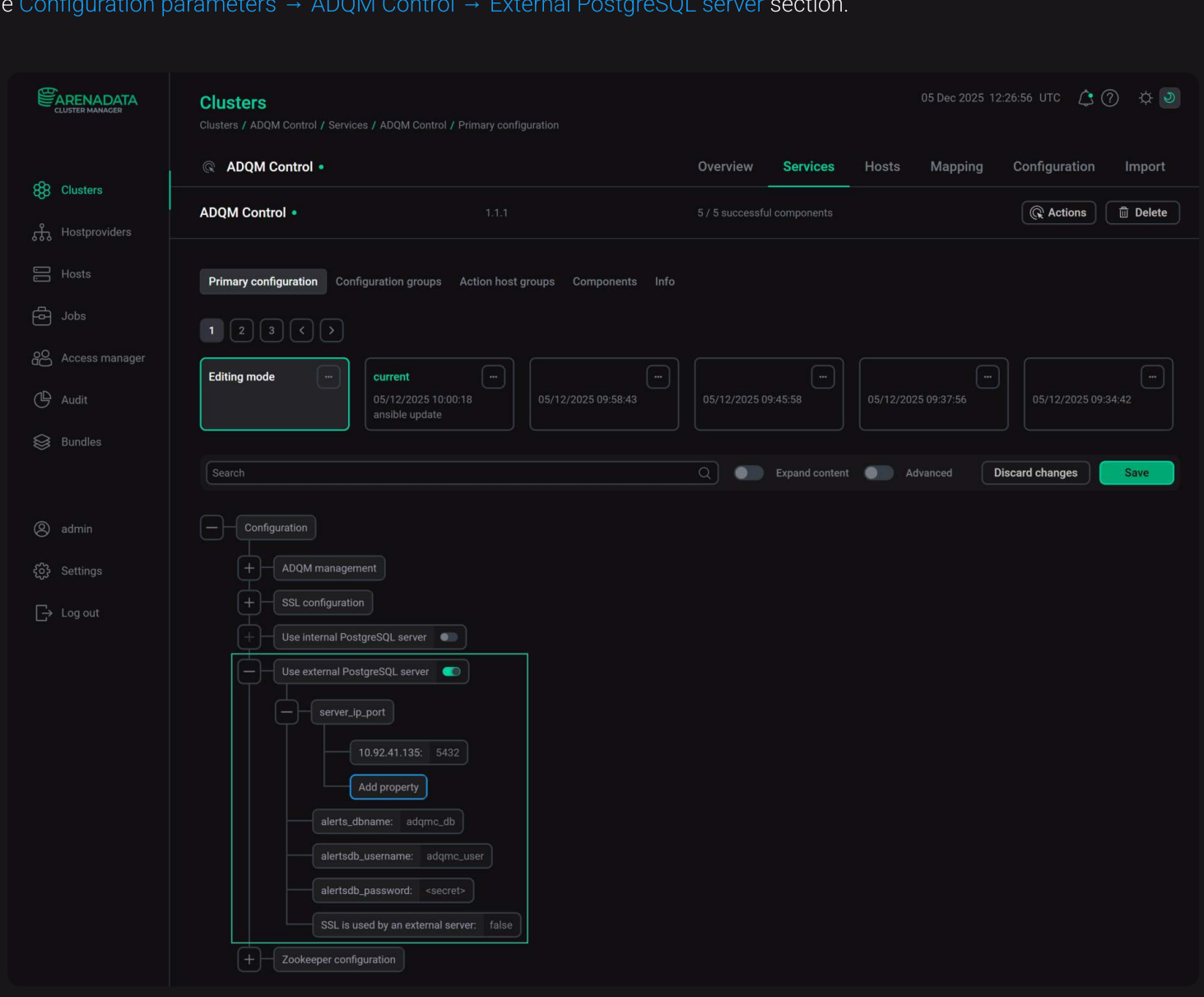
When using higher versions, compatibility testing is required to ensure correct system operation.

3. To enable access to the `<database_name>` database from an ADQM Control host as the `<user_name>` user, add an entry like the following to the *pg_hba.conf* file on the PostgreSQL server side:

```
host <database_name> <user_name> <adqmc_host_address> trust
```

where `<adqmc_host_address>` is an address of the ADQM Control host (for example, if an IPv4 address of the ADQM Control host is `10.92.41.87`, replace `<adqmc_host_address>` with `10.92.41.87/32`). You can find description of the `address` field and other fields of records in the *pg_hba.conf* file in the article The pg_hba.conf File.

4. On the configuration page of the ADQM Control service, activate the ***Use external PostgreSQL server*** toggle and fill in parameters in the list that opens. The assignment of parameters is described in the Configuration parameters → ADQM Control → External PostgreSQL server section.



ADQM Control parameters for connecting to an external PostgreSQL database

5. Click ***Save*** to save the ADQM Control configuration.

6. Install the ADQM Control service using the Install service action or execute the ***Reconfigure and restart*** action for the ADQM Control service if it is already installed.

Configure alerts in ADQM Control via web UI

Elena Dvoryadkina

Contents

To Table of Contents

System alerts
- Configure system alerts
- Types of system alerts

Internal alerts

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

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

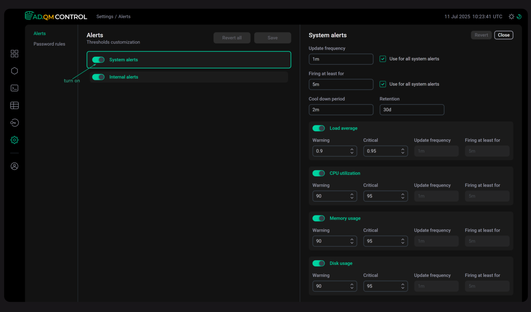
Alerts in ADQM Control are grouped into modules. Currently, ADQM Control can send and allows you to configure alerts of the System alerts and Internal alerts modules.

System alerts

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

Configure system alerts

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



Configure system alerts

System alert settings

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

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

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

Types of system alerts

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

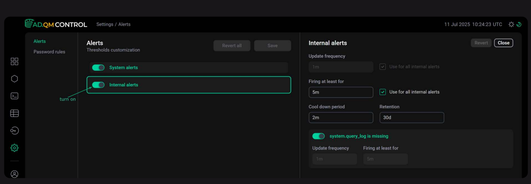
System alerts

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

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

Internal alerts

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



Configure internal alerts

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

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

View a dashboard

A. Vera Kolumbina

ADQM CONTROL

Contents

To Table of Contents

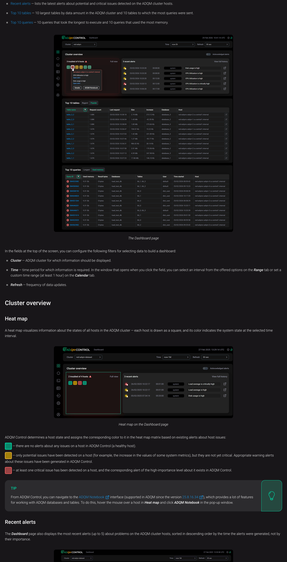
Cluster overview
- Heat map
- Recent alerts
- Acknowledged alerts
- Open alert details
- Go to the Cluster metrics page

Top 10 tables

Top 10 queries

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

- Heat map — graphically represents the statess 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

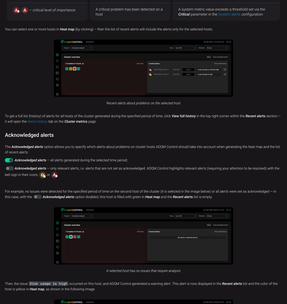
In the fields at the top of the screen, you can configure the following filters for selecting data to build a dashboard:

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

Cluster overview

Heat map

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



Heat map on the Dashboard page

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

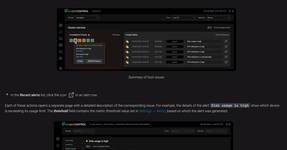
- there are no alerts about any issues on a host in ADQM Control (a healthy host);
- only potential issues have been detected on a host (for example, the increase in the values of some system metrics), but they are not yet critical. Appropriate warning alerts about these issues have been generated in 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.

TIP

From ADQM Control, you can navigate to the ADQM Notebook interface (supported in ADQM since the version 1.5.1 LTS), which provides a lot of features for working with ADQM databases and tables. To do this, hover the mouse over a host in *Heat map* and click *ADQM Notebook* in the pop-up window.

Recent alerts

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

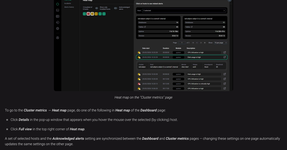


Recent alerts on the Dashboard page

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.alert 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.alert configuration

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



Recent alerts about problems on the selected host

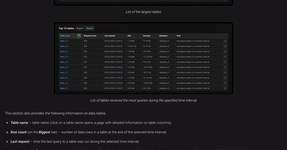
To get a full list (history) of alerts for all hosts of the cluster generated during the specified period of time, click **View full history** in the top-right corner within the *Recent alerts* section – it will open the Alerts page with the *Cluster metrics* page.

Acknowledged alerts

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

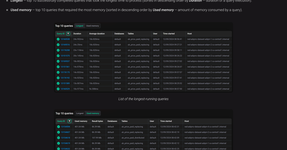
- **Acknowledged alerts** – all alerts generated during the selected time period.
- **Acknowledged alerts** (option disabled, this host is filled with green in *Heat map* and the *Recent alerts* list is empty).

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



A selected host has no issues that require analysis

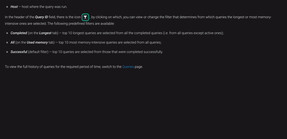
Then, the major **Disk usage is high** occurred on this host, and ADQM Control generated a warning alert. This alert is now displayed in the *Recent alerts* list and the color of the host is yellow in *Heat map* – a warning about potential issues.



Example of an unresolved issue on the host

If the disk usage issue has been analyzed and resolved, it no longer requires attention and you can edit the corresponding alert to the list of acknowledged alerts (on the page with alert settings). Then, the corresponding alert will not take into account the alert when monitoring the cluster health – it will display the host in green in *Heat map* again and exclude the alert from the *Recent alerts* list.

If the **Acknowledged alerts** option is disabled, the *Recent alerts* list shows the latest alerts among all alerts (active and "closed") generated during the specified time period, and all alerts affect the host map. For example, in the image below, all cluster hosts in *Heat map* are red — this means that at least one critical issue was detected on each of these hosts during the selected time period (the last month in this example). The *Recent alerts* list shows that there are no relevant alerts among the most recent alerts about problems on the selected host (seen1 icons are without the bell sign).



The Acknowledged alerts option is enabled

Open alert details

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

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

Summary of host issues

- In the *Recent alerts* list, click the icon of an alert case.

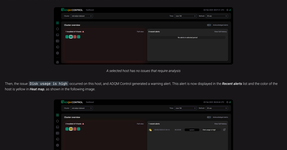
Each of these actions opens a separate page with a detailed description of the corresponding issue. For example, the details of the alert **Disk usage is high** show which device is exceeding its usage limit. The **threshold** field contains the metric threshold value set in Settings → Alerts, based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

Host issue details

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

Go to the Cluster metrics page

From *Heat map* on the *Dashboard* page you can go to the Heat map tab on the *Cluster metrics* page. It also shows the heat map matrix for an ADQM cluster and additionally displays general information about ClickHouse servers as well as a table with a list of all metrics (and not just the latest ones) related to the selected hosts on the right. You can get here more information about a specific alert by clicking the corresponding row in the table – alert details will appear below the row.



Heat map on the "Cluster metrics" page

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

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

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

Top 10 tables

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

- *Biggest* – 10 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).

List of the biggest tables

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* (set the *Biggest* tab) — number of 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* (set 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 (the icon opens the ADQM Notebook UI interface or the built-in ClickHouse web interface for this host, depending on the ADQM version – ADQM Notebook is supported starting with CTR 1.5.1 LTS);

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

- *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 information about them at 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 *Duration* – 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).

List of the longest running queries

List of queries used the most memory

For each query, the following information is also provided:

- *Query ID* – the query identifier (click on an identifier opens the *Query details* page where you can view the query text);
- *Average duration* (set the *Longest* tab) — average query execution time (statistics are calculated for queries whose text completely matches, or for parameterized queries, specific parameter values are taken into account);
- *Result bytes* (set 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 parameters are available:

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

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

Cluster metrics

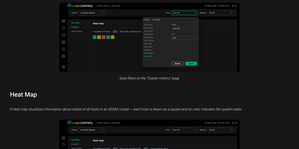
Contents

- Heat Map
  - Host state
  - Alerts related to a host
- Incidents
  - How incidents are generated
  - Settings
- Alerts history
  - Alert filters
  - Alert details

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

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

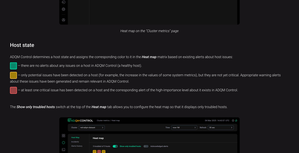
- **Cluster** – ADQM cluster for which data will be displayed.
- **Time** – time period for which information is required. In the window that opens when you click the field, you can select an interval from the offered options on the **Range** tab or set a custom time range on the **Calendar** tab.
-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



Data filters on the "Cluster metrics" page

Heat Map

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



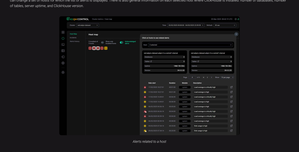
Heat map on the "Cluster metrics" page

Host state

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

- there are no alerts about any issues on a host in ADQM cluster (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 on these issues have been generated and remain relevant in ADQM Control;
- at least one critical issue has been detected on a host. Appropriate alerts with a critical level are created and still exist in ADQM Control.

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

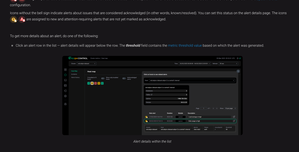


Heat map shows only troubled hosts

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

Alerts related to a host

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



Alerts related to a host

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

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

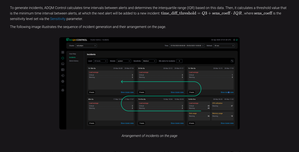
The first column of the table with a list of alerts contains an icon that indicates the severity level of an alert:

- – warning alert that means that a potential issue with an ADQM cluster that the issue has not reached the critical level yet). Generated when a system metric value exceeds a threshold set by the **Warning** parameter in the System alerts configuration.
- – high importance alert that means a critical issue on a host. Generated when a system metric value exceeds a threshold set as the **Critical** parameter in the System alerts configuration.

Some columns of the last-sign indicate alerts about issues that are considered acknowledged (in other words, known/resolved). You can set this status on the alert details page. The icons in the are assigned to new and attention-requiring alerts that are not yet marked as acknowledged.

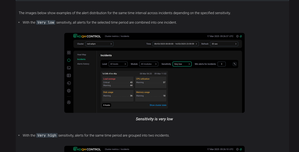
To get more details about an alert, do one of the following:

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



Alert details within the list

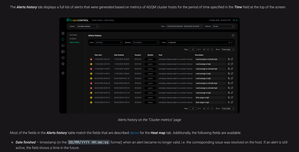
- Click the icon – within an alert row – this will take you to a separate page with alert details (the page has a unique address that you can share). The **Acknowledged alerts** option allows you to change the alert status – mark the corresponding issue as resolved or that it does not affect the cluster host map.



Alert details on a separate page

Incidents

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



Incidents on the "Cluster metrics" page

An incident is displayed as a block that includes:

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

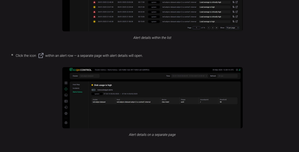
- **Show cluster view** link – refers to the Dashboard page with the Acknowledged alerts option enabled and the **Time** period equal to the incident interval. If an incident interval is less than an hour, ADQM Control automatically increases it to 1 hour by adding equal parts to each boundary (since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

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

How incidents are generated

To generate incidents, ADQM Control calculates time intervals between alerts and determines the interquartile range (IQR) based on this data. Then, it calculates a threshold value that is used to determine whether a new alert is related to the incident or not. The threshold is the next alert will be added to a new incident: **Start_diff_threshold = Q3 + max_coeff * IQR**, where **max_coeff** is the coefficient that can be set in the Settings section.

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



Arrangement of incidents on the page

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

- For a fixed time interval in the past, the structure of incidents is generated when the **Incidents** page is opened and remains fixed because the set of alerts for this time interval does not change.
- If you select a time interval relative to the current moment (for example, the interval **Last 12h**, **Last 1d**, **Last 1h**, etc.), the structure of incidents on the page may change, since the end boundary of the time interval shift over time, which means the set of active alerts may change (new alerts may occur or the time of previously-valid alerts may become outside the current time frame).

Settings

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

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

Example

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

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

Sensitivity is very low

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

Sensitivity is very high

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

Alerts history

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

Alerts history on the "Cluster metrics" page

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

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

The icons , , , indicate whether an issue requires attention or is set as acknowledged.

Alert filters

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

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

To reset all applied filters, click the icon (**Reset all filters**).

Alert details

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

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

Alert details within the list

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

Alert details on a separate page

Work with tables in ADQM Control

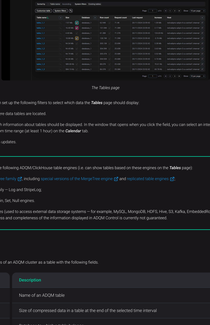
© Arenadata Software

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.

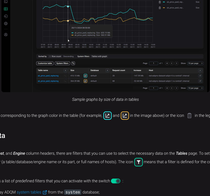
For integration table engines (used to access external data storage systems — for example, MySQL, MongoDB, HDFS, Hive, S3, Kafka, EmbeddedRocksDB, RabbitMQ, PostgreSQL), the correctness and completeness of the information displayed in ADQM Control is currently not guaranteed.

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

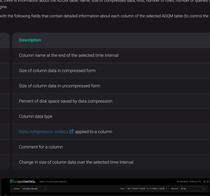
Additional fields with information about ADQM tables

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. Clicking on the icon opens ADQM Notebook at http://<adqm_host_name>:8123/notebook for ADQM starting with the version 21.8.13.24 or the built-in ClickHouse web interface at http://<adqm_host_name>:8123/play for earlier versions of ADQM, where <adqm_host_name> is the ADQM host FQDN (the system should be configured so that a can resolve the host FQDN into the IP address)
Engine	Table engine

Graphs

On the **Tables** page, you can get graphs to track the dynamics of changes in the **Size** and **Row count** metrics of ADQM tables (size of compressed data on disk and number of rows, respectively) over a selected period of time. Click the icon in the column of the corresponding metric – above the list of tables. ADQM Control will display a section with a graph built based on this metric of the ADQM table. You can add up to 5 graphs simultaneously.

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



Sample graphs by size of data in tables

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

Filter and sort data

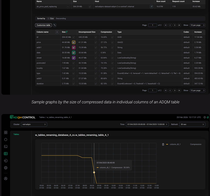
In the **Table name**, **Database**, **Host**, and **Engine** column headers, there are filters that you can use to select the necessary data on the **Tables** page. To set a filter, click the icon and enter a value or part thereof (or a table database/engine name or its part, or full names of hosts).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

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

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

It is also possible to sort data on the **Tables** page by values in any column of the table with the list of ADQM tables. To apply sorting by a column, click the icon , or in its header.

Above the list of tables, there is a panel that shows the sorting (**Sorted by**) and filtering (**System filters**, **Filtered by**) conditions currently applied to data.



Current conditions for sorting and filtering data on the Tables page

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

Table details

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

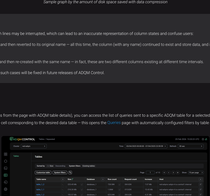
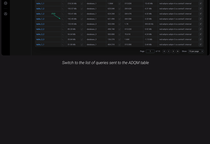


Table selection

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

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

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

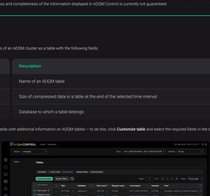


Information on table columns

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

On 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, or to analyze effect of applying data compression on the amount of disk space saved. To do this, use the icon in the **Size, Uncompressed Size** or **Compression** column, respectively. The graph area can display up to 7 graphs by individual columns and one graph by the entire table size at a time.

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



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

CAUTION

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

- a column was renamed and then reverted to its original name — at this time, the column (with any name) continued to exist and store data, and its metrics were collected;
- a column was dropped and then re-created with the same name – in 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

Work with queries in ADQM Control

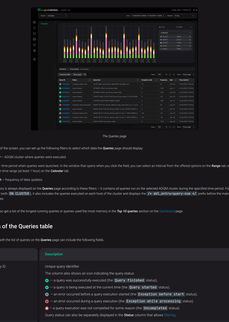
Anna Novikova

Contents

Fields of the Queries table
Sort and filter data
Query details
Graphs
- Duration
- User
- Kind

To Table of Contents

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 executed in your clusters and see various key performance metrics – for example, to analyze query performance, amounts of read/written data, resource usage, or how many queries occurred during some execution. Information on queries can also be presented in the form of graphs, which allow you to visualize the number of queries of different types, see how the total number of queries varies over time, and find the queries that are the most prominent in the period.



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** – the cluster whose queries you want to see.
- **Time** – time period when queries were launched. In the window that opens when you click the field, you can select an interval from the offered options on the **Range** tab or set a custom time range on the **Custom** tab.
- **Refresh** – frequency of data updates.

Only queries that were executed on the **Queries** page according to these filters – it contains all queries run on the selected ADQM cluster during the specified period. For each query, the **Query ID** (`2fc3a6f1-...`) and the **Query** text are displayed. Hover the pointer over the **Query ID** (`2fc3a6f1-...`) or `SELECT` to see further the next set of such queries.

You can also get a list of the longest-running queries or queries used the most metrics 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 contains the following fields:

Field	Description
Query ID	Unique query identifier. The column also shows an icon indicating the query status: - 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 during a query execution started on the `Execution before start` status). - An error occurred during a query execution (the `Execution write processing` status). - A query execution was not completed for some reason (the `Unconfirmed` status).
Query text	Query text. For a joining query, its text is truncated and ends as shown the inverse cluster over the column sort. A is displayed as `(query start) ILO (query end)`. You can see the full text of a query on a separate page or to open it, click the query in the **Query ID** column.
Status	Current status of a query. Possible values: - `Query started` – a query execution is in progress; - `Query finished` – a query was successfully executed; - `Execution before start` – an error occurred before a query execution; - `Execution write processing` – an error occurred during a query execution; - `Unconfirmed` – 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 tables currently applied and deleted. For example, if a query was executed 5 times, and then 2 tables were deleted, the 3 parameterized queries, specific parameter values are taken into account)
Databases	Names of databases accessed by a query (there may be multiple – for example, if a query uses `JOIN`)
Tables	Names of tables accessed by a query (there may be multiple – for example, if a query uses `JOIN`)
Used memory	Amount of memory consumed by a query
User	Name of an ADQM user who initiated a query execution. For queries executed on cluster hosts when running a distributed query, a user is not defined – the **User** column shows an empty string (`''`)
Time started	Query execution start time
Duration	Duration of a query execution or the current execution time (`now - time_started`) – if a query that has not yet been executed before a query is completed, a value in the **Duration** column will be replaced with the total query execution time since the query was launched and before the query was fully executed (the `Query finished` status).
Average duration	Average query execution time. Statistics are calculated for queries whose text completely matches (i.e. for parameterized queries, specific parameter values are taken into account).
Read bytes	Total amount of data read from all tables and tuples during a query. For distributed queries, this is the number of bytes read on all nodes. It can be different from the **Read bytes** value and the server initiator of the query sums of all received and local values.
Result bytes	Amount of memory used by a query result
Result rows	Number of rows in the result of a `SELECT` query or number of rows in an `INSERT` query
Read rows	Total number of rows read from all tables and table functions participated in a query. For distributed queries, this is the total number of rows read in 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: `WITH`, `SET`, `CREATE`, `INSERT`, `ALTER`, `DROP`, `SHOW`, `DELETE`, `RENAME`, `OPTIMIZE`, `GRANT`, `REVOKE`, `KILLQUERY`, `DESCRIBE`, `EXISTS`, `USE`, `SYSTEM`, `KILL`, `QUERY`, `BEGIN`, `COMMIT`, `ROLLBACK`, `SETTRANSACTION SNAPSHOT`, `ATTACH`, `DETACH`, `CHECK`, `TRUNCATE`, `EXTERNAL`, `OPTIMIZE`. `Select` – for `SELECT` queries; `Insert` – for `INSERT` queries; `Other` – for `CREATE`, `ALTER`, `DROP`, `SHOW`, `DESCRIBE`, `CHECK`, `USE`, `EXISTS` queries and queries that failed with the `SYNTAX_ERROR` or `INCORRECT_DATA` errors.

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

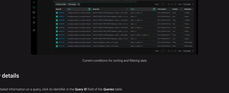
In the column headers of the table with the list of queries, there are filters that you can use to select specific data. To open a filter, click the icon (`⋮`). Filters are available in the following columns:

- **Query text** – enter a truncated query text (as it is displayed in a table) that appears when you hover the mouse cursor over a column cell, or its fragment.
- **Databases**, **Tables** – enter a name of a database or table. You can use multiple databases or tables separated by commas. For example, `db1,db2` in the **Tables** column. Filtering works only if the search is performed by one item from a list of databases or tables is in cell.
- **Exception code** – enter a numerical error code (you can find the desired code) on the ClickHouse® site or select a group of errors on the **Presets** tab.
- **Host**, **User** – you can enter names of multiple hosts or users, respectively (for queries initiated by a distributed query, filtering by an empty value in the **User** column is not supported).
- **Duration** – you can enter the minimum and maximum query execution time (in seconds).
- **Status**, **Type** – you can select a value from a query list. Open the drop-down list.

The icon (`⋮`) indicates that a filter for column value is applied. To reset all filter conditions, click the icon (`↻`).

Data sorting is available in the following columns: **Exception** (actually sorting is performed by error code), **Exception code**, **Used memory**, **Time started**, **Duration**, **Read bytes**, **Result bytes**, **Result rows**, **Read rows**, **Written rows**, and **Written bytes**. Click the column name to sort the data in ascending order (`↑`) or descending order (`↓`). If the column name is not highlighted, the data is not sorted in this column. To reset sorting, click the column name until the highlighting disappears.

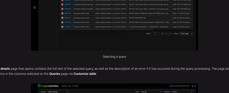
Above the table with queries, there is a panel that shows the sorting (**Sorted by**) and filtering (**Filtered by**) conditions currently applied to data:



Current conditions for sorting and filtering data

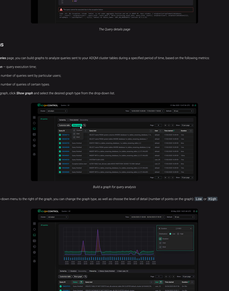
Query details

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 in the **Queries** table.



The Query details page

Graphs

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

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

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

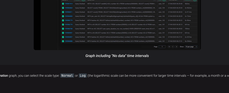
Build a graph for query analysis

In the drop-down menu to the right of the graph, you can change the graph type, as well as: choose the level of detail (number of points on the graph) – **Scale**; change the time period for which the graph is built – **Range**; and set the time interval for which the data is grouped – **Group by**.

Change the graph type and select the level of detail

Duration

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

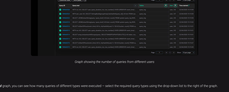


Graph of queries by execution duration

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

- **95th Quantile** – 95th quantile of query execution time.
- **Average** – average query execution time.
- **Extreme zone** (zone shaded in red if the **Average** line is above the **95th Quantile** line) – time interval when there was one or more long queries relative to most others. For example, if the **95th Quantile** value is `0.01` and the **Average** value is `0.08`, this means that 5% of queries (`1 - 0.95`) took more than `0.01` seconds, and there was a query with a duration of `0.08` (or more) seconds. Such queries can be considered as outliers and need to be checked for the **Duration** value and the query text.
- **Calm zone** (zone between the **Average** and **95th Quantile** lines, shaded in green) – time interval when duration of all queries was comparable. A peak in this zone means that all queries in a given time interval took longer than usual.
- **Query** – number of queries.
- **No data** (zone shaded in gray) – time interval for which there is no data about queries according to the applied filters. The **No data** item appears in the legend only if the graph contains one or more time intervals with missing data.

Example of a graph with no data on queries



Graph including "No data" time intervals

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

User

The **User** graph allows you to see the number of queries sent to the cluster by each user and evaluate this number relative to the number of queries from other users and to the total of all queries. The graph is built using data on queries from users whose names are listed in the filter by the **User** column. You can also specify user names in the field to the right of the graph (the list of names is automatically updated with the filter by the **User** column).

Graph showing the number of queries from different users

Kind

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

Graph showing the number of queries of different types

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

Configuration parameters

Contents

- ADQM Control
- ADPS
- Monitoring

To Table of Contents

This article describes parameters that can be configured for ADQM Control services via ADQM. To see about the configured services, refer to the configuration article or about the basic installation of ADQM.

NOTE

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

ADQM Control

ADQM management

Parameter	Description	Default value
ADQM username	Username to connect to ADQM	adqmc
ADQM password	Password for the ADQM username, which is stored in ADQM encrypt form. To change the password, enter a new value in plain text.	adqmc
ADQM default port	Default port # to be assigned to host an ADQM cluster	9000
ADQM secure port	Default port # to be assigned to host an ADQM cluster if it is a TLS one	9440
Connection type	Type of connection between ADQM and ADQM Control services. Valid values: TCP, SECURE_STRICT, SECURE_CERT, SECURE_FULL.	Insecure (TCP)

NOTE

If the ADQM parameter **ADQM_CONTROL_CONNECTION** is set to **HOST_PORT** (see the [ADQM configuration article](#)), then the ADQM Control service will connect to ADQM via the host and port specified in the **ADQM_CONTROL_HOST** and **ADQM_CONTROL_PORT** environment variables, respectively.

```
CREATE USER IF NOT EXISTS admin IDENTIFIED WITH sha256_password BY 'qwerty123';

GRANT ON CLUSTER default_cluster ALL WITH GRANT OPTION TO admin;
```

Backend

Parameter	Description	Default value
Backend timeout	Connection wait time interval for backend (timeout) (in seconds)	30
Enable CORS validation	Enables CORS validation	Enabled

SSL configuration

NOTE

The default value for all configured automatically (for configuration of HAProxy services, see the [HAProxy article](#)).

Parameter	Description	Default value
ADQM Control Backend - Enable SSL	Enables SSL for the Backend component	Disabled
ADQM Control ADQM Agent2 - Enable SSL	Enables SSL for the Agent component	Disabled
ADQM Control Auth Server (Keeper) - Enable SSL	Enables SSL for the Auth Server component	Disabled
ADQM Control Auth Server (Keeper) - Enable SSL	Enables SSL for the Auth Server component	Disabled
ZKHelper - Enable SSL	Enables SSL for the ZKHelper component	Disabled
ADQM Control Backend - Provide key file	Path to the file with the private key of the server SSL certificate for the Backend component	Will be added automatically
ADQM Control Backend - Certificate file	Path to the server SSL certificate for the Backend component	Will be added automatically
ZKHelper/Agent2 - Certificate file	Path to the server SSL certificate for the ZKHelper/Agent component	Will be added automatically
ZKHelper/Agent2 - Private key file	Path to the file with the private key of the server SSL certificate for the ZKHelper/Agent component	Will be added automatically
ZKHelper/Agent2 - Certificate authority file	Path to the file with the server SSL certificate authority for the ZKHelper/Agent component	Will be added automatically

Keeper

Parameter	Description	Default value
Session timeout	Connection wait time interval for session (timeout) (in seconds)	30
Enable CORS validation	Enables CORS validation	Enabled

Keeper configuration

Parameter	Description	Default value
ADQM Control Keeper - Enable SSL	Enables SSL for the Keeper component	Disabled
ADQM Control Keeper - Enable SSL	Enables SSL for the Keeper component	Disabled
ADQM Control Auth Server (Keeper) - Enable SSL	Enables SSL for the Auth Server component	Disabled
ADQM Control Auth Server (Keeper) - Enable SSL	Enables SSL for the Auth Server component	Disabled
ZKHelper/Agent2 - Private key file	Path to the file with the private key of the server SSL certificate for the ZKHelper/Agent component	Will be added automatically
ZKHelper/Agent2 - Certificate file	Path to the server SSL certificate for the ZKHelper/Agent component	Will be added automatically
ZKHelper/Agent2 - Certificate authority file	Path to the file with the server SSL certificate authority for the ZKHelper/Agent component	Will be added automatically

Keeper configuration

Parameter	Description	Default value
Backend timeout	Connection wait time interval for backend (timeout) (in seconds)	3000
Auth timeout	Connection wait time interval for auth (timeout) (in seconds)	3000
ADQM agent port	Port the Agents service listens on	8002
ADQM Keeper port	Port the Keeper service listens on	8003
ADQM Keeper port	Port the Keeper service listens on	8003
ADQM Keeper port	Port the Keeper service listens on	8003

Query navigation codes

Contents

Syntax Processing
Semantics
Resource Governing
Runtime Operations
Data Handling
Core Scripting
Networking
Concurrency and async
Distributed Execution
Core Control
Core Storage
System Governance
Interoperability
Features

Try Free — It's Instant

The Power and Ease of Query Navigation Codes

Syntax Processing

Control how the engine parses and validates query text.

Syntax and grammar

Function code	Example
SYNTAX.MODE	SYNTAX.MODE 'STRICT'
SYNTAX.VERSION	SYNTAX.VERSION 'V2'
SYNTAX.CASE	SYNTAX.CASE 'INSENSITIVE'
SYNTAX.QUOTES	SYNTAX.QUOTES 'DOUBLE'

API versions and limits

Function code	Example
API.VERSION	API.VERSION '2.0'
API.TIMEOUT	API.TIMEOUT 30
API.RETRY	API.RETRY 3

Specific rules and limits

Function code	Example
LIMIT.MAX.ROWS	LIMIT.MAX.ROWS 1000
LIMIT.MAX.SIZE	LIMIT.MAX.SIZE '10MB'
LIMIT.MAX.TIME	LIMIT.MAX.TIME 60
LIMIT.MAX.DEPTH	LIMIT.MAX.DEPTH 5

Semantics

Define how queries are interpreted and executed.

- Semantics control how queries are interpreted and executed.
- Examples: SEMANTICS.MODE 'STRICT', SEMANTICS.NULL.HANDLING 'IGNORE'.
- Semantics can be set globally or per query.

Execution and approach

Function code	Example
EXECUTION.MODE	EXECUTION.MODE 'PARALLEL'
EXECUTION.TIMEOUT	EXECUTION.TIMEOUT 30
EXECUTION.PRIORITY	EXECUTION.PRIORITY 'HIGH'

Type casts and type rules

Function code	Example
TYPE.CAST	TYPE.CAST 'NUMBER'
TYPE.CHECK	TYPE.CHECK 'STRICT'
TYPE.COERCE	TYPE.COERCE 'ON'

Joins, comparisons and comparisons

Function code	Example
JOIN.TYPE	JOIN.TYPE 'INNER'
JOIN.ON	JOIN.ON 'CUSTOMER_ID'
JOIN.USING	JOIN.USING 'ORDER_ID'

Aggregates and aggregation

Function code	Example
AGGREGATE.FUNC	AGGREGATE.FUNC 'SUM'
AGGREGATE.GROUP	AGGREGATE.GROUP 'REGION'
AGGREGATE.HAVING	AGGREGATE.HAVING 'TOTAL > 1000'

Runtime Operations

Control how queries execute at runtime.

- Runtime operations control how queries are executed.
- Examples: RUNTIME.MODE 'PRODUCTION', RUNTIME.LOG.LEVEL 'DEBUG'.
- Runtime operations can be set globally or per query.

Fallbacks and logic

Function code	Example
FALLBACK.MODE	FALLBACK.MODE 'ENABLED'
FALLBACK.VALUE	FALLBACK.VALUE 'DEFAULT'
FALLBACK.RETRY	FALLBACK.RETRY 3

Runtime operations

Function code	Example
RUNTIME.MODE	RUNTIME.MODE 'PRODUCTION'
RUNTIME.LOG.LEVEL	RUNTIME.LOG.LEVEL 'DEBUG'
RUNTIME.TIMEOUT	RUNTIME.TIMEOUT 30
RUNTIME.RETRY	RUNTIME.RETRY 3

Resource Governing

Manage and control how queries use system resources.

- Resource governing controls how queries use system resources.
- Examples: RESOURCE.LIMIT.MEMORY '1GB', RESOURCE.LIMIT.CPU '50%'.
- Resource governing can be set globally or per query.

Memory and disk space

Function code	Example
MEMORY.LIMIT	MEMORY.LIMIT '1GB'
DISK.SPACE	DISK.SPACE '10GB'

Query timeouts, limits and termination

Function code	Example
TIMEOUT	TIMEOUT 30
LIMIT.ROWS	LIMIT.ROWS 1000

Rate and concurrency limits

Function code	Example
RATE.LIMIT	RATE.LIMIT '100/MIN'
CONCURRENCY.LIMIT	CONCURRENCY.LIMIT 5

Caching policy, and invalidation

Function code	Example
CACHE.MODE	CACHE.MODE 'ENABLED'
CACHE.TTL	CACHE.TTL 3600
CACHE.INVALIDATE	CACHE.INVALIDATE 'ON_CHANGE'

Filesystem

Control filesystem access and operations on the server's filesystem.

- Filesystem controls how queries access and operate on the server's filesystem.
- Examples: FILESYSTEM.ACCESS 'READ_ONLY', FILESYSTEM.PATH '/data/queries'.
- Filesystem can be set globally or per query.

Search/find and operations

Function code	Example
FIND.FILE	FIND.FILE '*.txt'
FIND.DIR	FIND.DIR 'logs'
FIND.RECURSIVE	FIND.RECURSIVE 'data'

Options, path and file management

Function code	Example
OPEN.FILE	OPEN.FILE 'data.txt'
CLOSE.FILE	CLOSE.FILE 'data.txt'
READ.FILE	READ.FILE 'data.txt'
WRITE.FILE	WRITE.FILE 'data.txt'

Size and limits management

Function code	Example
DISK.SPACE	DISK.SPACE '/data'
FILE.SIZE	FILE.SIZE 'data.txt'
LIMIT.FILE.SIZE	LIMIT.FILE.SIZE '10MB'

Create and delete operations

Function code	Example
CREATE.FILE	CREATE.FILE 'data.txt'
DELETE.FILE	DELETE.FILE 'data.txt'
CREATE.DIR	CREATE.DIR 'logs'
DELETE.DIR	DELETE.DIR 'logs'

Data Health

Check the health and status of data in the database, including data validity and integrity.

Integrity and checksum

Function code	Example
CHECKSUM.VERIFY	CHECKSUM.VERIFY 'data.txt'
INTEGRITY.CHECK	INTEGRITY.CHECK 'table1'
CHECKSUM.CALC	CHECKSUM.CALC 'data.txt'

Schema validation and check

Function code	Example
SCHEMA.VALIDATE	SCHEMA.VALIDATE 'schema.json'
SCHEMA.CHECK	SCHEMA.CHECK 'table1'
SCHEMA.VERSION	SCHEMA.VERSION '1.0'

Core Scripting

Control how queries are executed and scripted.

- Core scripting controls how queries are executed and scripted.
- Examples: SCRIPT.MODE 'INTERACTIVE', SCRIPT.TIMEOUT 30.
- Core scripting can be set globally or per query.

Core and control

Function code	Example
SCRIPT.MODE	SCRIPT.MODE 'INTERACTIVE'
SCRIPT.TIMEOUT	SCRIPT.TIMEOUT 30
SCRIPT.ENV	SCRIPT.ENV 'PRODUCTION'

Specific functions

Function code	Example
SCRIPT.RUN	SCRIPT.RUN 'script.sql'
SCRIPT.STOP	SCRIPT.STOP 'script.sql'
SCRIPT.STATUS	SCRIPT.STATUS 'script.sql'

Networking

Control how queries access the network.

- Networking controls how queries access the network.
- Examples: NETWORK.MODE 'ENABLED', NETWORK.TIMEOUT 30.
- Networking can be set globally or per query.

Connection and session

Function code	Example
CONNECT	CONNECT 'db1.example.com'
DISCONNECT	DISCONNECT 'db1'
SESSION.ID	SESSION.ID 'abc123'

Protocol and operations

Function code	Example
HTTP.GET	HTTP.GET 'https://api.example.com/data'
HTTP.POST	HTTP.POST 'https://api.example.com/data'
FTP.GET	FTP.GET 'ftp://ftp.example.com/file.txt'
FTP.PUT	FTP.PUT 'file.txt' 'ftp://ftp.example.com/file.txt'

Distributed Execution

Control how queries are executed across multiple servers or nodes.

Coordination

Function code	Example
NODE.ID	NODE.ID 'node1'
CLUSTER.NAME	CLUSTER.NAME 'mycluster'
LEADER.ELECT	LEADER.ELECT 'node1'

Replica state and management

Function code	Example
REPLICA.SYNC	REPLICA.SYNC 'node2'
REPLICA.STATUS	REPLICA.STATUS 'node2'
REPLICA.LAG	REPLICA.LAG 'node2'

Queries and distribution

Function code	Example
QUERY.DISTRIBUTE	QUERY.DISTRIBUTE 'ON'
QUERY.NODE	QUERY.NODE 'node1'

Distributed query execution

Function code	Example
QUERY.EXECUTE	QUERY.EXECUTE 'SELECT * FROM table1'
QUERY.STATUS	QUERY.STATUS 'query1'
QUERY.CANCEL	QUERY.CANCEL 'query1'

Core Storage

Control how queries interact with persistent storage (databases or files).

Data and management

Function code	Example
DATA.FILE	DATA.FILE 'customers.dat'
DATA.DIR	DATA.DIR '/data'

Queries, tables and management

Function code	Example
TABLE.CREATE	TABLE.CREATE 'customers'
TABLE.DROP	TABLE.DROP 'customers'
TABLE.ALTER	TABLE.ALTER 'customers'

Metadata and structure

Function code	Example
SCHEMA.GET	SCHEMA.GET 'customers'
SCHEMA.SET	SCHEMA.SET 'customers'

Caching and background operations

Function code	Example
CACHE.CLEAR	CACHE.CLEAR 'customers'
INDEX.REBUILD	INDEX.REBUILD 'customers'

System Governance

Control how queries interact with the system and its resources.

API, operations

Function code	Example
API.VERSION	API.VERSION '2.0'
API.TIMEOUT	API.TIMEOUT 30
API.RETRY	API.RETRY 3

Configuration and settings

Function code	Example
CONFIG.SET	CONFIG.SET 'key' 'value'
CONFIG.GET	CONFIG.GET 'key'
CONFIG.RESET	CONFIG.RESET 'key'

Environment

Function code	Example
ENV.SET	ENV.SET 'VAR' 'value'
ENV.GET	ENV.GET 'VAR'
ENV.UNSET	ENV.UNSET 'VAR'

Access Control

Control who can access and modify data and resources.

- Access control controls who can access and modify data and resources.
- Examples: ACCESS.GRANT 'user1' 'table1' 'SELECT', ACCESS.REVOKE 'user1' 'table1' 'INSERT'.
- Access control can be set globally or per query.

Authentication

Function code	Example
AUTH.LOGIN	AUTH.LOGIN 'user1' 'password'
AUTH.LOGOUT	AUTH.LOGOUT 'user1'
AUTH.CHECK	AUTH.CHECK 'user1'

Permissions

Function code	Example
GRANT	GRANT 'user1' 'table1' 'SELECT'
REVOKE	REVOKE 'user1' 'table1' 'INSERT'
CHECK.PERMISSION	CHECK.PERMISSION 'user1' 'table1' 'UPDATE'

Roles and groups

Function code	Example
ROLE.CREATE	ROLE.CREATE 'admin'
ROLE.DROP	ROLE.DROP 'admin'
ROLE.ASSIGN	ROLE.ASSIGN 'user1' 'admin'
ROLE.REVOKE	ROLE.REVOKE 'user1' 'admin'

Interoperability

Control how queries interact with other systems and languages.

Data exchange and import

Function code	Example
IMPORT.DATA	IMPORT.DATA 'data.csv'
EXPORT.DATA	EXPORT.DATA 'data.csv'
CONVERT.FORMAT	CONVERT.FORMAT 'data.csv' 'data.json'

Foreign data and operations

Function code	Example
FOREIGN.QUERY	FOREIGN.QUERY 'SELECT * FROM table1'
FOREIGN.TABLE	FOREIGN.TABLE 'table1'
FOREIGN.CONNECT	FOREIGN.CONNECT 'db1.example.com'

Interop and operations

Function code	Example
INTEROP.CALL	INTEROP.CALL 'function' 'arg1' 'arg2'
INTEROP.RESULT	INTEROP.RESULT 'result'

Shell and operations

Function code	Example
SHELL.RUN	SHELL.RUN 'ls -l'
SHELL.OUTPUT	SHELL.OUTPUT 'output'

Machine learning

Function code	Example
ML.TRAIN	ML.TRAIN 'model1' 'data.csv'
ML.PREDICT	ML.PREDICT 'model1' 'data.csv'

Concurrency

Control how queries execute concurrently and manage shared resources.

Locks and transactions

Function code	Example
LOCK.ACQUIRE	LOCK.ACQUIRE 'table1' 'EXCLUSIVE'
LOCK.RELEASE	LOCK.RELEASE 'table1'
LOCK.STATUS	LOCK.STATUS 'table1'
TRANSACTION.BEGIN	TRANSACTION.BEGIN
TRANSACTION.COMMIT	TRANSACTION.COMMIT
TRANSACTION.ROLLBACK	TRANSACTION.ROLLBACK

Async and operations

Function code	Example
ASYNC.RUN	ASYNC.RUN 'query1'
ASYNC.WAIT	ASYNC.WAIT 'query1'
ASYNC.STATUS	ASYNC.STATUS 'query1'

Threads and operations

Function code	Example
THREAD.CREATE	THREAD.CREATE 'worker1'
THREAD.JOIN	THREAD.JOIN 'worker1'
THREAD.STATUS	THREAD.STATUS 'worker1'

Features

Control which features are enabled and how they behave in queries.

Modes and control

Function code	Example
MODE.SET	MODE.SET 'STRICT'
MODE.GET	MODE.GET
MODE.RESET	MODE.RESET

Properties

Function code	Example
FEATURE.ENABLE	FEATURE.ENABLE 'NEW_PARSER'
FEATURE.DISABLE	FEATURE.DISABLE 'OLD_API'
FEATURE.LIST	FEATURE.LIST

Query codes

Function code	Example
QUERY.CODE.GET	QUERY.CODE.GET 'SUBMIT_ORDER'
QUERY.CODE.SET	QUERY.CODE.SET 'SUBMIT_ORDER' 'SELECT * FROM orders'

Transaction

Function code	Example
BEGIN.TRANSACTION	BEGIN.TRANSACTION
COMMIT.TRANSACTION	COMMIT.TRANSACTION

Machine learning

Function code	Example
ML.TRAIN	ML.TRAIN 'model1' 'data.csv'
ML.PREDICT	ML.PREDICT 'model1' 'data.csv'

System Errors

Control how errors are handled and reported during query execution.

Exceptions and error handling

Function code	Example
ERROR.RAISE	ERROR.RAISE 'Invalid input'
ERROR.CATCH	ERROR.CATCH 'ERR_001'
ERROR.LOG	ERROR.LOG 'Error occurred'
ERROR.CLEAR	ERROR.CLEAR

Debug and control

Function code	Example
DEBUG.LOG	DEBUG.LOG 'Debug message'
DEBUG.TRACE	DEBUG.TRACE 'ON'
DEBUG.DUMP	DEBUG.DUMP 'variable1'
DEBUG.ASSERT	DEBUG.ASSERT 'condition'

Session and operations

Function code	Example
SESSION.START	SESSION.START
SESSION.END	SESSION.END
SESSION.ID	SESSION.ID
SESSION.STATUS	SESSION.STATUS

ADQM Control monitoring metrics

Elena Dvoryadkina

ADQM CONTROL

Contents

To Table of Contents

System metrics

ADQM Control service metrics

The Monitoring service installed in an ADQM Control cluster collects two groups of metrics: system metrics from hosts and ADQM Control service metrics.

You can view metrics in your browser in the Prometheus format (ports and endpoints for access to metrics are described below), as well as use the Prometheus and Grafana web interfaces.

System metrics

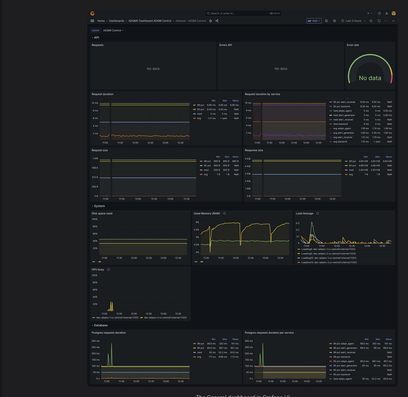
System metrics indicate general characteristics of cluster hosts, usually related to resource consumption: for example, CPU utilization, disk capacity, memory usage, I/O performance, and other parameters.

To view system metrics of ADQM Control hosts in the Prometheus format, enter in the address bar of your browser `http://<adqmc_host_ip>:11203/metrics`, where `11203` and `/metrics` are a port number and endpoint to listen for system metrics, which are set up in the ***Node Exporter settings*** section of the Monitoring service configuration.

In the Grafana interface, system metrics for ADQM Control hosts are displayed in the ***System*** dashboard and in the ***System*** row of the ***General*** dashboard.

The ***General*** dashboard also contains the ***Database*** section with the ***Postgres requests duration*** and ***Postgres requests duration per service*** graphs based on the `postgres_client_duration_seconds_bucket` metric – duration of queries to the PostgreSQL database.

The General dashboard in Grafana



The General dashboard in Grafana UI

ADQM Control service metrics

ADQM Control service metrics allow you to monitor and analyze various parameters of ADQM Control operation and performance: for example, availability of service components, types and frequency of errors, response time for user requests, frequency of alert generation and alert types, and other metrics.

Use the addresses listed in the table below to explore monitoring metrics of the ADQM Control service components in the Prometheus format. Provided addresses contain default ports for accessing metrics of service components. You can change these ports in the ***Network configuration*** section of the ADQM Control service configuration.

ADQM Control service component	Address to access component metrics
Agents	http://<host_ip>:5002/api/v1/metrics
Alert Generator	http://<host_ip>:5001/api/v1/metrics
Alert Receiver	http://<host_ip>:12322/api/v1/metrics
Alertmanager	http://<host_ip>:9093/metrics
Backend	http://<host_ip>:5555/api/v1/metrics

The tables below describe the *ADQM Control* service metrics, grouped as they are presented on dashboards in the Grafana interface.

Adqm_agent

Metrics of the *Agents* component show which information about ADQM clusters is collected and stored by ADQM Control.

Metric name	Description
adqm_agent_hosts	Number of ADQM cluster hosts
adqm_agent_databases	Number of ADQM databases
adqm_agent_tables	Number of ADQM tables
adqm_agent_columns	Number of columns in ADQM tables
adqm_agent_queries	Number of queries run in ADQM
adqm_agent_queries_normalized	Number of normalized (linked to tables) queries
jobs_total	Number of jobs performed by a service. Job statuses: `success`, `failed`. Job types: `alerts_cleanup`, `check_query_log`, `hosts_collection`, `job_log_cleanup`, `queries_collection`, `queries_duration_aggregation`, `queries_duration_collection`, `queries_normalization`, `tables_collection`
adqm_agent_queries_unnormalizable	Number of queries that cannot be normalized
adqm_agent_queries_unnormalized	Number of unnormalized queries
adqm_agent_queries_queue_size	Size of the current queue of queries for normalization
adqm_agent_normalization_cache_hit_ratio	Coefficient of hitting normalization tables in the cache
adqmc_database_corruption	Indicates issues with the consistency of the ADQM Control database (the ***Missing**** panels at the bottom of the ***Adqm_agent*** dashboard)

The Adqm_agent dashboard in Grafana UI

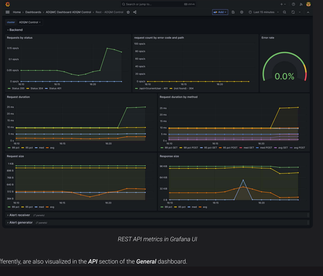
Alerts

Metric name	Description
alertgenerator_alerts_lifetime_bucket	Alert lifetime (the period between the time an alert is sent and the time the alert is no longer considered valid)
alertgenerator_alerts_total	Number of generated alerts
alertgenerator_fired_alerts_total	Number of alerts sent to Alert Manager
alertreceiver_alerts_received_total	Number of alerts received by Alert Receiver
alertgenerator_alerts_resend_total	Number of alerts resent to Alert Manager

Rest

The ***Rest*** dashboard contains the ***Backend***, ***Alert receiver***, and ***Alert generator*** rows, where panels display the following metrics for the corresponding service components.

Metric name	Description
http_request_count_total	Number of requests
http_request_duration_seconds_bucket	Duration of requests in seconds
http_request_size_bytes_bucket	Size of requests in bytes
http_response_size_bytes_bucket	Size of responses in bytes



REST API metrics in Grafana UI

REST API metrics, grouped differently, are also visualized in the ***API*** section of the ***General*** dashboard.

ADQM Control releases

Elena Dvoryadkina

Contents

To Table of Contents

1.2.1
1.2.0
1.1.1
1.1.0
1.0.0

1.2.1

1.2.1.b2

Date: 18.03.2026

New features

- Added support for Astra Linux 1.8.4 Voronezh
- Added support for Astra Linux 1.8.4 Orel

Bug fixes

- Fixed an issue related to Monitoring when ADQM Control was not working with enabled SSL

1.2.1.b1

Date: 02.03.2026

Improvements

- Added the ability to access the ADQM Notebook web interface from the Dashboard and Tables pages
- Large numbers (such as the number of data rows in a table and the number of queries sent to a table) are now displayed in a human readable format

1.2.0

Date: 04.02.2026

Improvements

- Updated permissions in the role-based access model of ADQM Control
- Added support for filtering queries by multiple exception codes
- Added support for filtering queries by exception code presets
- Implemented various backend and frontend optimizations

Misc/Internal

- Introduced ADQM Control Maintenance CLI
- Removed the Zookeeper service

1.1.1

Date: 10.09.2025

New features

- Added support for Alt Linux 10.2
- Added support for Astra Linux 1.7.6 Voronezh

Misc/Internal

- Updated submodules
- Increased the minimum version of ADCM to 2.7.1

1.1.0

Date: 24.07.2025

New features

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

1.0.0

Date: 17.12.2024

New features

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

Contents

To Table of Contents

ADPG

ADPG

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

ADPG configurations

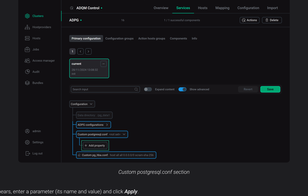
TIP

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

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

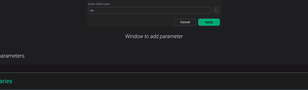
Custom postgresql.conf

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



Custom postgresql.conf section

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

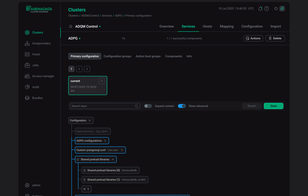


Window to add parameter

You can add multiple parameters.

Shared preload libraries

Section for setting the `shared_preload_libraries` parameter of the *postgresql.conf* configuration. ADCM Control requires the `timescaledb` and `timescaledb_toolkit` extensions. If you need to add other extensions (for example, `pg_stat_statements`), use the icon: →



The 'Shared preload libraries' section

Custom pg_hba.conf

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

SSL configuration

NOTE

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

Parameter	Description	Default value
Enable SSL	Enables SSL encryption of incoming traffic for the ADPG service	Disabled
Certificate file	Path to the server SSL certificate file	/etc/adpg14/server.crt
Private key file	Path to the file with the private key of the server SSL certificate	/etc/adpg14/server.key
Certificate authority file	Path to the CA certificate file	/etc/adpg14/ca.crt

Contents

To Table of Contents

ADQM Control

ADQM Control

ADQM management

Parameter	Description	Default value
ADQM username	Username to connect to ADQM	adqmc
ADQM user password	Password for the `ADQM username` user to connect to ADQM (should have a minimum length of 8 characters and should not contain `#`, `]`, `{`, `}`, `%`, `$`, `^`, `/`, `<`, `>`, `,`, `?`, `:`, or `;`)	–
ADQM default port	Default port if it is not exported from an ADQM cluster	9000
ADQM secure port	Default port for SSL-secured communication with ADQM if it is not exported from an ADQM cluster	9440
Connection type	Type of connection (insecure or secure) between ADQM Control and ADQM. Possible values: `Insecure (TCP)`, `Secure (TCP secure)`	Insecure (TCP)

NOTE

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

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


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

Backend

Parameter	Description	Default value
Session timeout	ADQM Control web interface session timeout (in seconds)	1800
Enable CSRF validation	Enables CSRF validation	Enabled

SSL configuration

NOTE

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

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

Network configuration

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

Connection settings

Parameter	Description	Default value
Backend connection pool size	Maximum number of connections to a PostgreSQL database of the Backend service	8
ADQM agent connection pool size	Maximum number of connections to a PostgreSQL database of the Agents service	8
Statement timeout	Timeout for queries and transactions (in seconds)	120
Cleanup timeout	Server timeout (in seconds): abort any database query that runs longer than the specified time	86400
Migration timeout	Database migration timeout (in seconds)	3600
Prometheus timeout	Prometheus connection timeout (in seconds)	60
Alertmanager timeout	Alertmanager connection timeout (in seconds)	60

Polling rate

Parameter	Description	Default value
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
Check query_log interval	Frequency of checking query_log (in seconds)	60

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
Tables cache size	Size of the table cache	32768
Table cache invalidate interval	Time interval of the table cache invalidation (in seconds)	28800
Maximum revision per table in cache	Maximum number of revisions per table in the cache	100
Queries cache size	Size of the query cache	8192
Batch size of queries normalization	Query batch size for normalization	1024
Capacity of existing tables cache	Capacity of the existing tables cache	500000000
Existing tables cache invalidate period	Period of the existing table cache invalidation (in seconds)	604800
Table's existing period for existing tables cache calculation	Period of a table's existence (in seconds) for the existing table cache calculation	7776000
Maximum of parallel workers	Maximum number of parallel workers	0

Internal PostgreSQL server

NOTE

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

Parameter	Description	Default value
Alerts dbname	Name of a database to be created for use by ADQM Control services	adqmc
Alertsdb username	Name of a user with access to the `Alerts dbname` database	adqmc
Alertsdb password	Password for the `Alertsdb username` user to access the `Alerts dbname` database (should have a minimum length of 8 characters and should not contain `"`, `#`, `]`, or `\`)	–

External PostgreSQL server

NOTE

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

Parameter	Description	Default value
server_ip_port	Click ***Add property*** to specify an IP address (as property name) and port (as property value) 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 `Alertsdb username` user to access the `Alerts dbname` database (should have a minimum length of 8 characters and should not contain `"`, `#`, `]`, or `\`)	–
SSL is used by an external server	Specifies whether SSL is used by an external server	Disabled

Contents

To Table of Contents

Monitoring

Monitoring

Prometheus settings

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

Grafana settings

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

Node Exporter settings

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

SSL configuration

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

Contents

To Table of Contents

Zookeeper

Zookeeper

Main

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

zoo.cfg

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

SSL configuration

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

Metrics

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

zookeeper-env.sh

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