AD.B ES

Arenadata DB Enterprise Services

Arenadata DB Enterprise Services (ADBES) is a solution that comes with the Enterprise edition of ADB and provides binary backup management and real-time query monitoring for that DBMS.

Version CURRENT Language: EN

Contents

Arenadata DB Control overview

Daria Barysheva

Contents

To Table of Contents

Features

Arenadata DB Control (ADB Control) is a real-time Arenadata DB query monitoring system.

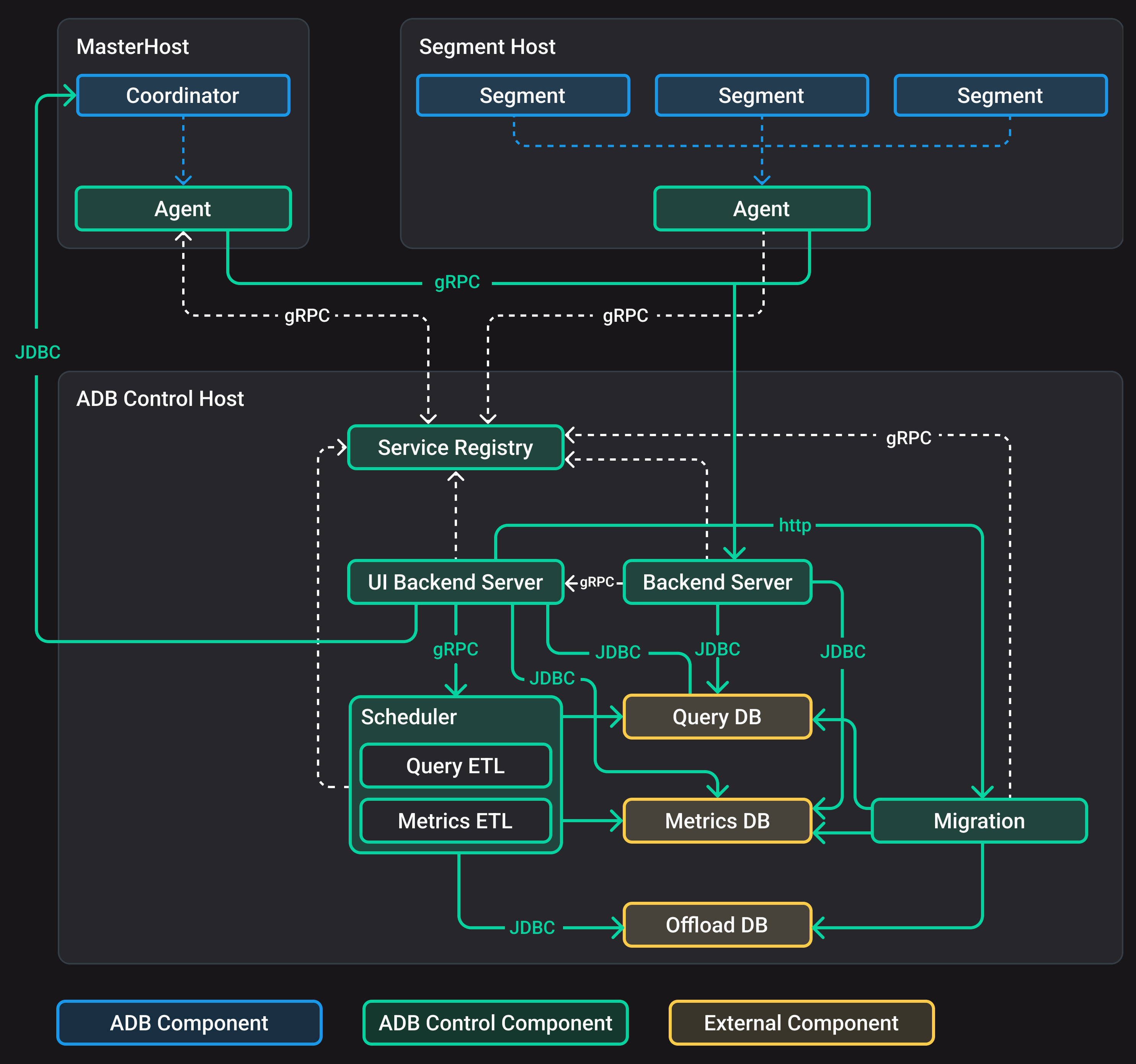
ADB Control allows you to perform the following functions:

- Work with multiple ADB clusters.
- Monitor SQL commands and transactions, including inner queries.
- Monitor active and idle sessions with ability to terminate them.
- View information on system resource consumption at the query and transaction levels.
- Build different charts for tracking the dynamic changes of system metrics over the selected time period.
- Cancel running transactions.
- Analyze statistics on resource group usage with ability to configure resource groups via the ADB Control web interface.
- Move running transactions to another resource group.
- View a query plan with ability to track the execution progress.
- Offload ADB Control metrics and events to an external database (PostgreSQL/ADB).
- Audit ADB relation accesses (in clusters used for monitoring), as well as user authorizations and other operations in ADB Control.

> **NOTE**
>
> Currently, ADB Control can be installed only offline.

Architecture

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



ADB Control architecture

ADB Control architecture is based on the following components:

- **Agent**. Application for receiving and filtering metrics from the ADB master and segments. Agents are installed one per ADB cluster host, i.e. one agent collects metrics from all segments within a single host.
- **Service Registry**. Registry of ADB Control services. Responsible for service discovery in ADB Control. Due to Service Registry, ADB Control discovers available agents (there is no pre-configured agent map in ADB Control). And agents, in turn, find necessary ADB Control services to send metrics. Since ADB Control can work with multiple ADB clusters at the same time — when you add a cluster, new agents are also registered via Service Registry.
- **UI Backend Server**. Backend server of the ADB Control web interface. Handles requests from Backend Server.
- **Backend Server**. Horizontally scalable service for processing and aggregating metrics from agents. It also writes query/metric information to the Query DB/Metrics DB databases and sends updates to the UI Backend Server.
- **Scheduler**. Runs system jobs on a schedule:
  - Cleanups old metrics and other data in ADB Control.
  - Offloads ADB Control metrics and audit events to an external database (Offload DB).
  - Actualizes command and transaction statuses.
  - Aggregates information on system resource consumption.
  - Obtains metrics on resource groups.
- **Migration**. Provides automatic migration of schemas and data in databases with which ADB Control interacts (Query DB, Metrics DB, Offload DB).
- **Query DB**. PostgreSQL database that performs the following functions:
  - Query logging.
  - Transaction logging.
  - Storage of system metric aggregates by transactions and queries.
  - Audit.
  - Statistics on database relations.
  - Storage of ADB Control system tables with information on users, clusters, system settings, and so on.
- **Metrics DB**. ClickHouse database that is used to store the following information:
  - System metrics obtained from agents.
  - History of resource consumption within resource groups (based on `gp_resgroup_*`).
- **Offload DB**. External PostgreSQL/ADB database that is used to offload metrics and audit data from ADB Control.

External storage usage

- By default, Query DB and Metrics DB come with an ADB ES bundle. Alternatively, you can connect both of them to ADB Control as external storage systems:
  - Query DB (see Database in the list of ADB ES services) — select `External postgres parameters` in the ***Database type*** field and fill in the ***External postgres parameters*** section when configuring the Database service (see Configuration parameters → Database).
  - Metrics DB (see Clickhouse in the list of ADB ES services) — select `External Clickhouse parameters` in the ***Database type*** field and fill in the ***External Clickhouse parameters*** section when configuring the Clickhouse service (see Configuration parameters → Clickhouse).
- Starting with ADB Control 4.7.5, you need to create users in external databases and grant them permissions manually. For more details, see Configure external databases for ADB Control and ADBM.

Arenadata DB Backup Manager overview

Daria Barysheva

Contents

To Table of Contents

Features

Arenadata DB Backup Manager (ADBM) is a fault-tolerant system for ADB binary backups management, which is built on the base of pgbackrest.

ADBM allows you to perform the following functions:

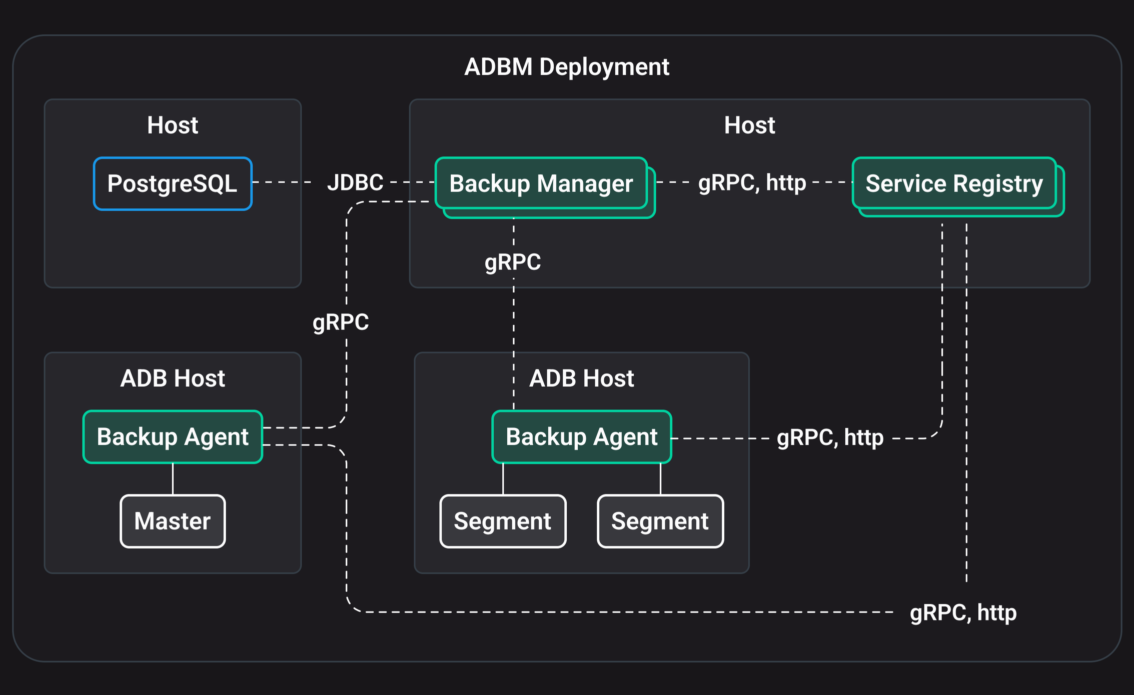
- Configure backup policies including separate schedules for different backup types.
- Create database backups according to the configured schedule and on demand.
- Maintain a list of backups with the ability to search for them and view details (including a cluster topology).
- Cleanup backups according to existing policies on a schedule and manually.
- Create restore points according to the configured schedule and manually.
- Restore databases at the moment of the selected restore point.
- Log all system actions with the ability to view details of the failed actions.
- Work with multiple ADB clusters of different types (with/without standby and mirroring).
- Use S3 and Posix compatible repositories to store backups.

NOTE

Currently, ADBM can be installed only offline.

Architecture

The high-level architecture view of ADBM is shown below.



ADBM architecture

ADBM architecture is based on the following components:

- **Backup Manager**. Orchestrates cluster actions and performs background work according to the configured schedules: creates backups and recovery points, cleans up old and invalid data, etc. Backup Manager and ADB clusters interact via Backup Agents.
- **Backup Agent**. Agents are installed one per ADB cluster host. They are responsible for segment data management:
  - Run `pgbackrest` commands.
  - Manage `pgbackrest` configuration.
  - On master host — manage ADB cluster operations (for example, cluster restart).
- **Service Registry**. Responsible for service discovery in ADBM. Due to Service Registry, ADBM discovers available agents (there is no pre-configured agent map in ADBM). And Backup Agents, in turn, find necessary ADBM services to send responses. Since Backup Manager can work with multiple ADB clusters at the same time — when you add a cluster, new agents are also registered via Service Registry.
- **PostgreSQL**. It is used to store the following data:
  - Cluster-wide locks to ensure the exclusivity of the actions to be launched.
  - Information about the current active action state. If Backup Manager stops or fails — the current action metadata can be retrieved from PostgreSQL.
  - Historical data on backups and configurations.

  By default, it is deployed via the Docker container. The external PostgreSQL database can also be used.

NOTE

- To use external PostgreSQL, select `External postgres parameters` in the ***Database type*** field and fill in the ***External postgres parameters*** section when configuring the Database service in ADB ES (for more details, see Configuration parameters → Database → External postgres parameters).
- Starting with ADBM 1.7.3, you need to create users in the external database and grant them permissions manually. For more details, see Configure external databases for ADB Control and ADBM.

Concepts

ADBM implements **point-in-time recovery (PITR)** — the ability to restore databases to the selected point in time. This approach is based on several concepts, which are summarized below.

Backup

Backup is a consistent copy of ADB cluster that can be used to restore databases in case of hardware and other failures. ADBM supports the following backup types:

- **Full**. When a full backup is created, the entire database content is backed up. The first backup running in ADBM within each timeline always has the full type, even if according to the schedule (or user choice — when launching a backup manually) the first should be a differential or incremental backup. A full backup can be used to restore data directly as it does not depend on any external files from other backups. The advantage of full backups is that all files are quickly restored (compared to other types). However, it is not recommended to create full backups regularly (hourly or daily) as they take a lot of time to be generated and occupy significant disk space.
- **Differential**. A differential backup contains only those database files that have changed since the last full backup was launched. In comparison with full backups, differential backups are faster and require less disk space. However, to restore data from a backup of this type, it is necessary to copy both its contents and files from the last full backup.
- **Incremental**. An incremental backup contains only those database files that have changed since the last backup of any other type was launched (full, differential, or incremental). Of all types, incremental backups are the fastest and occupy minimal disk space. However, their restoring takes longer because it is necessary to extract files from the last full and, if available, differential backup, and then apply all incremental backups sequentially.

WAL

Write-ahead log (WAL) is a standard mechanism to ensure that no committed changes are lost. All changes to data files (that contain tables, indexes, etc.) are written sequentially to the WAL. Afterwards, a background process writes the changes into the main database cluster files. In case of failures, the WAL can be replayed to make the database consistent.

WAL is broken up into individual 64 MB files called **segments**. Every record in a segment has a 16-digit logical sequence number (LSN) that is used to find this record by offset in the current segment file. A segment name, in turn, consists of the timeline number and LSN of the first segment record.

Using WAL significantly reduces the number of disk writes since there is no need to flush data pages to disk on every transaction commit. Only WAL files should be flushed to disk to guarantee that a transaction is committed.

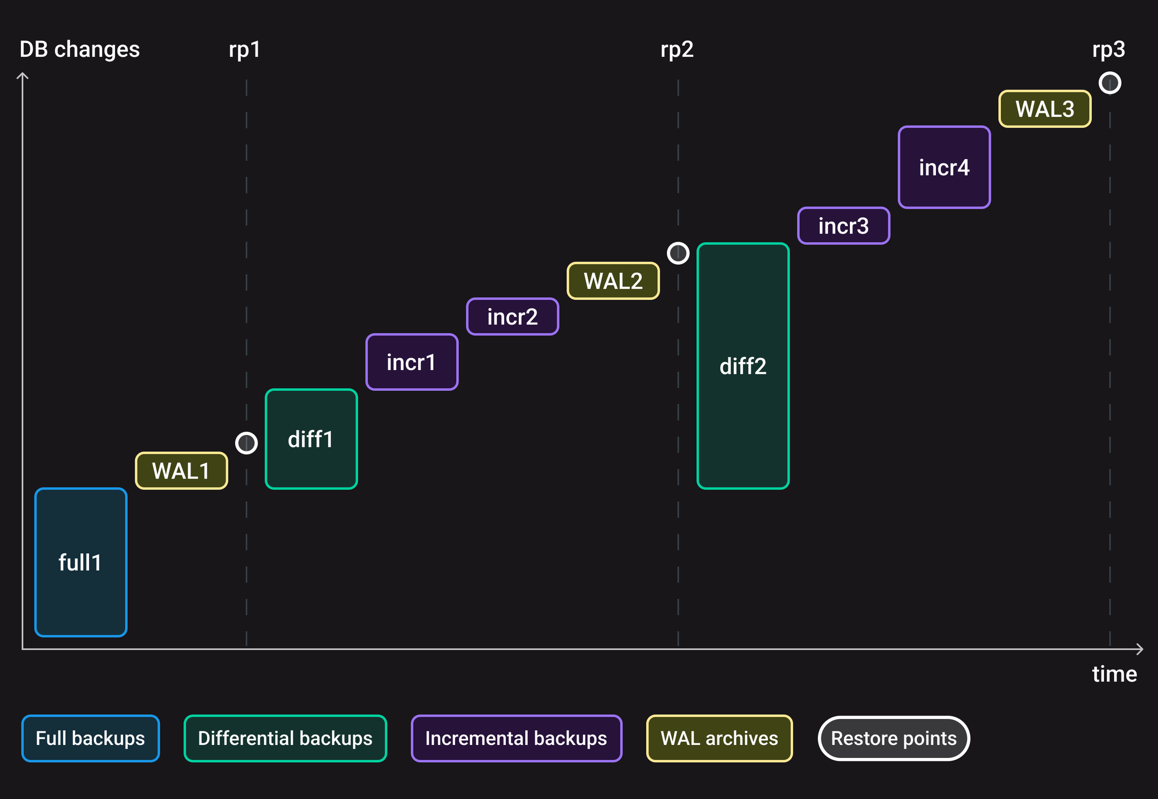
WAL plays a central role in the mechanism of point-in-time recovery (PITR) that is used in ADBM. By archiving the WAL data, ADBM allows you to restore databases at moment of any restore point covered by the available WAL data. In fact, the restore process requires two steps:

- Install a prior backup (or backups).
- Replay the WAL records, written after this backup, up to the specified named point.

Restore point

Restore point means a named recovery point, which is the minimum unit of data consistency granularity in ADB clusters. After a restore point is created (on a schedule or manually), you can restore the database state at the moment of the point creation. For each existing restore point, ADBM stores information on which backups should be used and which WAL segments should be reproduced to return the database state at the creation time of this point.

To better understand the mechanism of restore points, look at the figure below.



Example with restore points and different backup types

The following table explains which backups and WAL files will be used if you select any of the shown restore points when running the restore action.

Logic of restore points

Selected restore point	Objects to be used for recovery
rp1	full1, WAL1
rp2	full1, diff1, incr1, incr2, WAL2
rp3	full1, diff2, incr3, incr4, WAL3

Timeline

Timeline is a mechanism that is used in ADBM to distinguish the WAL series generated after the database recovery at the specified restore point from those created before the restore action was applied.

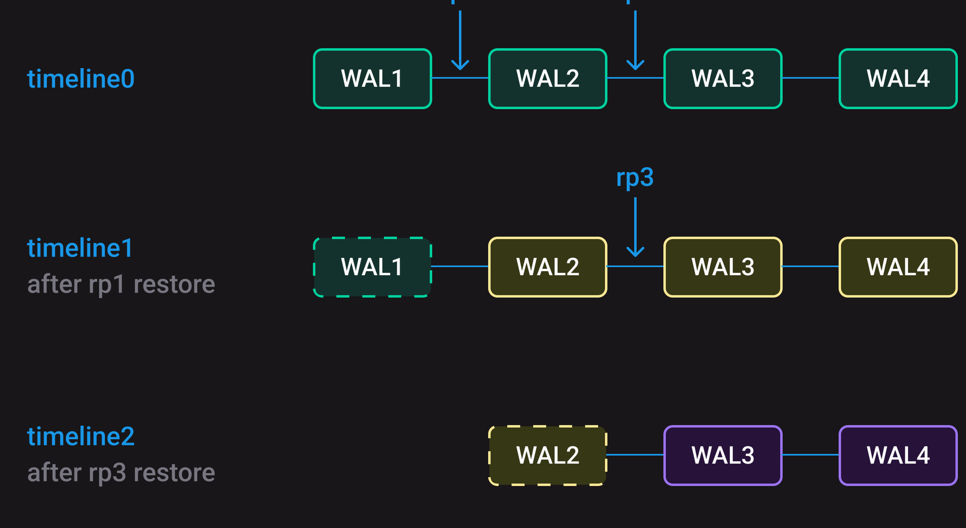
Suppose that to recover the database at the creation time of some restore point, after loading one or more backups it is necessary to reproduce `WAL1` (the number is used to simplify). After the archive recovery completes, WAL will continue to populate. If the WAL archives are added to the same directory, the subsequent WAL records will overwrite those that exist in the database original history (`WAL2`, `WAL3` and so on). This, in turn, will result in the failure of database recovery for other restore points added before recovery.

To avoid the conflicts described, ADBM initializes a new timeline number after each data recovery to identify post-recovery WAL archives. Timelines are numbered starting with `0`, with each successful restore action increasing the number by one. This number is recorded in the names of the directories that store WAL archives and backups divided by segments (so-called **stanza**). With each data recovery, all necessary directories are created with a new timeline number in the name, and all subsequent WAL records are placed into new directories without overwriting files in the directories that have the previous number.

IMPORTANT

Since newly created directories initially do not contain backups, the first backup after each data restore action will be of type `full`, even if a differential or incremental backup should be launched first according to the schedule (or user choice — when launching a backup manually).

The following figure illustrates the timeline logic in a simplified way. After you restore data at the moment of the `rp1` point, `WAL1` is replayed and `timeline1` is created. Subsequent WAL records are added to new directories. The next restore action creates `timeline2` and so on. Note that after creation of `timeline1` and `timeline2` it is still possible to restore data at the moment of the `rp2` point — without applying the timeline concept, this is impossible.



Timeline logic

Quick start

The typical sequence of actions when working with ADBM via the web interface includes the following steps:

1. Connect to ADBM.
2. Create a configuration, which will be used to create backups and perform other related actions.
3. Run a backup manually or wait for the automatic backup launch on a schedule. If necessary — view a list of backups.
4. Create a restore point manually or wait for the automatic restore point creation on a schedule.
5. Run the auxiliary operations if necessary:
   - Cleanup backups
   - Verify backups
   - Terminate the running backup/cleanup action
   - Restart the latest terminated backup
6. Restore data from a backup if it is necessary to restore the database state at the moment of one of existing restore points.

On each of the steps, you can view the details of running actions, including causes of the failures (if they are failed).

Migration of ADB Control and ADBM metadata during ADB ES installation

Daria Barysheva

Contents

To Table of Contents

Starting with ADB 6.30.0.1, ADB Control and ADBM come with a separate ADB Enterprise Services (ADB ES) bundle. On the ADB side, only agents are left: ADBM agents and ADBC agents. During initial ADB ES cluster installation, you should perform the following steps for a proper ADB upgrade (to version 6.30.0.1), including ADB Control and ADBM metadata migration from ADB to ADB ES.

IMPORTANT

- The following steps should be performed one time during the initial ADB ES installation and ADB upgrade to version 6.30.0.1 if ADB previously used ADB Control or ADBM. If these services were not installed in your ADB cluster before, none of the following steps is required — instead, perform the standard ADB cluster upgrade and offline installation of ADB ES (then, add ADBC and ADBM agents to ADB, import ADB ES configuration to ADB, and install agents in ADB).
- The minimal ADB version for upgrade is 6.27.1.63.

Prerequisites

Names of the external PostgreSQL databases for ADB Control and ADBM services in the ADB ES cluster (regardless of whether these databases were added earlier or will be created when you follow the steps listed below) should be generated according to the following pattern:

- ADB Control — `<prefix>_adcc`
- ADBM — `<prefix>_adbm`

The `<prefix>` value should be equal for both services. Subsequently, that value should be specified in the Database service parameter ***External postgres parameters → Database name prefix***. Examples of valid database names: `external_adcc` and `external_adbm` (where `external` is a prefix).

If external databases were previously created with names that do not meet the above-mentioned pattern, perform renaming after ADB upgrade before ADB ES installation (see Step 3. Rename external PostgreSQL databases).

Step 1. Migrate metadata stored in PostgreSQL

IMPORTANT

- You should perform step 1 if an external PostgreSQL database was not used for ADB Control or ADBM in ADB. If an external database was used for both services, you should proceed to the next step without performing metadata migration. If an external database was not used for any of the services — perform step 1 only for that service.
- Since ADB ES provides one common service for storing PostgreSQL data in ADB Control and ADBM (namely Database), if different PostgreSQL servers were previously used to store external databases for these services, it is necessary to select one PostgreSQL server, transfer both external databases to it, and then specify its settings in the Database service. Also, both databases should have the same owner.

1. Depending on which PostgreSQL target databases you plan to use in ADB ES, perform the following:
   - **External PostgreSQL databases:** — perform steps 1 — 3 from the ***PostgreSQL*** section of the Configure external databases for ADB Control and ADBM article.
   - **Internal ADPG databases that come with an ADB ES bundle** — add one Database service to ADB ES, map its components to hosts, and perform service installation. Before installation, on the service configuration page, leave the default value `Internal postgres database parameters` in the ***Database type*** parameter.

2. Migrate metadata from previously used internal databases to new ones (external or delivered in ADB ES). For this, you can use the pg_dump utility on the host where target databases are deployed. Example:

```
$ sudo -u postgres pg_dump -h <host> -p <port> --format=custom --no-owner --no-privileges -U <user> -f /tmp/<dump>.dump <database_name>
```

   where:

   - `-h <host>` — IP address of the host where the internal database is deployed (from which you perform metadata migration).
   - `-p <port>` — port number for connection to the internal database (by default, `5433`).
   - `--format=custom` — specifies creation of a compressed dump suitable for the pg_restore utility.
   - `--no-owner` — excludes owner information from a dump (owners will be set during restore).
   - `--no-privileges` — excludes privilege information from a dump (can be restored separately if needed).
   - `-U <user>` — username to connect to the internal database (by default, `postgres`).
   - `-f /tmp/<dump>.dump` — path to a dump. Replace `<dump>` by the target file name.
   - `<database_name>` — name of the internal database (by default, `adbm` for ADBM and `adcc` for ADB Control).

   IMPORTANT

   - You need to create separate dumps for ADB Control and ADBM since their metadata is stored in different databases.
   - When running the utility, you will be asked for a password to connect to the internal database. The default password is `123` (however, in your environment it may be different — see a note below).
   - In the example, the default credentials are used. However, for ADB Control, you should replace them with those you specified when configuring this service in your ADB cluster (see ***ADB Control → Database parameters*** in the ADB documentation versions prior to 6.30.0). For ADBM, use default values listed above.

3. Restore data from each dump you created. For this, on the host where target databases are deployed, run the pg_restore utility:

```
$ sudo -u postgres pg_restore -p <target_port> -d <target_db> --clean --no-owner /tmp/<dump>.dump
```

   where:

   - `-p <target_port>` — port number for connection to the target database. In case you use the internal database that comes with ADB ES, port number is `5433`.
   - `-d <target_db>` — name of the target database. In case you use the internal database that comes with ADB ES, database name is `adcc` for ADB Control and `adbm` for ADBM.
   - `--clean` — drop existing database objects before restoring.
   - `--no-owner` — do not restore information on database object ownership (owners will be set manually in the next step).

4. Ensure that metadata is loaded to the target database. You can do this by comparing the number of rows in the key tables of the source and target databases:

```
SELECT count(*) FROM <schema>.<table>;
```

   Query example for ADBM:

```
SELECT count(*) FROM adbm.adbm_restore_point;
```

5. Check that the correct owner is set for the `adcc` (in case of ADB Control) and `adbm` (in case of ADBM) schemas in target databases. If needed, change the owner manually. Example for `adbm`:

```
$ sudo -u postgres psql -p <target_port> -d <target_db> -c "ALTER SCHEMA adbm OWNER TO <target_owner>;"
```

   where `<target_owner>`:

   - For external databases — the `<user_name>` value that you used when performing SQL commands from the Configure external databases for ADB Control and ADBM article.
   - For internal ADPG databases that come with an ADB ES bundle — `postgres`.

6. For the `quartz` schema in ADBM, you should restore ownership for all its tables and sequences (if privileges were lost during migration).
   For tables:

```
DO $$ DECLARE r record;
BEGIN
    FOR r IN SELECT tablename FROM pg_tables WHERE schemaname = 'quartz' LOOP
        EXECUTE 'ALTER TABLE quartz.' || quote_ident(r.tablename) || ' OWNER TO <target_owner>;';
    END LOOP;
END $$;
```

   For sequences:

```
DO $$ DECLARE r record;
BEGIN
    FOR r IN SELECT sequence_name FROM information_schema.sequences WHERE sequence_schema = 'quartz' LOOP
        EXECUTE 'ALTER SEQUENCE quartz.' || quote_ident(r.sequence_name) || ' OWNER TO <target_owner>;';
    END LOOP;
END $$;
```

   where `<target_owner>` is the same as in the previous step.

Step 2. Upgrade ADB clusters

For all ADB clusters that were monitored in the previous version of ADB Control (before upgrade):

1. Remove ADBM from ADB using the Uninstall service action. In the ***Erase data*** field, set `false`.
2. Remove ADB Control from ADB using the Uninstall service action. In the ***Erase data*** field, set `false`.
3. Perform ADB cluster upgrade to 6.30.0.1.

Step 3. Rename external PostgreSQL databases

If external databases were previously created with names that do not meet the above-mentioned naming pattern, you should rename them before performing the next steps. However, renaming changes the Liquibase checksums, which can adversely affect future updates. To avoid this, after renaming databases, execute the following command in each external PostgreSQL database (ADB Control and ADBM) to reset checksums:

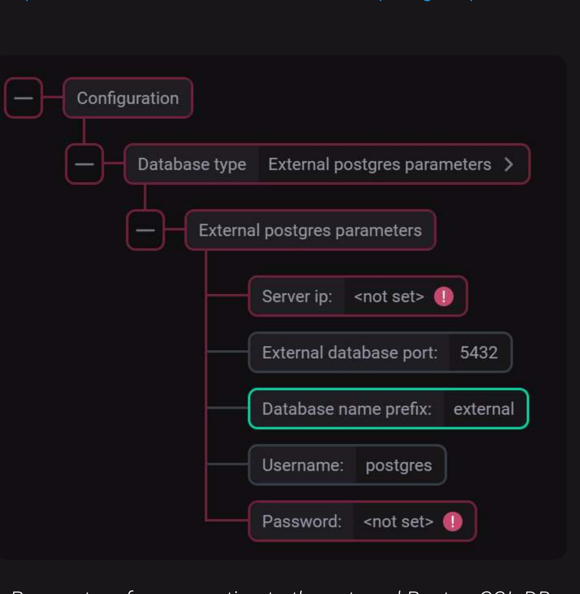
```
UPDATE DATABASECHANGELOG SET MD5SUM = NULL;
```

Step 4. Attach external PostgreSQL databases to ADB ES

IMPORTANT

Perform step 4 only if you plan to use external PostgreSQL (not included in ADB ES).

1. Add one Database service to ADB ES and map its components to hosts.
2. On the Database service configuration page, select `External postgres parameters` in the ***Database type*** field and fill in all parameters in the list that opens. The assignment of parameters is described in the Configuration parameters → Database → External postgres parameters section.



Parameters for connecting to the external PostgreSQL DB

3. Click ***Save*** to save the Database configuration.

Step 5. Install ADB ES and agents in ADB

1. In the ADB Control database, to where you migrated metadata (external or internal), run the following query to delete data about the existing ADB cluster (needed to prevent conflicts in the future):

```
DELETE FROM adcc.system_user WHERE cluster_name = '<cluster_name>';
```

   where `<cluster_name>` is an ADB cluster name.

2. In ADB ES, add other services (except Database) and map their components to hosts.
3. Perform ADB ES installation using the **Install** cluster action.
4. Add the ADBC agents and ADBM agents services to all ADB clusters that were monitored in the previous version of ADB Control.
5. Import ADB ES configuration to ADB clusters.
6. Install ADBC agents and ADBM agents in ADB clusters.

Step 6. Perform post-installation actions

1. In case of errors with GUC management in the ADB Control web interface after upgrade — on the host with ADB Control UI, add the following record to /etc/hosts:

```
<master_IP> <master_hostname>
```

   Since the master ADB Control agent is registered in Eureka using the ADB master hostname, there can be resolution errors when ADB Control UI tries to reach the agent to manage GUC settings.

2. On the ADB master host, run the following command to remove all existing configuration files of the pgbackrest utility:

```
$ gpssh -f arenadata_configs/arenadata_all_hosts.hosts -v -e 'rm -rf /home/gpadmin/arenadata_configs/pgbackrest/*'
```

3. Using the web interface of the ADBM service (installed in ADB ES), run the Verify action to recreate configuration files.
4. Using the web interface of the ADBM service (installed in ADB ES), run the Create configuration action to reapply the current backup configuration.

Step 7. Check results

As a successful result of the previous steps, all ADB clusters registered for monitoring in the previous version of ADB Control (before upgrade) should be displayed on the Configuration → Clusters page of the ADB Control web interface.

Offline installation

To Table of Contents

This topic describes **offline** installation of ADB Enterprise Services (ADB ES). This can be applied to the hosts without access to the Internet.

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

NOTE

ADB ES is compatible with ADB Enterprise version 6.30 and later. The installation sequence can be one of the following:

- Install ADB ES first and then import its configuration to your ADB Enterprise cluster.
- Install ADB Enterprise without ADB Control and ADBM agents, then install ADB ES and import the ADB ES configuration in your ADB cluster.

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

Step 1.

Install ADCM

Step 2.

Prepare hosts

Step 3.

Install ET cluster

Step 4.

Install ADB ES cluster

Step 5.

Install ADB cluster

Step 6.

Install ADB ES monitoring

Create a cluster

Daria Barysheva

Contents

To Table of Contents

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.

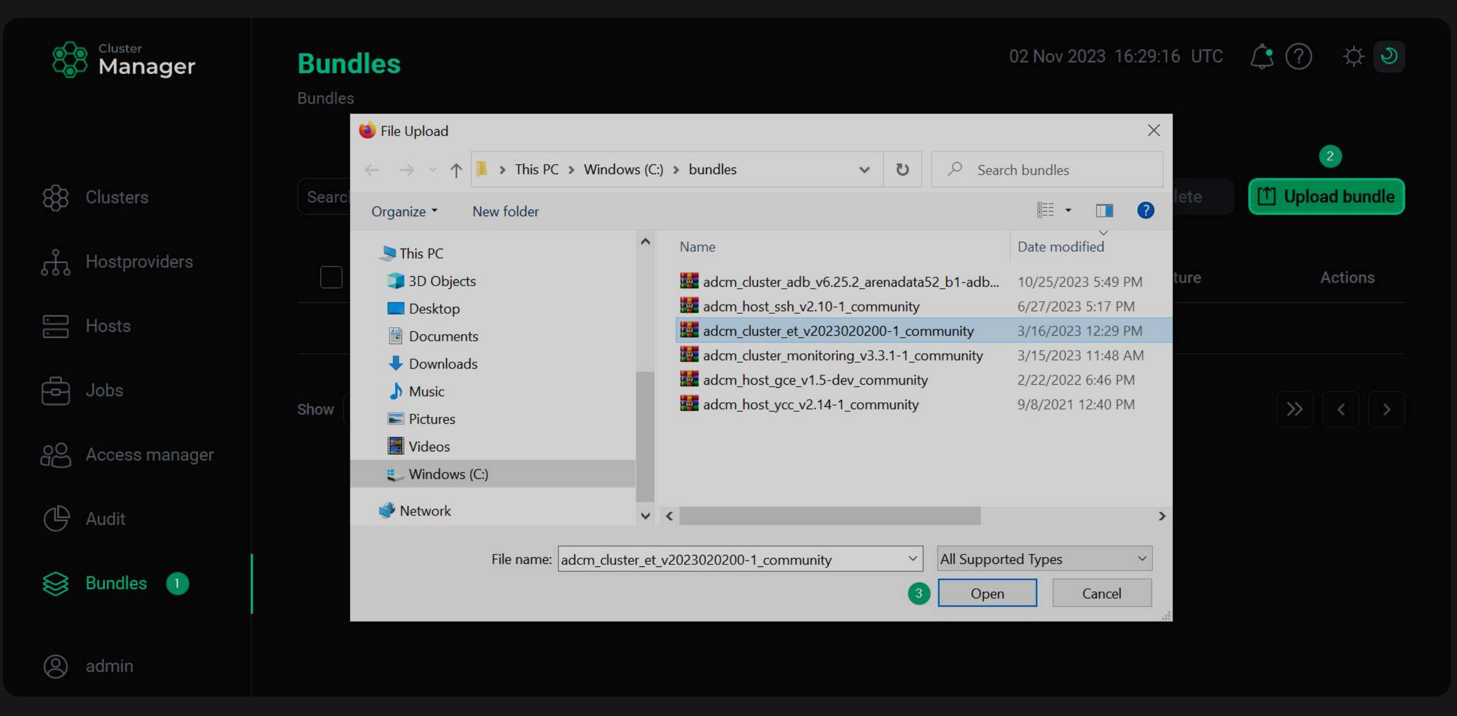
Step 1. Get the Enterprise version of a cluster bundle

Please get a cluster bundle from the Arenadata support team.

Step 2. Upload a cluster bundle to ADCM

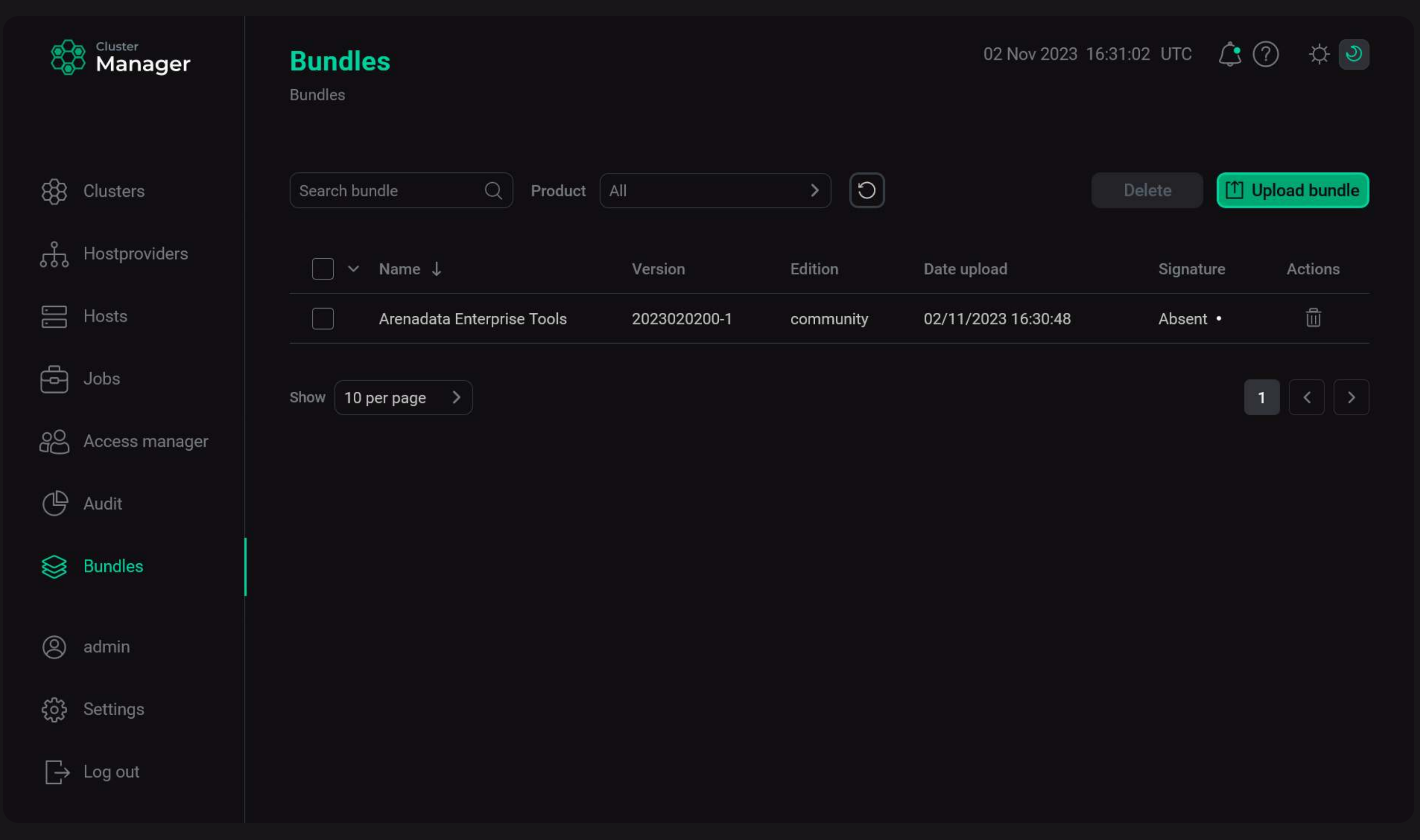
To upload a bundle to ADCM, follow the steps:

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

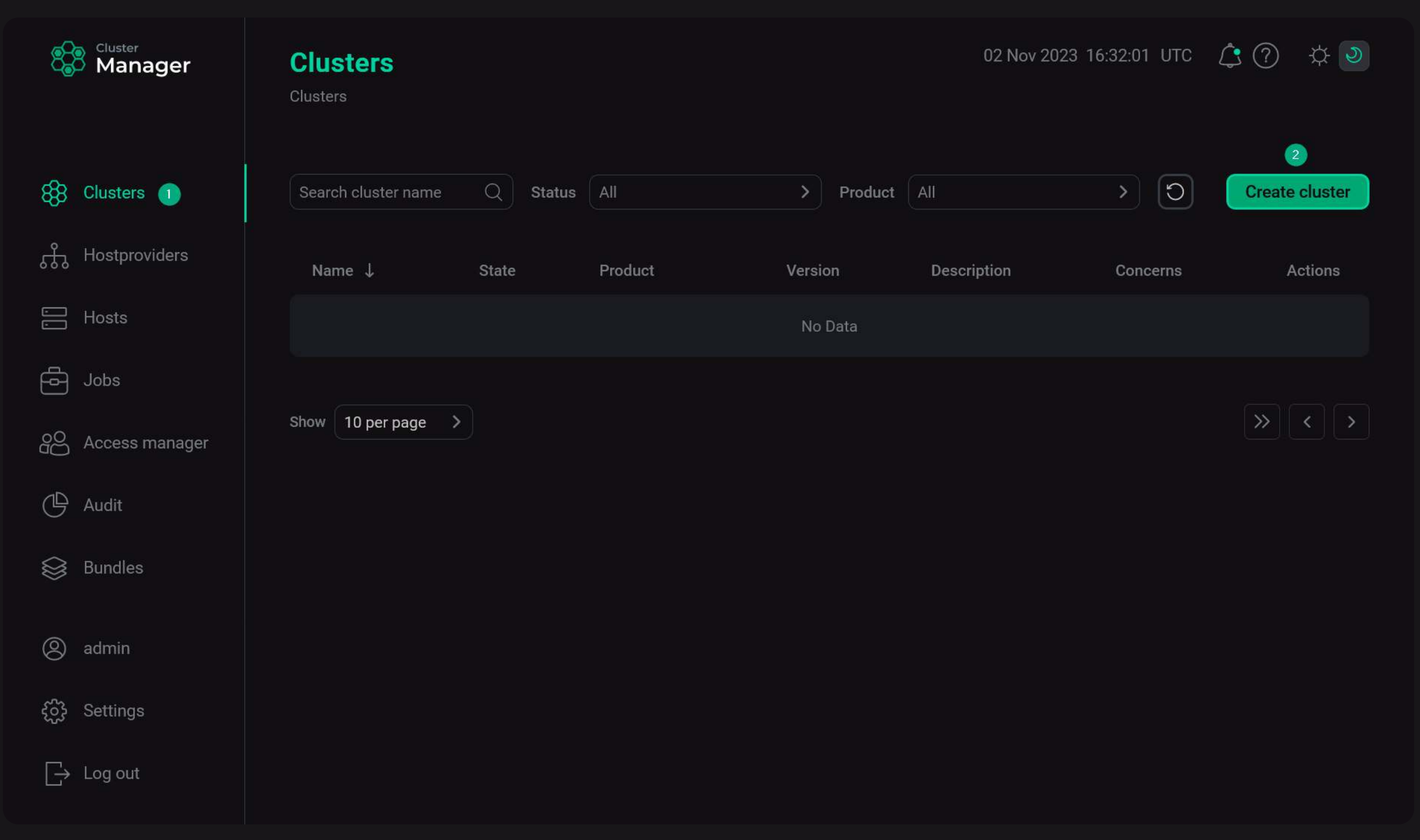
Step 3. Create a cluster based on the uploaded bundle

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.

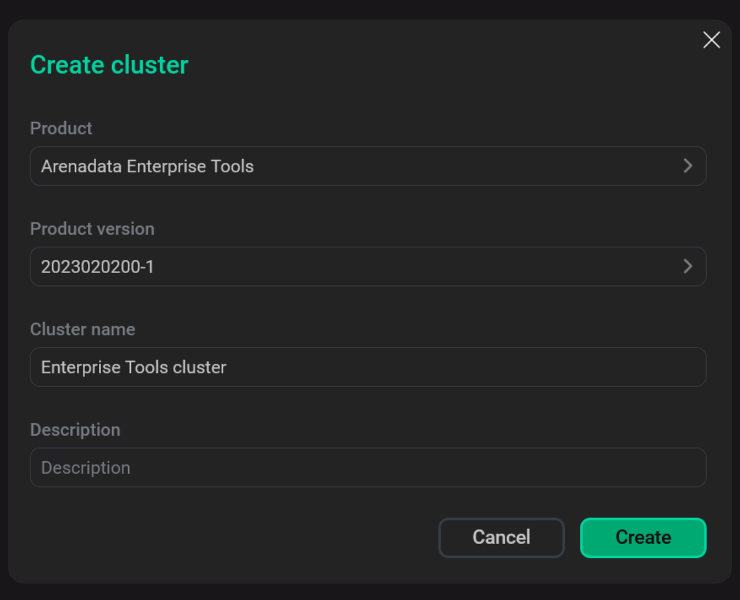
The steps for cluster creation are given below:

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



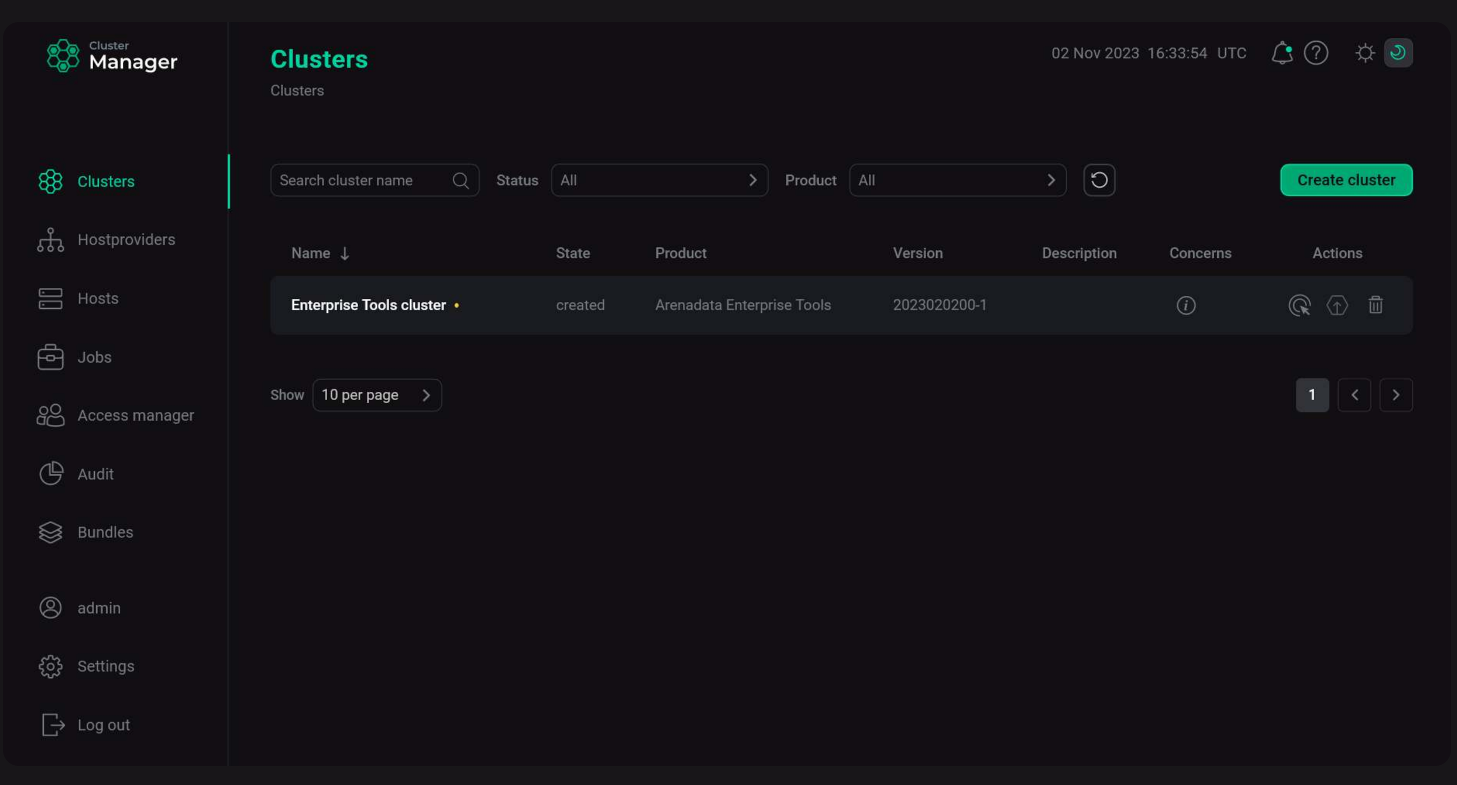
Switch to cluster creation

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

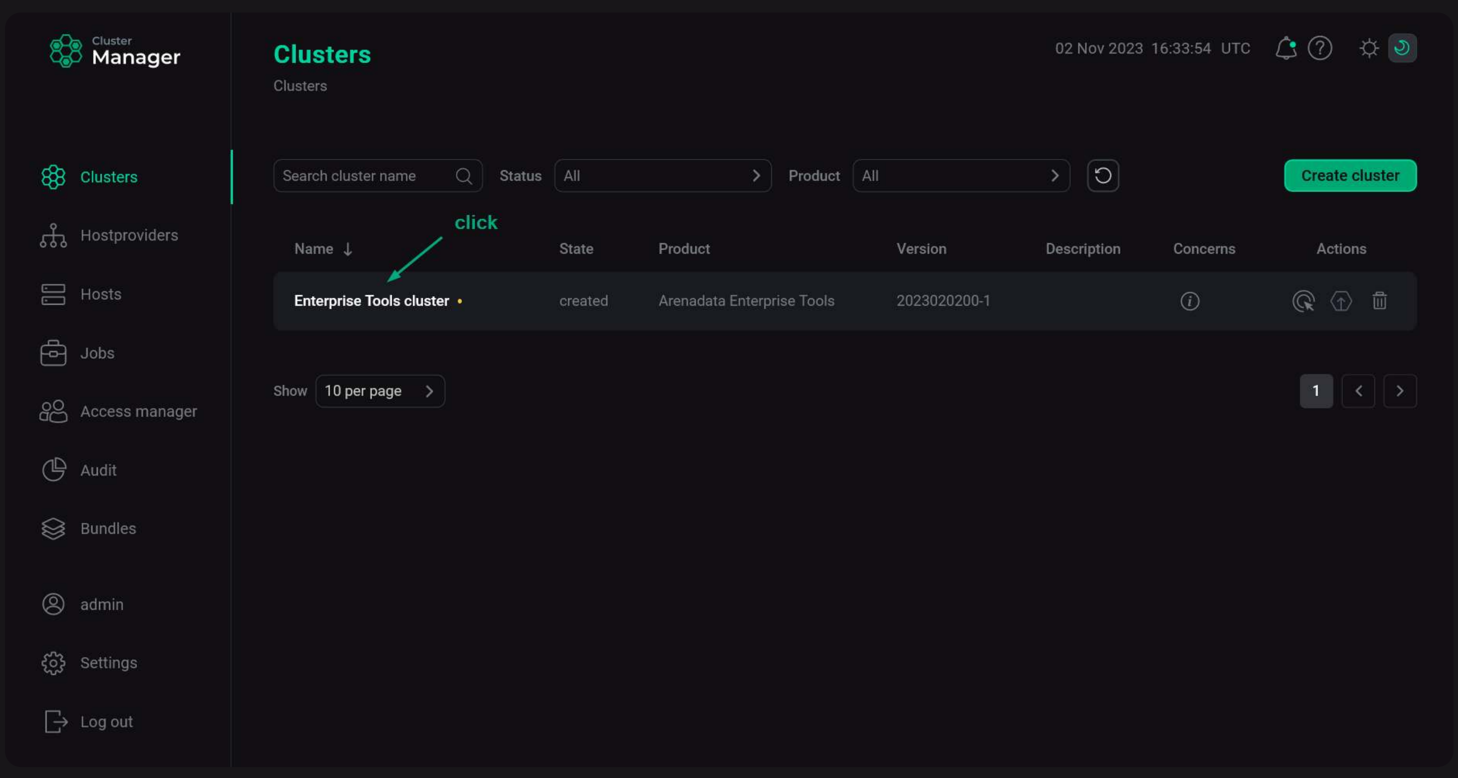
Add services

Daria Barysheva

To Table of Contents

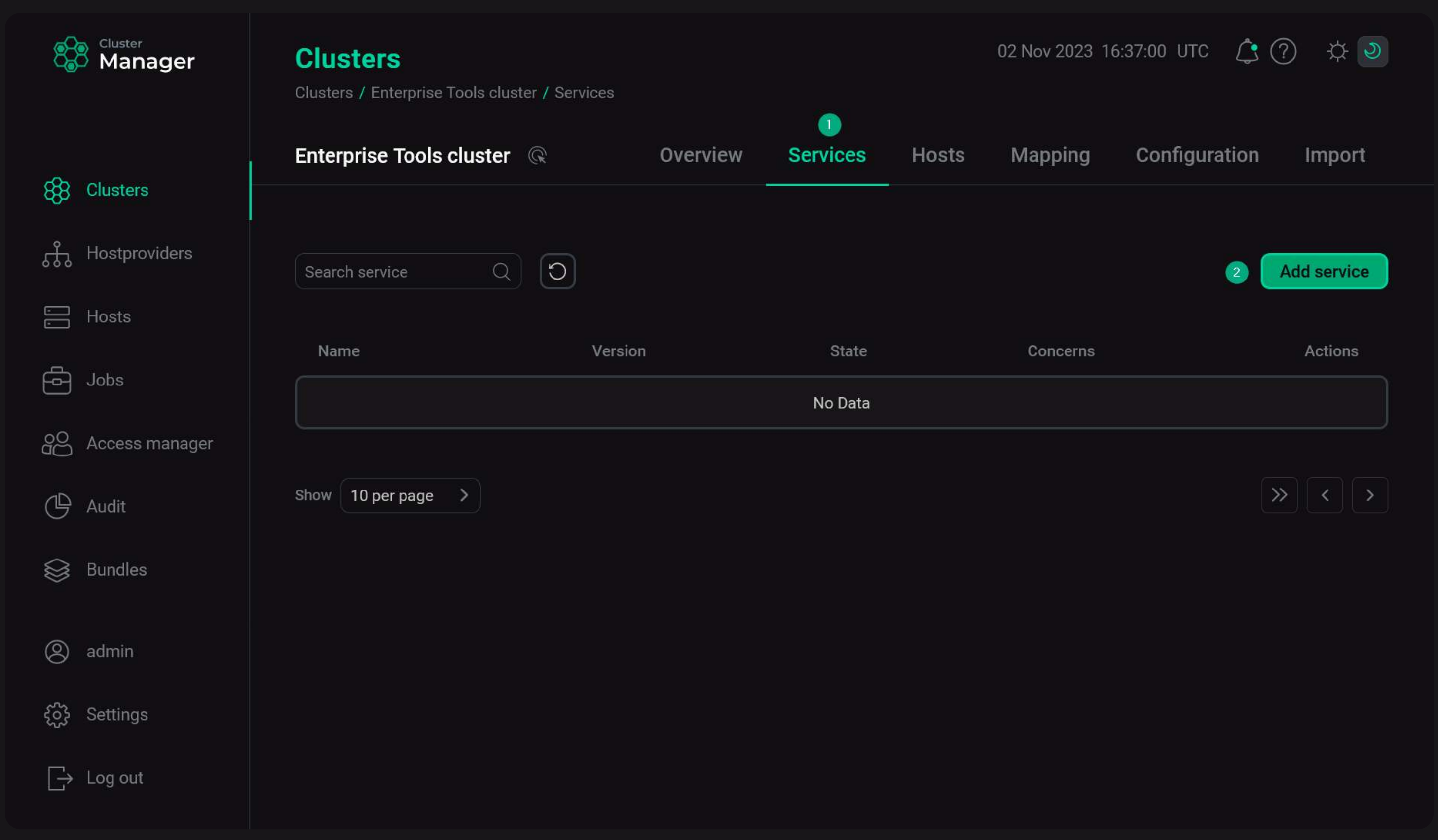
In ADCM, a **service** means a software that performs some function. The steps for adding services to a cluster are listed below:

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



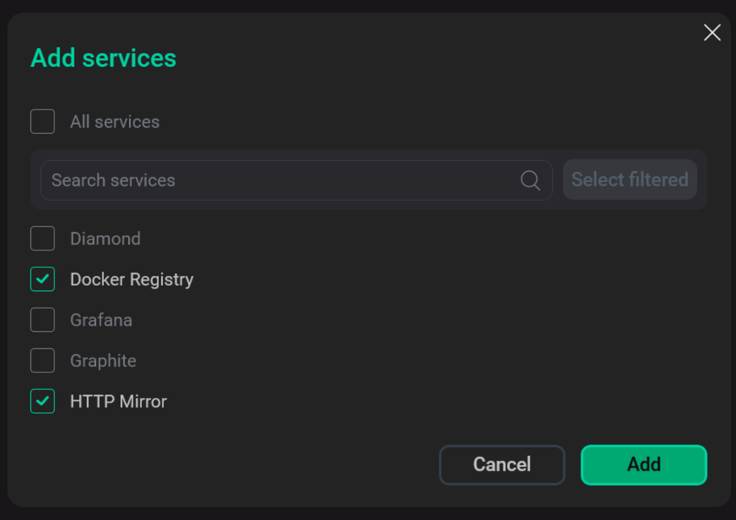
Select a cluster

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



Switch to adding services

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



Select services

The brief description of available services is listed below.

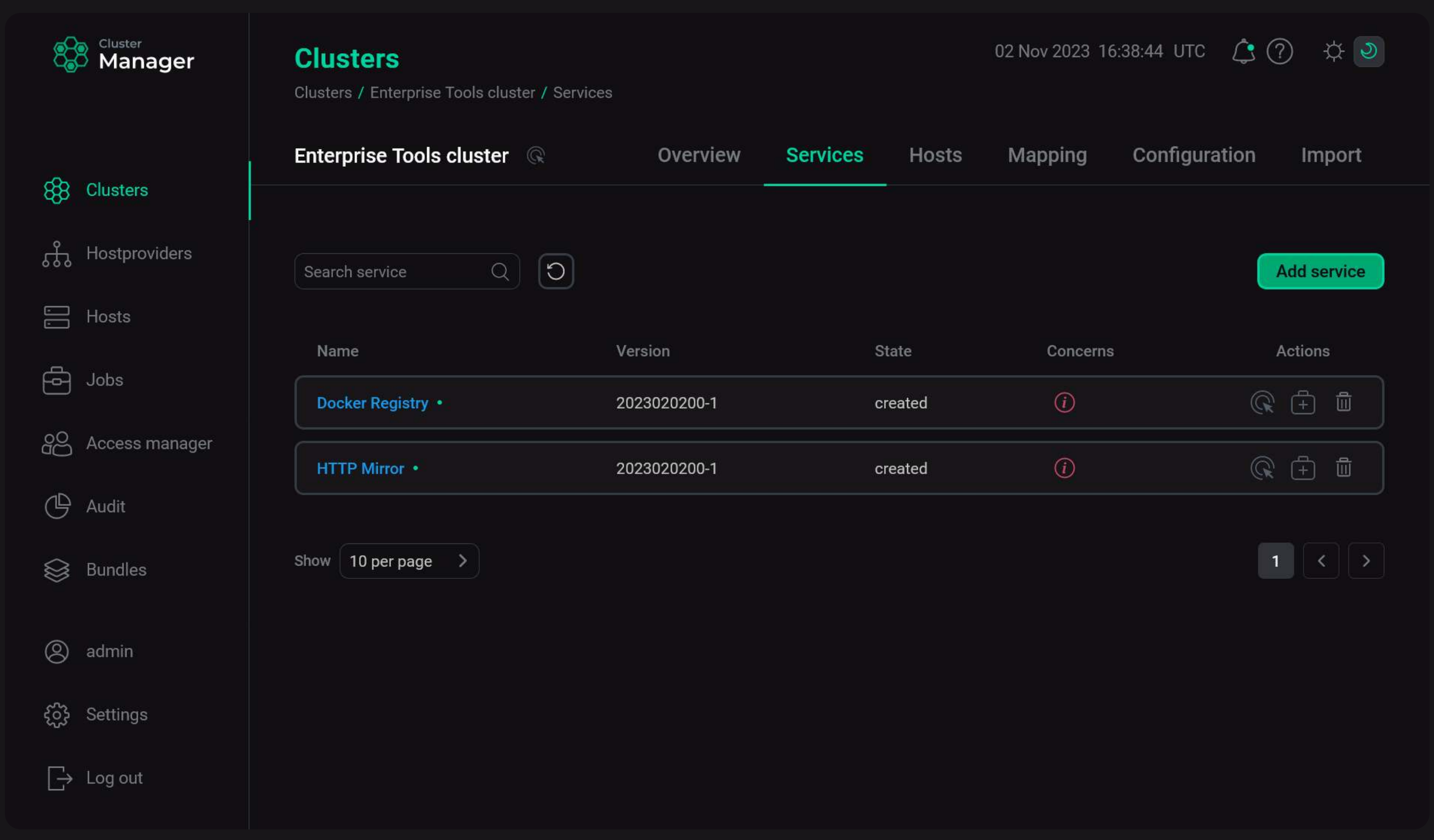
Services that should be added to the Enterprise Tools cluster

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

NOTE

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

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



The result of successful adding services to a cluster

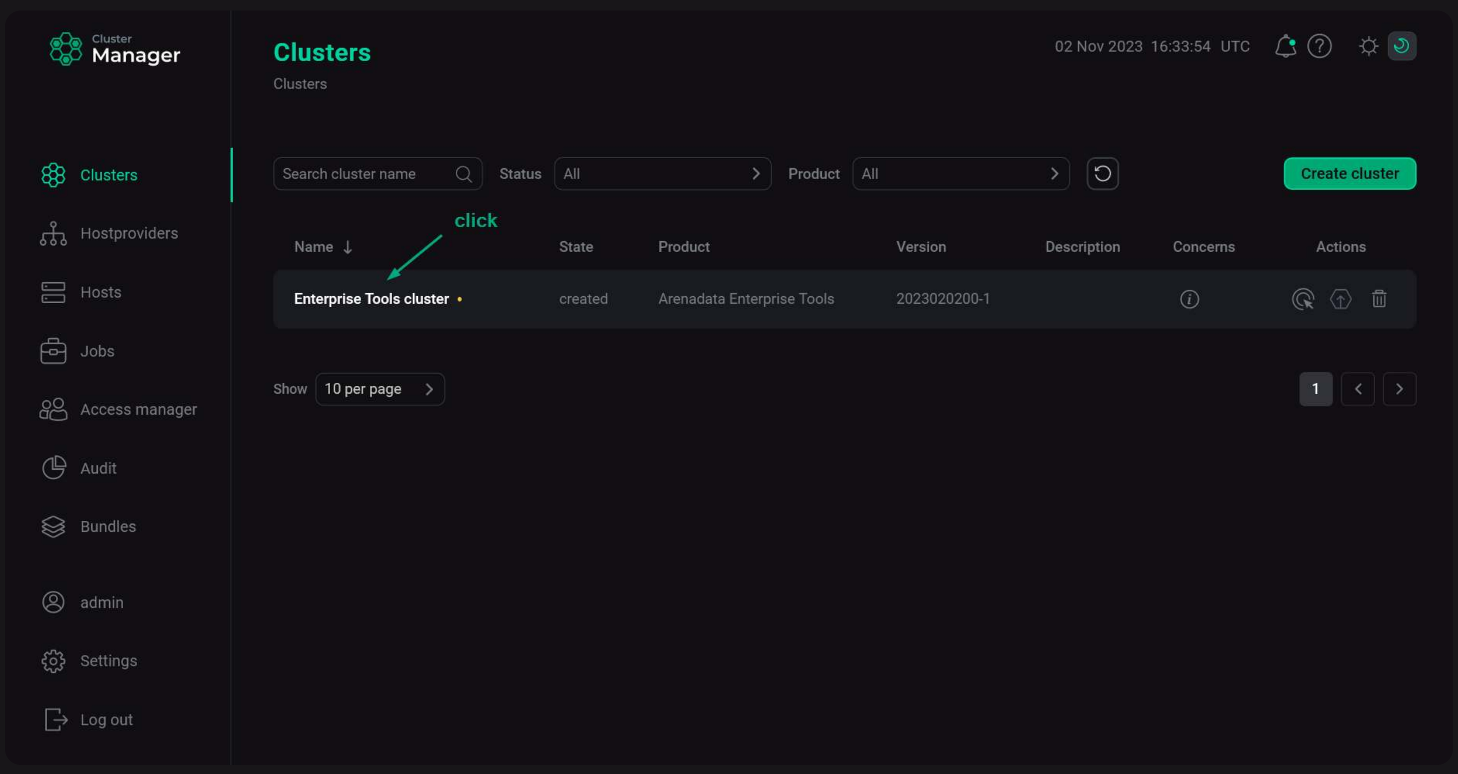
Add hosts to a cluster

Daria Barysheva

To Table of Contents

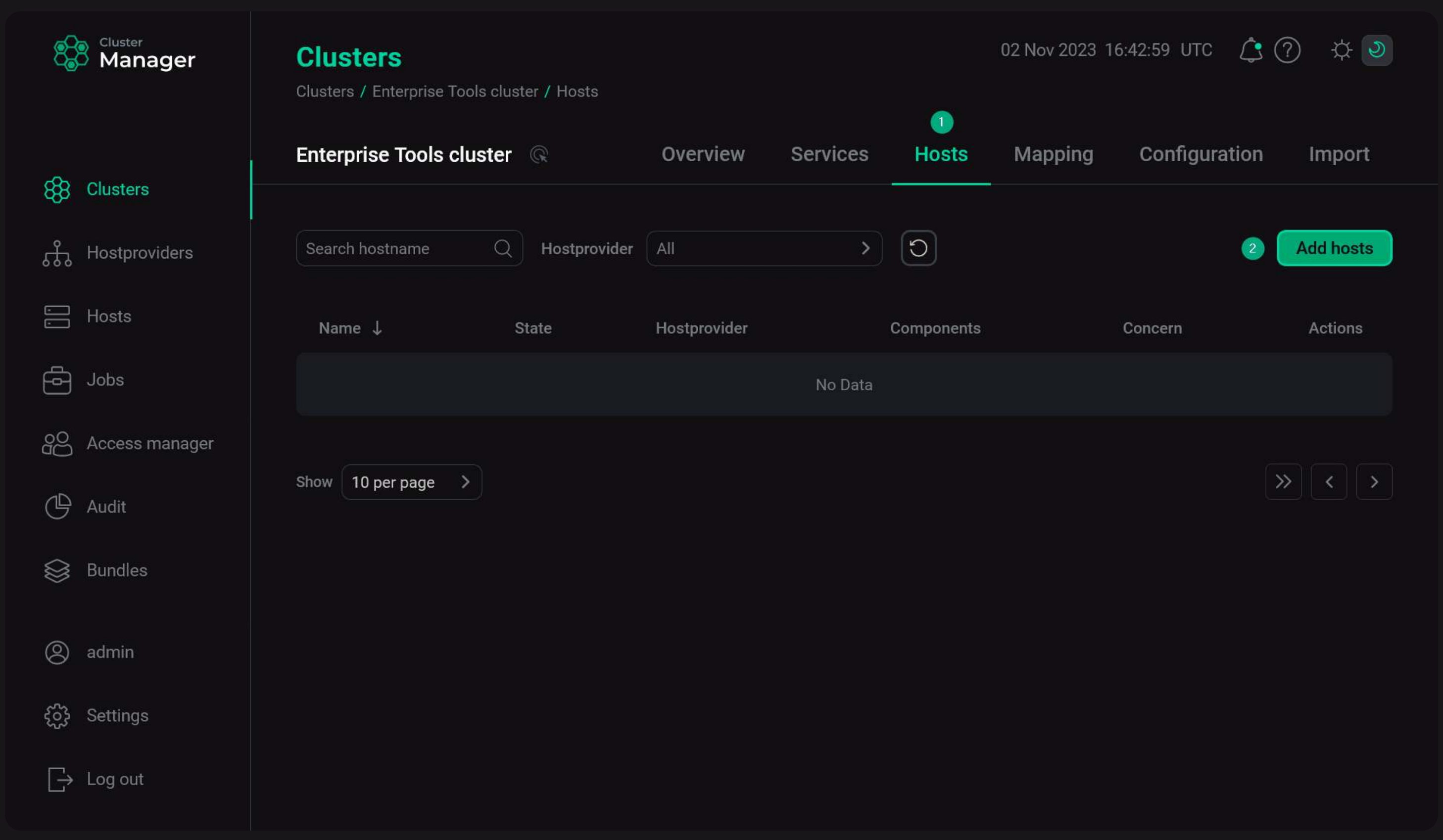
The steps for adding hosts to a cluster are listed below:

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



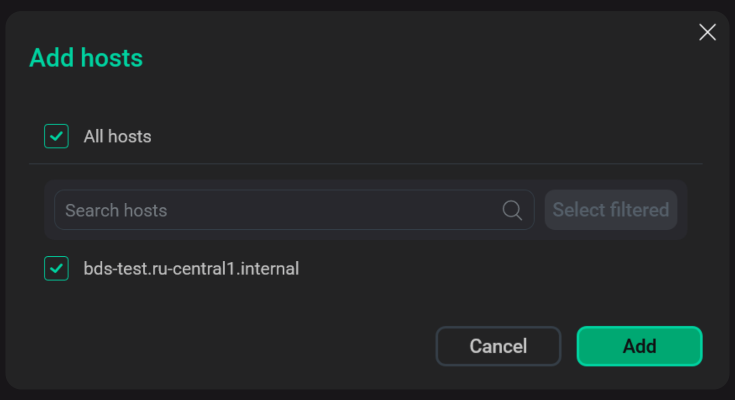
Select a cluster

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



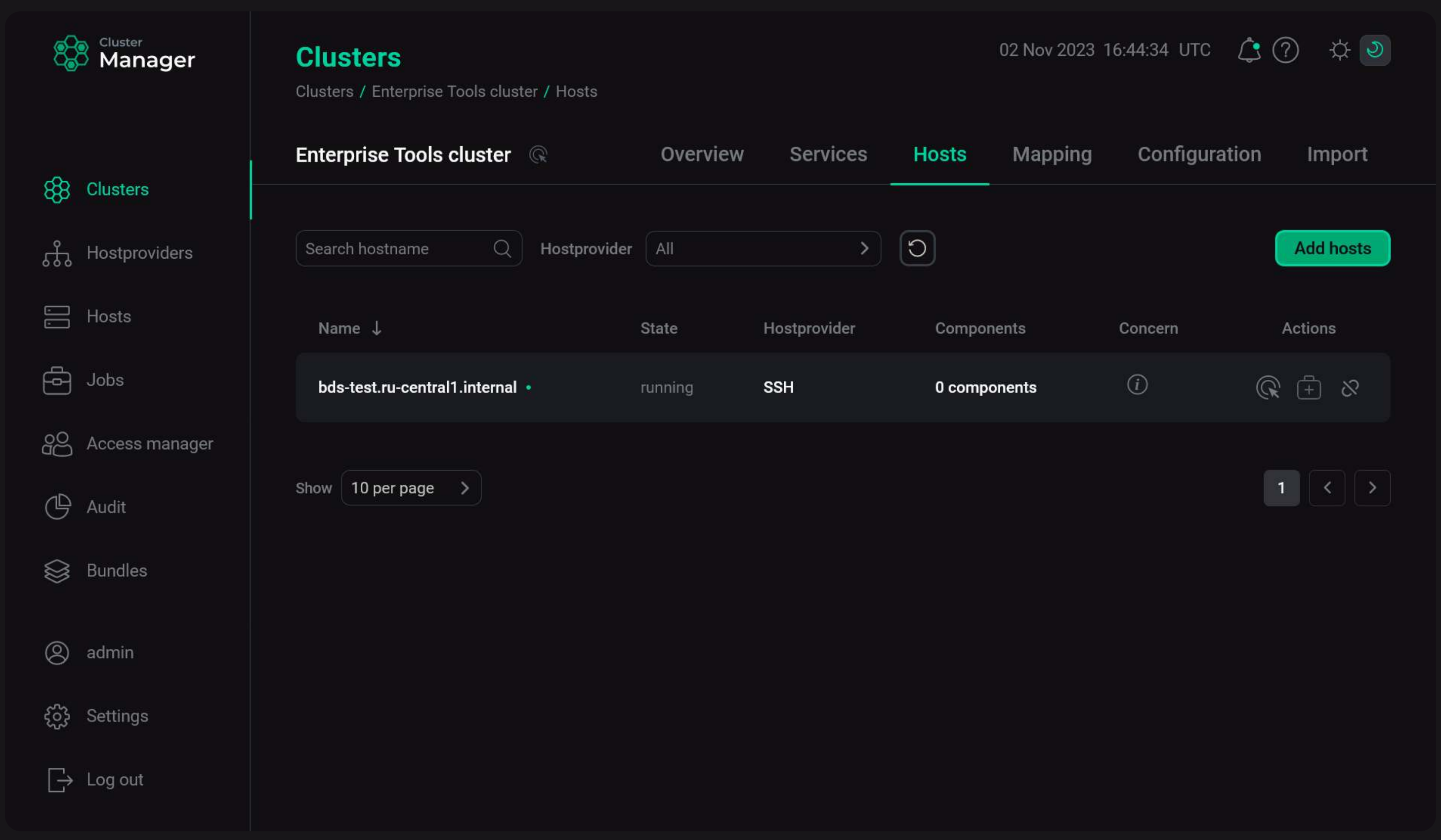
Switch to adding hosts

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



Select hosts

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



The result of successful adding hosts to a cluster

Add components

Daria Barysheva

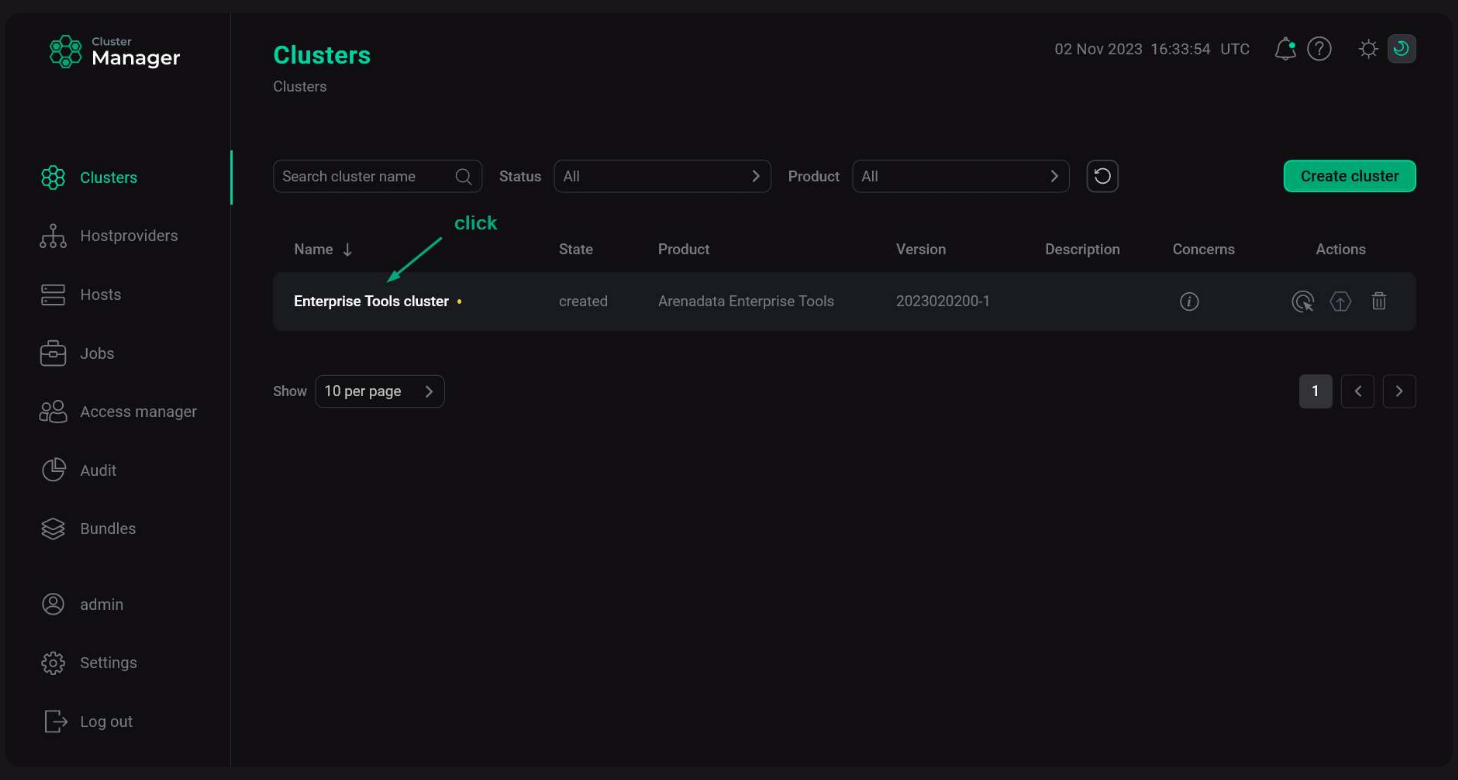
To Table of Contents

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

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

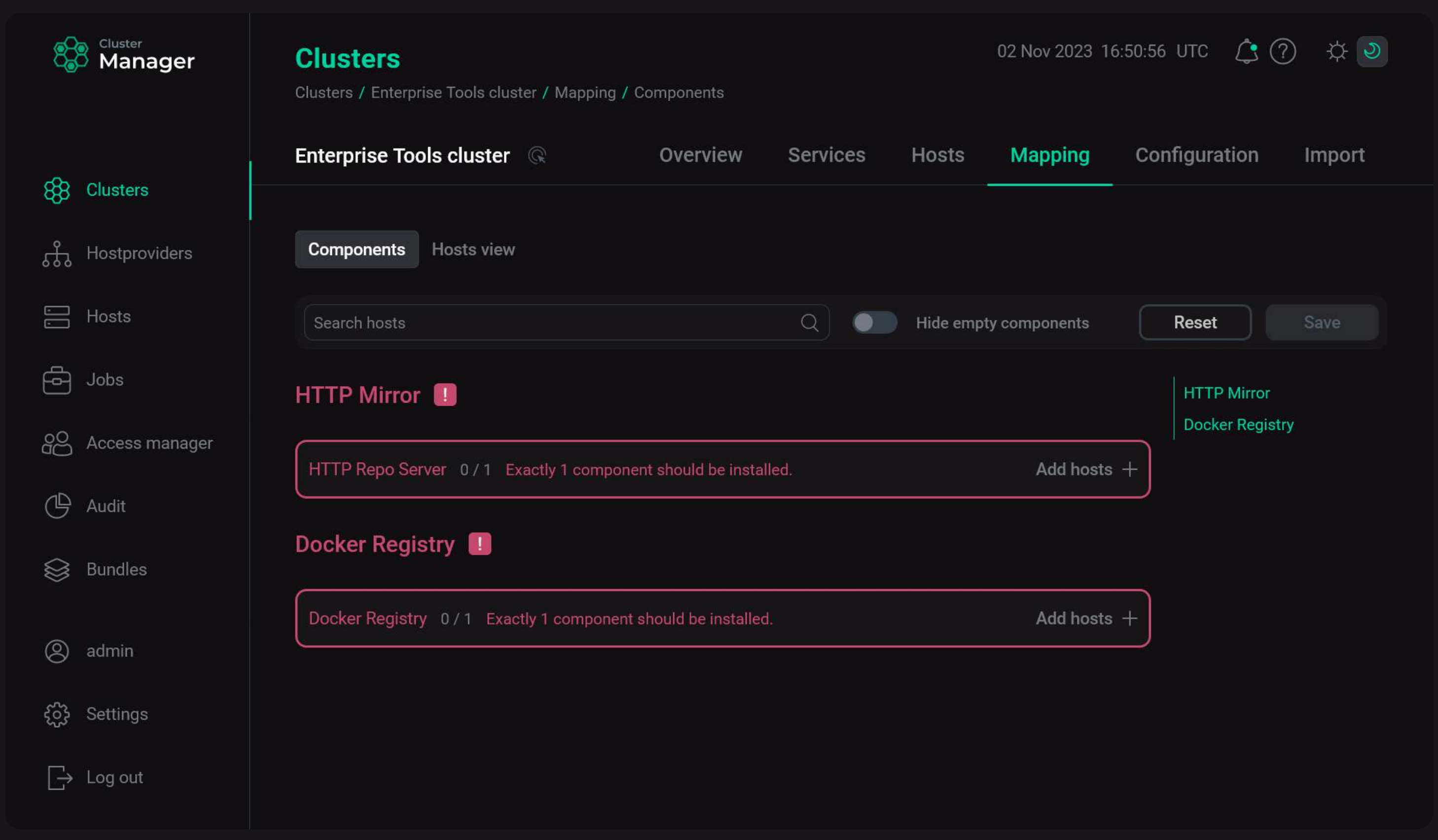
The steps for adding components to hosts are listed below:

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



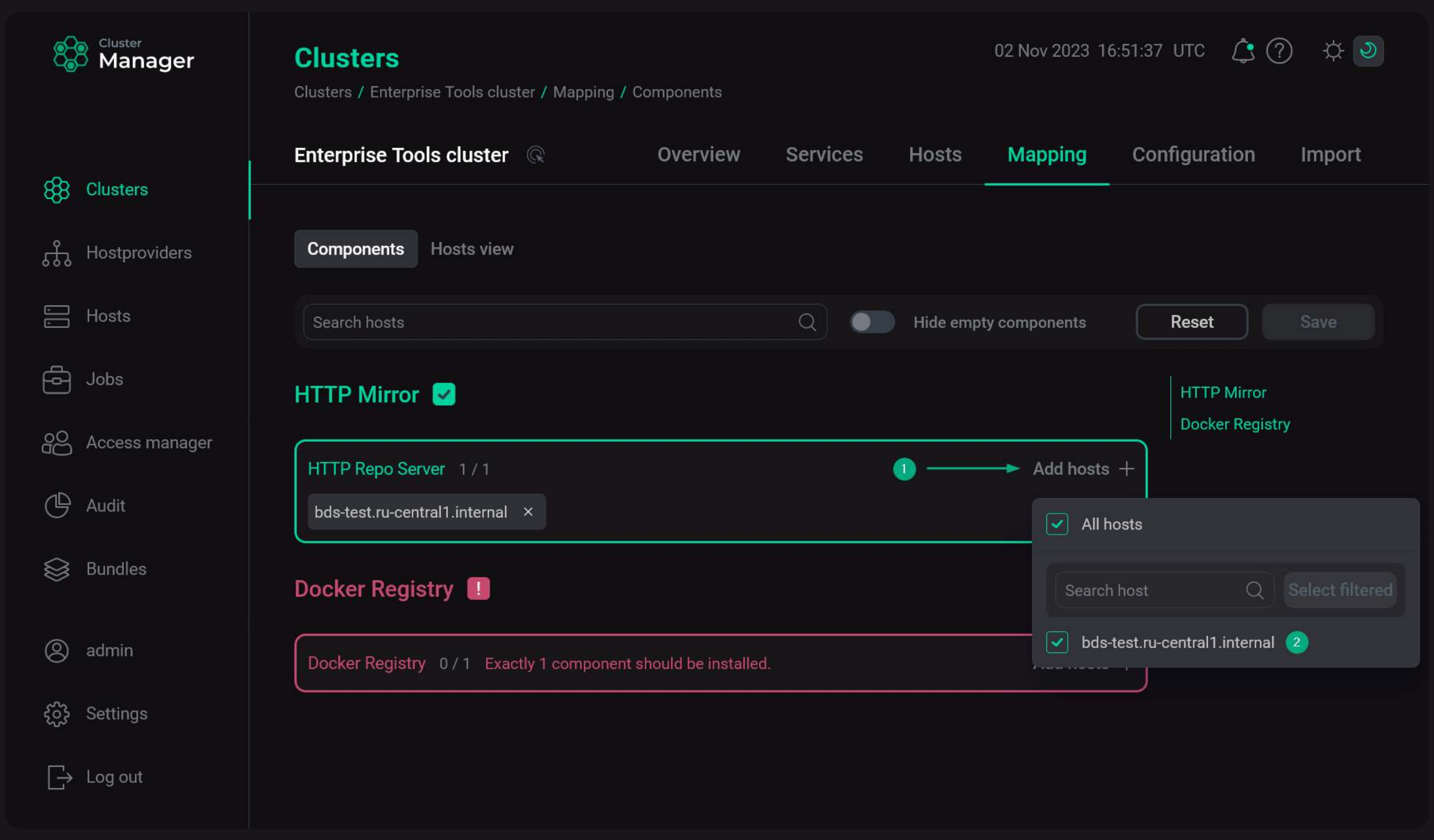
Select a cluster

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



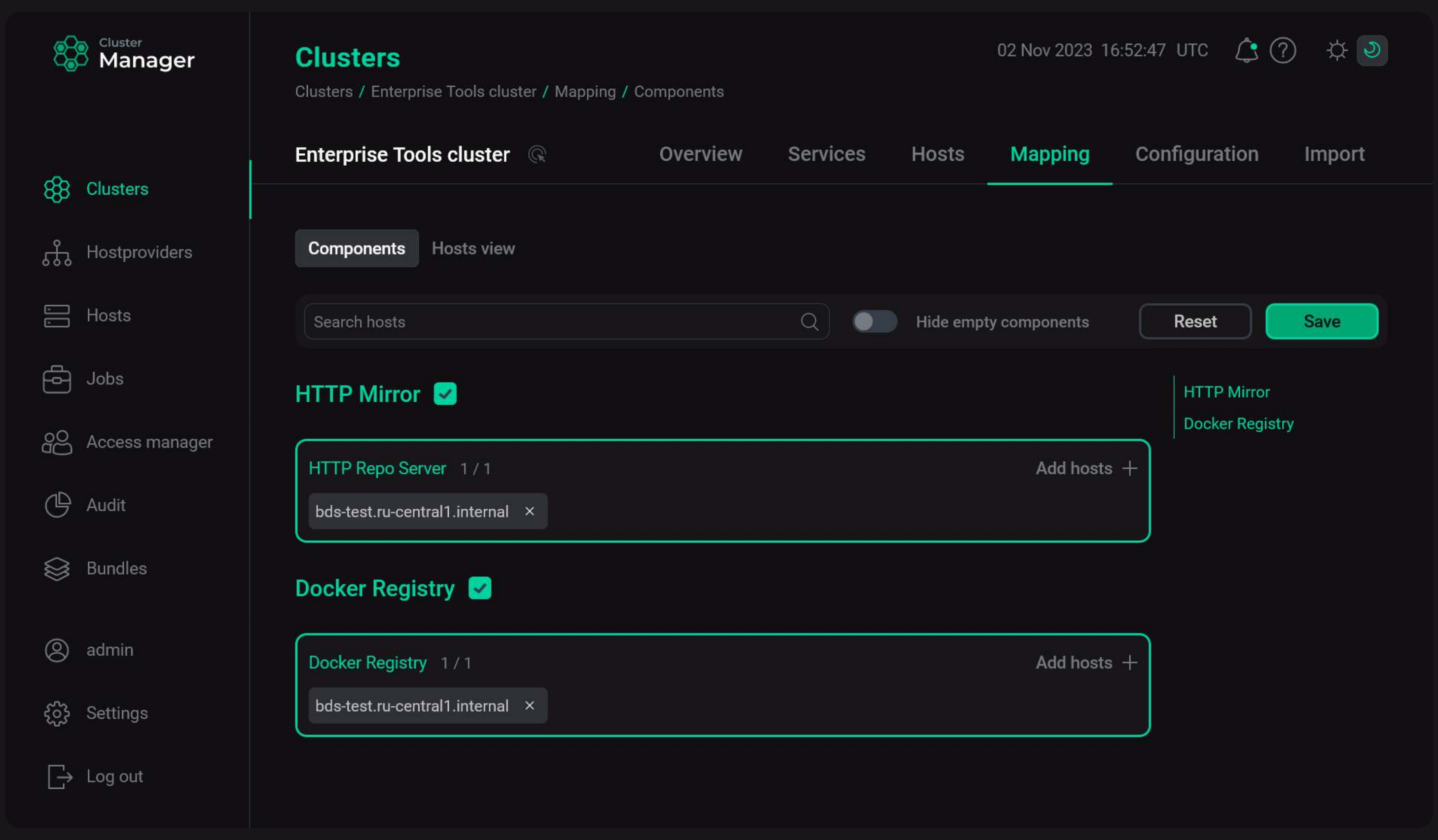
Switch to mapping service components

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



Select hosts for a component

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



Save mapping of components

Install a cluster

Daria Barysheva

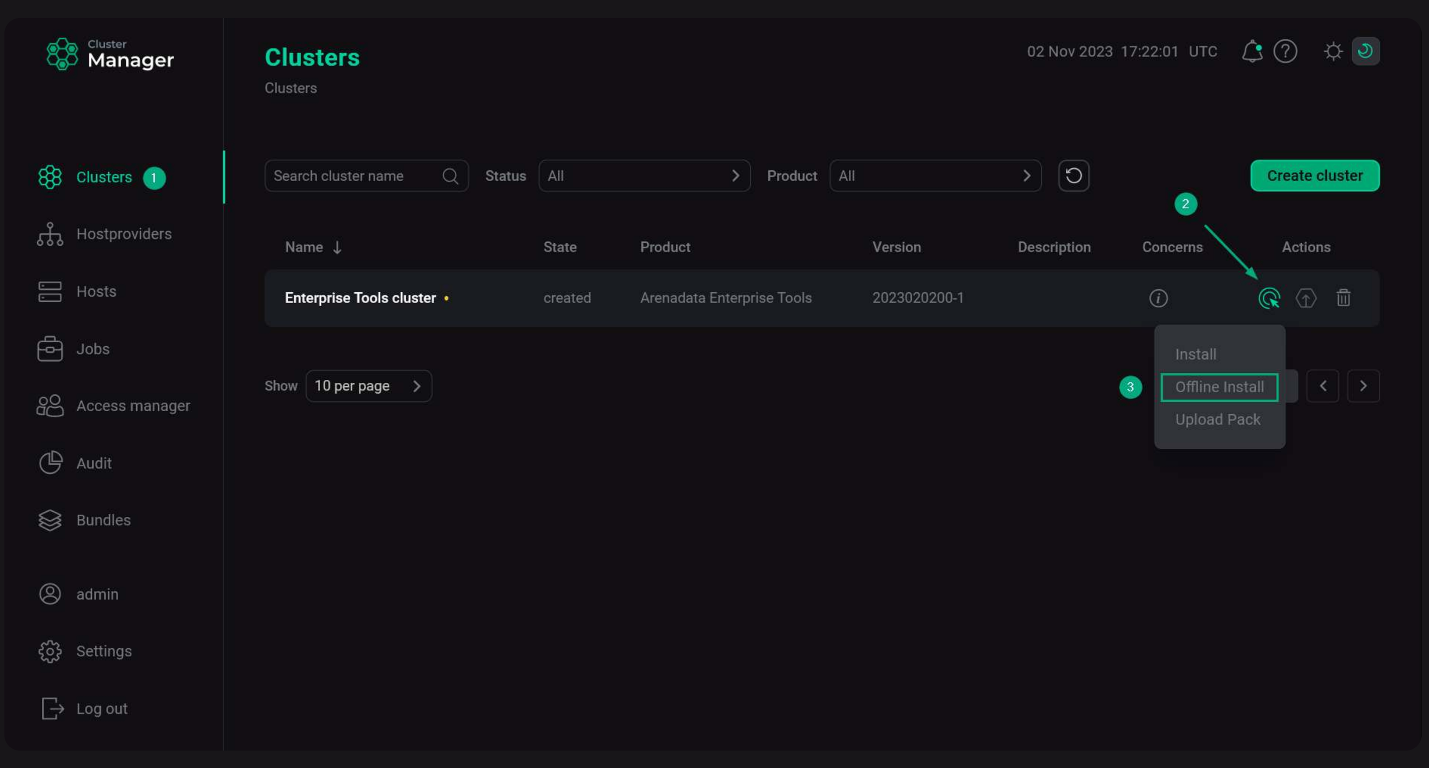
Contents

To Table of Contents

Step 1. Run installation

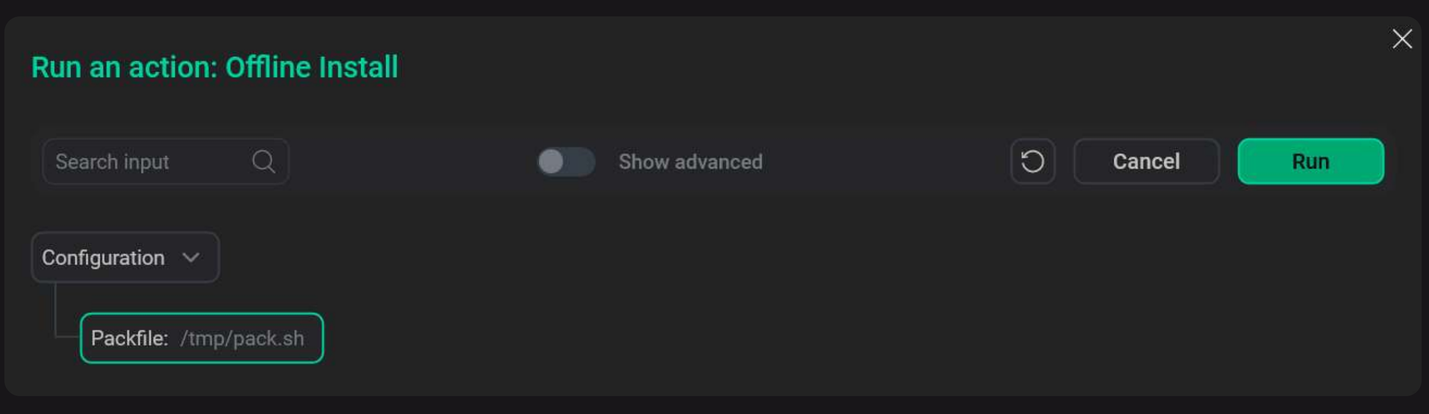
The last step of adding a new Enterprise Tools cluster is to install its services. To do this:

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



Switch to cluster installation

2. In the opened window, enter a full path to the shell script that is received from the Arenadata support team and click ***Run***.

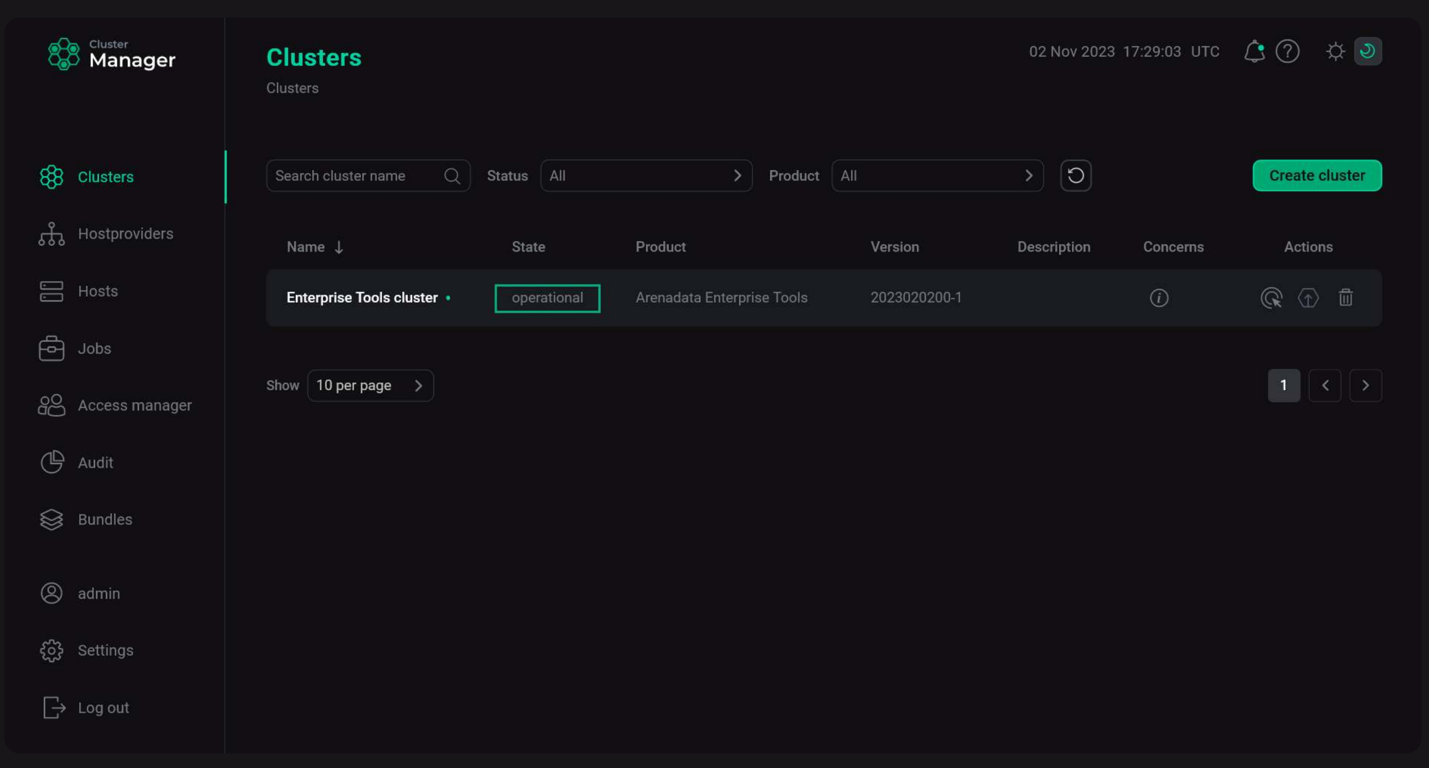


Verify installation

3. Wait until the installation process is completed.

NOTE

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

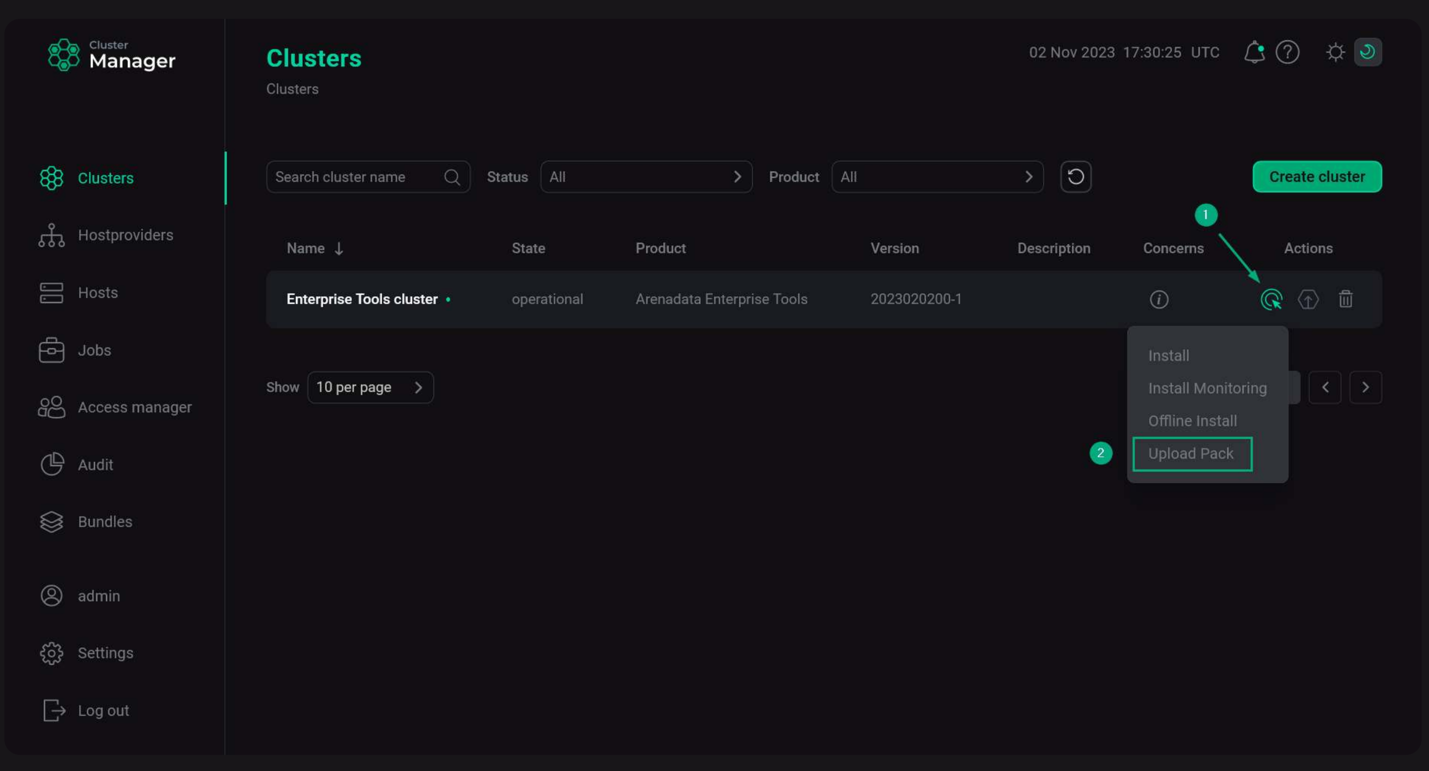


Cluster installation is completed

Step 2. Upload packages for offline installation

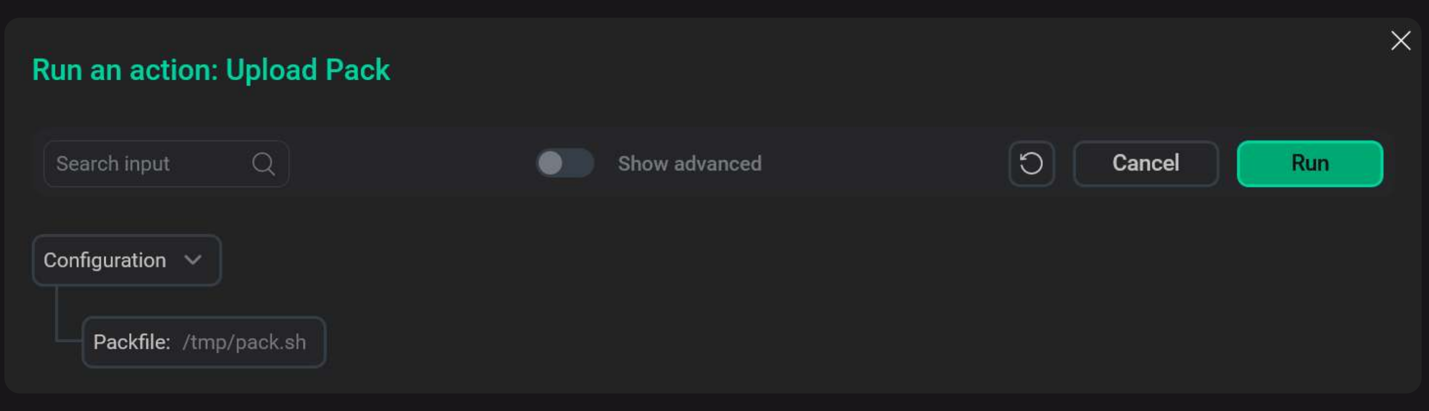
In addition to installing the cluster, you should process the shell script that is received from the Arenadata support team. It is necessary to enable offline installation of products in the future. To do this:

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



Switch to uploading packages

2. In the opened window, enter a full path to the shell script received from the Arenadata support team and click ***Run***.



Verify upload

3. Wait until the upload process is completed.

NOTE

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

Step 3. Verify results

The steps for checking the installation results are listed below:

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

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

The result should look like this:

```
tcp6       0      0 :::443                  :::*                    LISTEN      1694/docker-proxy-c
tcp6       0      0 :::81                   :::*                    LISTEN      1708/docker-proxy-c
```

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

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

The following example displays the data that can be placed in this folder for ADB ES installation.

```
total 32
drwxr-xr-x 8 nobody root 4096 Mar 17 12:33 .
drwxr-xr-x 3 nobody root 4096 Mar 17 12:32 ..
drwxr-xr-x 3 nobody root 4096 Mar 17 12:32 ADBM
drwxr-xr-x 3 nobody root 4096 Mar 17 12:33 ADCC
drwxr-xr-x 3 nobody root 4096 Mar 17 12:32 ADMPROM
drwxr-xr-x 3 nobody root 4096 Mar 17 12:32 ADPG-CORE
drwxr-xr-x 3 nobody root 4096 Mar 17 12:32 ADQM
drwxr-xr-x 3 nobody root 4096 Mar 17 12:32 eureka
```

Create a cluster

Anton Vasilev

Contents

To Table of Contents

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.

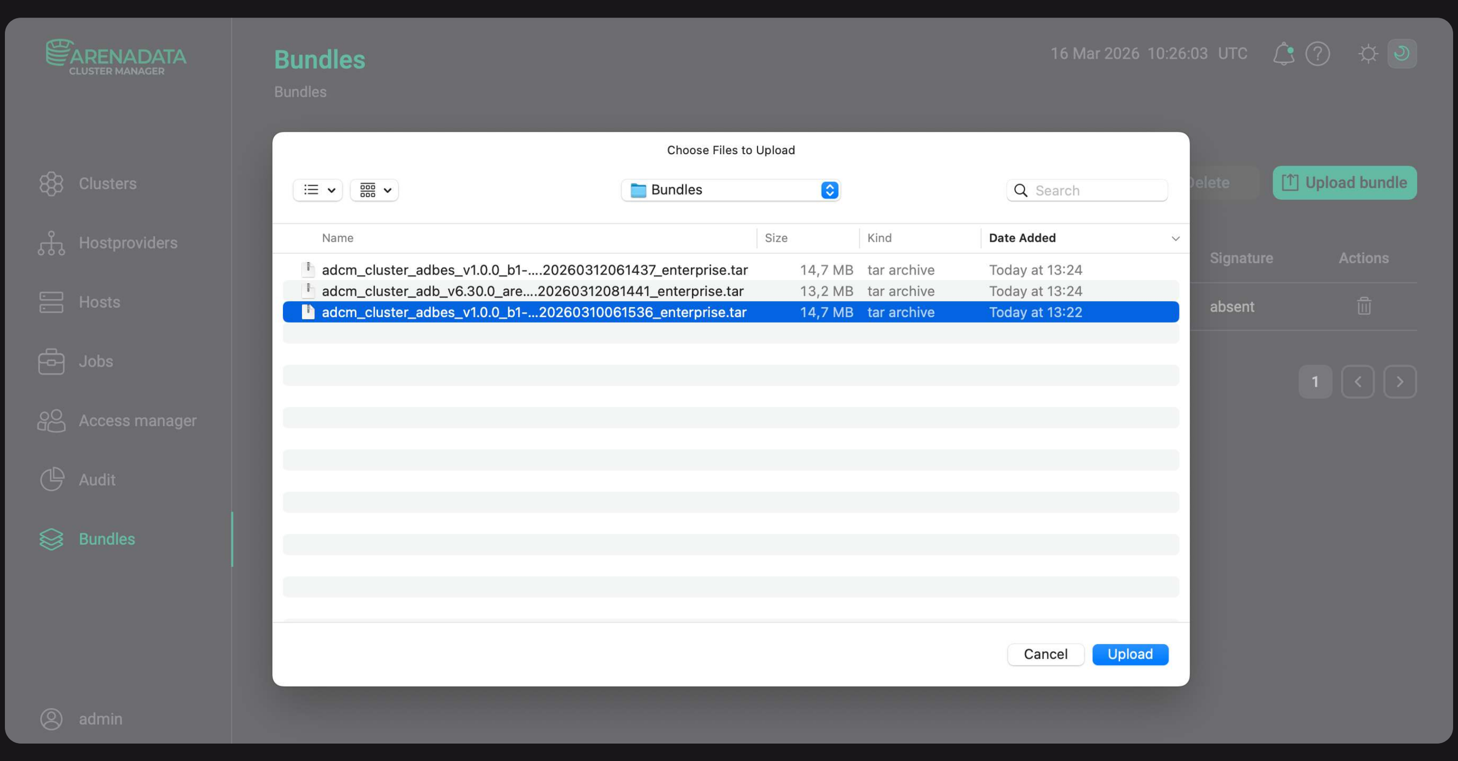
Step 1. Get the Enterprise version of a cluster bundle

Please get a cluster bundle from the Arenadata support team.

Step 2. Upload a cluster bundle to ADCM

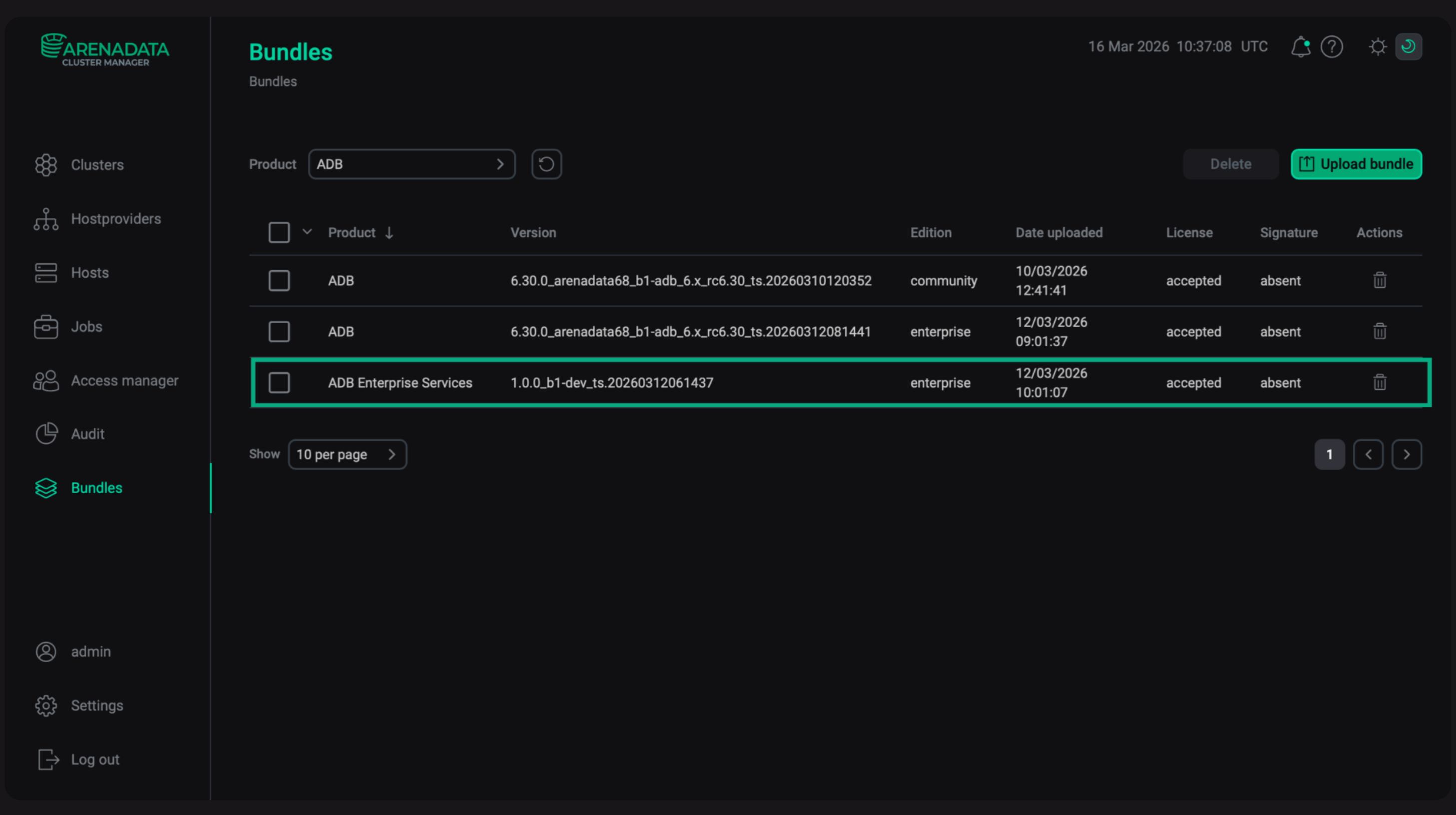
To upload a bundle to ADCM, follow the steps:

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

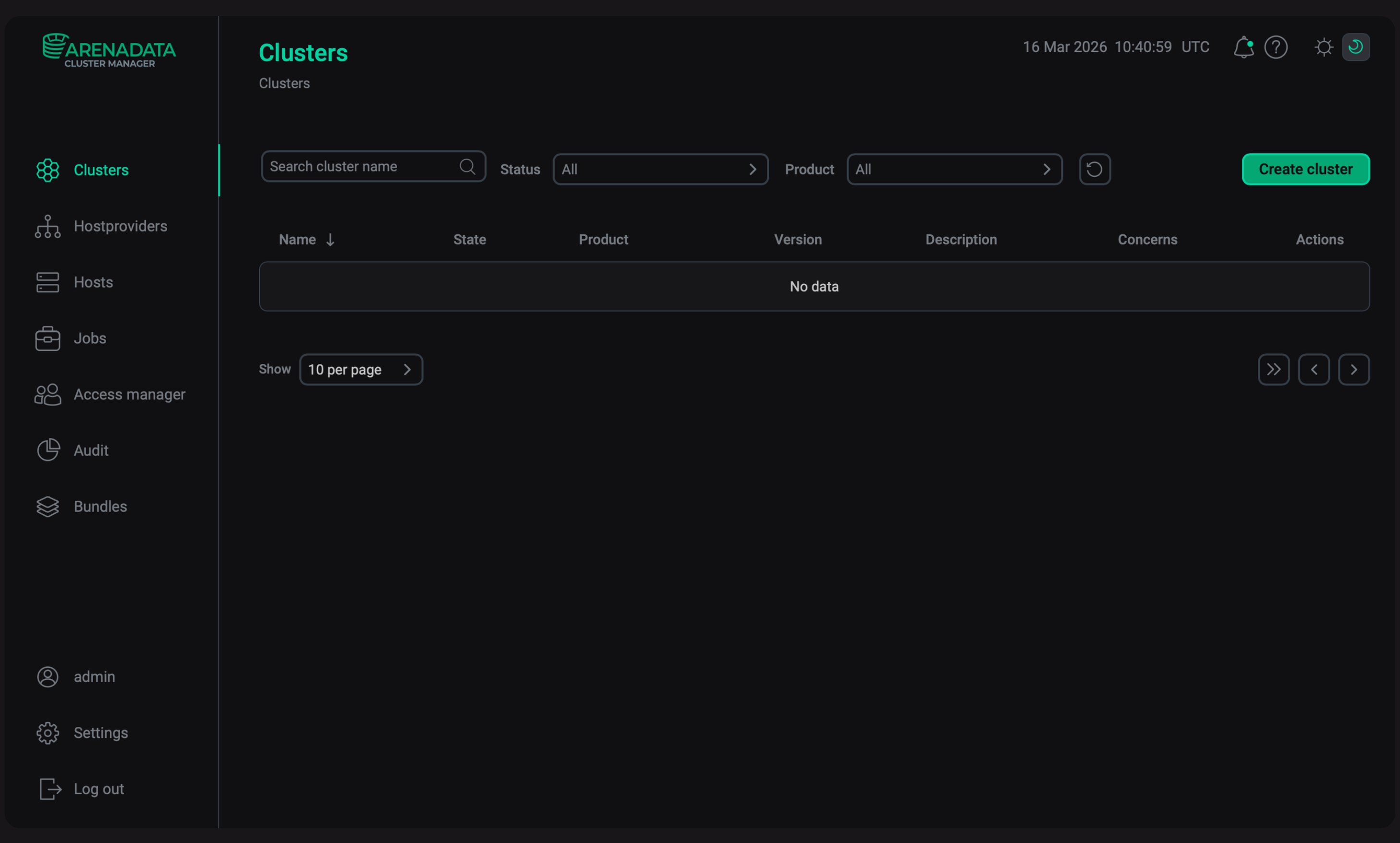
Step 3. Create a cluster based on the uploaded bundle

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.

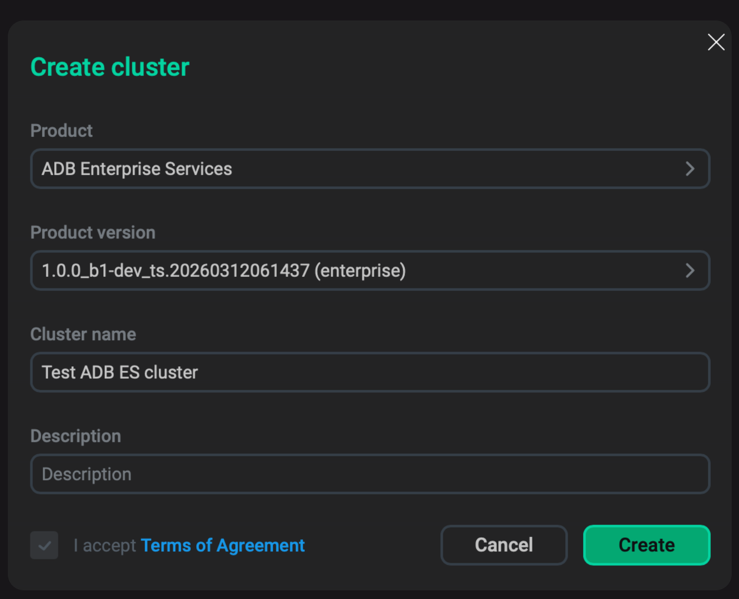
The steps for cluster creation are given below:

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



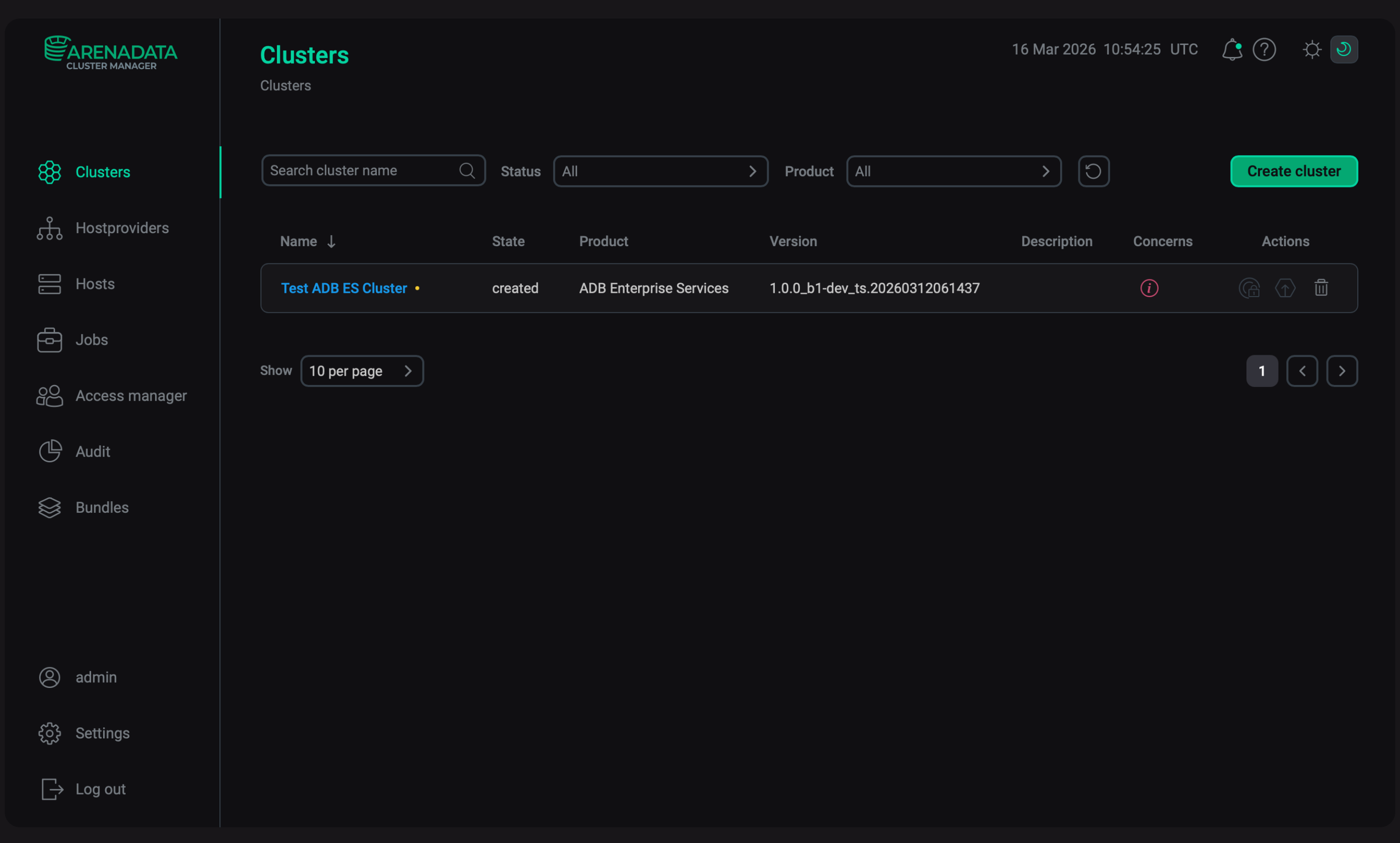
Switch to cluster creation

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

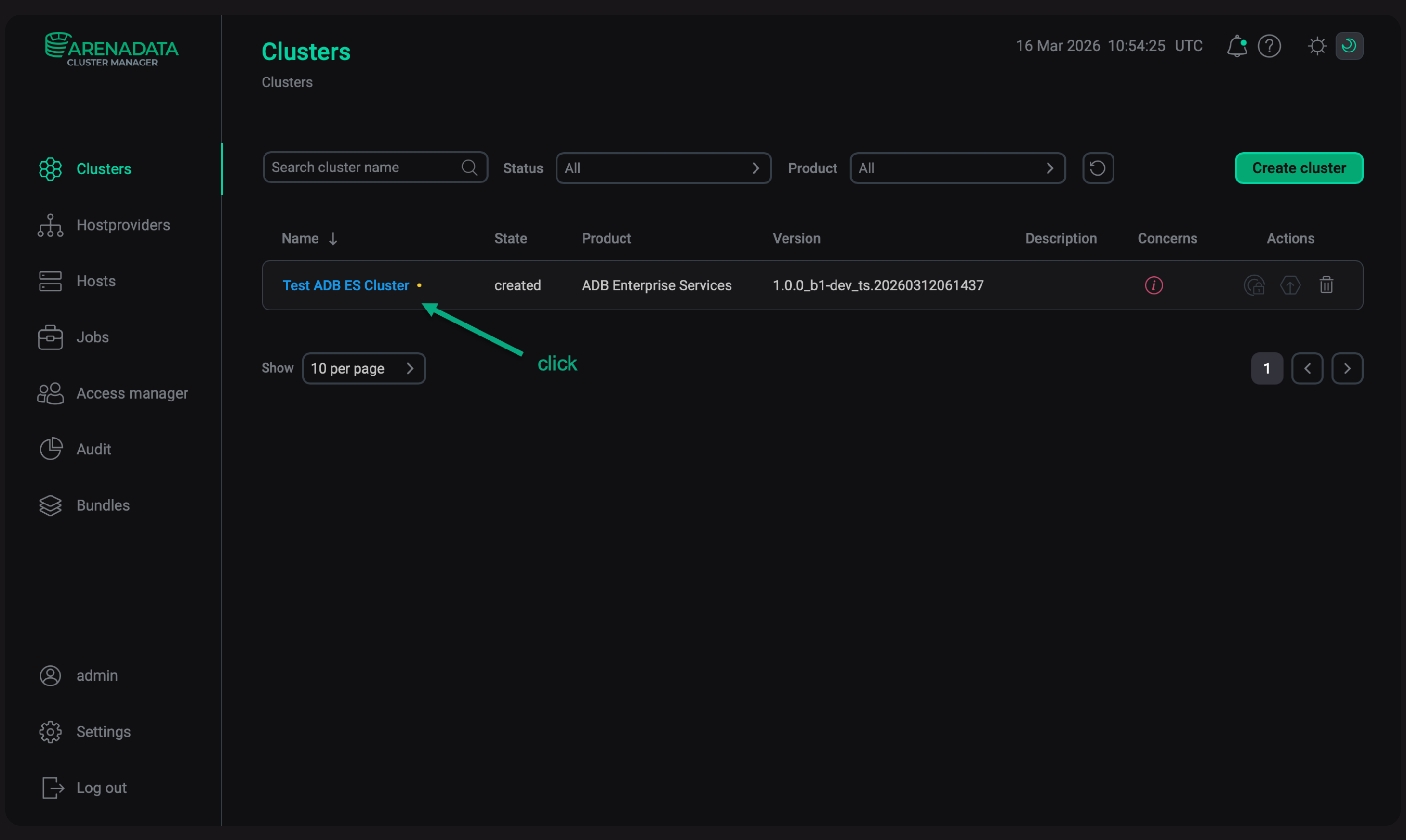
Add services

Anton Vasilev

To Table of Contents

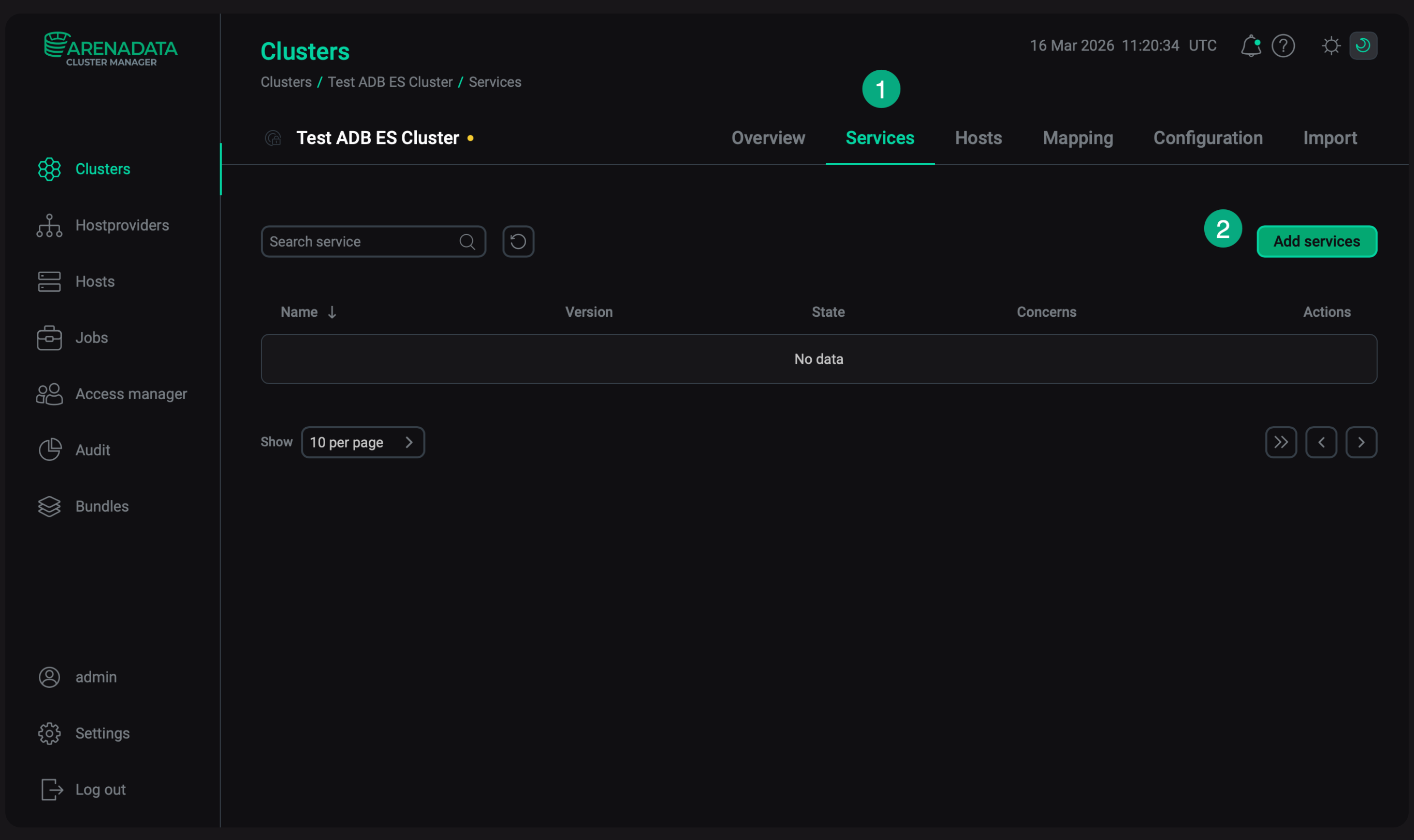
In ADCM, a **service** means a software that performs some function. Examples of services for ADB ES clusters: ADBM and ADB Control. The steps for adding services to a cluster are listed below:

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



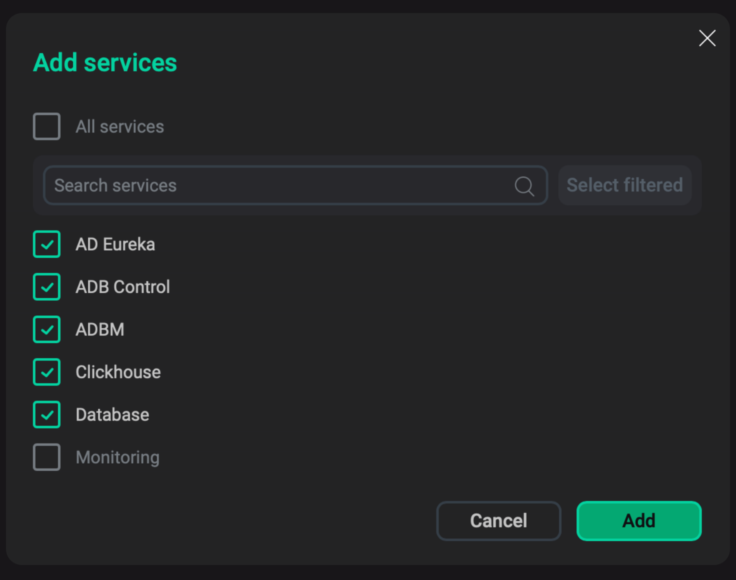
Select a cluster

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



Switch to adding services

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



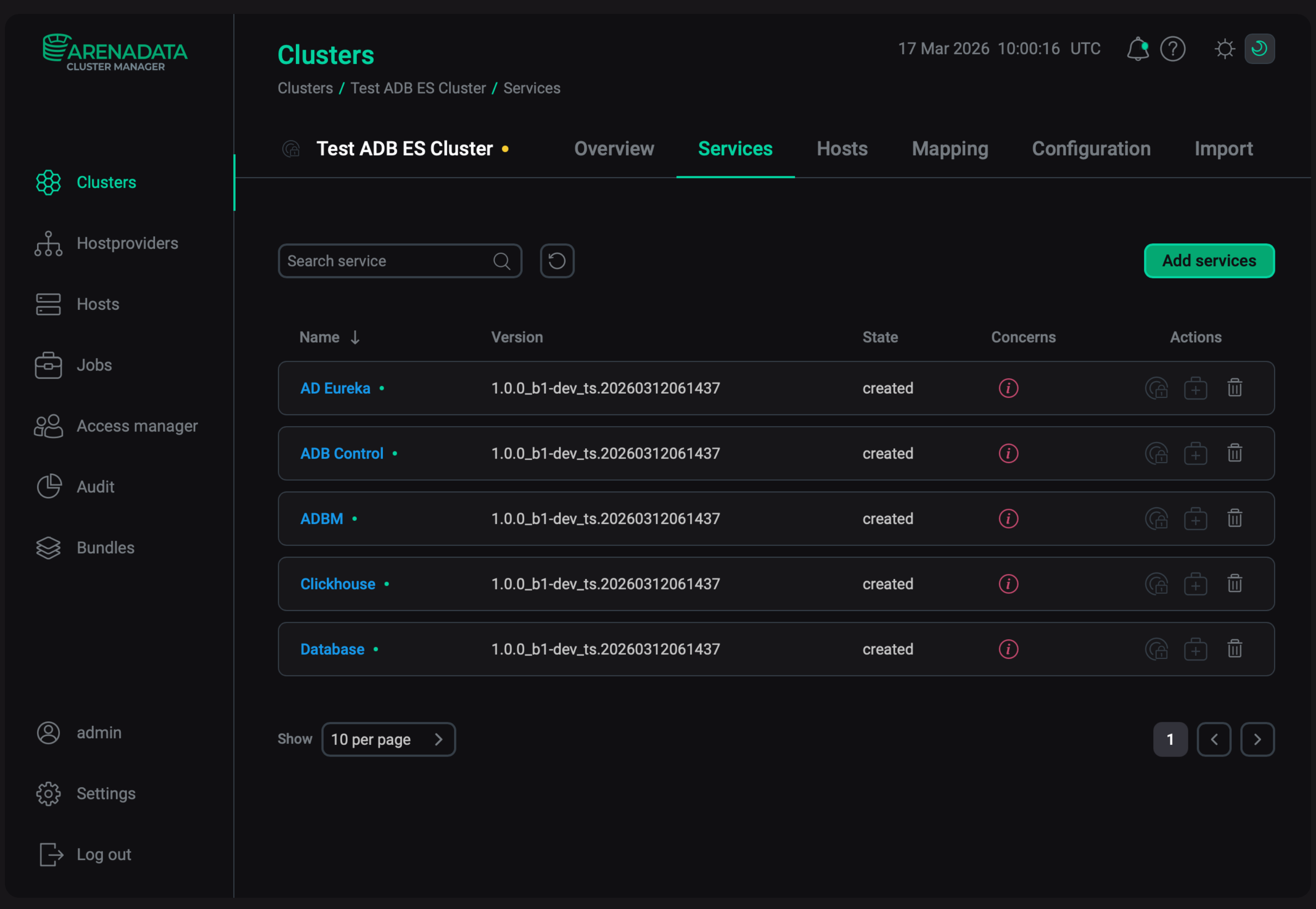
Select services

The brief description of available services is listed below.

Services that can be added to an ADB ES cluster

Service	Mandatory	Purpose
AD Eureka	Yes	Service registry, responsible for service discovery in ADB Control and ADBM
Clickhouse	Yes	ClickHouse DBMS used in ADB Control. By default, Arenadata QuickMarts (ADQM) is installed
Database	Yes	PostgreSQL used by ADB Control and ADBM. By default, Arenadata Postgres (ADPG) is installed
ADB Control	Yes	Arenadata DB Control (ADB Control) — a service for monitoring, diagnostics, and management of queries in ADB clusters.
ADBM	No	Arenadata DB Backup Manager (ADBM) — a service for ADB binary backup management
Monitoring	No	A service that monitors the cluster hosts and the ADB ES services (listed above). Steps for installing the Monitoring service are described in a separate section

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



The result of successful adding services to a cluster

NOTE

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

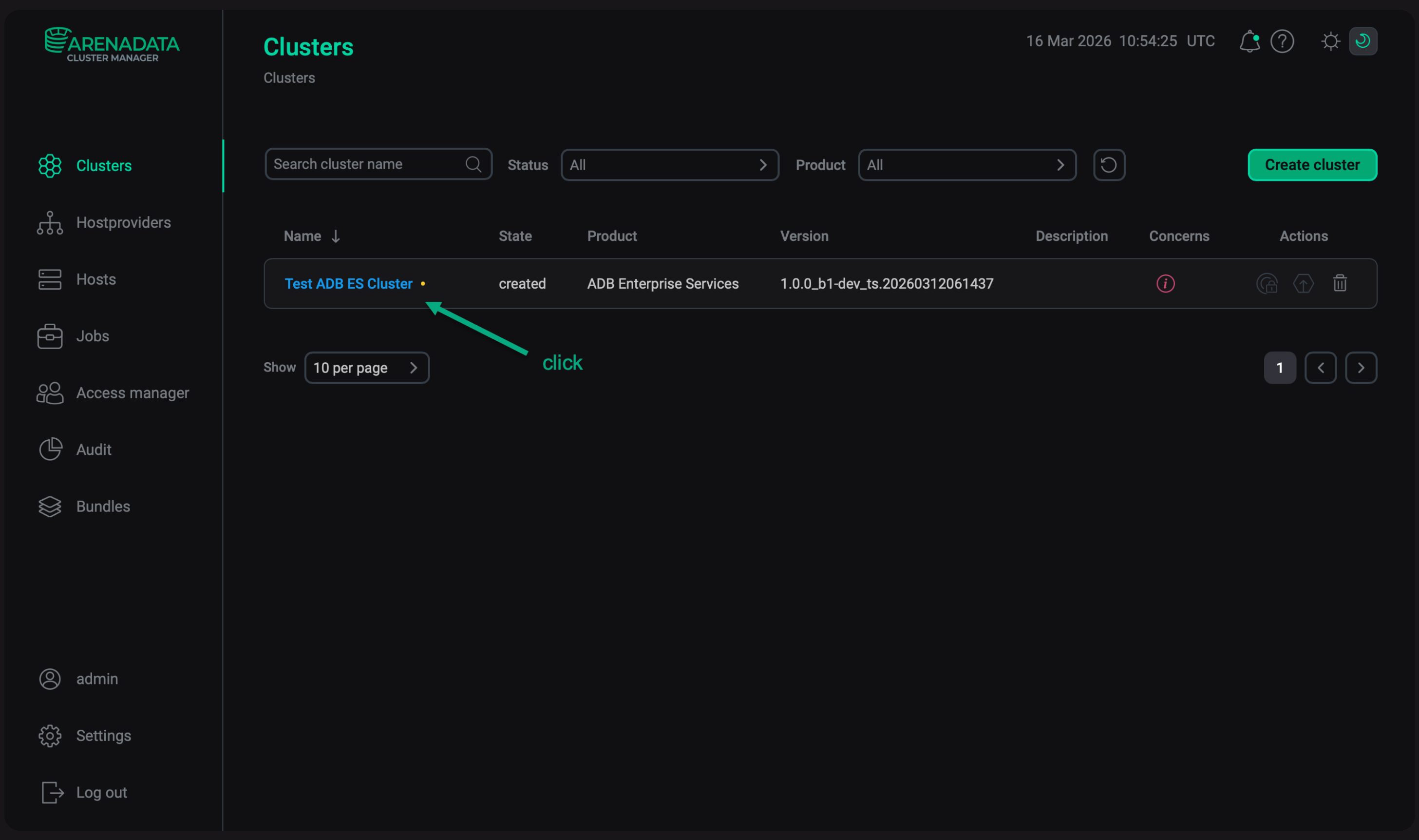
Add hosts to a cluster

Anton Vasilev

To Table of Contents

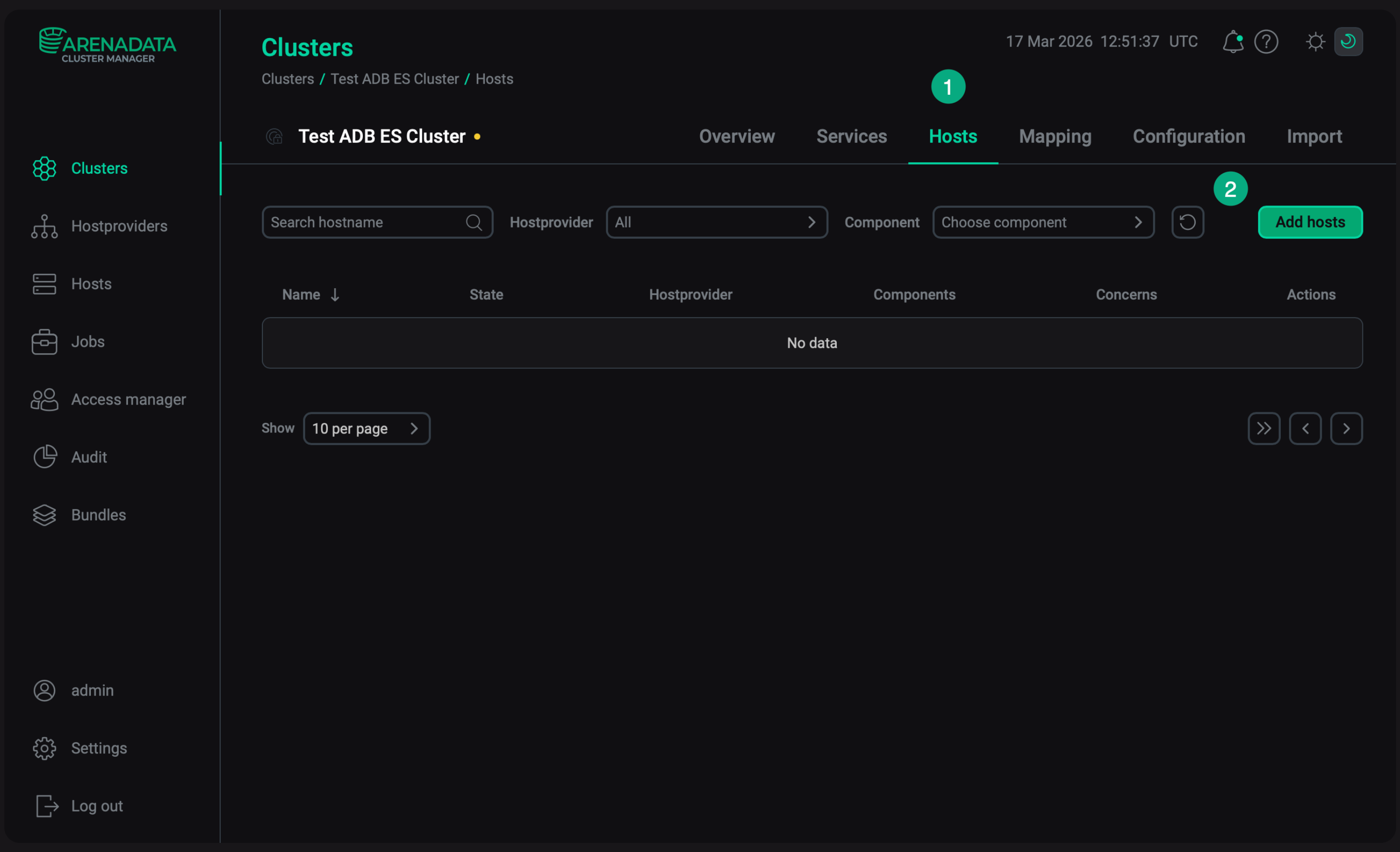
The steps for adding hosts to a cluster are listed below:

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



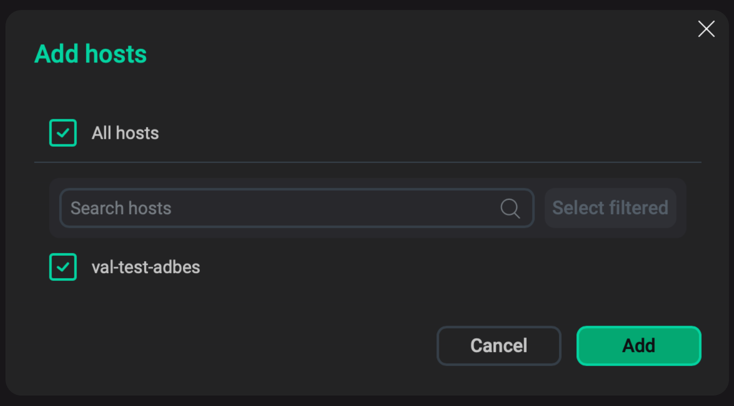
Select a cluster

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



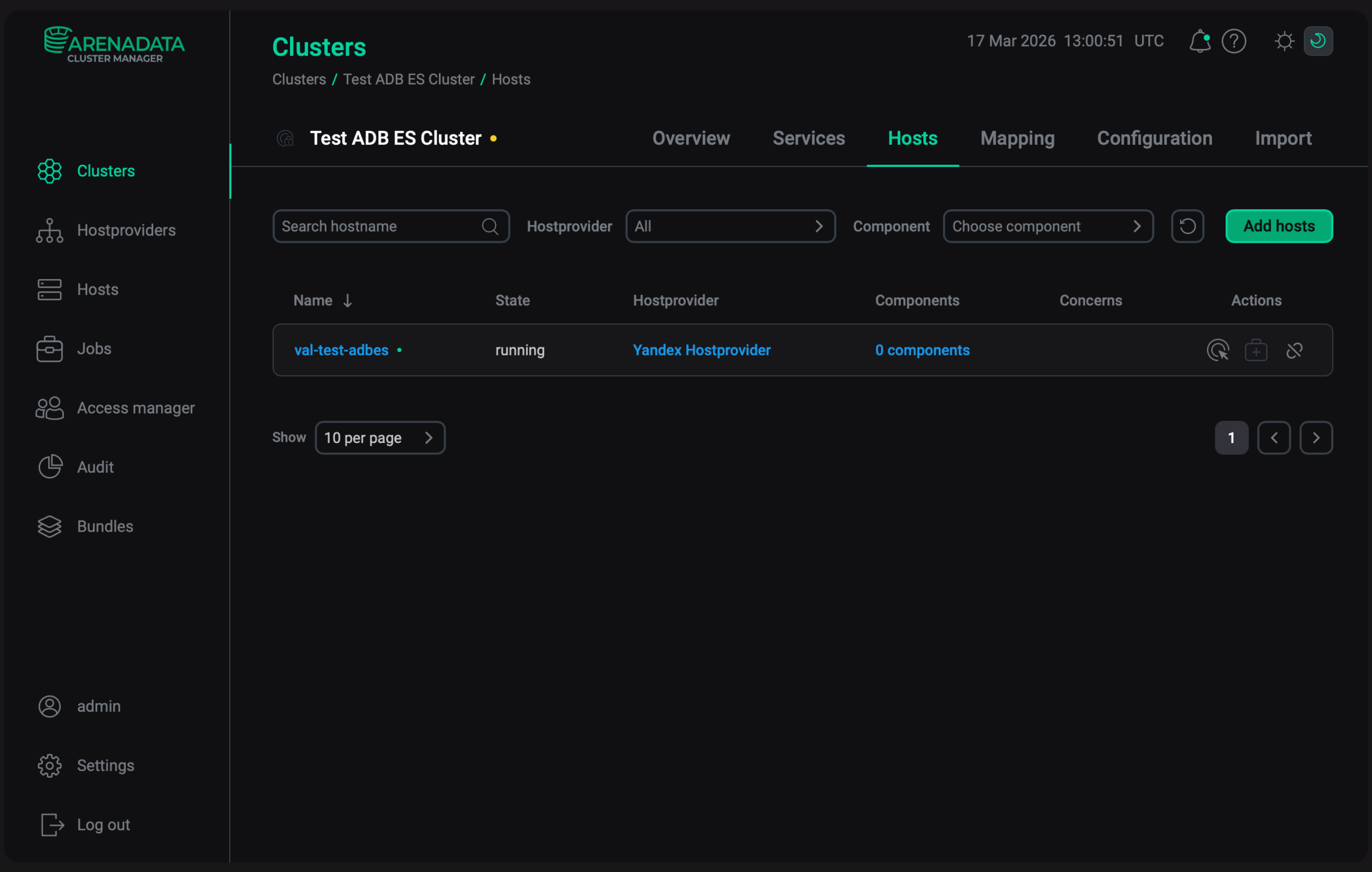
Switch to adding hosts

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



Select hosts

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



The result of successful adding hosts to a cluster

Add components

Anton Vasiliev

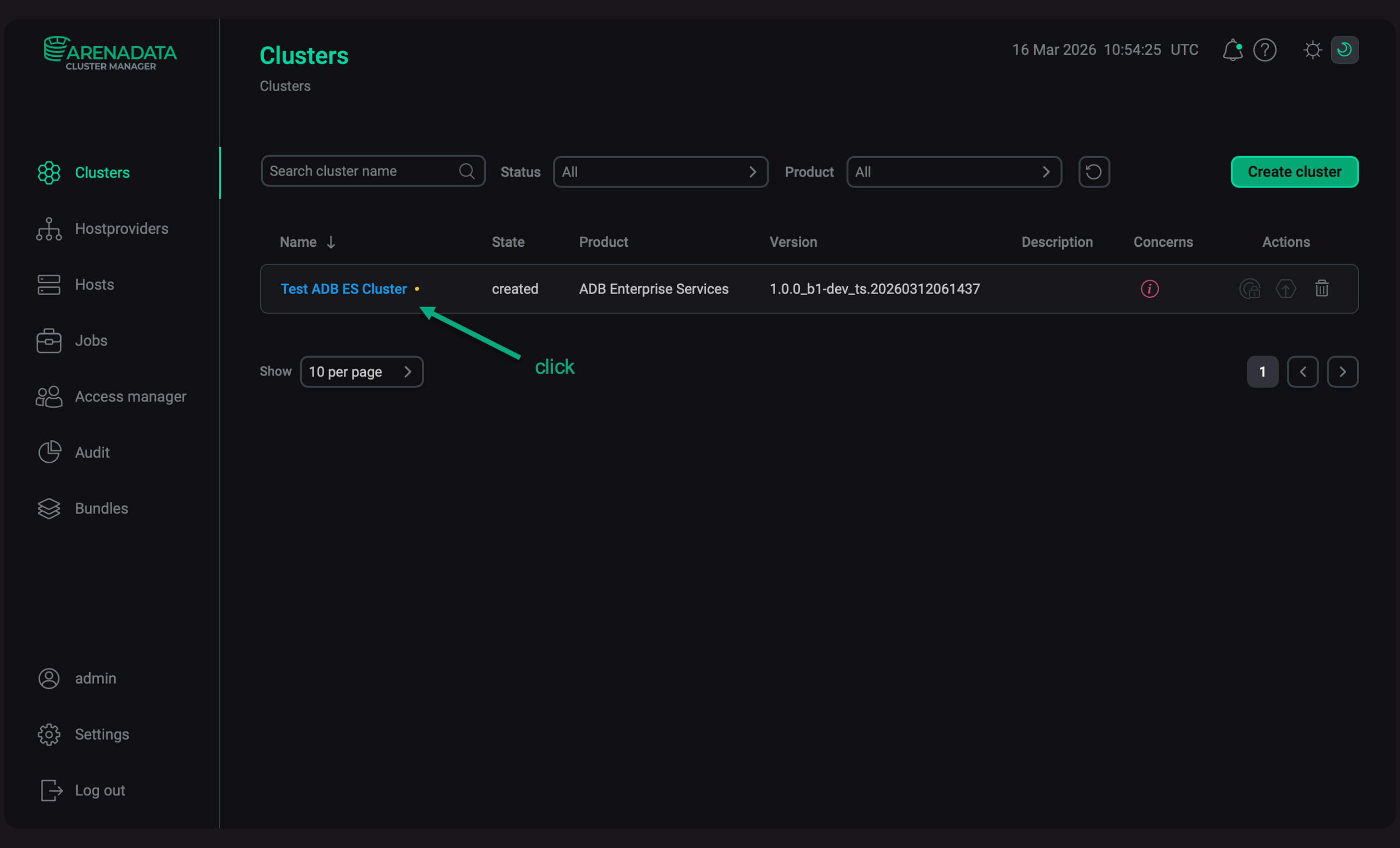
To Table of Contents

In ADCM, a **component** means a part of a service that should be deployed on one or several cluster hosts. For example, the ADB Control service consists of the ADB Control Server, ADB Control UI, ADB Control Scheduler, and ADB Control Migration components.

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

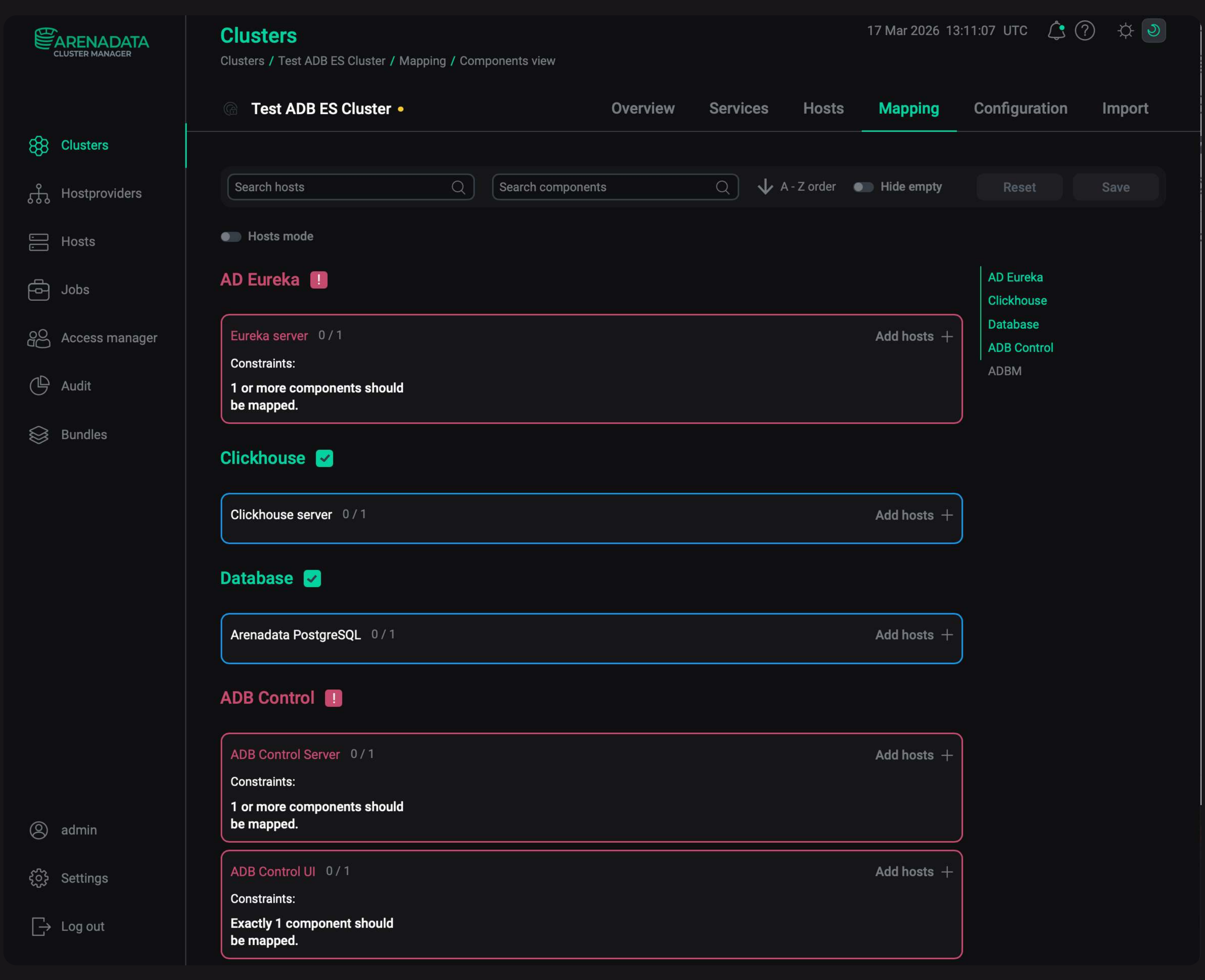
The steps for adding components to hosts are listed below:

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



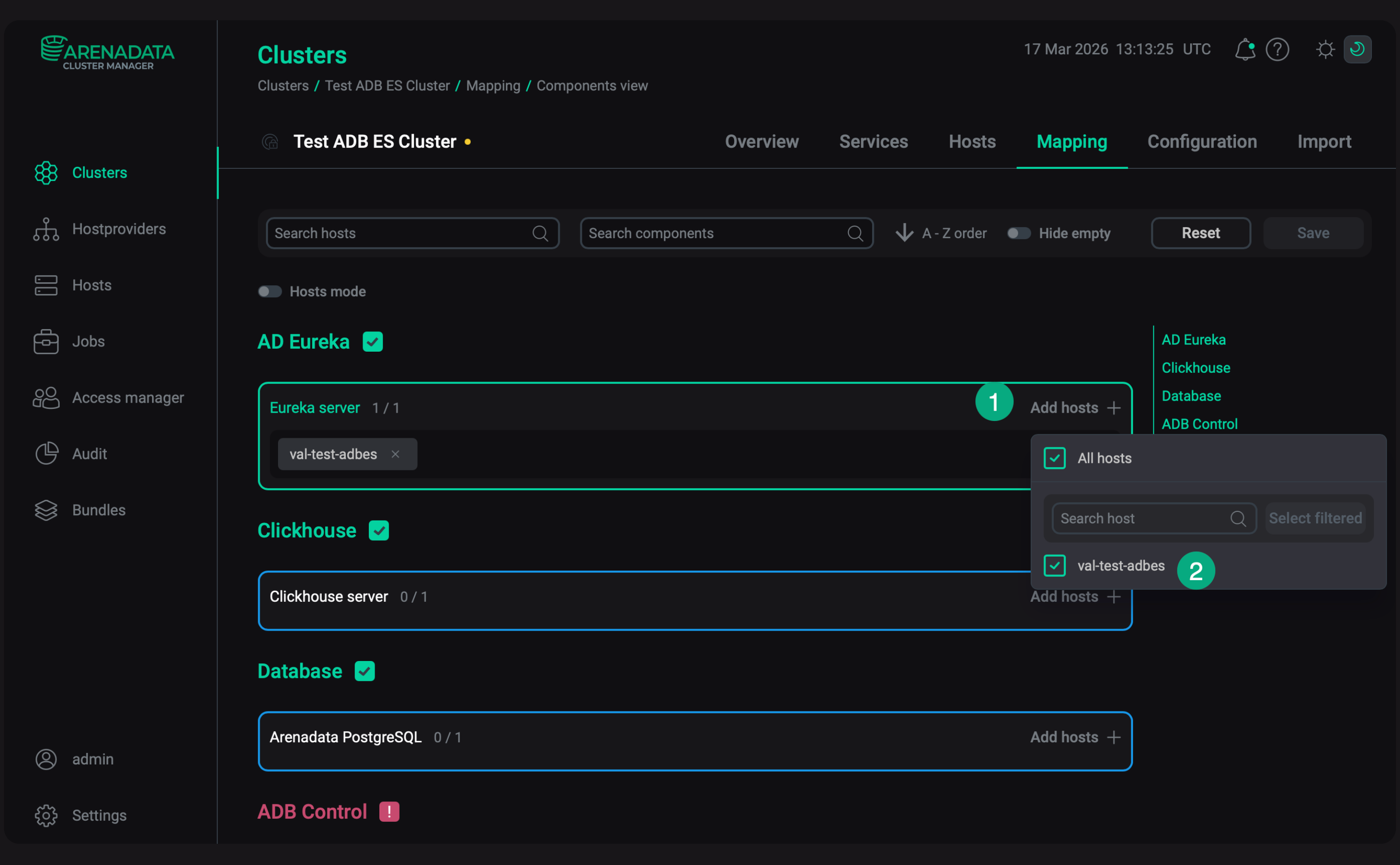
Select a cluster

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



Switch to mapping service components

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



Select hosts for a component

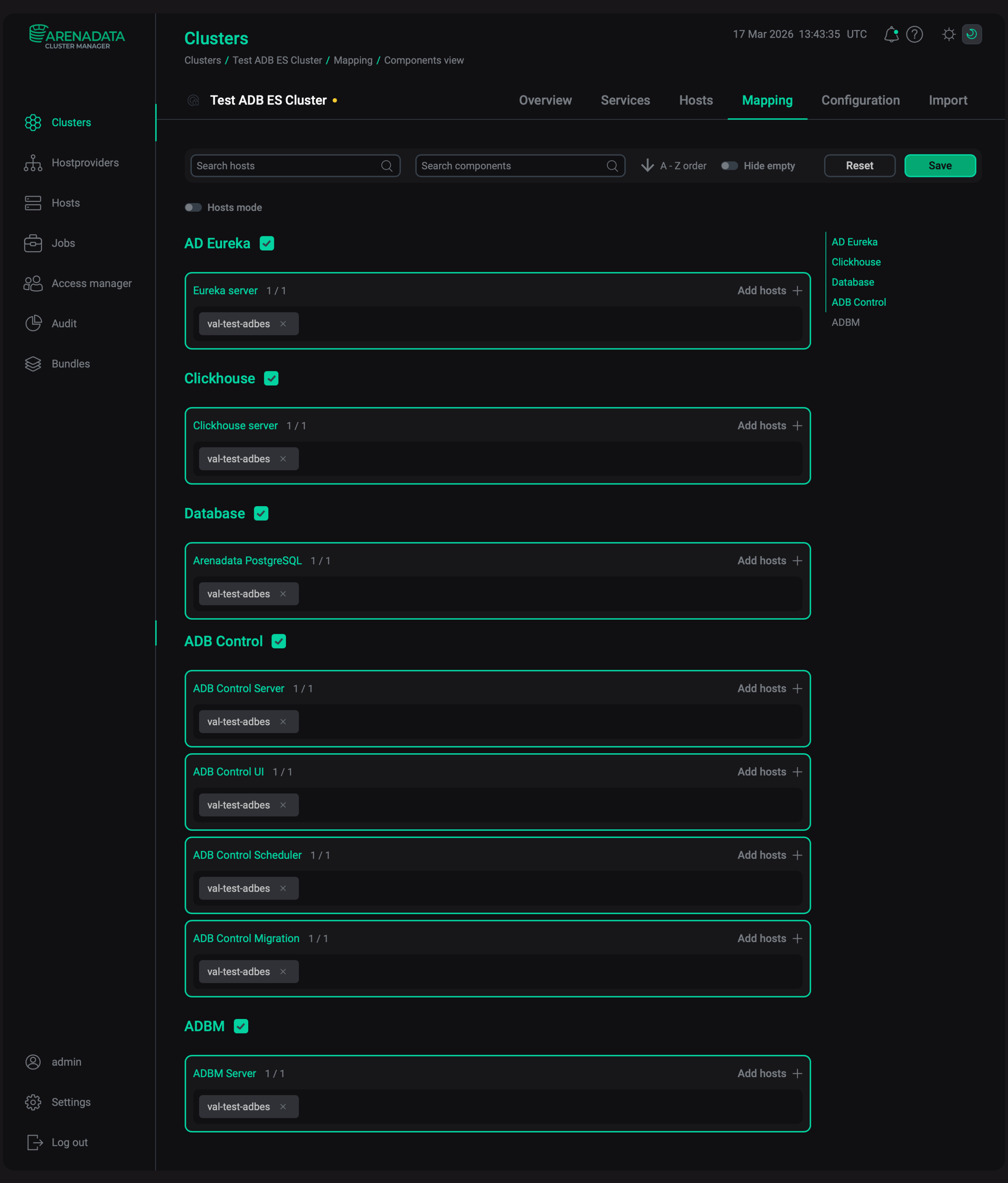
Mandatory components

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

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

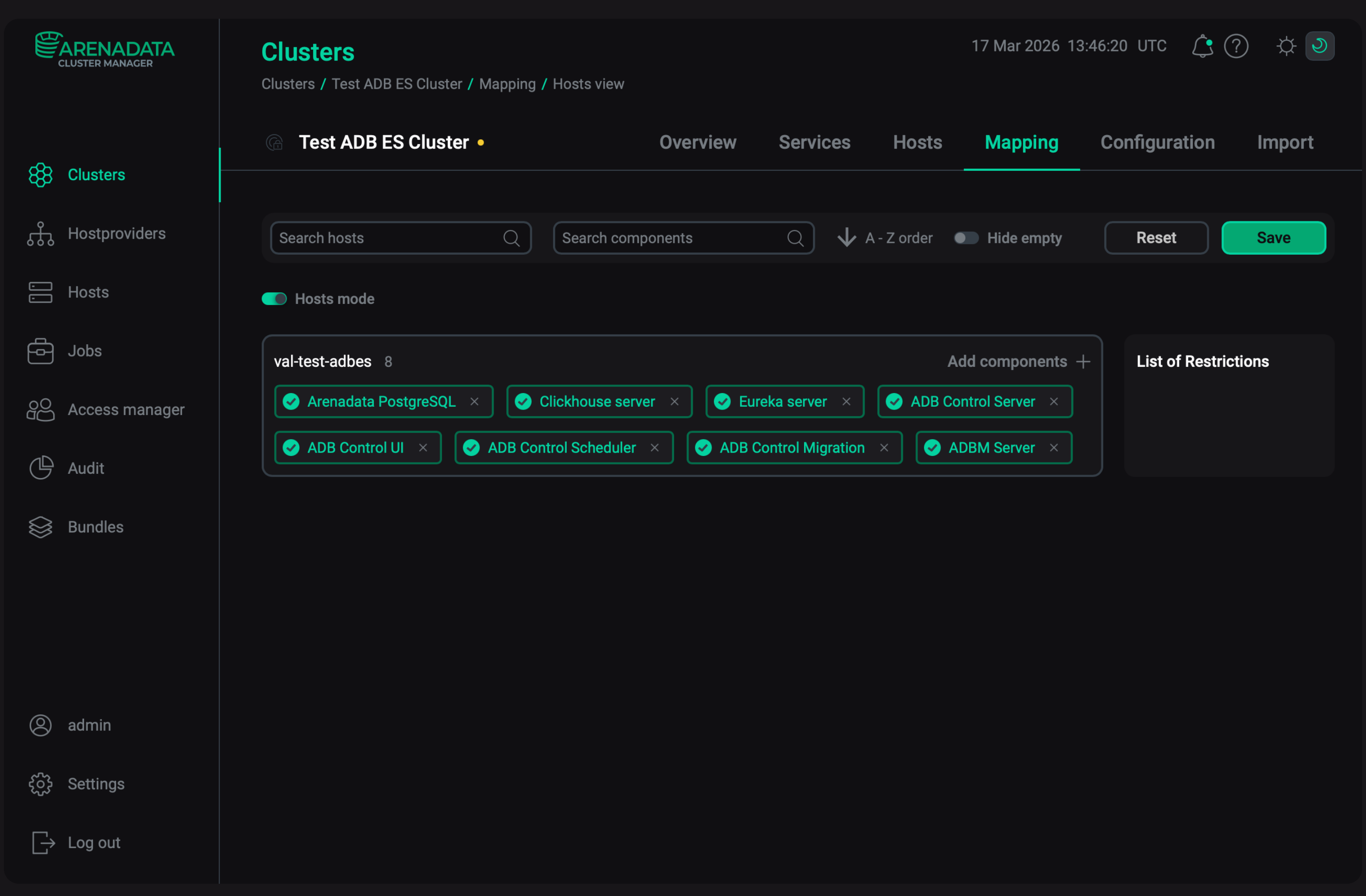
Any of the ADB ES components can be installed on a separate host, or you can install all of them on a single host. It is highly recommended to install ADB ES components on a host separate from ADB.

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.



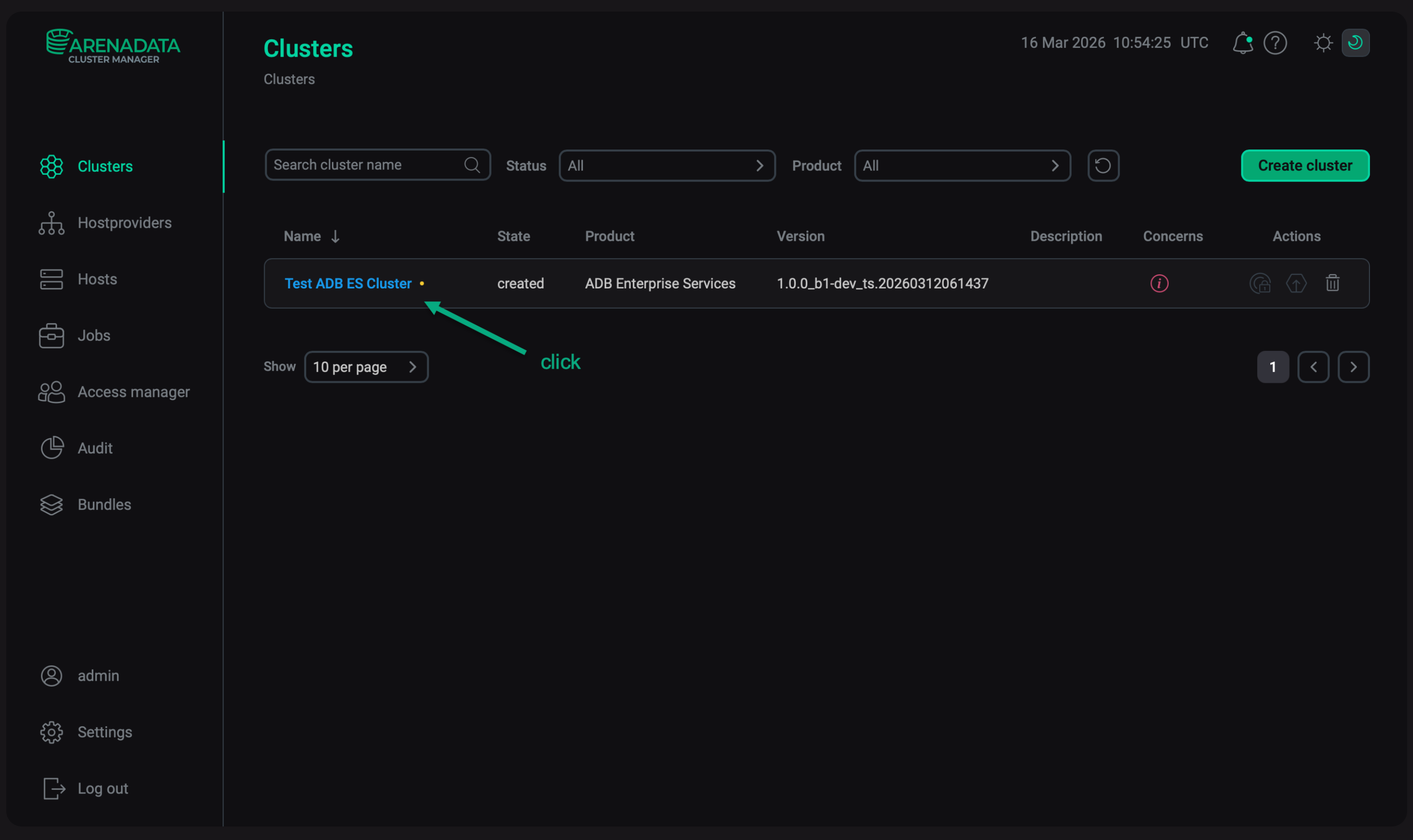
Use mapping by hosts

Configure services

Anton Vasilev

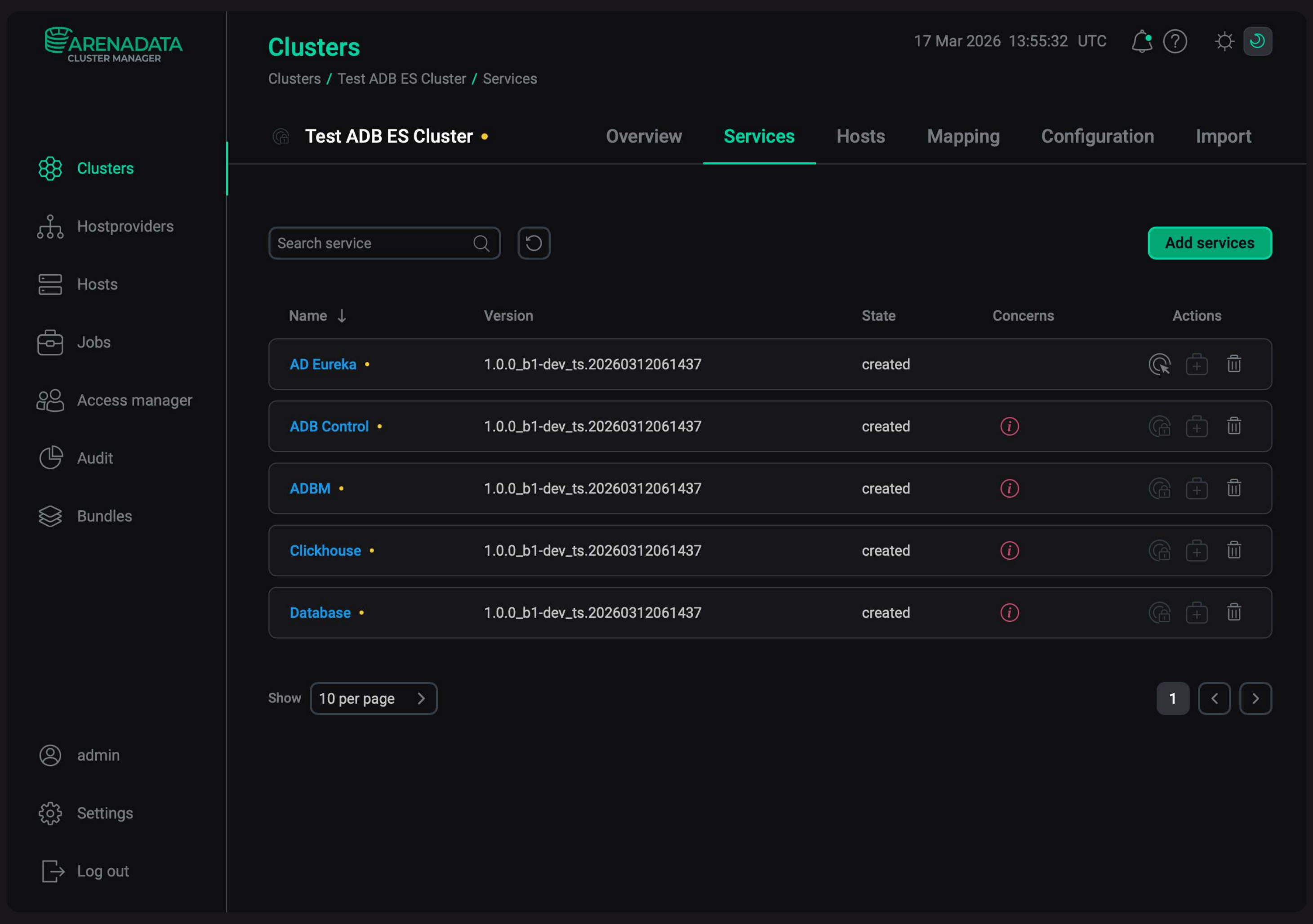
To Table of Contents

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



Select a cluster

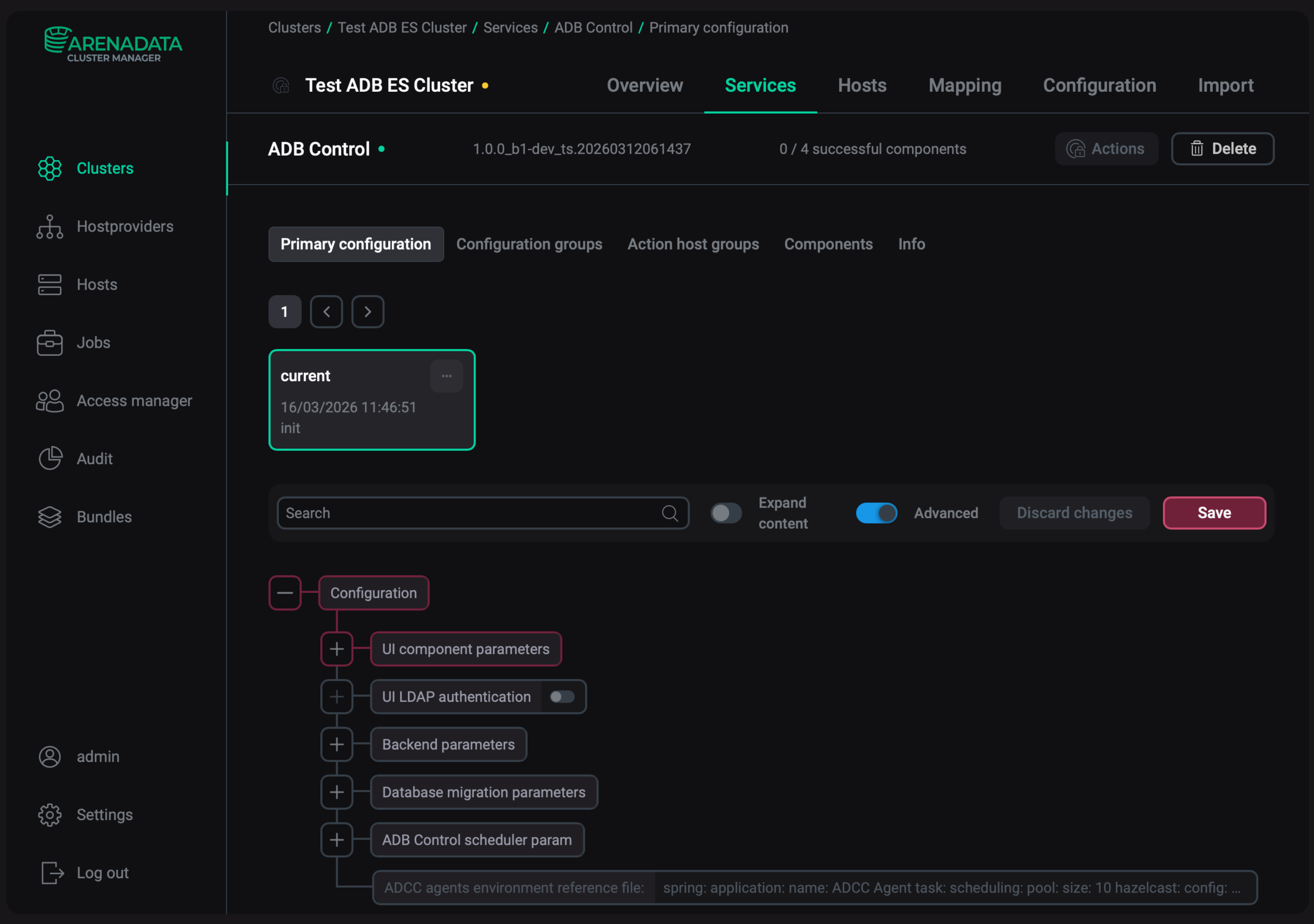
2. Open the ***Services*** tab on the cluster page, find the row that contains the desired service, and click its name in the ***Name*** column.



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

Pay attention to the following parameters:

- ***Database*** → ***Database type***. Select ***Internal postgres database parameters*** and configure the DBMS user password to use the Arenadata Postgres (ADPG) 16 provided by the ADB ES bundle. Or select ***External postgres parameters*** and specify the IP address and credentials to connect to an external PostgreSQL server.
- ***Clickhouse*** → ***Database type***. Select ***Internal Clickhouse database parameters*** and configure the DBMS user password to use the Arenadata QuickMarts (ADQM) instance provided by the ADB ES bundle. Or select ***External Clickhouse parameters*** and specify the IP address and credentials to connect to an external ClickHouse server.

NOTE

For more information about configuration parameters of the services, see Configuration parameters.

Configure a cluster

Anton Vasilev

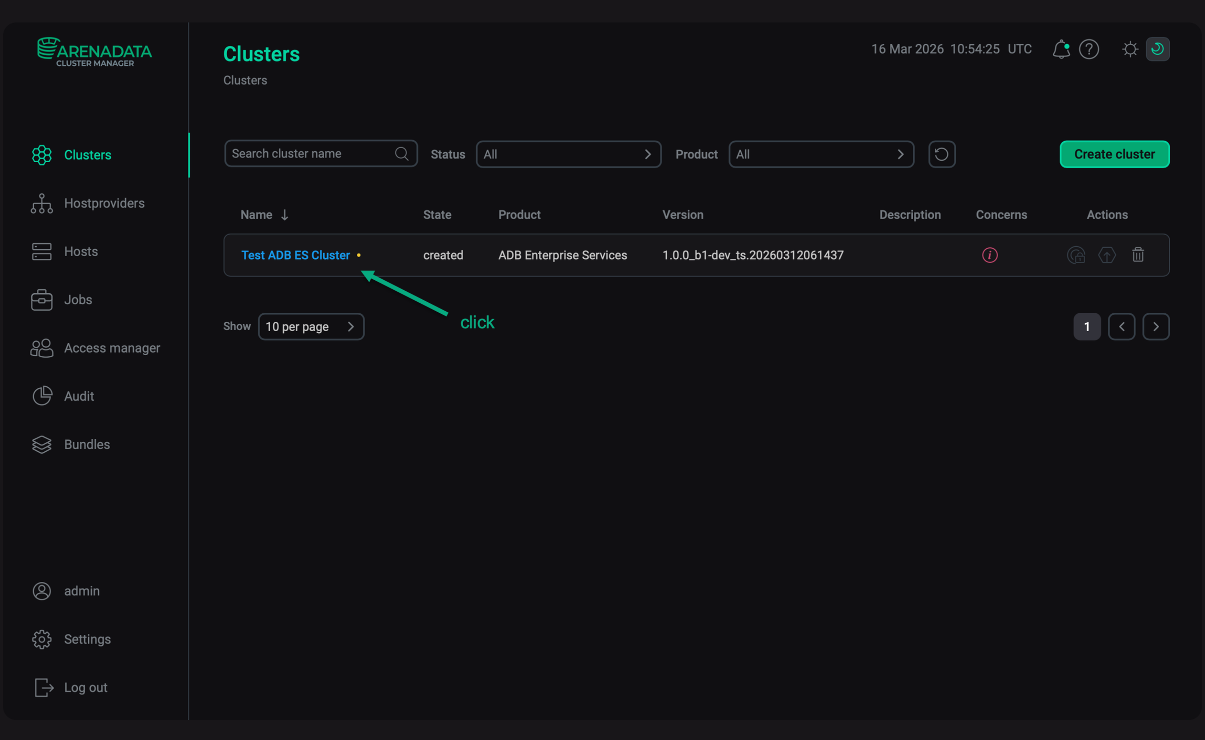
Contents

To Table of Contents

> **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.

> **CAUTION**
>
> Fill the ***Use custom JAVA_HOME for cluster*** field at your own risk.

Available configuration parameters are listed below:

- Paths to the repositories that are used during the installation process — depending on the cluster operating system (ALT Linux, Ubuntu, Astra Linux, or RED OS):
  - ***Set up Monitoring repo*** — use the repository that is described in the ***Monitoring repo url*** field.
  - ***Monitoring repo url*** — the repository that is required to install monitoring.

- ***Vault configuration*** — a switcher that determines whether to use [HashiCorp Vault](#) to store secrets of ADB ES services. When the switcher is activated, the following parameters become available.

Name	Description	Default
Address of the Vault server	The Vault server address (e.g. `192.168.1.120` or `vault.domain.com`)	—
Port of the Vault server	The Vault server port. Should be specified if the service is published on a port other than HTTPS (443)	443
Internal CA certificate in PEM format (optional)	The internal CA certificate in PEM format that signed the Vault server's certificate	—
Vault namespace (optional)	The Vault [namespace](#)	—
Writer authentication method for the Vault server	The [authentication method](#) that is used to connect to the Vault server with write access. Possible values: • `token` • `approle`	approle
Writer Vault token	The application token that is used to connect to the Vault server with write access. Required only if the ***Writer authentication method for the Vault server*** parameter is set to `token`	—
Writer Vault role ID	The role identifier provided for applications to connect to the Vault server with write access. Required only if the ***Writer authentication method for the Vault server*** parameter is set to `approle`	—
Writer Vault secret ID	The secret identifier associated with the ***Writer Vault role ID*** role that is used by applications to connect to the Vault server with write access. Required only if the ***Writer authentication method for the Vault server*** parameter is set to `approle`	—
Reader authentication method for the Vault server	The [authentication method](#) that is used to connect to the Vault server with read access. Possible values: • `token` • `approle`	approle
Reader Vault token	The application token that is used to connect to the Vault server with read access. Required only if the ***Reader authentication method for the Vault server*** parameter is set to `token`	—
Reader Vault role ID	The role identifier provided for applications to connect to the Vault server with read access. Required only if the ***Reader authentication method for the Vault server*** parameter is set to `approle`	—
Reader Vault secret ID	The secret identifier associated with the ***Reader Vault role ID*** role that is used by applications to connect to the Vault server with read access. Required only if the ***Reader authentication method for the Vault server*** parameter is set to `approle`	—
Mount point for ADB ES cluster secrets	A mount point that is accessible for read/write operations with the provided ADB ES cluster authentication parameters. Each service will store secrets on the Vault server at the following path: *<Mount point for ADB ES cluster secrets>/adbes/<cluster_name>/service/<service_name>/<component>*	—
Login mount point	A custom authentication mount point. If empty, equals to the authentication method selected in the ***Writer authentication method for the Vault server*** (write) or ***Reader authentication method for the Vault server*** (read) field	—

> **IMPORTANT**
>
> - Before editing Vault integration parameters, ensure that Vault is installed and configured.
> - Each time you edit and save parameters in the ***Vault integration*** section, run the cluster action ***Reconfigure Vault integration***.
> - Currently, Vault can be used to store authentication parameters of ADB Control and ADBM.

- ***SSL configuration*** — this section is read-only: you can use it to check whether SSL is enabled and configured. To enable and configure SSL in the ADB ES cluster, use the ***Manage SSL*** cluster action, which is available after the cluster is installed.

- ***Timeout to unlock package manager*** — parameters of this section are intended to set the timeout of the DNF software package manager unlocking wait. Displayed if the ***Advanced*** flag is enabled. In Ubuntu, after the host reboot, automatic update starts, which locks DNF.

Name	Description	Default
retries	The maximum number of retries	3
delay	The time delay between retries (in seconds)	10

- ***Use custom JAVA_HOME for cluster*** — a switcher that determines whether to use a custom Java path for ADB ES services that use Java (for example, AD Eureka, ADB Control, and ADBM). Displayed if the ***Advanced*** flag is enabled. When the switcher is activated, the ***JAVA_HOME*** field becomes available. Enter a valid Java path in that field. The selected location should contain a *bin* folder with a Java binary file, for example, */usr/lib/jvm/java-17-openjdk-amd64*. Installation will fail if there is no Java at the specified path.
  If the ***JAVA_HOME*** value is empty, one of the following default paths is used (depending on the operating system).

Ubuntu

/usr/lib/jvm/java-17-openjdk-amd64

RHEL

/usr/lib/jvm/jre-17-openjdk

AltLinux

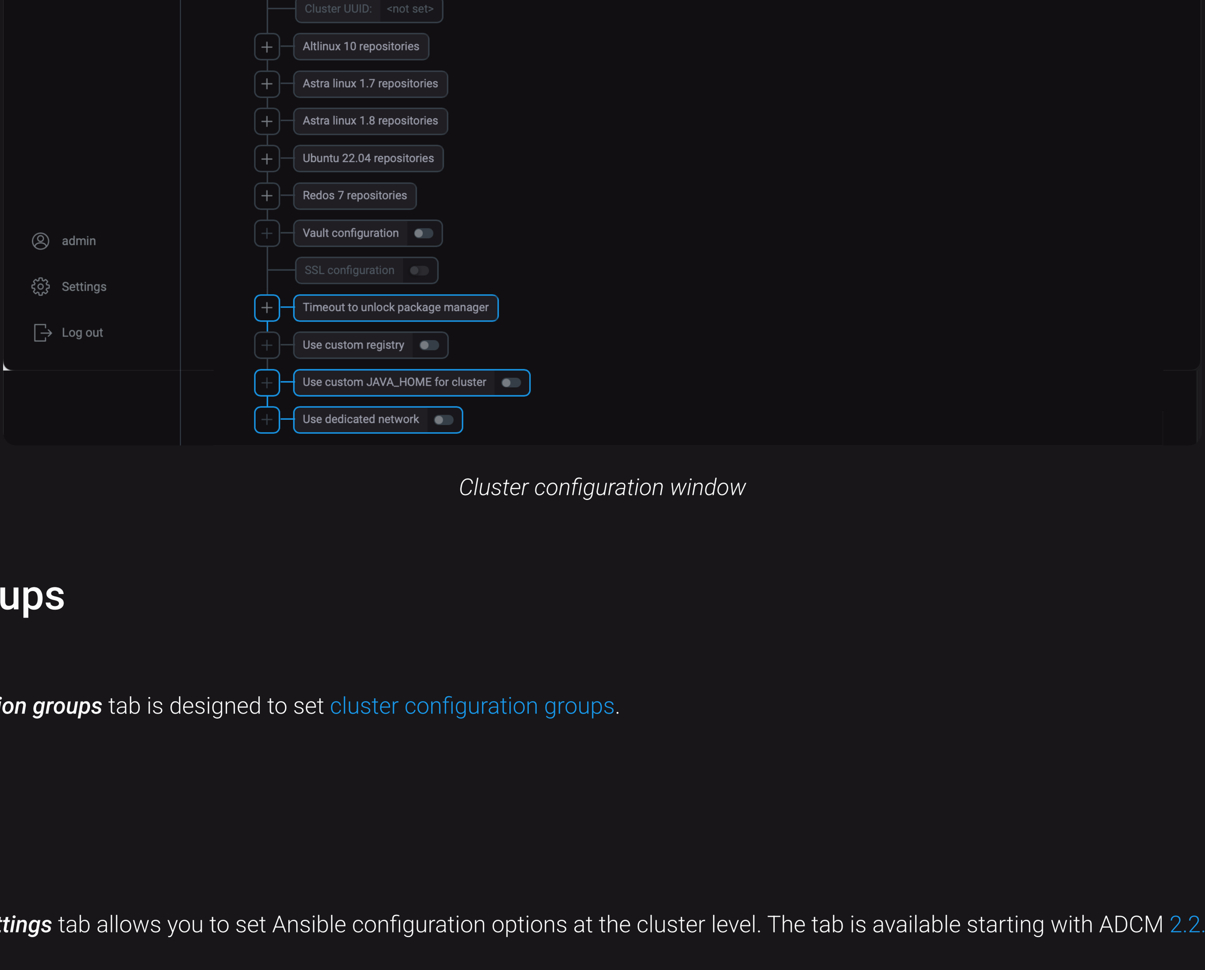
/usr/lib/jvm/jre-17

RED OS

/usr/lib/jvm/jre-17-openjdk

> **IMPORTANT**
>
> - If you plan to use custom Java (such as Axiom), Java 17 is required for installation.
> - For Astra Linux, the ***JAVA_HOME*** field is mandatory.

- ***Use dedicated network*** — a switcher that determines whether to use a dedicated network for interconnection of ADB ES components. Displayed if the ***Advanced*** flag is enabled. When the switcher is activated, the ***Dedicated network*** field becomes available. The field requires the `<subnet>/<netmask>` notation, for example `192.168.1.0/24`. If this parameter is not set, then ADCM uses addresses with the default route. A host must have exactly one address in the range. Run the ***Reconfigure & Restart*** ADB Control action to apply changes.
  If you use SSL certificates with an IP address in the `SAN` section, the IP addresses of the ADB ES components can change in the dedicated network. In this case, you will need to issue a new certificate specifying the new IP address.



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

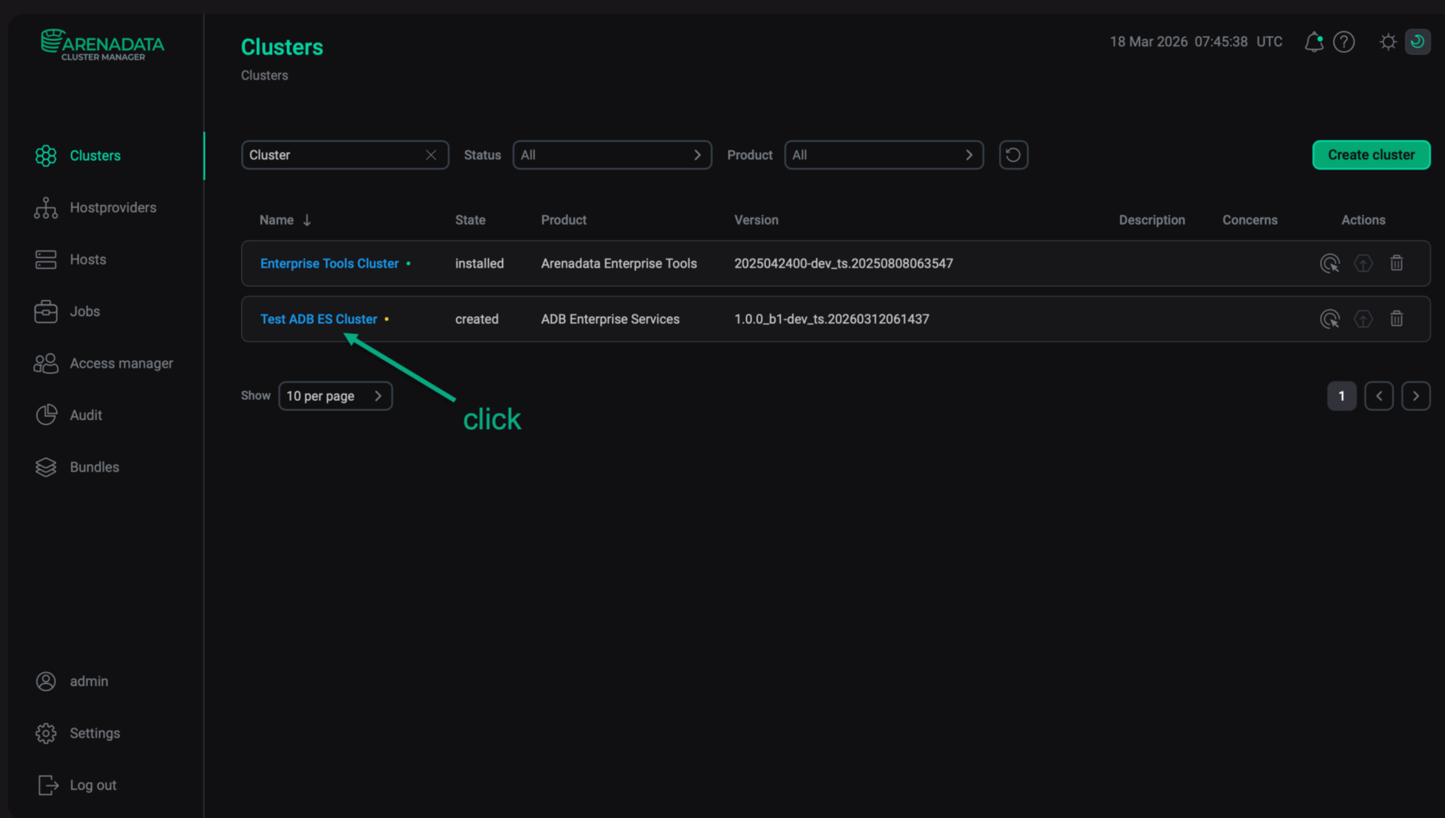
Import ET settings

Anton Vasilev

To Table of Contents

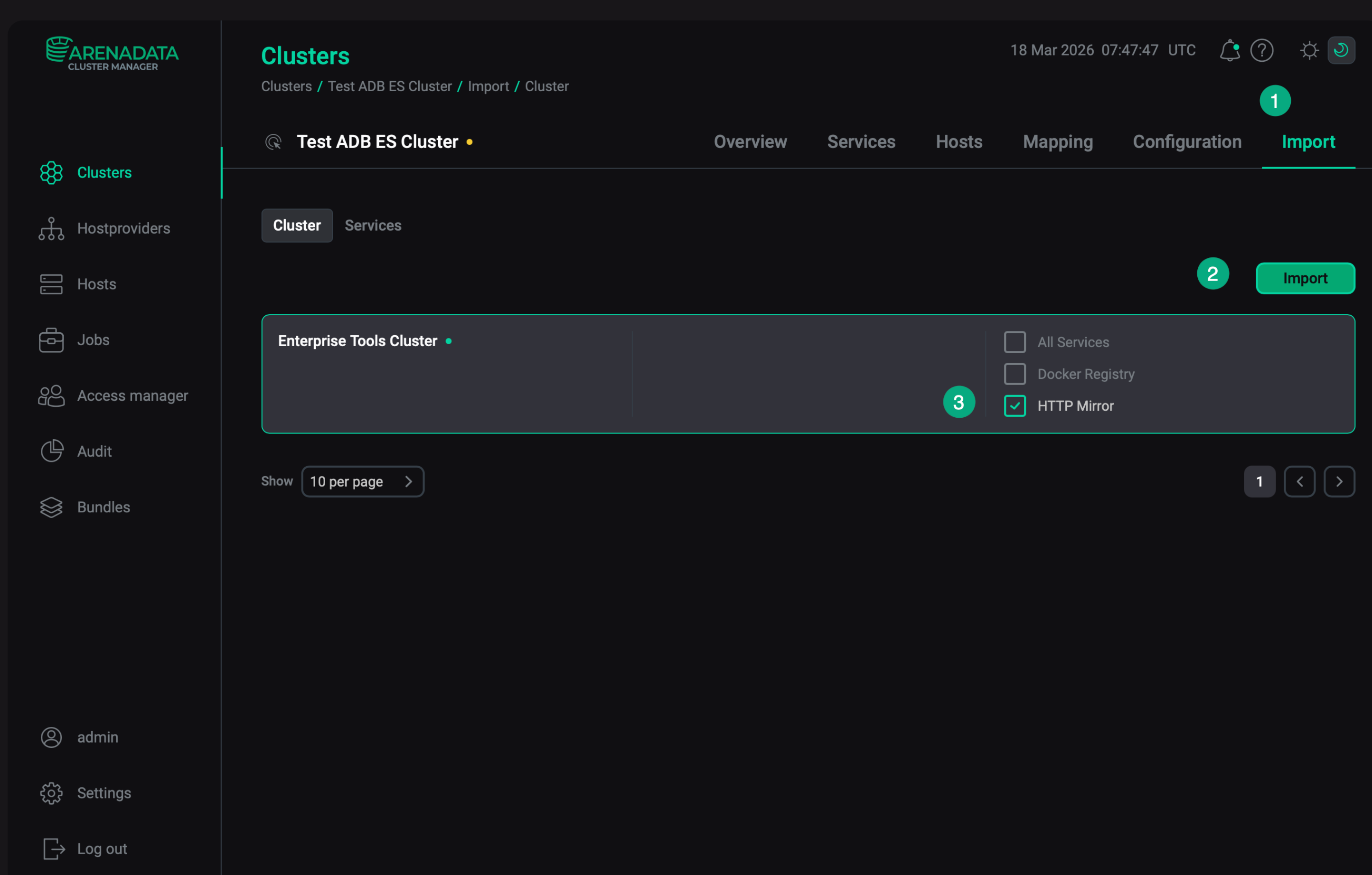
For a new cluster to be installed offline, it is necessary to integrate this cluster with the previously created Enterprise Tools (ET) cluster:

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



Select a cluster

2. Open the ***Import*** tab in the cluster menu.
3. Select the ***HTTP Mirror*** service of the previously created ET cluster.
4. Click ***Import***.



Integration with an 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

Anton Vasilev

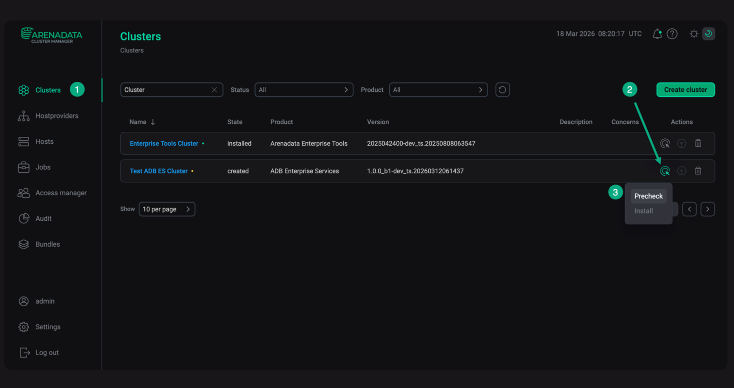
Contents

To Table of Contents

Step 1. Precheck

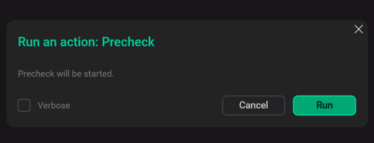
Before installing a new cluster it is recommended to run the ***Precheck*** action. By doing this, you make sure that all necessary repositories are available, the cluster does not contain hosts with the same IP addresses, and all other prerequisites are met. Do the following:

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



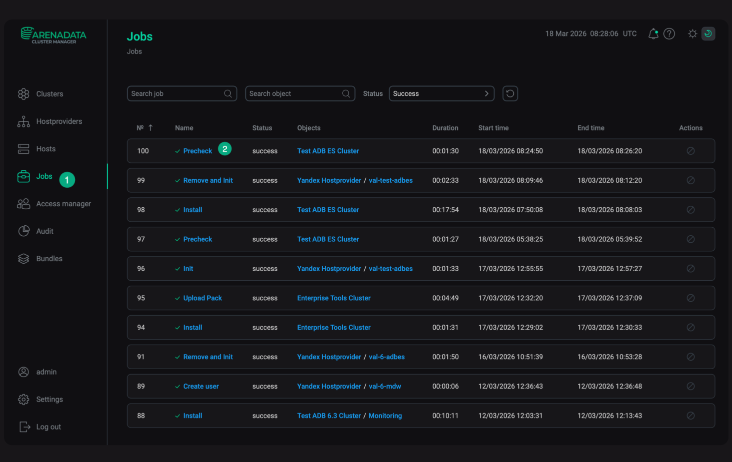
Run the cluster precheck

2. Verify the requested action in the opened window.



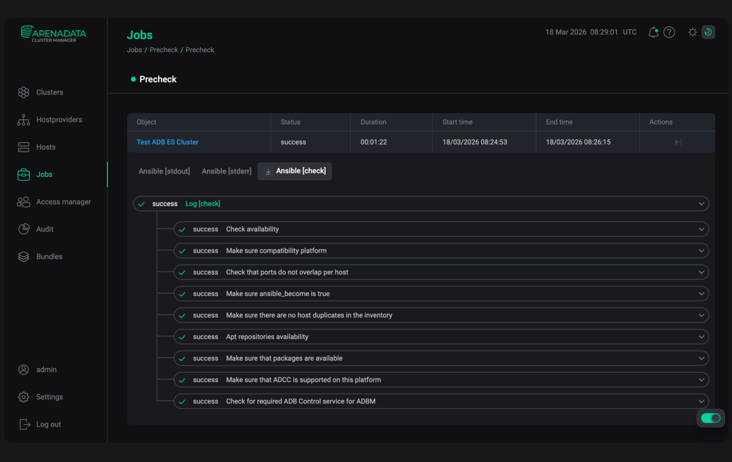
Verify the action

3. Open the ***Jobs*** page and wait until the running job is completed. After this, click the first item in the job list.



View jobs

4. View logs in the window that opens. To download logs, click ***Download***.



View logs

Step 2. 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.

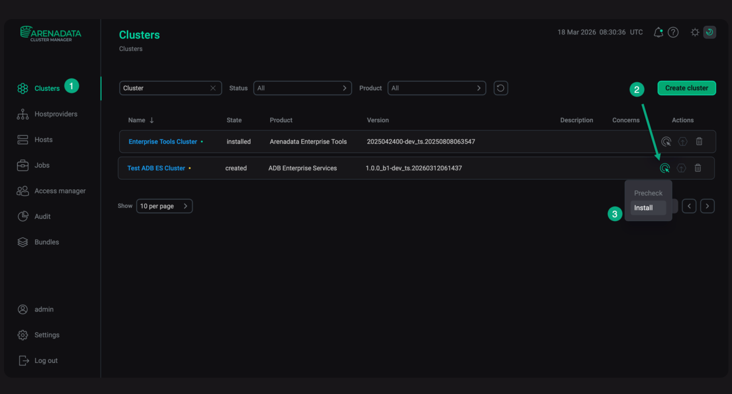
In both cases each service starts automatically after installation.

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

Whole cluster

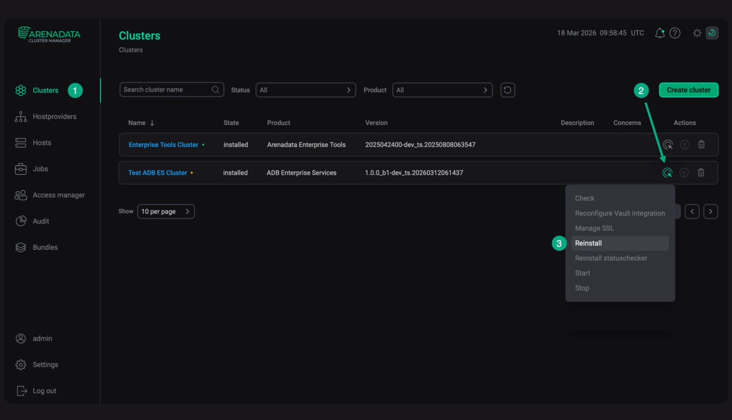
To install all cluster services within a single action, follow the steps:

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. Verify the requested action in the opened window.
3. Wait until the job is completed.
4. If some service fails during installation, but the preceding services are installed successfully, you can try to fix errors and restart installation of all failed services. To do this, select the ***Reinstall*** action in the ***Actions*** column on the ***Clusters*** page.



Installation restart

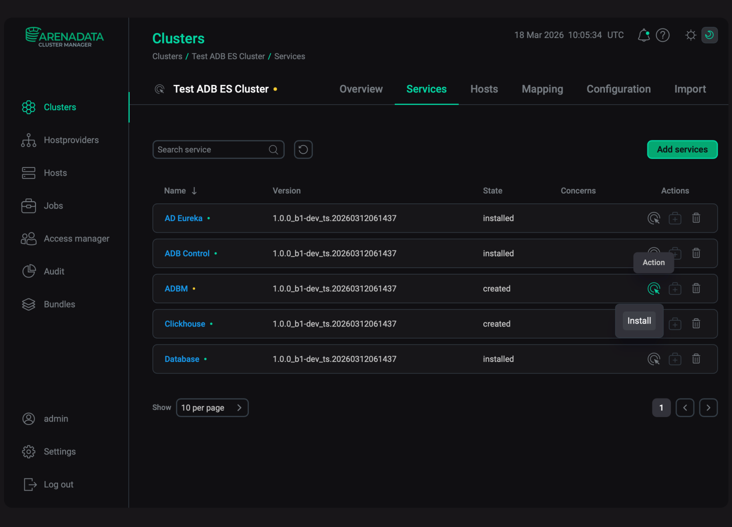
Single services

The recommended way to install ADB ES is to use the ***Install*** cluster action (as described above). Alternatively, you can install all services manually, one by one. If you choose this method, follow the installation order below:

1. ***Database***, ***Clickhouse***, and ***AD Eureka*** (in any order)
2. ***ADB Control***
3. ***ADBM***

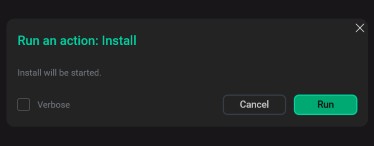
To install services one by one, follow the steps:

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



Switch to installation of single services

2. Verify the action in the opened window.



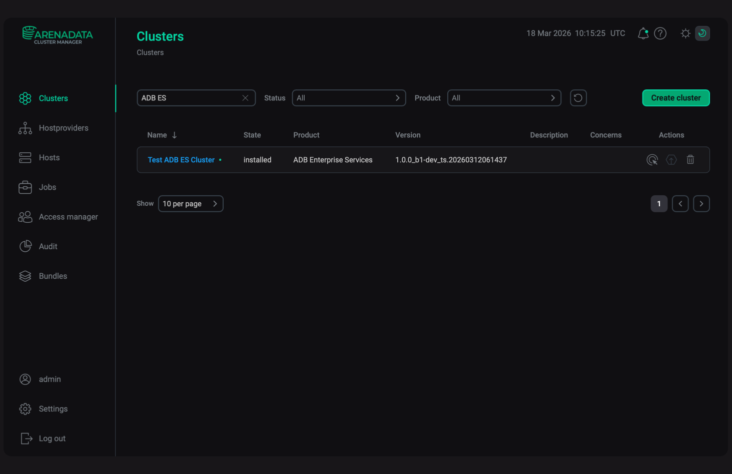
Verify the action

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

Step 3. Check the cluster state after installation

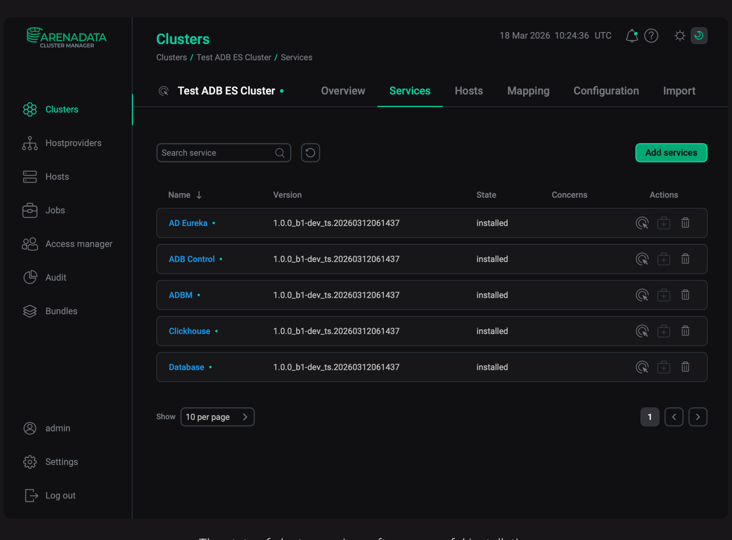
As a result of successful installation, the cluster and its services change their state according to the following rules:

- 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

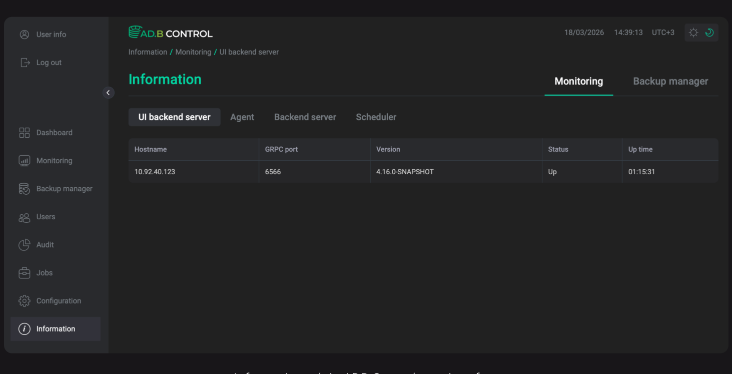
Step 4. Check the ADB Control web interface

After ADB ES installation is complete, check the ADB Control web interface:

1. In a browser, enter `<adb_control_host>:8890` where `adb_control_host` is the hostname or IP address of a host that you have mapped to ADB Control and `8890` is the default port for the ADB Control UI. This port can be changed in the ADB Control service configuration, in ***UI component parameters → Listening port***.

> **NOTE**
>
> The IP address and port of ADB Control are also listed on the ***Info*** tab of the ADB Control service.

2. To sign in, use the default credentials:
   - ***Username:*** admin
   - ***Password:*** 1234
     When you sign in for the first time, you will be prompted to change the default password.
3. Open the ***Information*** tab and check the version, status, and uptime of the installed components.



Information tab in ADB Control user interface

You can now import the ADB ES cluster configuration to ADB.

Install monitoring

Anton Vasilev

Contents

To Table of Contents

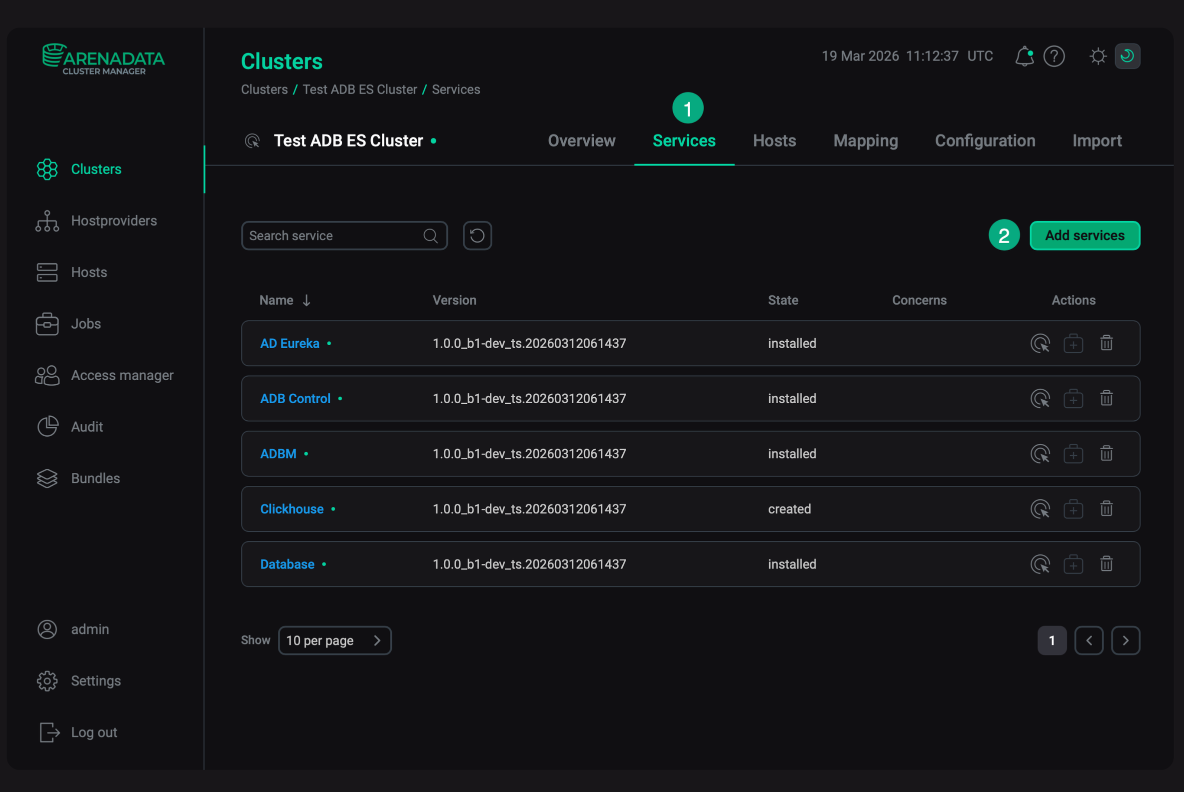
Overview

Prometheus, a monitoring toolkit, collects and stores monitoring metrics. It receives metrics from a predefined set of sources called **exporters**.

Grafana collects metrics from Prometheus and visualizes them as charts. Grafana in ADB ES includes a set of pre-configured dashboards that reflect the state of your cluster, databases, and hosts.

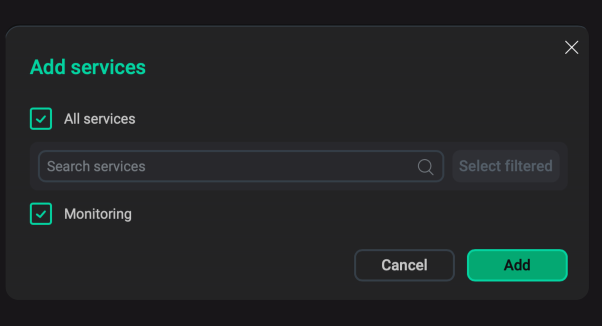
Step 1. Add the service

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



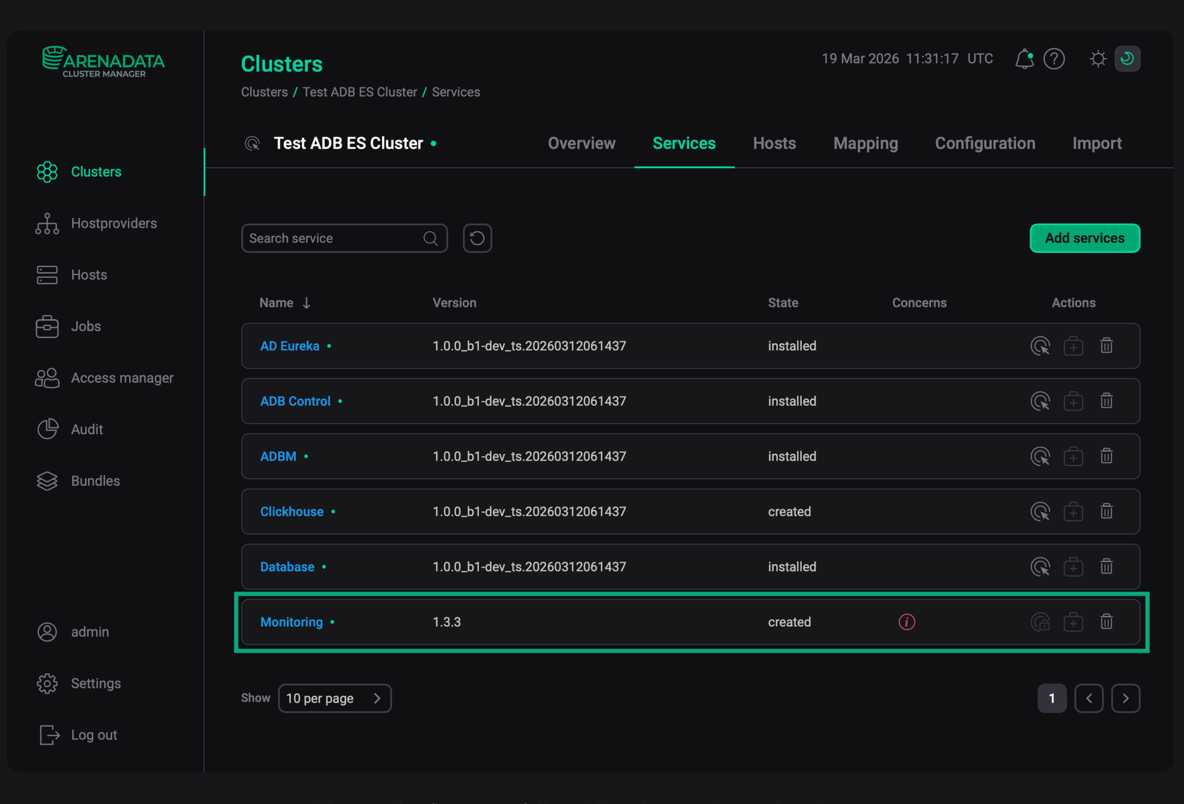
Switch to adding services

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



Select the service

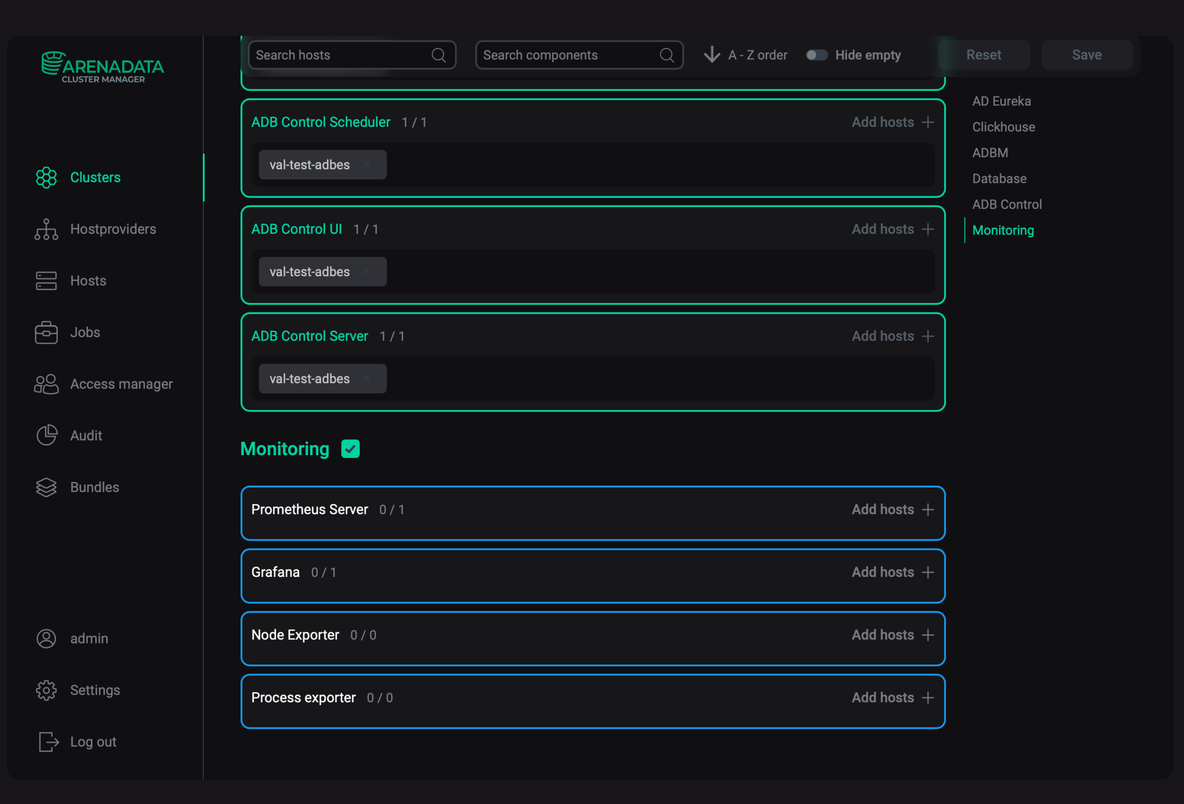
As a result, the added service is displayed on the ***Services*** tab.



The result of successfully adding the service to the cluster

Step 2. Add components

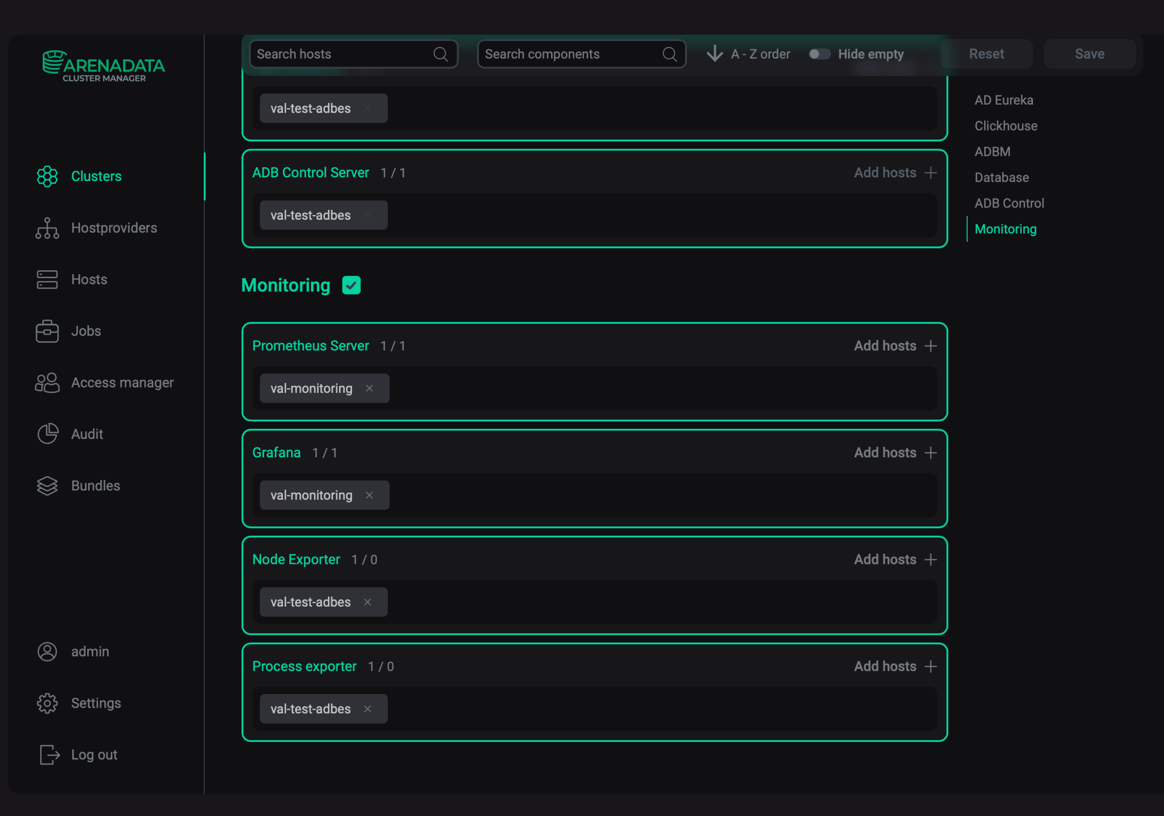
1. On the cluster page, open the ***Mapping*** tab to proceed to mapping service components to cluster hosts. Scroll down to the ***Monitoring*** section.



Switch to mapping service components

2. Assign hosts to components of the Monitoring service — click ***Add hosts*** and select the hosts:

   - The ***Prometheus Server*** and ***Grafana*** components are recommended to be installed on a host separate from ADB ES components. This setup ensures independent monitoring access even during an ADB ES failure.
   - For the ***Node Exporter*** component, select hosts from which you want to collect hardware and operating system metrics. In practice, this means installing it on every host in the cluster.
   - For the ***Process Exporter*** component, select hosts where ADBM and ADB Control are installed.
     For example, map the components as depicted in the image below.



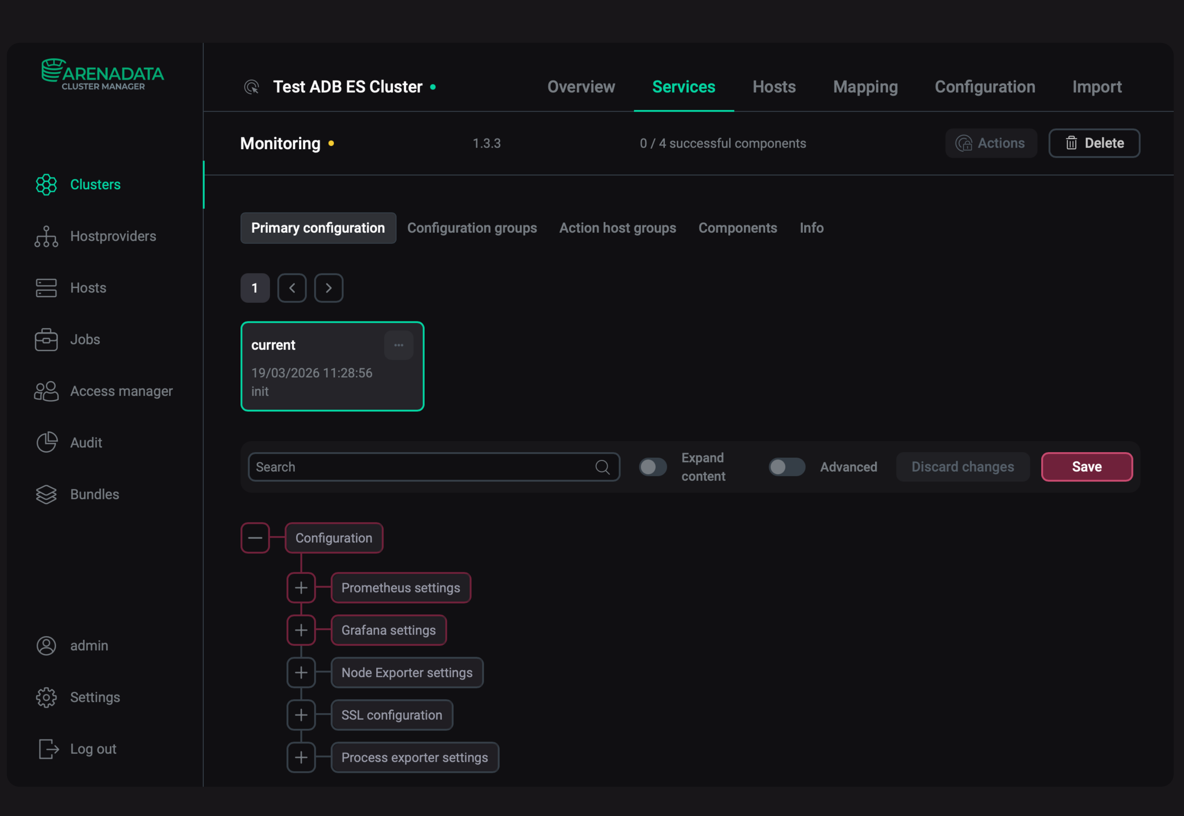
Mapping of monitoring components

3. Click ***Save*** to save the mapping.

Step 3. Configure the service

1. Open the ***Services*** tab on the cluster page and click the ***Monitoring*** service name in the ***Name*** column.
2. On the page that opens, fill in the service configuration parameters — see the ***Monitoring*** section in the Configuration parameters article for parameter descriptions. Fields highlighted in red are required.
   You will need the password from the ***Grafana administrator's password*** field to open the Grafana web interface and view dashboards.

   If you need to access the Prometheus web interface, create a user account for it using ***Prometheus settings → Prometheus users to login/logout to Prometheus***.

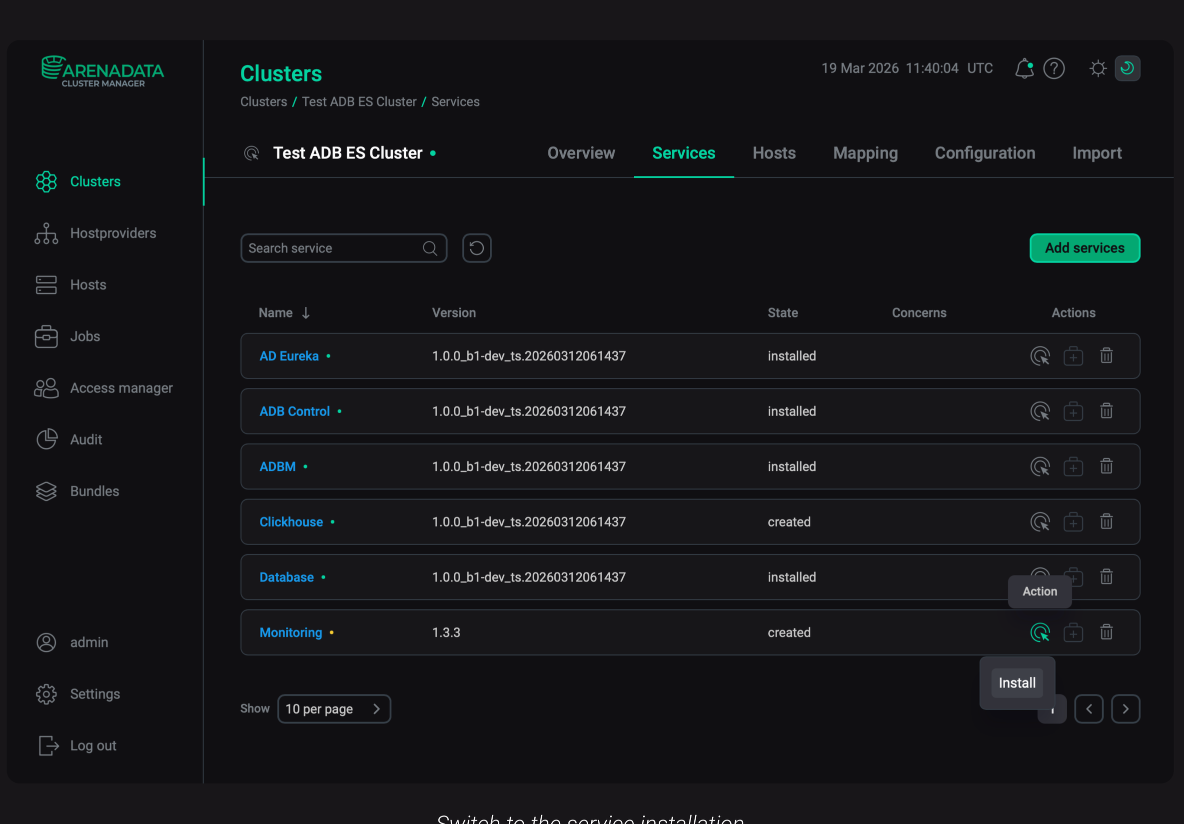


Configure the Monitoring service

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

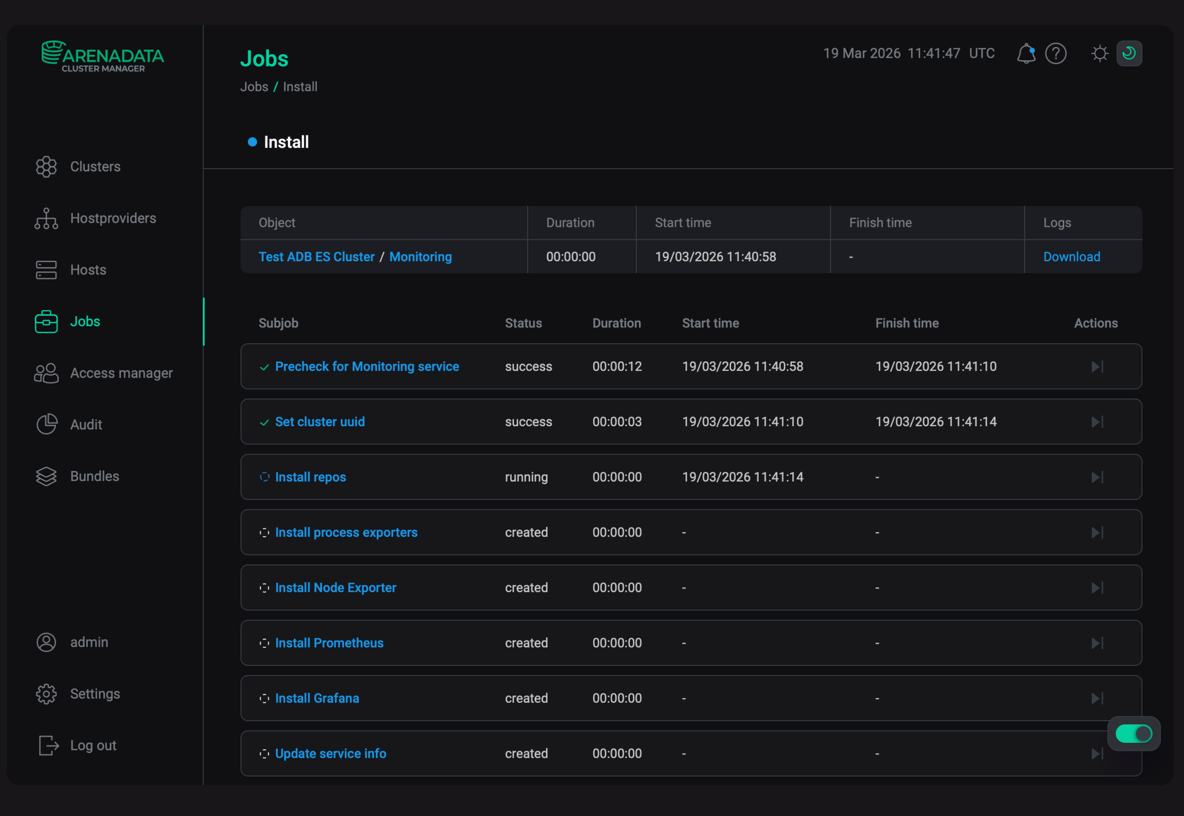
Step 4. Install the service

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` to `installed`.
   To view the service installation progress 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

Refer to ADCM documentation for description of the ***Jobs*** tab.

View results

To check that the configured exporters send metrics to Prometheus, open the Prometheus web interface as described in View metrics in Prometheus.

To check that Grafana collects metrics from Prometheus for its dashboards, open the Grafana web interface as described in Open Grafana.

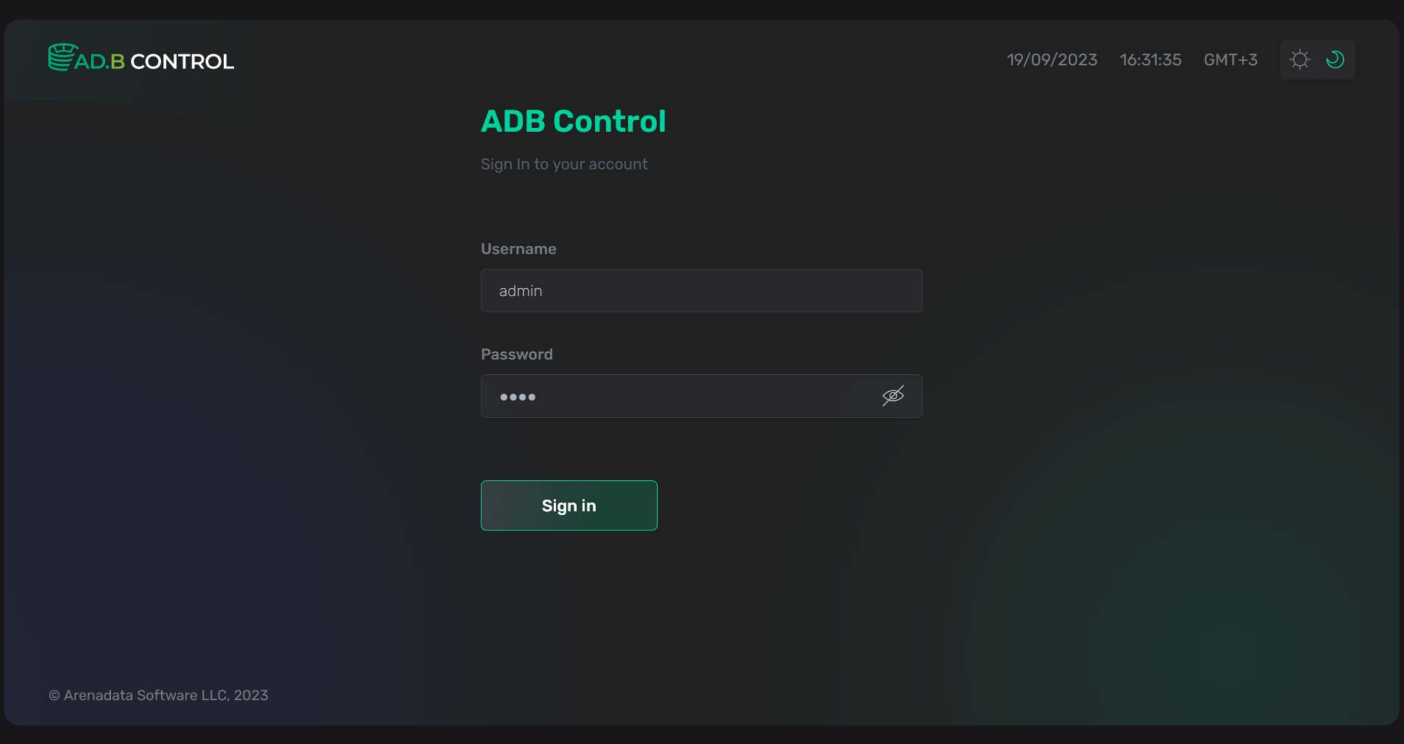
Connect to ADB Control and ADBM

Daria Barysheva

To Table of Contents

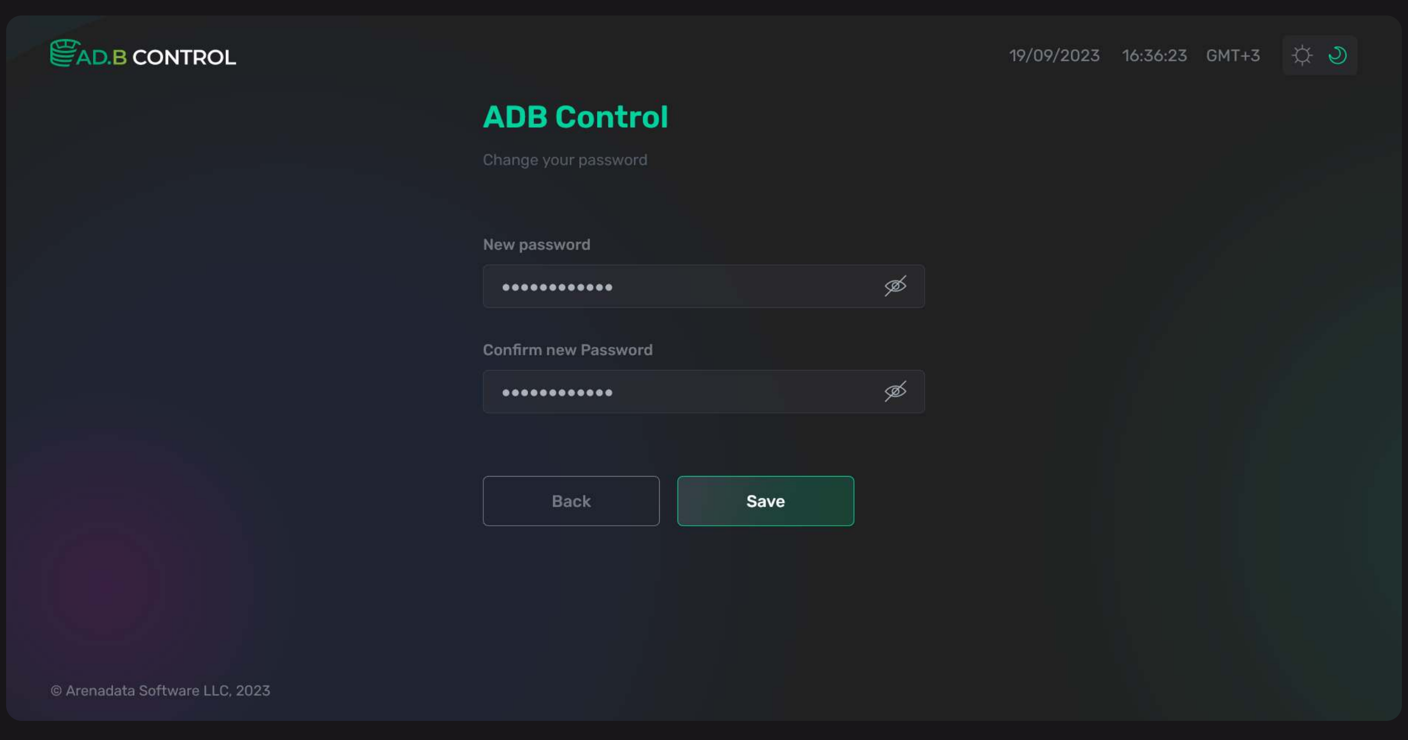
To connect to the ADB Control and ADBM web interface, follow the steps:

1. In the browser bar, enter the IP address of the host where ADB Control is deployed. To specify the port number, use the value of the Listening port parameter from the ***UI component parameters*** section defined when configuring the ADB Control service during the ADB ES cluster installation (`8890` by default).
2. In the window that opens, enter the default user credentials and click ***Sign in***:
   - ***Username:*** admin
   - ***Password:*** 1234



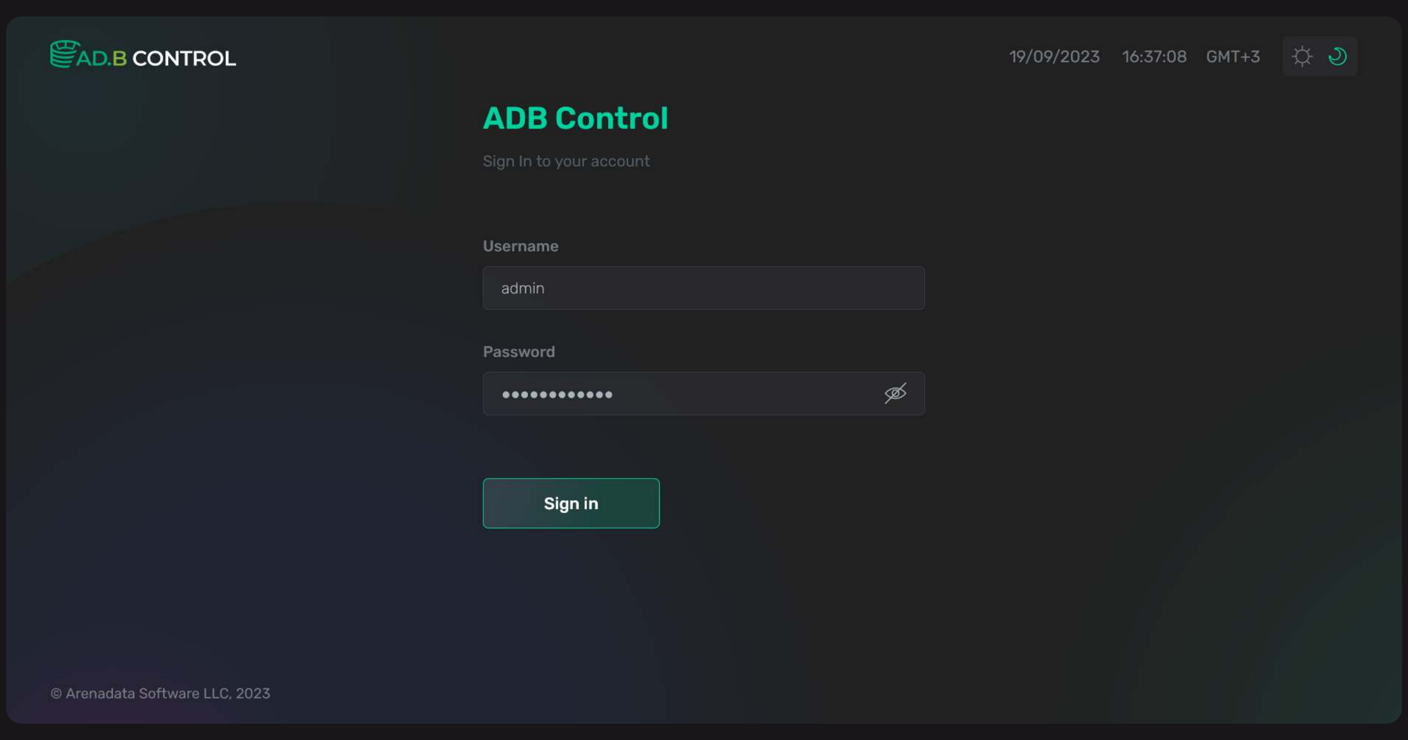
Log in to ADB Control

3. When you first log in, you should change a password for the `admin` user. Enter a new password in the ***New password*** field and confirm it in the ***Confirm new Password*** field. Click ***Save***. Note that your password must be greater than or equal to 12 characters by default. Any characters other than spaces are allowed to enter.



Change a password

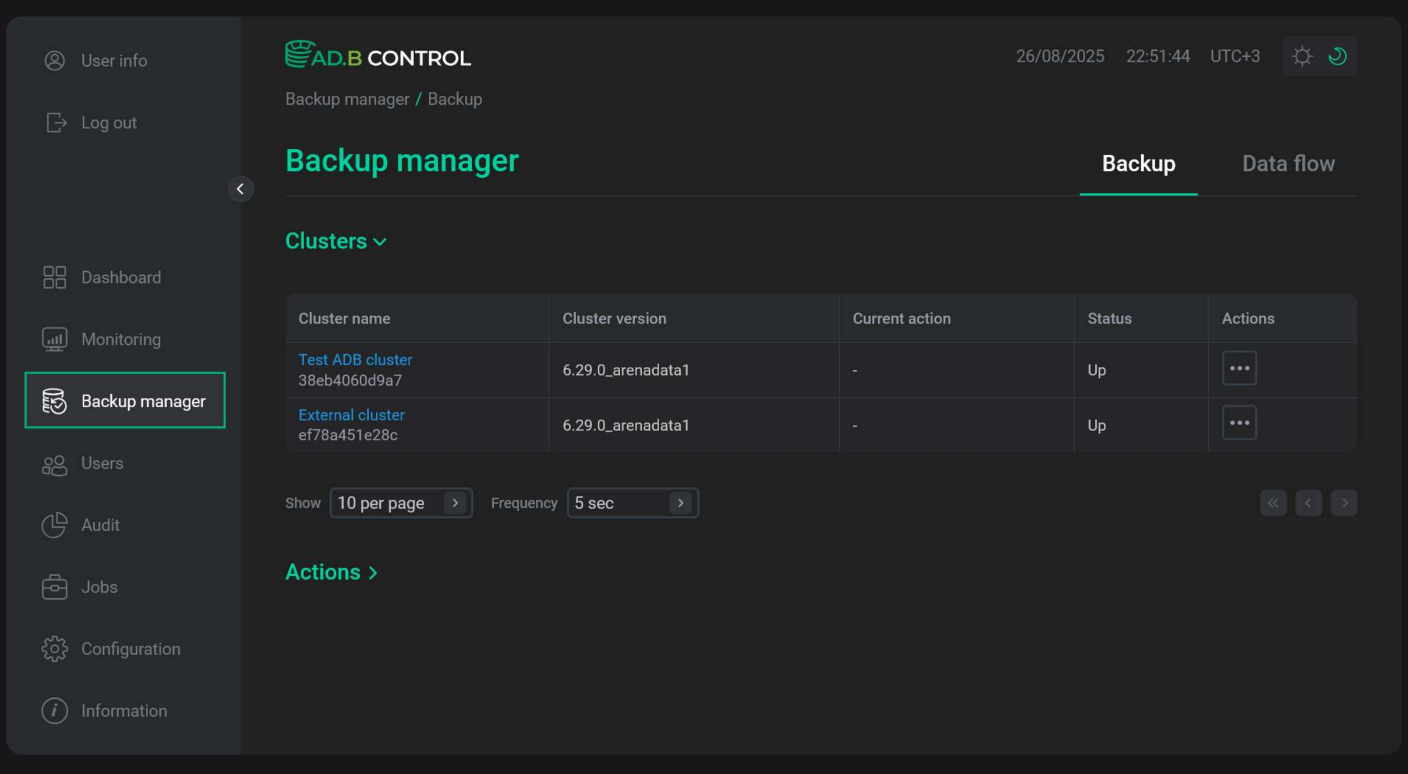
4. After saving a password, re-enter your login and a new password. Note that when a certain number of unsuccessful login attempts is reached, the user will be blocked — permanently or temporarily, depending on the settings.



Log in to ADB Control with a new password

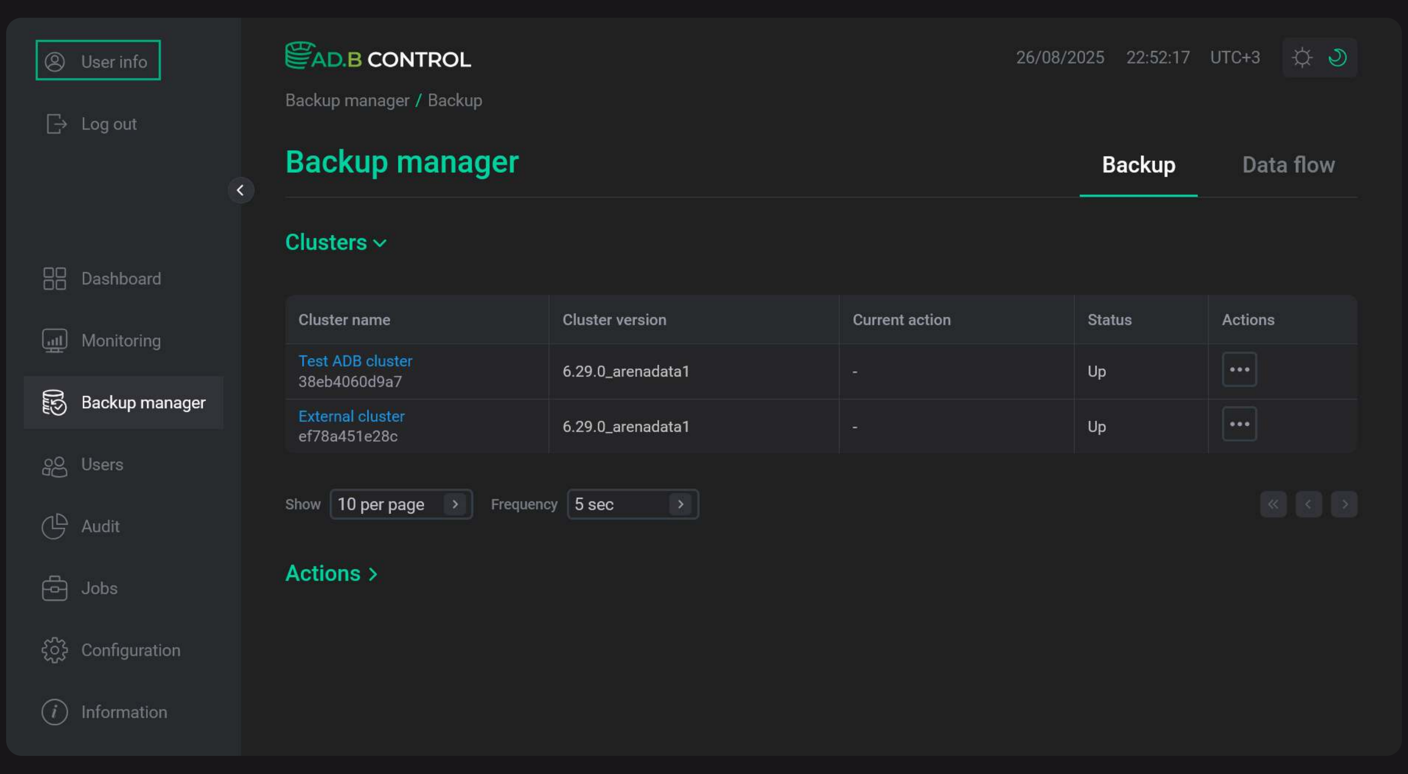
5. To start working with Arenadata DB Backup Manager (ADBM), on the next page, select the ***Backup manager*** item in the left navigation menu.

> **NOTE**
>
> The ***Backup manager*** page is available to users with the `Administrator`, `ADBM Administrator`, or `Owner` role if ADBM is installed in the ADB ES cluster.



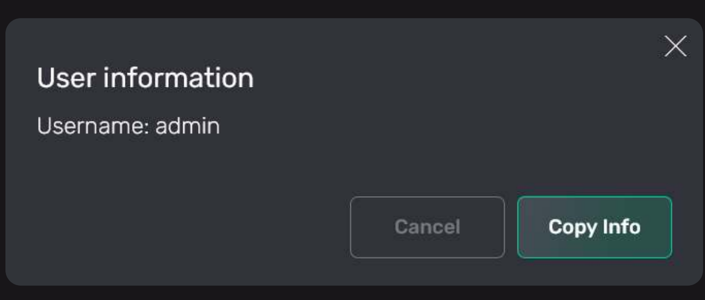
The "Backup manager" page

6. If you later need to get information about the current user, select the ***User info*** tab in the left navigation menu.



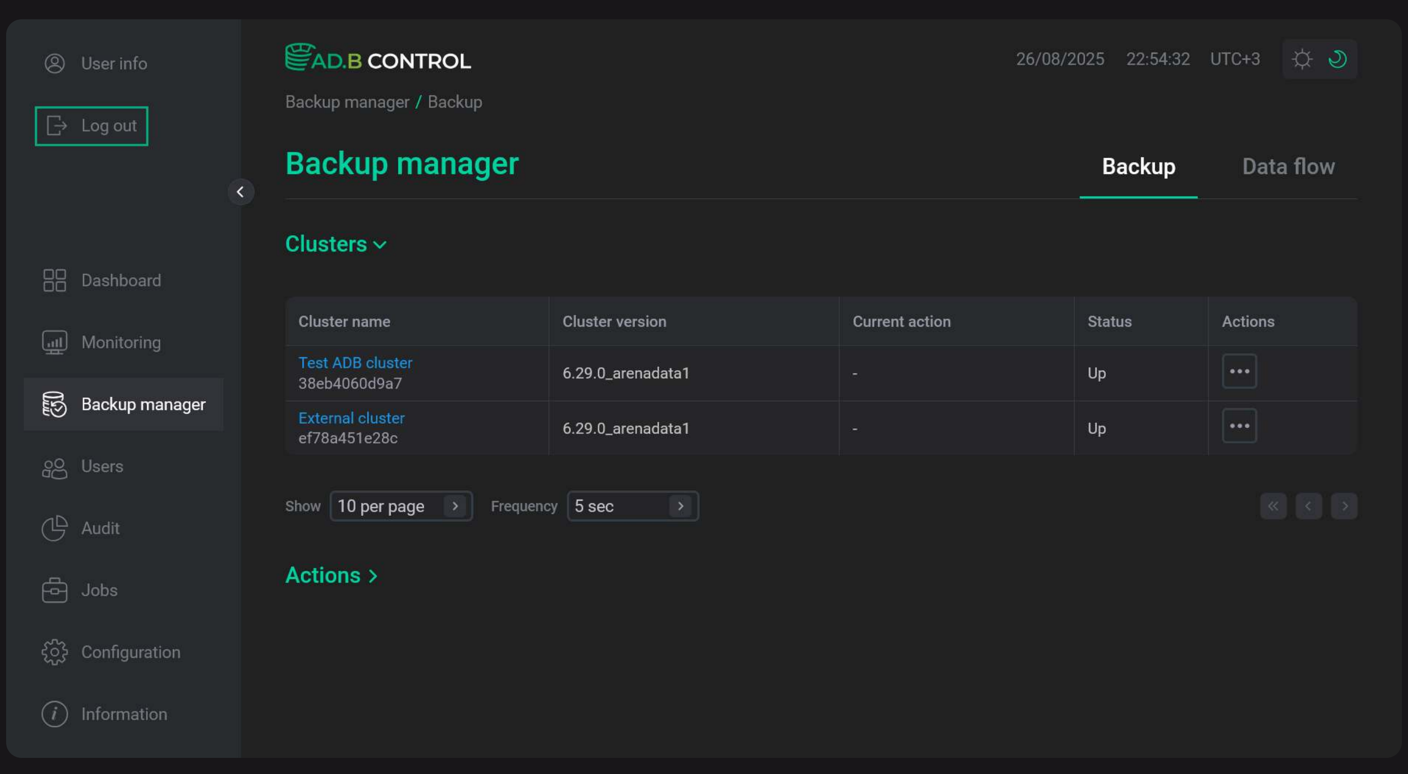
View user information

The modal window that opens displays the current user name, which can be copied by clicking ***Copy info***.



Copy user information

7. To log out from ADB Control, click ***Log out*** in the left navigation menu.



Log out from ADB Control

- To get information on how to add new users and configure access rights for them, refer to the following articles:
  - Basic authentication
  - LDAP authentication
  - Authorization
- You can change security policies (password length limits, authorization attempt limit, blocking rules) on the ***Configuration → Security*** page. For more information, see Configure security policies.

Basic authentication

Daria Barysheva

Contents

To Table of Contents

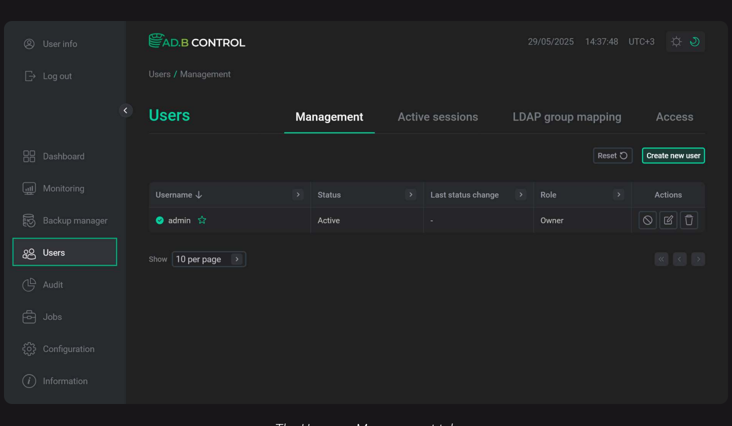
When basic authentication is used, users log in to ADB Control under the accounts previously created for them in the system. On the creation step, every user is assigned a role, which defines a set of permissions in ADB Control.

To manage users, you can use the ***Management*** tab on the ***Users*** page in the ADB Control web interface.

> IMPORTANT
>
> The operations listed below are available only to users with the `Owner` role. For more information on roles and permissions, see Authorization.

View a list of users

The ***Users*** → ***Management*** tab displays the following information on users.



The Users → Management tab

The "Users → Management" tab fields

Field	Description
Username	A user name. Depending on the user role and status (see below), the column can contain the following icons: — user is active; — user is blocked; — user is temporarily blocked; — user has the `Owner` role.
Status	A user status. Possible values: ▪ `Active` — user is active. ▪ `Blocked` — user is blocked. Users get this status after blocking manually or automatically according to the security policies. ▪ `Temporary blocked` — user is temporarily blocked according to the security policies.
Last status change	The last status change timestamp (see ***Status*** above) in the `DD/MM/YYYY HH:mm:ss` format
Role	A user role. Possible values: ▪ `ADBM Administrator` — ADBM administrator. ▪ `Administrator` — ADB Control administrator. ▪ `Advanced User` — user with advanced permissions. ▪ `Owner` — system owner. ▪ `Regular User` — ordinary user.

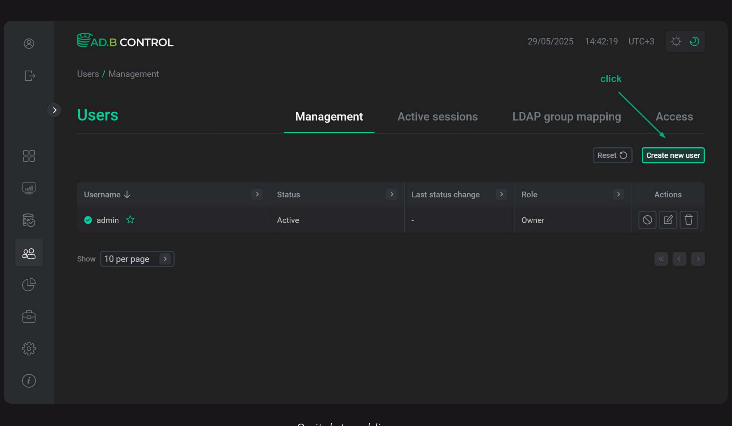
In the column headers of the table with a list of users, there are filters that you can use to select specific data. To open a filter, click the > icon. For those columns where the set of possible values is limited (e.g. ***Status***), you can select a value from the drop-down list. For some columns (e.g. ***Username***), the search value should be entered. For columns that show date and time (e.g. ***Last status change***), the time range can be selected from the calendar.

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

Add a user

To add a new user, follow the steps:

1. Click ***Create new user*** on the ***Users*** → ***Management*** tab.

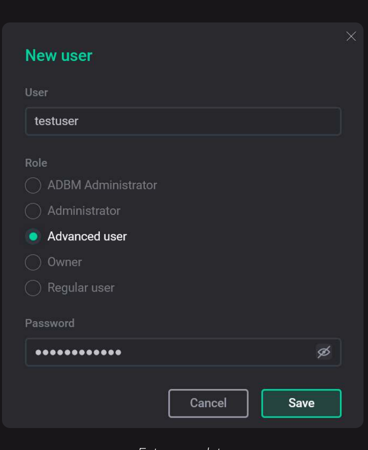


Switch to adding a user

2. In the window that opens, fill in the following fields:

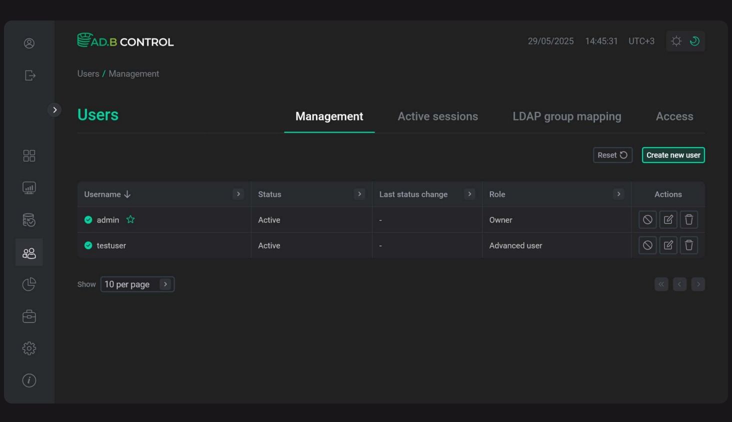
   - ***User*** — user name;
   - ***Role*** — user role;
   - ***Password*** — user password.

> TIP
>
> For information on login and password requirements, you can hover the mouse over the ? icon, which appears to the right of the ***User*** and ***Password*** fields during their filling.



Enter user data

3. Click ***Save***. As a result, a new user is displayed on the ***Users*** → ***Management*** tab.

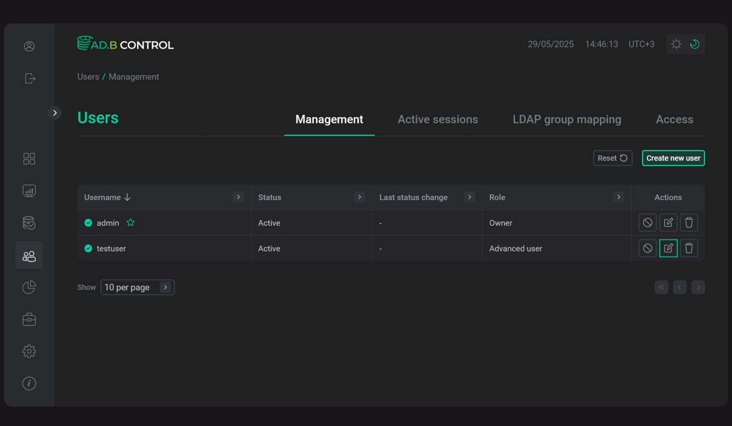


User is added

Edit a user

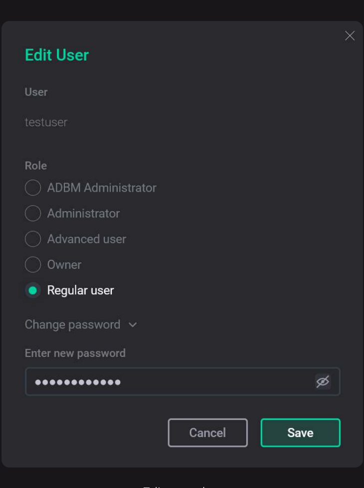
To edit user data, follow the steps:

1. Click the icon in the ***Actions*** column on the ***Users*** → ***Management*** tab.



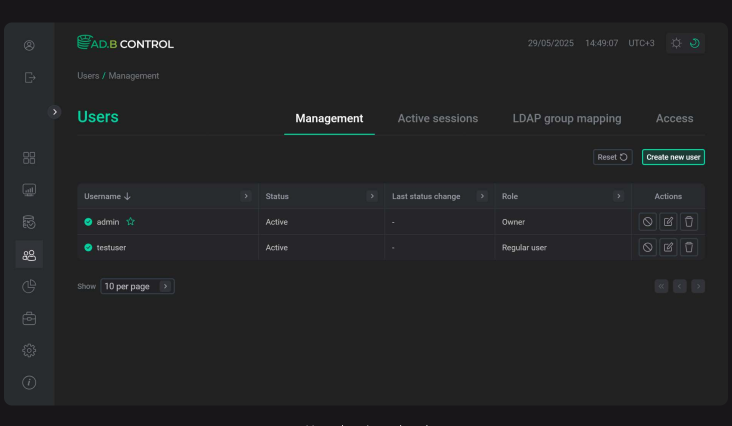
Switch to editing a user

2. In the window that opens, edit necessary fields. You can change a user role in the ***Role*** drop-down list and specify a new password in the ***Enter new password*** field (which is displayed after you open the ***Change password*** section). The ***User*** field value cannot be changed.



Edit user data

3. Click ***Save***. As a result, the user data is updated on the ***Users*** → ***Management*** tab.

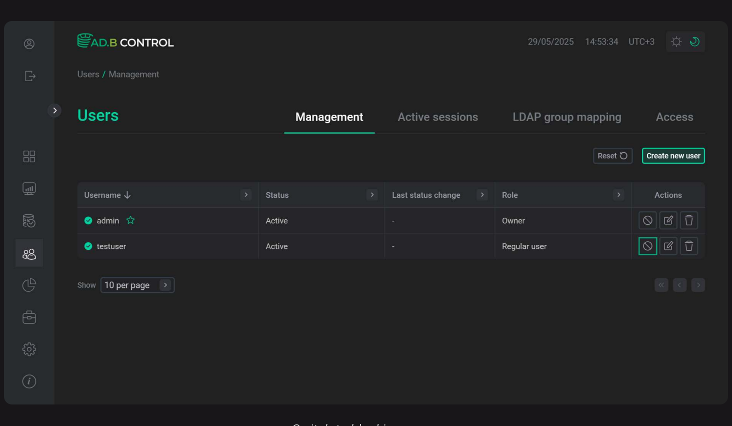


User data is updated

Block a user

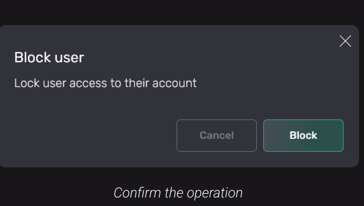
To block a user, follow the steps:

1. Click the icon in the ***Actions*** column on the ***Users*** → ***Management*** tab.



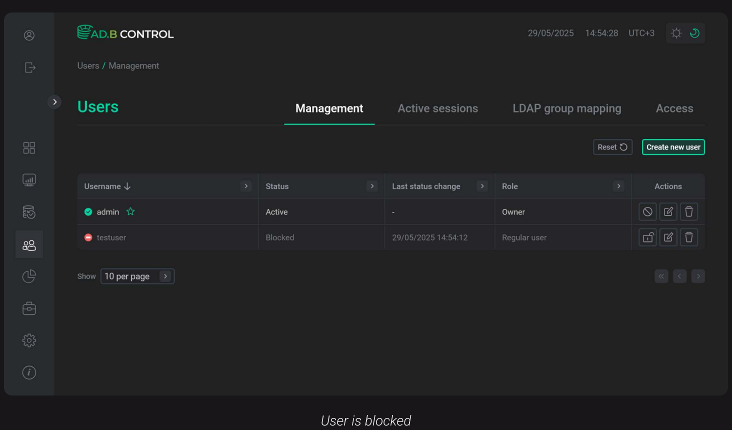
Switch to blocking a user

2. In the window that opens, confirm the operation by clicking ***Block***.



Confirm the operation

As a result, the user status is changed from ***Active*** to ***Blocked***. Note that the ***Last status change*** field is also updated on the ***Users*** → ***Management*** tab.

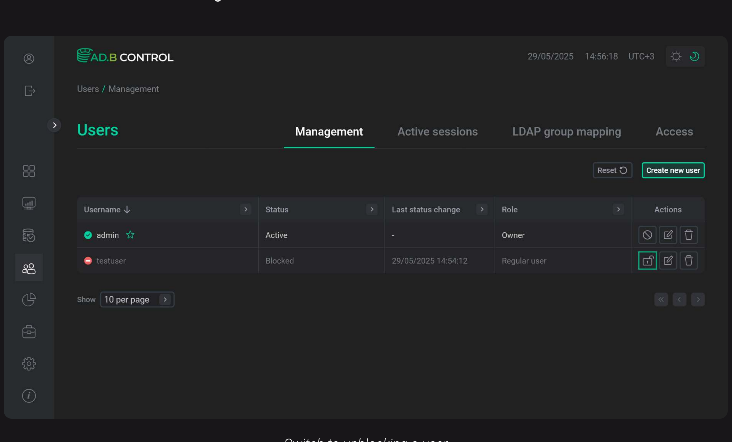


User is blocked

Unblock a user

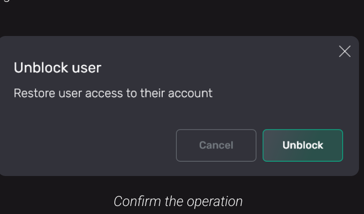
To unblock a previously blocked user, follow the steps:

1. Click the icon in the ***Actions*** column on the ***Users*** → ***Management*** tab.



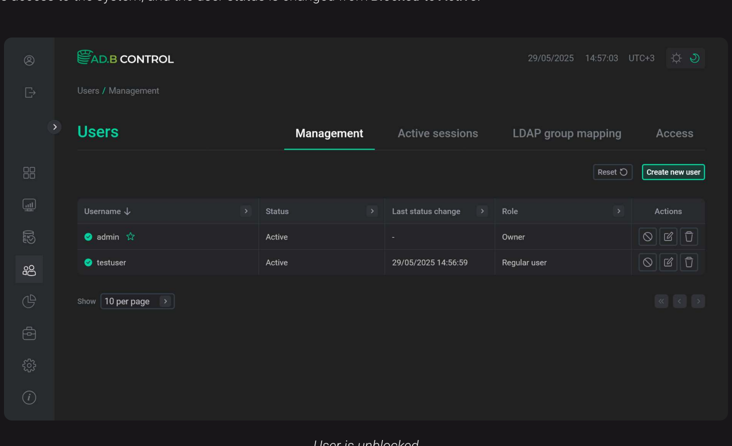
Switch to unblocking a user

2. In the window that opens, confirm the operation by clicking ***Unblock***.



Confirm the operation

As a result, the user regains access to the system and the user status is changed from ***Blocked*** to ***Active***.

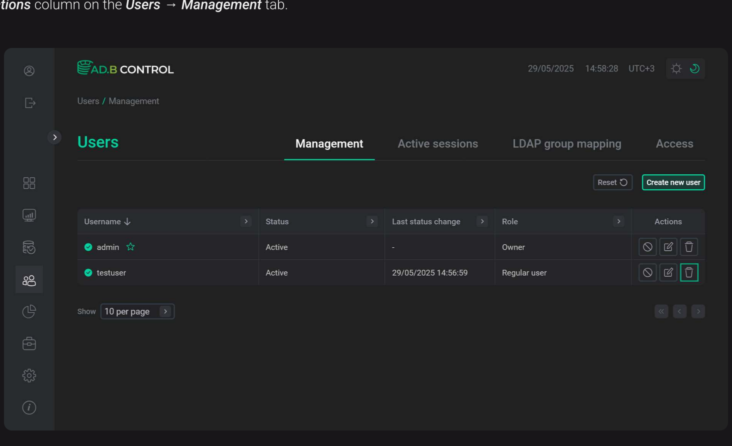


User is unblocked

Delete a user

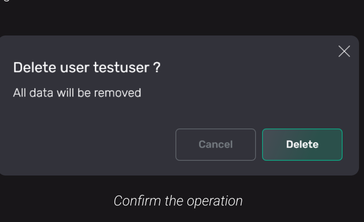
To delete an existing user, follow the steps:

1. Click the icon in the ***Actions*** column on the ***Users*** → ***Management*** tab.



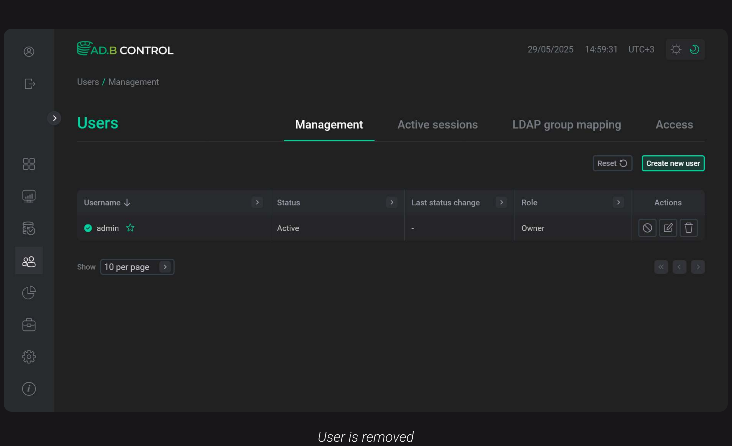
Switch to deleting a user

2. In the window that opens, confirm the operation by clicking ***Delete***.



Confirm the operation

As a result, the user data is removed from the system and no longer displayed on the ***Users*** → ***Management*** tab.



User is removed

LDAP authentication

Daria Barysheva

To Table of Contents

ADB Control supports Lightweight Directory Access Protocol (LDAP) user authentication. Unlike basic authentication, you do not need to add users manually via ADB Control when using LDAP. User accounts and passwords are checked on the selected LDAP server.

To configure LDAP authentication, follow the steps:

1. Ensure the target LDAP server is configured and available. ADB Control supports two LDAP implementations: Microsoft Active Directory (MS AD) and 389 Directory server as a part of FreeIPA.

2. Open the ADB Control service configuration page in ADCM.

3. Activate the ***UI LDAP authentication*** switcher.

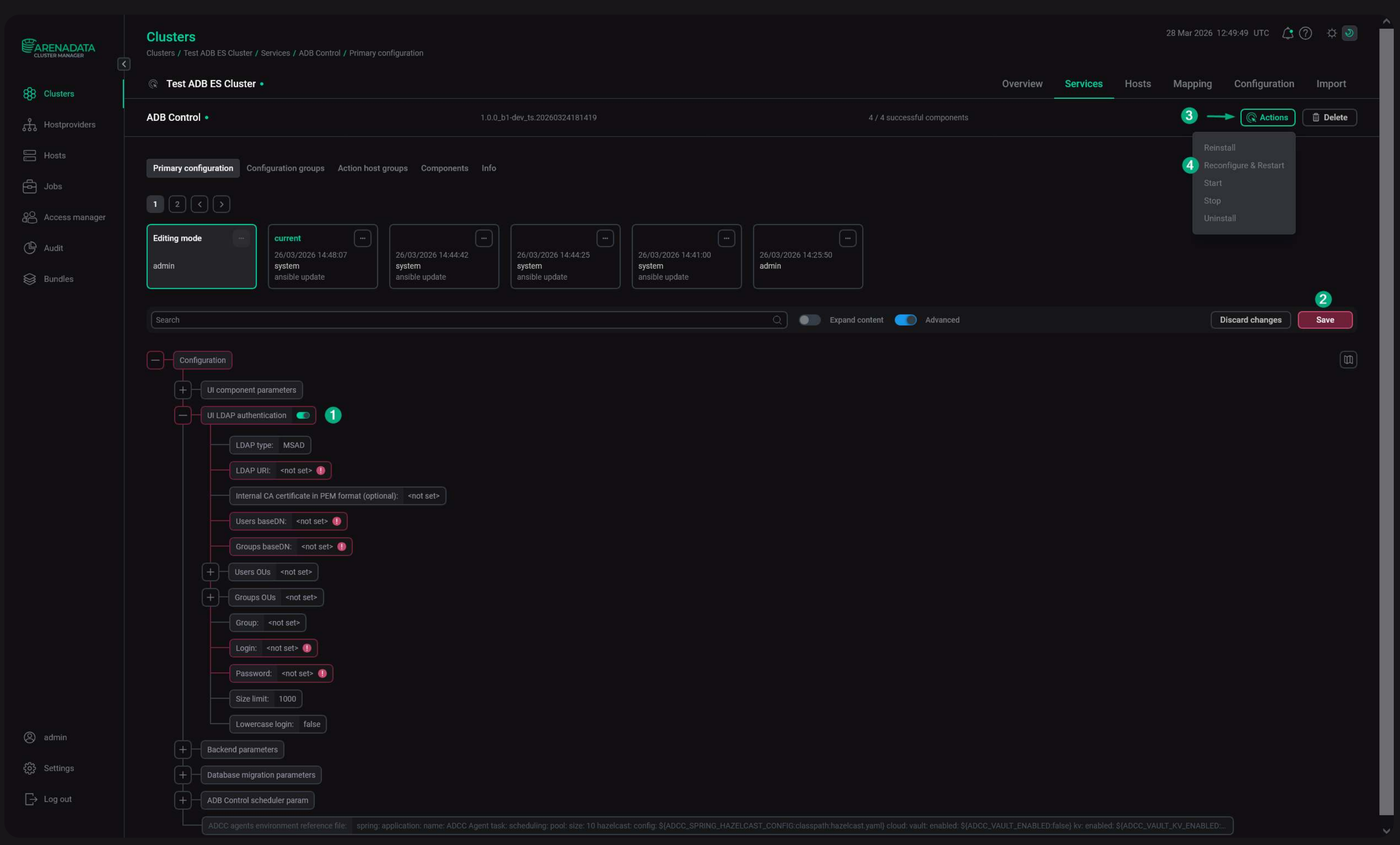
4. Fill in the following fields.

Field	Description	Example in MS AD
LDAP type	A type of the LDAP server. Possible values: ■ `MSAD` — Microsoft Active Directory; ■ `FreeIPA` — 389 Directory server as a part of FreeIPA.	MSAD
LDAP URI	A URI for connecting to the LDAP server. Requires the following format: `<scheme>://<host>:<port>`. For example, `ldap://example.com:389`, `ldaps://example.com:636`. If a port number is not specified, the standard ports are used: `389` for LDAP and `636` for LDAPS	ldap://10.92.2.66:389
Internal CA certificate in PEM format (optional)	The internal CA certificate or certificate chain in PEM format that signed the LDAP server certificate	—
Users baseDN	A limitation of the search scope of objects in the LDAP directory that is used in users search requests	dc=ad,dc=ranger-test
Groups baseDN	A limitation of the search scope of objects in the LDAP directory that is used in groups search requests	dc=ad,dc=ranger-test
Users OUs	A list of OUs for user search. If the option is set, users will be searched only in the specified OUs, otherwise — within full search scope specified in ***Users baseDN***. Enter each `OU` value on a separate line using the `ou=<name>` format (e.g. `ou=users1`)	ou=Peoples
Groups OUs	A list of `OU` for group search. If the option is set, groups will be searched only in the specified OUs, otherwise — within full search scope specified in ***Groups baseDN***. Enter each `OU` value on a separate line using the `ou=<name>` format (e.g. `ou=groups1`)	ou=Groups
Group	A group name that is used for service requests to the LDAP server	—
Login	A user name that is used for service requests to the LDAP server	cn=admin,dc=ad,dc=ranger-test
Password	A user password that is used for service requests to the LDAP server	A password of the `admin` user
Size limit	A maximum number of records that can be returned by the LDAP server	1000
Lowercase login	Whether to convert user names to lowercase	false

IMPORTANT

- Different LDAP implementations use different names for object types and identifiers. It is recommended to refine the settings format for a specific LDAP implementation from the LDAP server administrator.
- To use LDAP over SSL (LDAPS) starting with ADB Control 4.10.3, do the following:
  - Use the `ldaps` protocol in the ***LDAP URI*** field.
  - Fill in the ***HTTPS parameters enable*** section.
  - Fill in the ***Internal CA certificate in PEM format (optional)*** field. The certificate should be replicated to all hosts used by ADB Control (imported to *truststore.jks*).

5. Click ***Save***. Then apply the ***Reconfigure & Restart*** action to the ADB Control service.



Configure LDAP authentication for ADB Control via ADCM

6. In the ADB Control web interface, map LDAP user groups to ADB Control roles. This is necessary for LDAP users to get the appropriate permissions to work in ADB Control. For more information, see Authorization.

If all steps are completed successfully, users can connect to ADB Control using the accounts registered for them on the LDAP server.

Authorization

[illegible]

Contents

[illegible]

Roles and access permissions

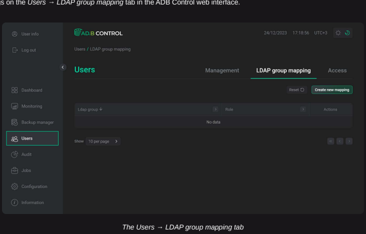
[illegible]

[illegible]						
Kill all queries	Ability to terminate and cancel all transactions	No	No	Yes	Yes	No
View resource groups	Ability to view all resource groups (including detailed information)	No	No	Yes	Yes	No
Change resource group configuration	Ability to change resource group configuration	No	No	Yes	Yes	No
View relation audit page	Ability to view the Audit → Relations page where statistics on the number of access attempts to ADB relations are displayed	No	No	Yes	Yes	No
View secured relation audit	Ability to view a table with secured statistics on ADB relations in a separate modal window	No	No	Yes	Yes	No
View Operations audit page	Access to the Audit → Operations page where statistics on launches of various operations in ADB Control is displayed	No	No	Yes	Yes	No
View Authorization audit page	Access to the Audit → Authorizations page where information on user authorizations in ADB Control/ADB is displayed	No	No	Yes	Yes	No
View ADB jobs	Access to the Jobs → ADB Control page where information on scheduled (*Schedule*) and completed (*Audit*) jobs in ADB Control is displayed	No	No	Yes	Yes	No
View backup manager jobs	Access to the Jobs → Backup manager page where information on scheduled (*Schedule*) and completed (*Audit*) jobs in ADB Backup Manager is displayed	No	No	Yes	Yes	Yes
View backup manager page	Access to the *Backup manager* page with the ability to view information on clusters, actions, backups, and restores	No	No	Yes	Yes	Yes
Run backup manager actions	Ability to run actions in ADB Backup Manager	No	No	Yes	Yes	Yes
Create backup manager configuration	Ability to create and edit configurations in ADB Backup Manager	No	No	Yes	Yes	Yes
View ADB information page	Access to the Information → ADB Control page where the current status of ADB Control components is displayed	No	No	Yes	Yes	No
View backup manager information page	Access to the Information → Backup manager page where the current status of ADB Backup Manager components is displayed	No	No	Yes	Yes	Yes
View configuration page	Access to the *Configuration* page with the ability to manage clusters and configure policies for jobs and security	No	No	No	Yes	No
View users page	Access to the *Users* page with the ability to manage users (create, edit, block, delete), set mappings with LDAP groups, and set advanced permissions	No	No	No	Yes	No
View sessions	Access to the Monitoring → Sessions page functionality	No	No	Yes	Yes	No
Manage user sessions	Ability to terminate sessions of ADB Control users on the *Users → Active sessions* page	No	No	Yes	Yes	No

Configure role mappings with LDAP groups

If asic authenticatio is used, roles are assigned to users at the reatio step. However, if DAP authenticatio is configured, you need to map user groups of the selected LDAP server to roles in ADB Control. It allows users to get appropriate permissions in the monitoring system.

You can manage such mappings on the *Users → LDAP group mapping* tab in the ADB Control web interface.

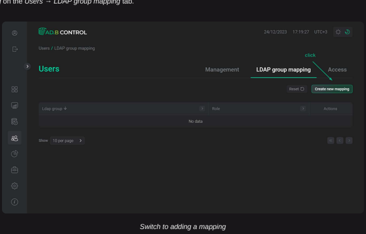


The Users → LDAP group mapping tab

Add a mapping

To add a new mapping, follow the steps:

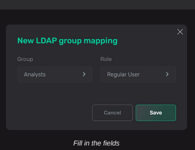
1. Click *Create new mapping* on the *Users → LDAP group mapping* tab.



Switch to adding a mapping

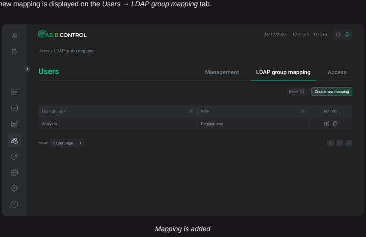
2. In the window that opens, fill in the following fields:
   - *Group* — a user group name on the LDAP server. A drop-down list with group names becomes available after the DAP authenticatio is successfully configured.
   - *Role* — a l in ADB Control.

> IMPORTANT
>
> Each LDAP group can be mapped only to one role in ADB Control.



Fill in the fields

3. Click *Save*. As a result, a new mapping is displayed on the *Users → LDAP group mapping* tab.

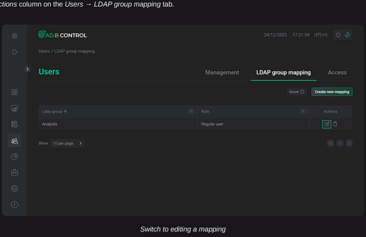


Mapping is added

Edit a mapping

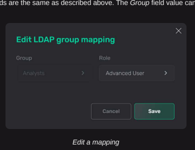
To edit an existing mapping, follow the steps:

1. Click the icon in the *Actions* column on the *Users → LDAP group mapping* tab.



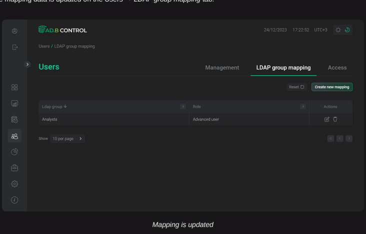
Switch to editing a mapping

2. In the window that opens, edit necessary fields. All fields are the same as described above. The *Group* field value cannot be changed.



Edit a mapping

3. Click *Save*. As a result, the mapping data is updated on the *Users → LDAP group mapping* tab.

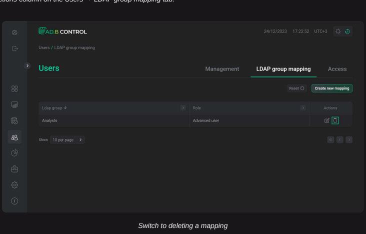


Mapping is updated

Delete a mapping

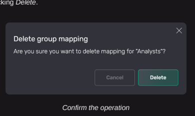
To delete an existing mapping, follow the steps:

1. Click the icon in the *Actions* column on the *Users → LDAP group mapping* tab.



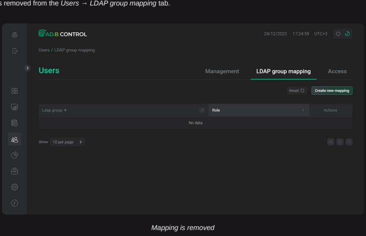
Switch to deleting a mapping

2. In the window that opens, confirm the operation by clicking *Delete*.



Confirm the operation

As a result, the mapping is removed from the *Users → LDAP group mapping* tab.



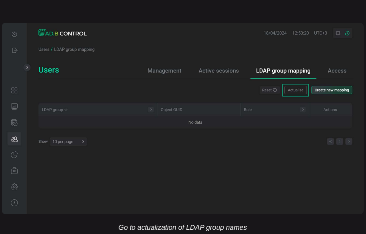
Mapping is removed

Actualize LDAP group names

> NOTE
>
> If DAP authenticatio is configured, LDAP groups are synchronized automatically each time when ADB Control is being started. The following feature is designed for on-demand updates of LDAP group names.

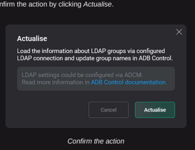
Starting with ADB Control 7 , you can actualize LDAP group names in the previously reate mappings by running synchronization with an LDAP server on the *Users → LDAP group mapping* page. To run synchronization:

1. Ensure that LDAP authentication for ADB Control is successfully onfigure and ADB Control roles are appe to LDAP groups on the *Users → LDAP group mapping* page.
2. Click *Actualise*.



Go to actualization of LDAP group names

3. Read the information in the window that opens and confirm the action by clicking *Actualise*.

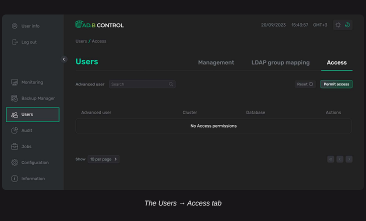


Confirm the action

Set advanced permissions

Users with the `Advanced User` role can be granted extended access rights to specific ADB databases. With such rights, users are able to view all transactions and commands running in the respective databases (regardless of who launched them) and interrupt or cancel them if necessary (see `View some queries` and `Kill some queries` permissions in the table bov).

You can manage such advanced permissions on the *Users → Access* tab in the ADB Control web interface.

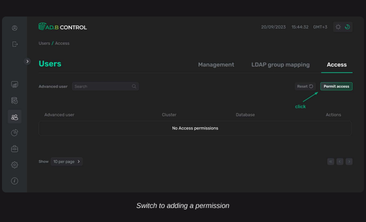


The Users → Access tab

Add a permission

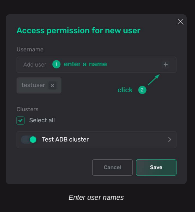
To add a new permission, follow the steps:

1. Click *Permit access* on the *Users → Access* tab.



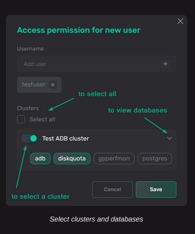
Switch to adding a permission

2. In the window that opens, fill in the following information:
   - In the *Username* field, specify one or several users. To add each user, add a user name and then click +. As a result, a list of users is displayed under the *Username* field. You can delete wrong records if necessary.



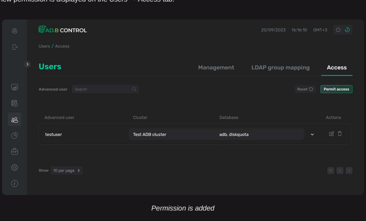
Enter user names

   - In the *Clusters* section, select ADB clusters and databases to be accessed by the users added in the previous step. To select all clusters with all databases, set the *Select all* flag (it is set by default).



Select clusters and databases

3. Click *Save*. As a result, a new permission is displayed on the *Users → Access* tab.

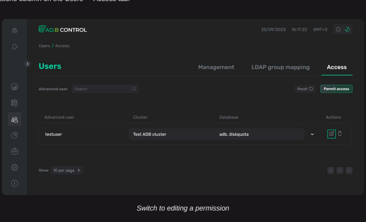


Permission is added

Edit a permission

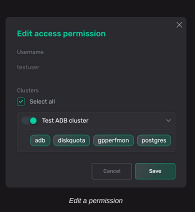
To edit an existing permission, follow the steps:

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



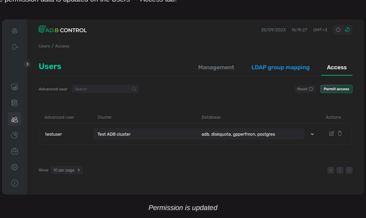
Switch to editing a permission

2. In the window that opens, edit necessary fields. All fields are the same as described bov . The *Username* field value cannot be changed.



Edit a permission

3. Click *Save*. As a result, the permission data is updated on the *Users → Access* tab.

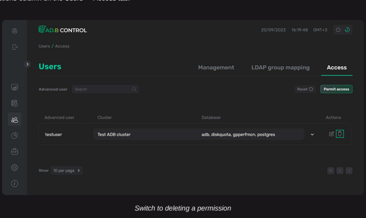


Permission is updated

Delete a permission

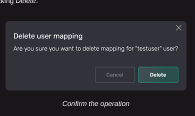
To delete an existing permission, follow the steps:

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



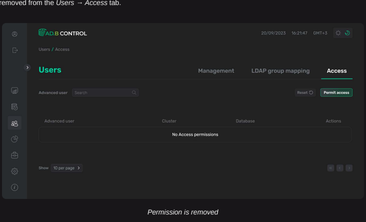
Switch to deleting a permission

2. In the window that opens, confirm the operation by clicking *Delete*.



Confirm the operation

As a result, permission is removed from the *Users → Access* tab.



Permission is removed

User sessions

Daria Barysheva

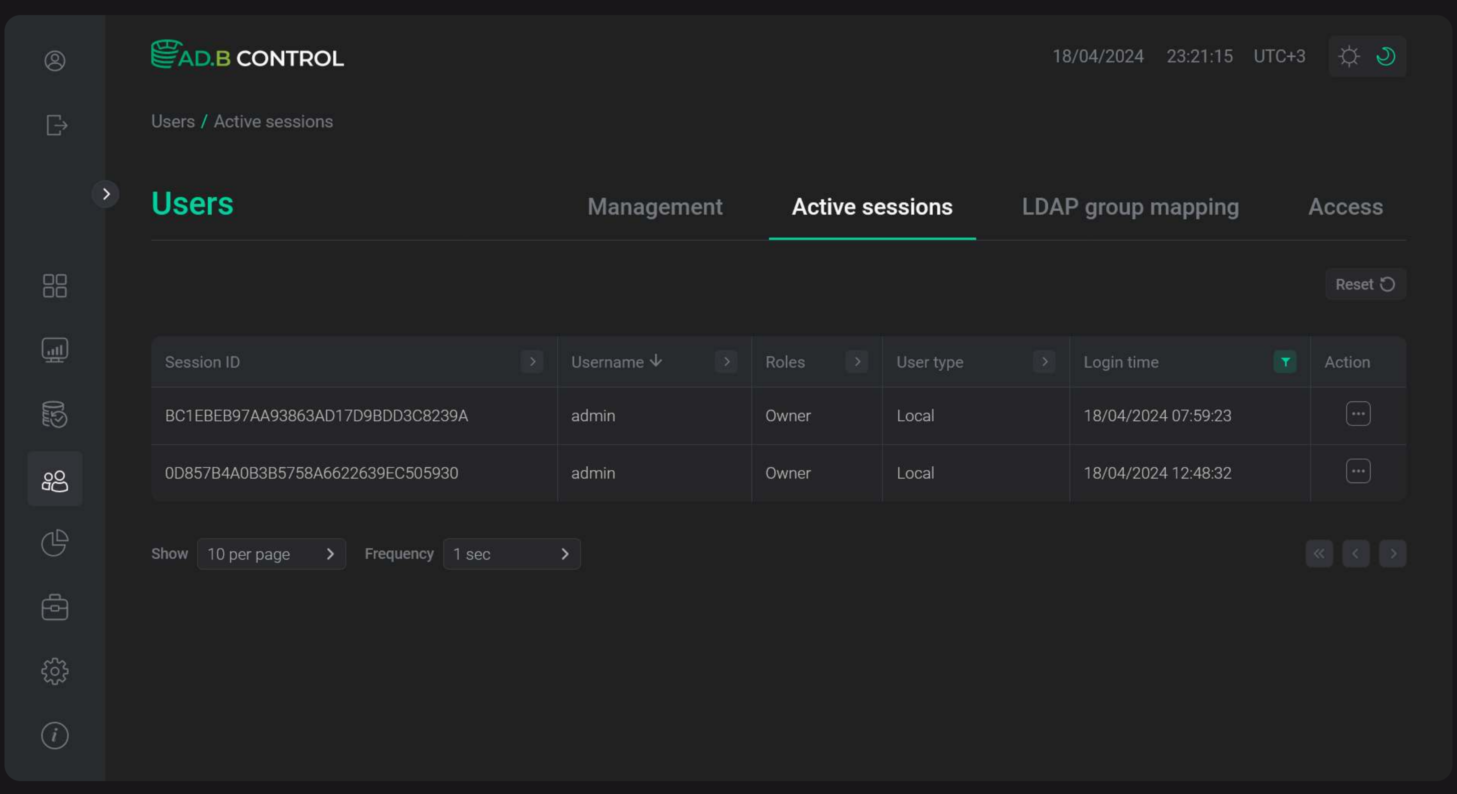
Contents

To Table of Contents

The ***Users → Active sessions*** tab is designed to monitor active sessions of ADB Control users. This page also allows you to terminate selected sessions.

View a list of sessions

On the ***Users → Active sessions*** tab, there is a table with the following information on sessions.



List of sessions

The "Users → Active sessions" tab fields

Field	Description
Session ID	A session identifier
Username	A name of the user who connected to ADB Control
Roles	A role of the user who connected to ADB Control
User type	A type of the user who connected to ADB Control. Possible values: ▪ `AD` — an LDAP user; ▪ `Local` — an ADB Control user who uses basic authentication; ▪ `System` — a system user.
Login time	A session start timestamp

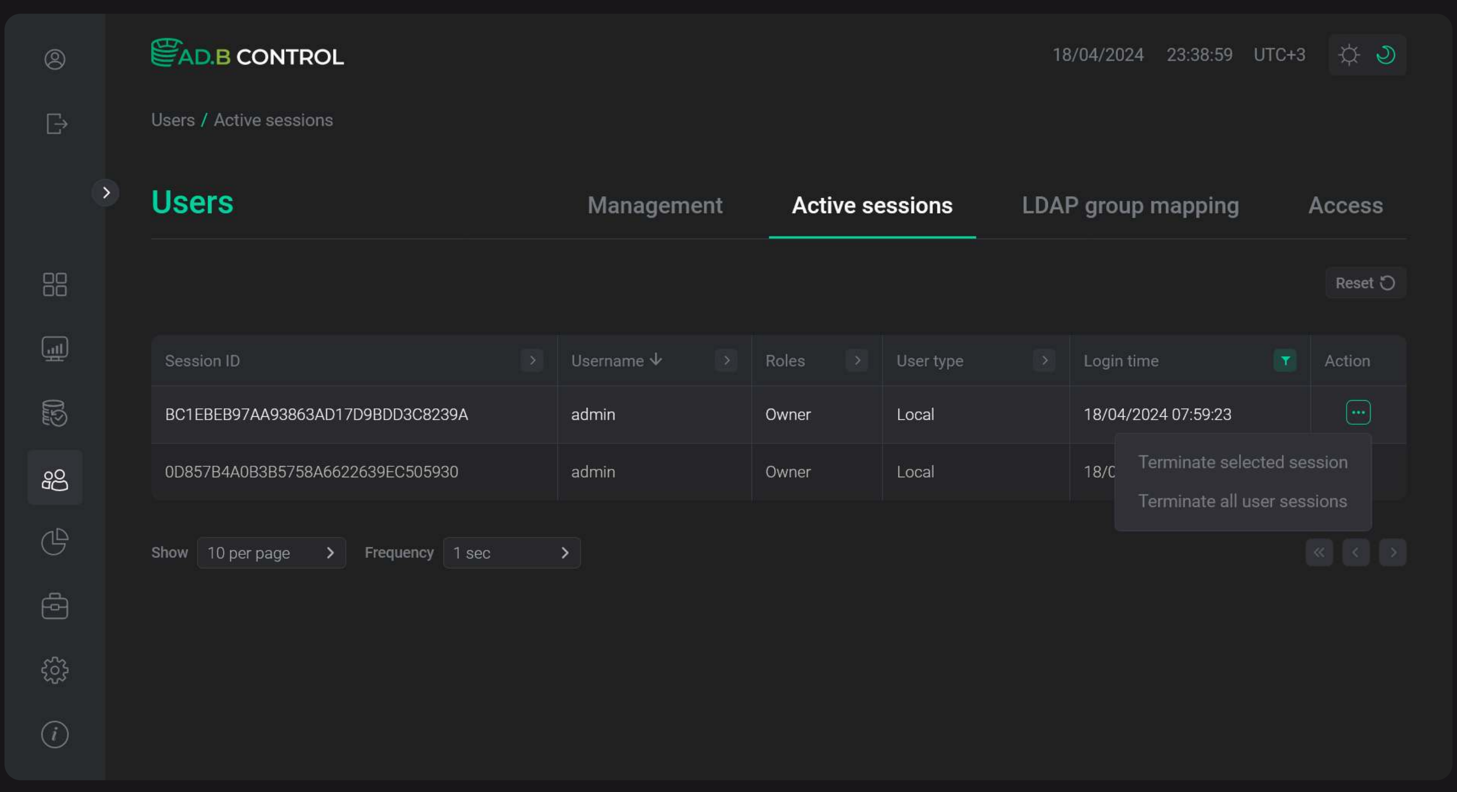
In the column headers of the table with a list of sessions, there are filters that you can use to select specific data. To open a filter, click the icon. For those columns where the set of possible values is limited (e.g. ***Roles***), you can select a value from the drop-down list. For some columns (e.g. ***Session ID***), the search value should be entered. For columns that show date and time (e.g. ***Login time***), the time range can be selected from the calendar.

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

Terminate sessions

> **NOTE**
>
> Termination of user sessions is available for ADB Control users with appropriate permissions (see `Manage user sessions` in the Authorization article).

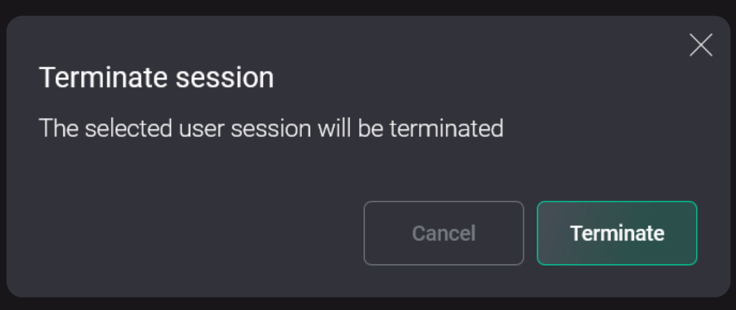
To terminate a session, click in the ***Actions*** column on the ***Users → Active sessions*** tab.



Go to terminating a session

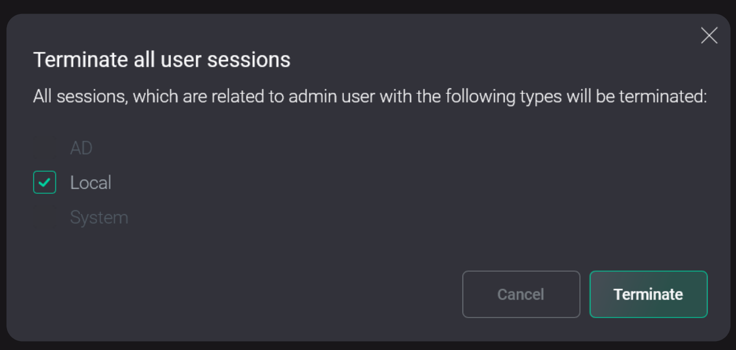
Select one of the options:

- ***Terminate selected session*** — terminates only one selected session. To confirm the action, click ***Terminate*** in the window that opens.



Confirm termination of one session

- ***Terminate all user sessions*** — terminates all sessions of the user who started the selected session. In the window that opens, you can set the user type (see ***User type*** above). To confirm the action, click ***Terminate***.



Confirm termination of several sessions

Cluster actions

Anton Vasilev

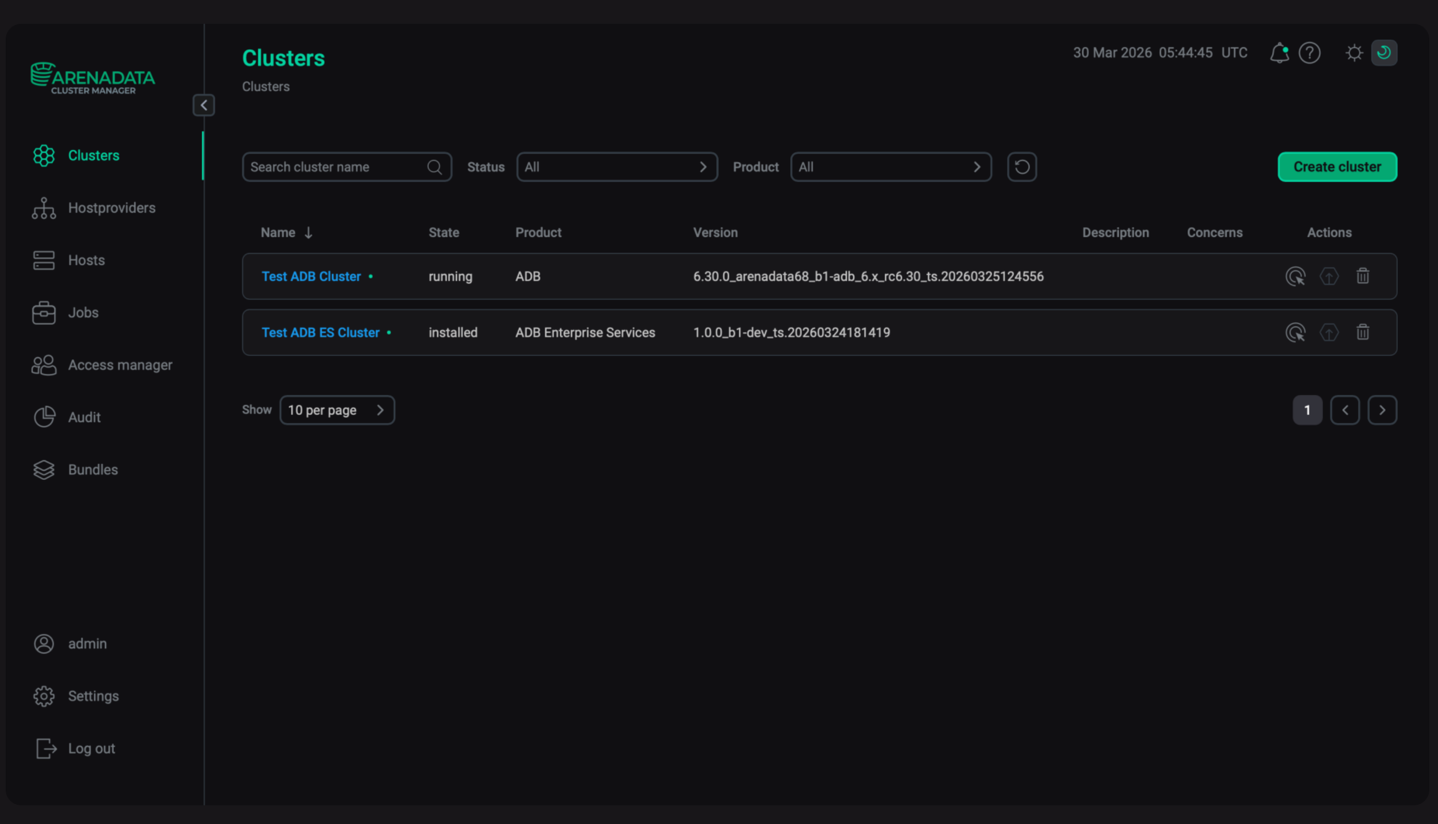
Contents

To Table of Contents

This article describes the actions available for the ADB ES cluster in the ADCM UI.

Overview

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

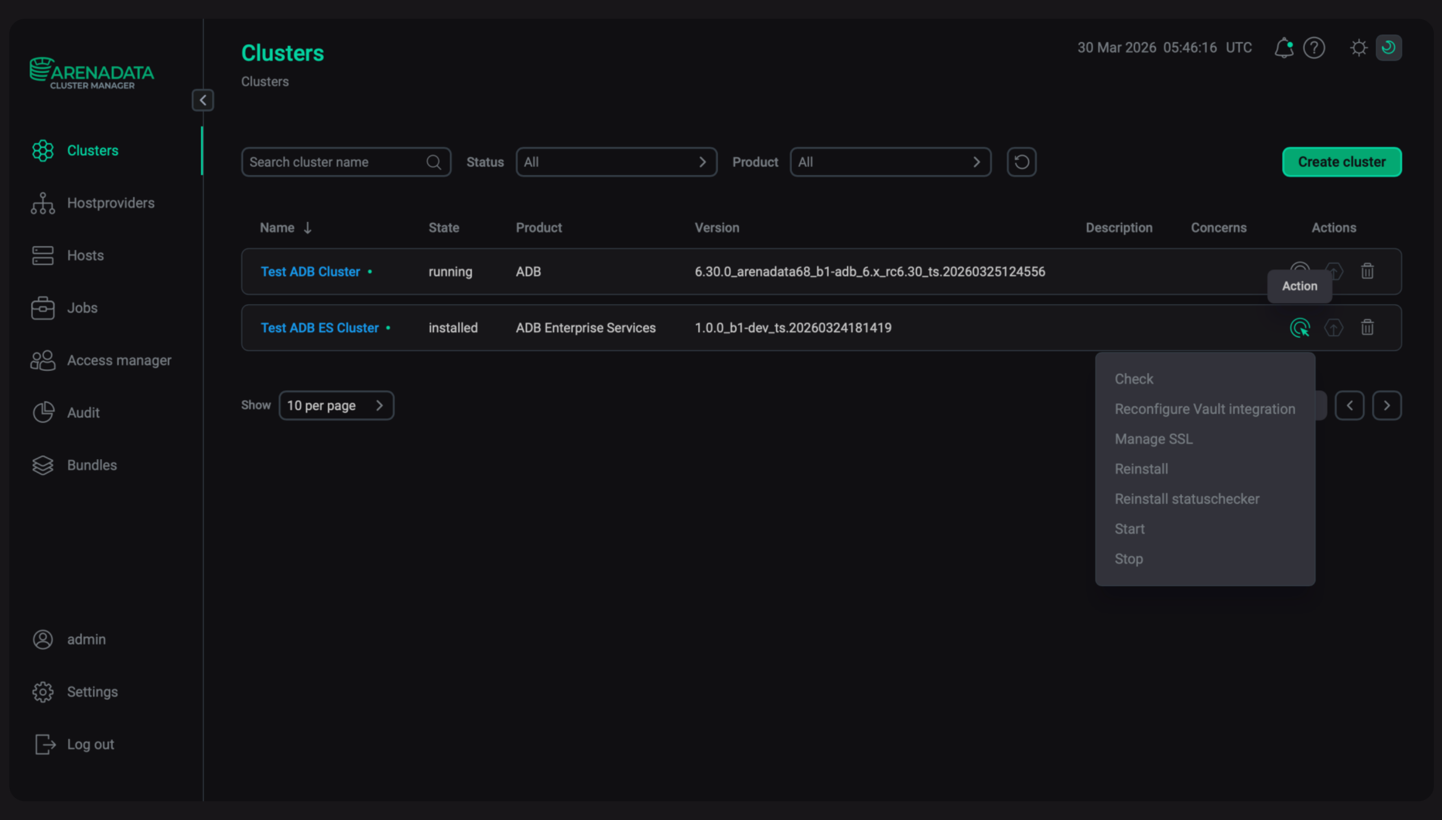


The Clusters page

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

The ***Actions*** column shows icons for managing the cluster:

- — opens the drop-down list that offers actions to manage the cluster.



Open a list of available cluster actions

- — indicates whether a new version of a bundle is available and allows you to upgrade the cluster.
- — deletes information about the cluster from ADCM (it does not remove ADB ES or make any changes to hosts that belong to the cluster).

IMPORTANT

When upgrading ADB ES, follow the order:

1. Upgrade an ADB ES cluster.
2. Upgrade agents on the ADB side (see the ***Reinstall*** service action for ADBC agents and ADBM agents).

A set of cluster actions (available after clicking) depends on the current ADB ES cluster status.

Available actions depending on the ADB ES status

Status	Condition	Available actions
created	The ADB ES cluster was created via ADCM, but not installed yet	- Precheck - Install
installed	The ADB ES cluster was successfully installed via ADCM	- Check - Reconfigure Vault integration - Reinstall - Reinstall statuschecker - Start - Stop

For information on the ***Precheck*** and ***Install*** actions, see Install a cluster. The actions for an installed cluster are described below.

Check

The ***Check*** action verifies that all hosts, components, and services are configured according to the ADB ES cluster requirements (similarly to what the Precheck action does). Additionally, it checks the status of each ADB ES service.

Reconfigure Vault integration

The ***Reconfigure Vault integration*** action is used to apply changes to the Vault integration parameters. These parameters are available on the ***Configuration*** tab of the cluster page and allow you to store secrets of ADB ES services in HashiCorp Vault.

IMPORTANT

- Before editing Vault integration parameters, ensure that Vault is installed and configured.
- Each time you edit and save parameters in the ***Vault integration*** section, run the cluster action ***Reconfigure Vault integration***.
- Currently, Vault can be used to store authentication parameters of ADB Control and ADBM.

Reinstall

The ***Reinstall*** action reinstalls the ADB ES cluster.

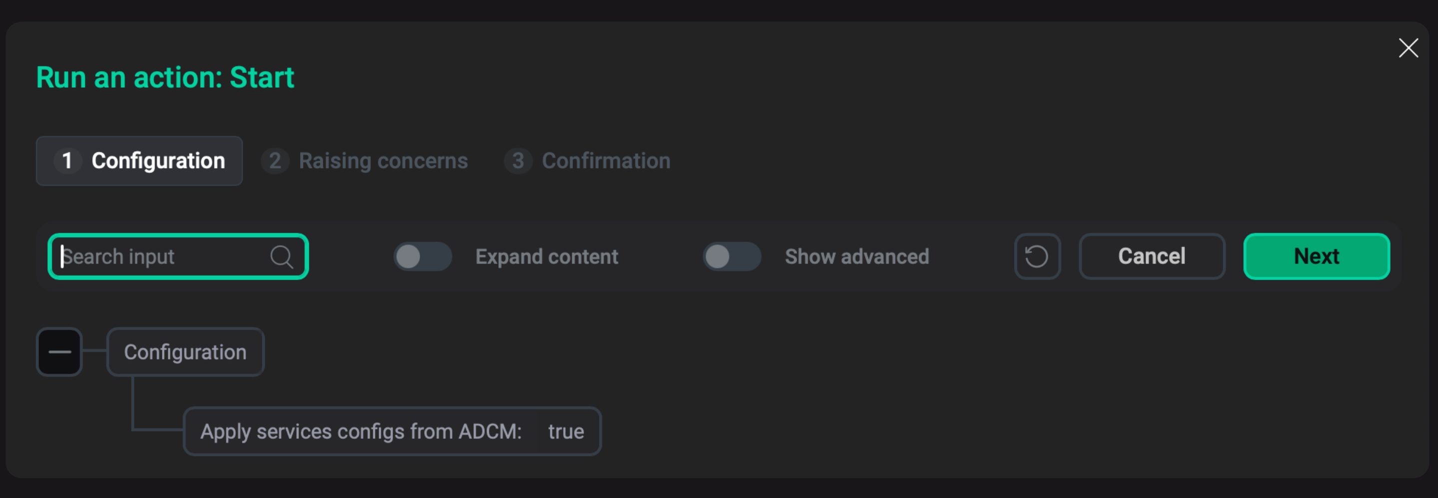
Reinstall statuschecker

The ***Reinstall statuschecker*** action reconfigures and restarts the statuschecker for all cluster services. Use this action when migrating a cluster to a new ADCM server.

Start

The ***Start*** action starts all services in the ADB ES cluster.

After you select the action, a dialog box opens where you can set the value for the ***Apply services configs from ADCM*** option. Enable this option to apply all changes made in the configurations of the services. Otherwise, the services will just start without applying the changes.



The Start window

Stop

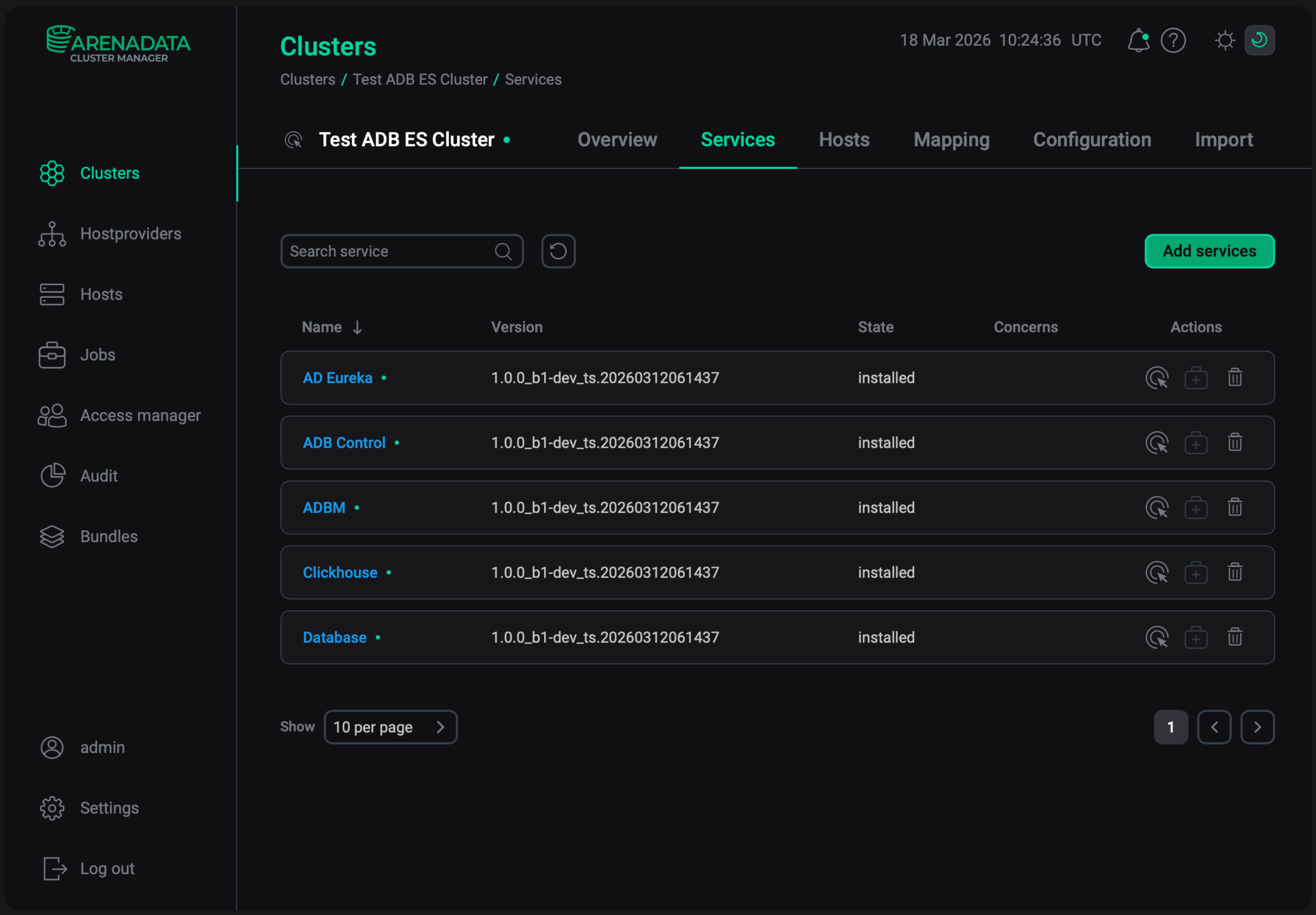
Stops all services of the ADB ES cluster.

Service actions

Anton Vasilev

To Table of Contents

ADCM UI provides actions to manage services. The actions are available on the ***Services*** tab. To open this tab, click an ADB ES cluster name on the ***Clusters*** page and switch to the ***Services*** tab.



ADB ES cluster services in the ADCM interface

Refer to ADCM documentation for an overview of the ***Services*** tab and common actions.

Refer to the following articles that describe actions of ADB ES cluster services:

- AD Eureka
- ADB Control
- ADBM
- Clickhouse
- Database
- Monitoring

AD Eureka service actions

Anton Vasilev

Contents

To Table of Contents

This article describes the actions available for the AD Eureka service in the ADCM web interface.

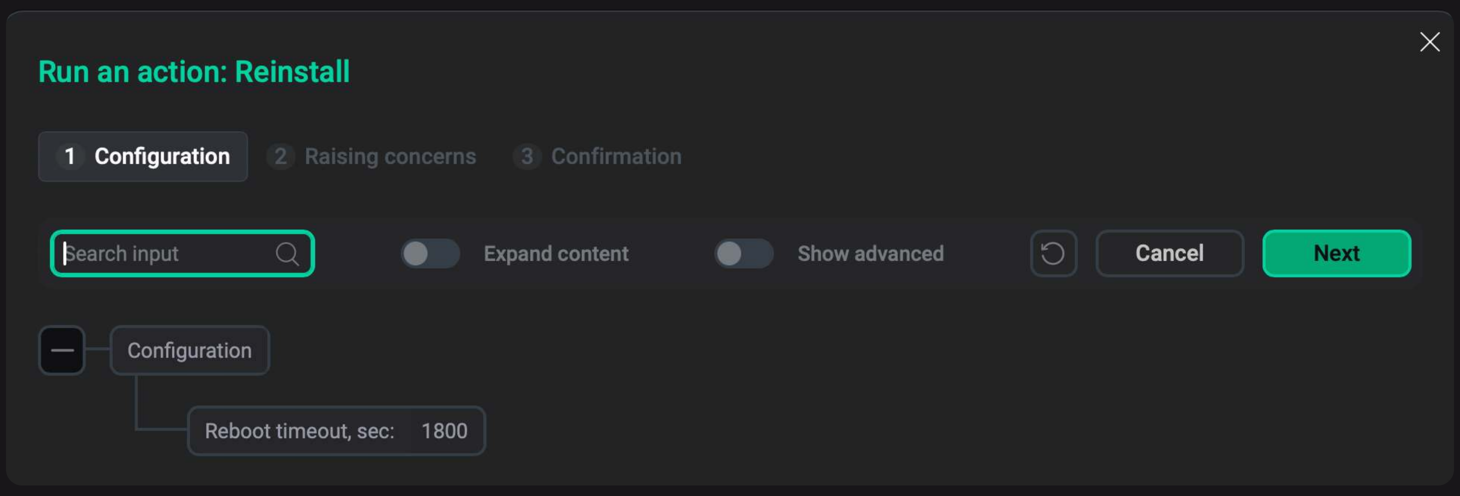
Install

Installs the AD Eureka service. This action is available only if a service has been added but not yet installed.

Reinstall

Reinstalls the AD Eureka service.

On the ***Configuration*** tab of the window that opens, in the ***Reboot timeout, sec*** field, enter the timeout to wait for hosts to restart (in seconds). The default value is `1800`. The value must be at least `600`.

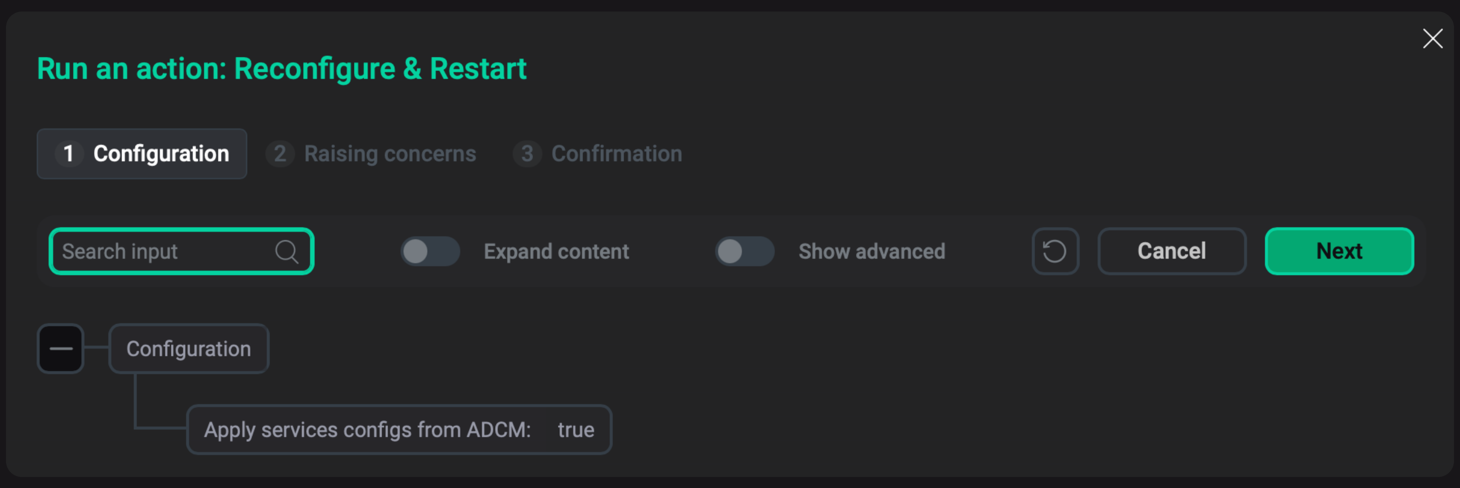


The Reinstall window

Reconfigure & Restart

Applies the current configuration to the service and restarts the service.

In the window that opens, use the ***Apply services configs from ADCM*** flag to determine whether to apply the current service configuration (`true`) or just restart the service (`false`).

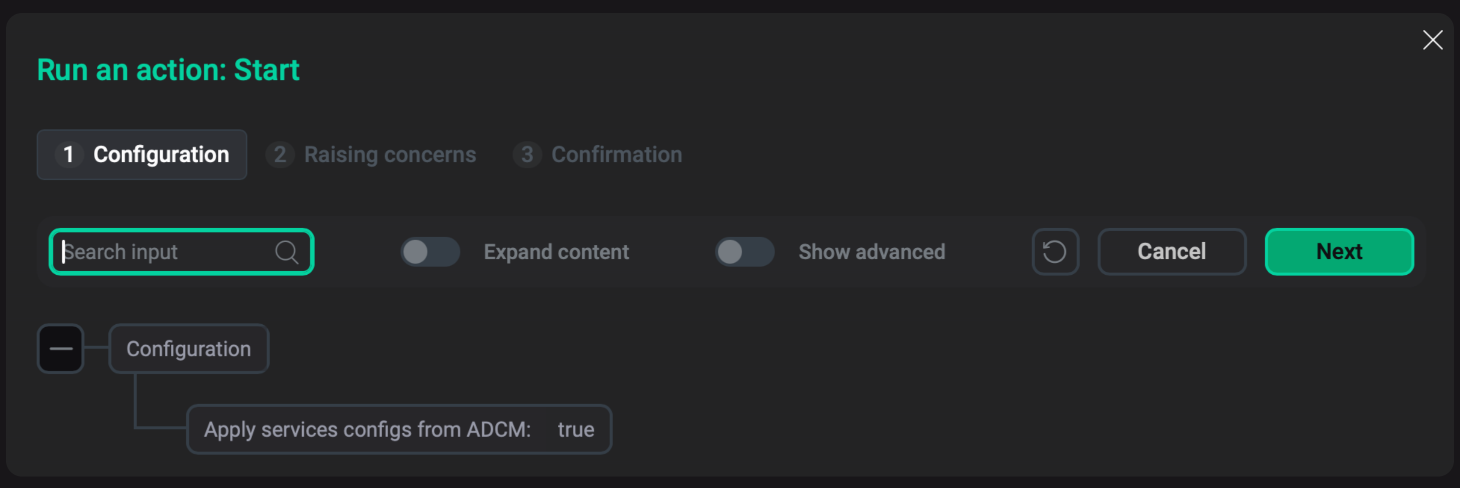


The Reconfigure & Restart window

Start

Starts the AD Eureka service.

In the window that opens, use the ***Apply services configs from ADCM*** flag to determine whether to apply the current service configuration (`true`) or just start the service (`false`).



The Start window

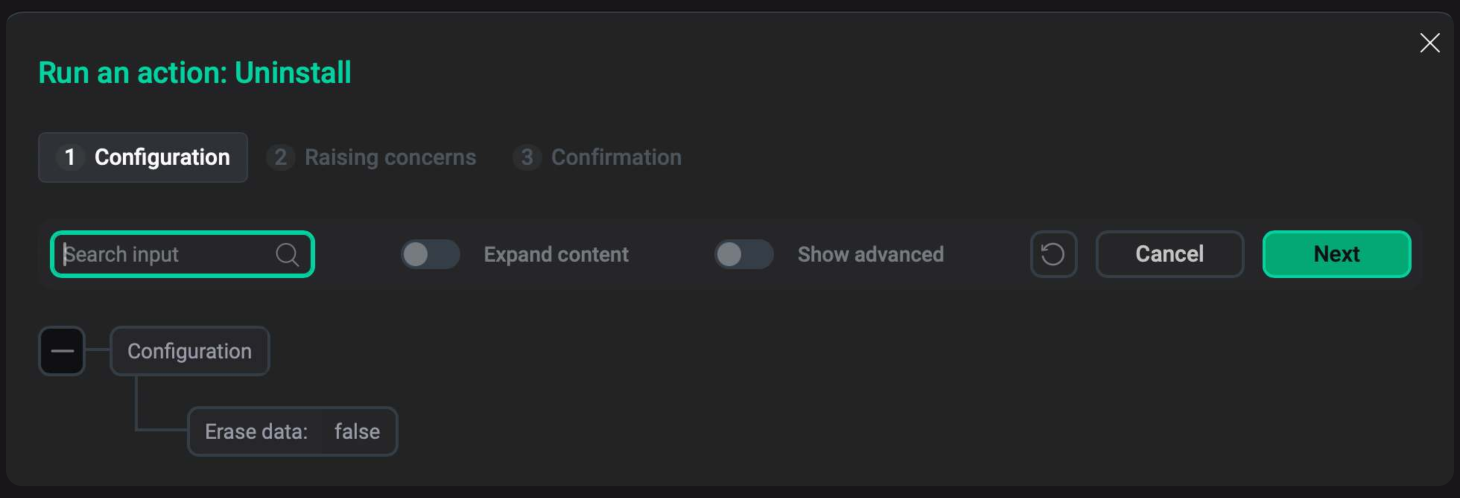
Stop

Stops the AD Eureka service.

Uninstall

Uninstalls the AD Eureka service.

In the window that opens, use the ***Erase data*** flag to select whether you need to delete files associated with a service.



The Uninstall window

ADB Control service actions

Anton Vasilev

Contents

To Table of Contents

This article describes the actions available for the ADB Control service in the ADCM web interface.

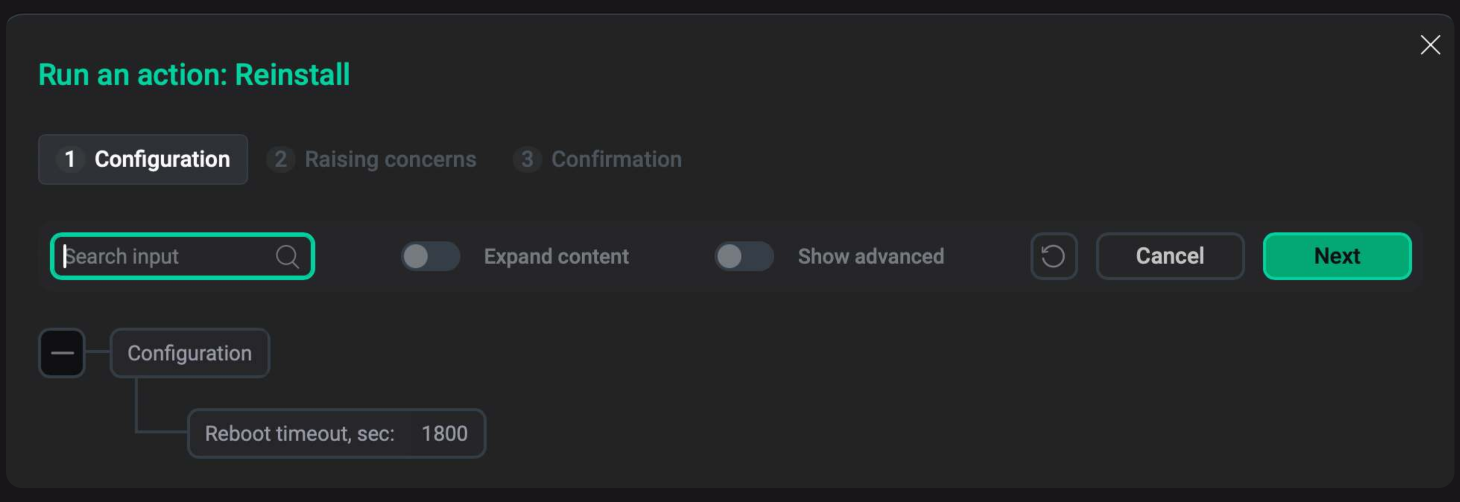
Install

Installs the ADB Control service. This action is available only if a service has been added but not yet installed.

Reinstall

Reinstalls the ADB Control service.

On the ***Configuration*** tab of the window that opens, in the ***Reboot timeout, sec*** field, enter the timeout to wait for hosts to restart (in seconds). The default value is `1800`. The value must be at least `600`.

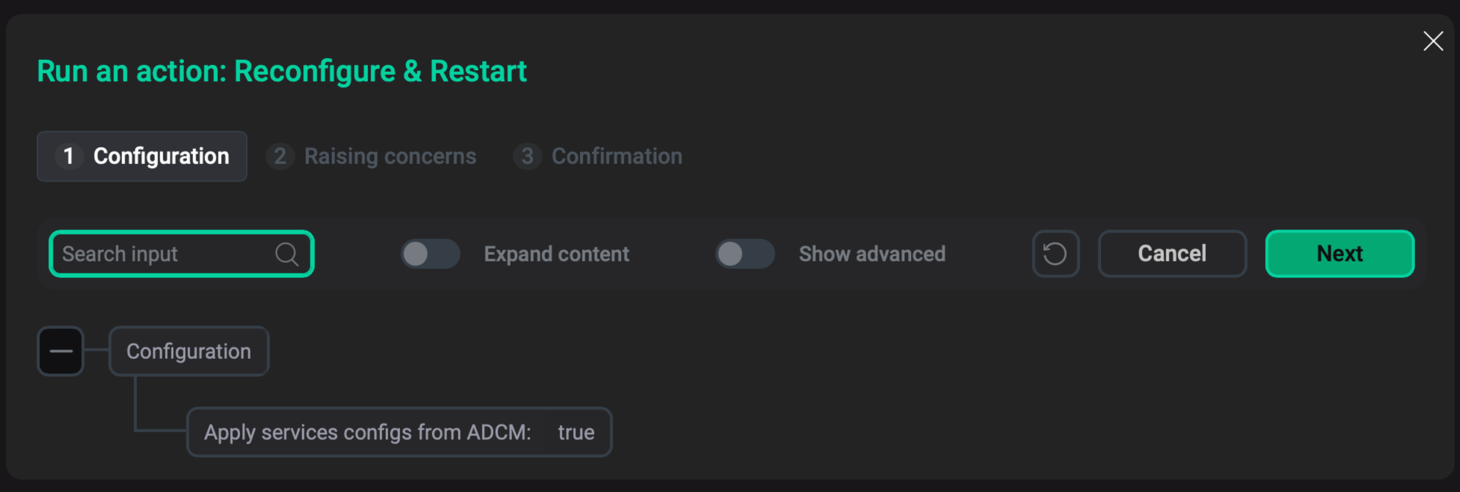


The Reinstall window

Reconfigure & Restart

Applies the current configuration to the service and restarts the service.

In the window that opens, use the ***Apply services configs from ADCM*** flag to determine whether to apply the current service configuration (`true`) or just restart the service (`false`).

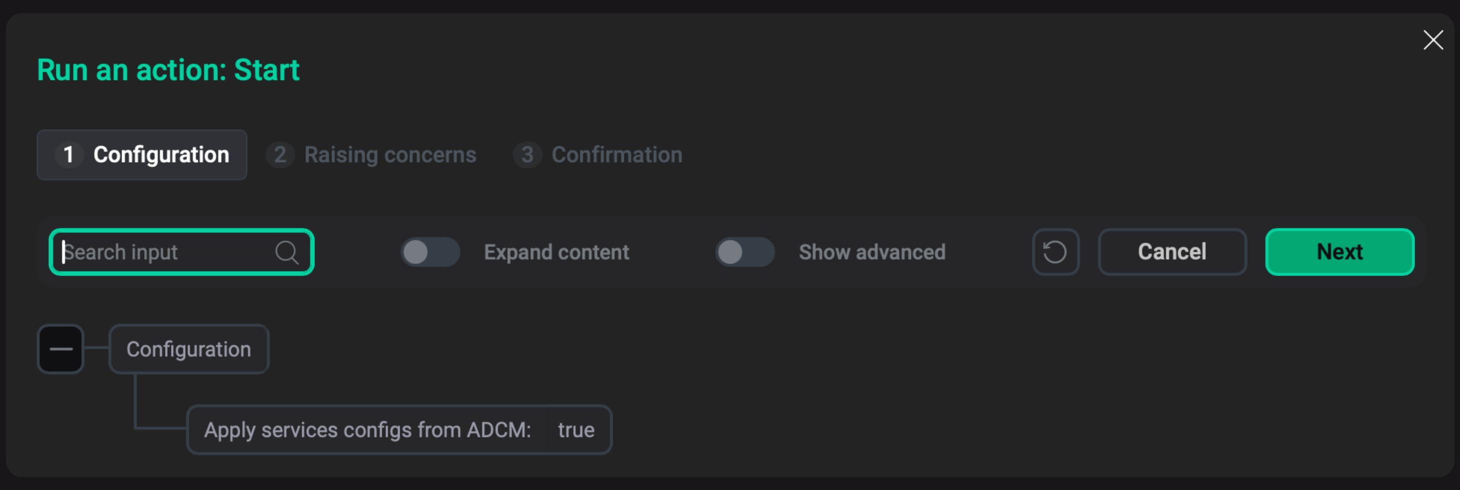


The Reconfigure & Restart window

Start

Starts the ADB Control service.

In the window that opens, use the ***Apply services configs from ADCM*** flag to determine whether to apply the current service configuration (`true`) or just start the service (`false`).



The Start window

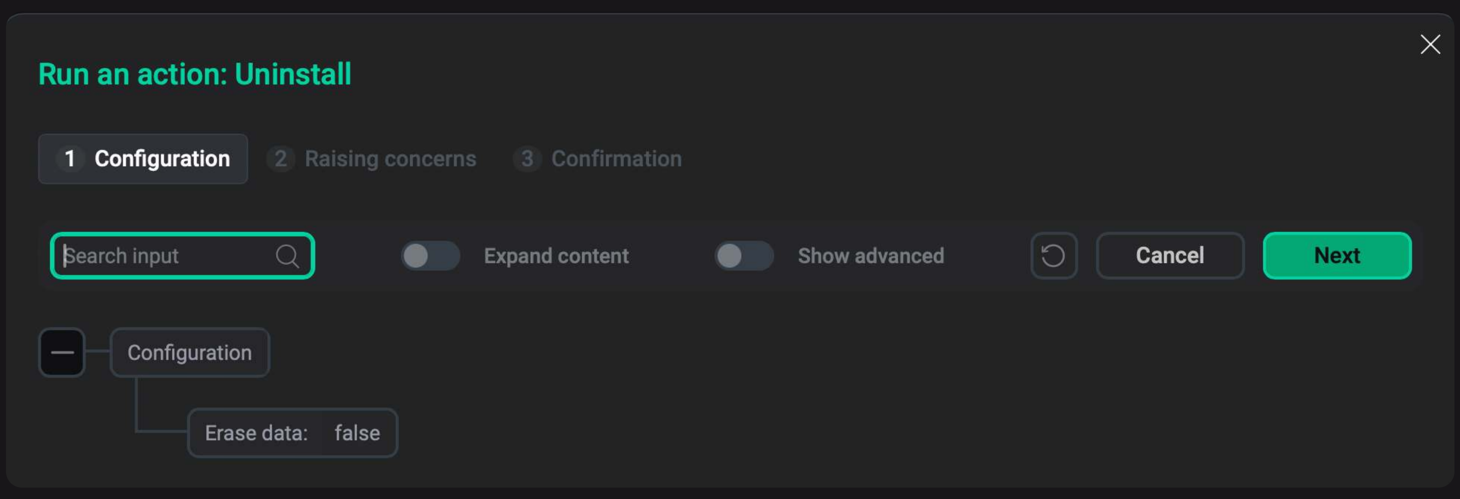
Stop

Stops the ADB Control service.

Uninstall

Uninstalls the ADB Control service.

In the window that opens, use the ***Erase data*** flag to select whether you need to delete files associated with a service.



The Uninstall window

NOTE

The ADBM service depends on ADB Control. If you want to uninstall ADB Control, you have to uninstall ADBM first.

ADBM service actions

Anton Vasilev

Contents

To Table of Contents

This article describes the actions available for the Arenadata DB Backup Manager (ADBM) service in the ADCM web interface.

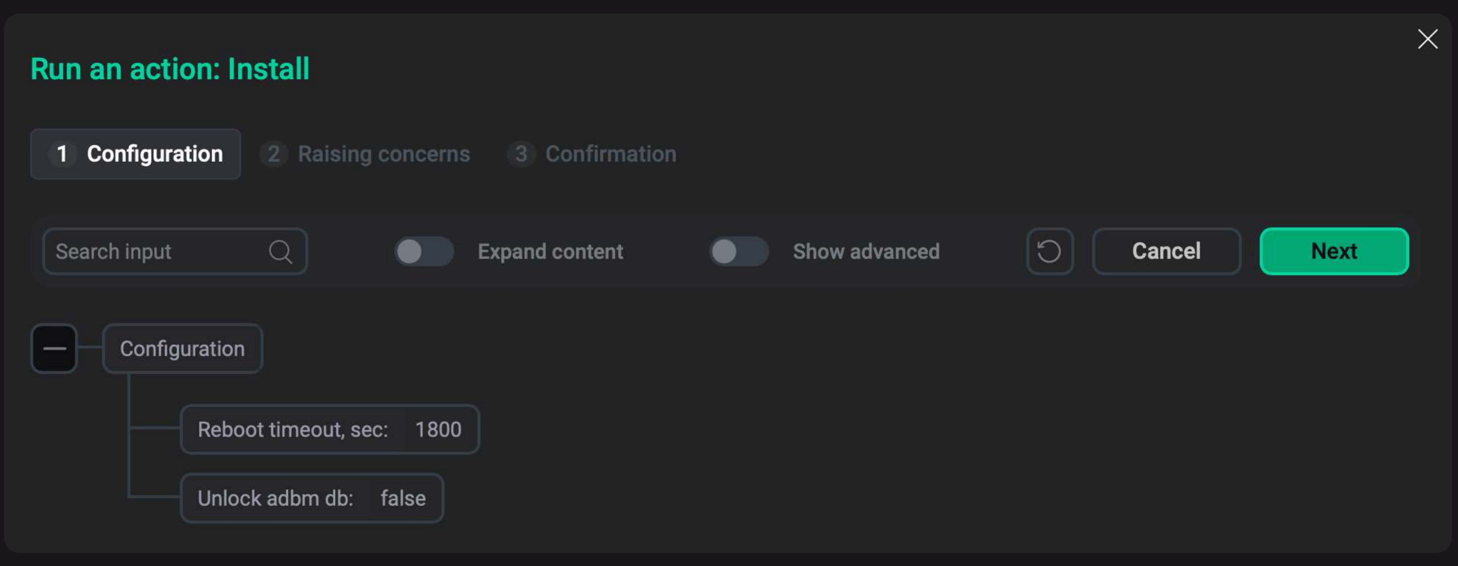
Install

Installs the ADBM service. This action is available only if a service has been added but not yet installed.

Reinstall

On the *Configuration* tab of the window that opens, configure:

- ***Reboot timeout, sec*** — enter the timeout to wait for hosts to restart (in seconds). The default value is `1800`. The value must be at least `600`.
- ***Unlock adbm db*** — use the flag to determine what to do with database locks that may occur if a migration failed. When the flag is set to `true`, ADBM releases the locks and makes the database available for migrations. The default value is `false`: preserve locks to avoid potentially conflicting migrations.

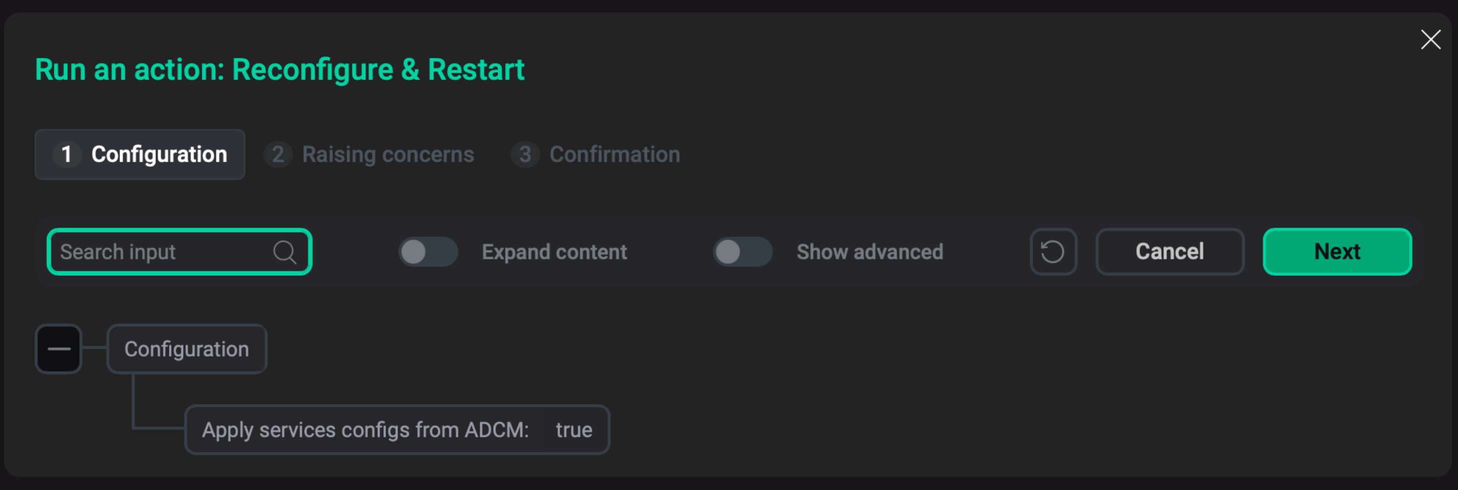


The Reinstall window

Reconfigure & Restart

Applies the current configuration to the service and restarts the service.

In the window that opens, use the ***Apply services configs from ADCM*** flag to determine whether to apply the current service configuration (`true`) or just restart the service (`false`).

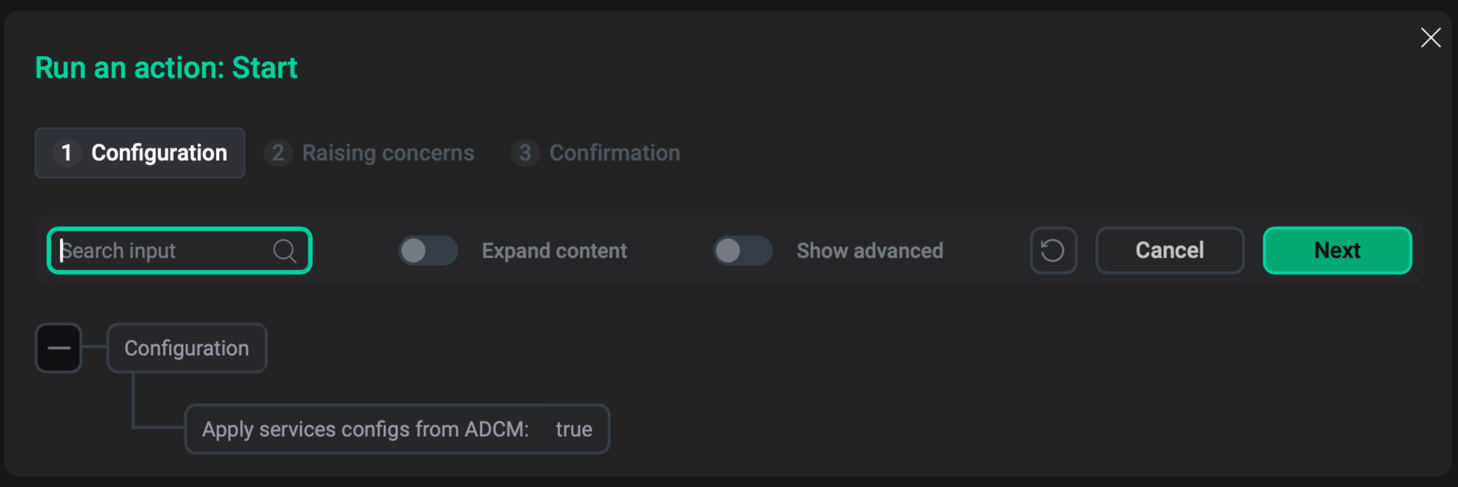


The Reconfigure & Restart window

Start

Starts the ADBM service.

In the window that opens, use the ***Apply services configs from ADCM*** flag to determine whether to apply the current service configuration (`true`) or just start the service (`false`).



The Start window

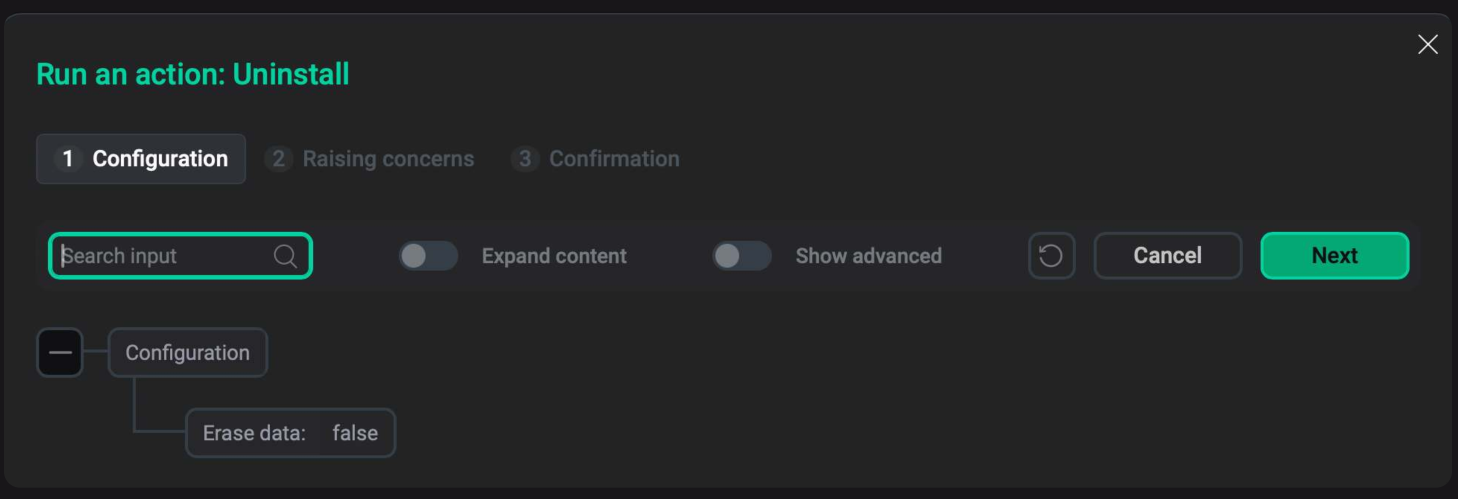
Stop

Stops the ADBM service.

Uninstall

Uninstalls the ADBM service.

In the window that opens, use the ***Erase data*** flag to select whether you need to delete files associated with a service.



The Uninstall window

Clickhouse service actions

Anton Vasilev

Contents

To Table of Contents

This article describes the actions available for the Clickhouse service in the ADCM web interface.

> IMPORTANT
>
> The full list of actions is available only when the service uses an internal ClickHouse database (ADQM). When an external database is used, the following actions are available:
>
> - Install
> - Uninstall
>
> Note that after changing the database type, you need to reinstall the Clickhouse service. Until that, the wrong actions are shown (for the previous database type).

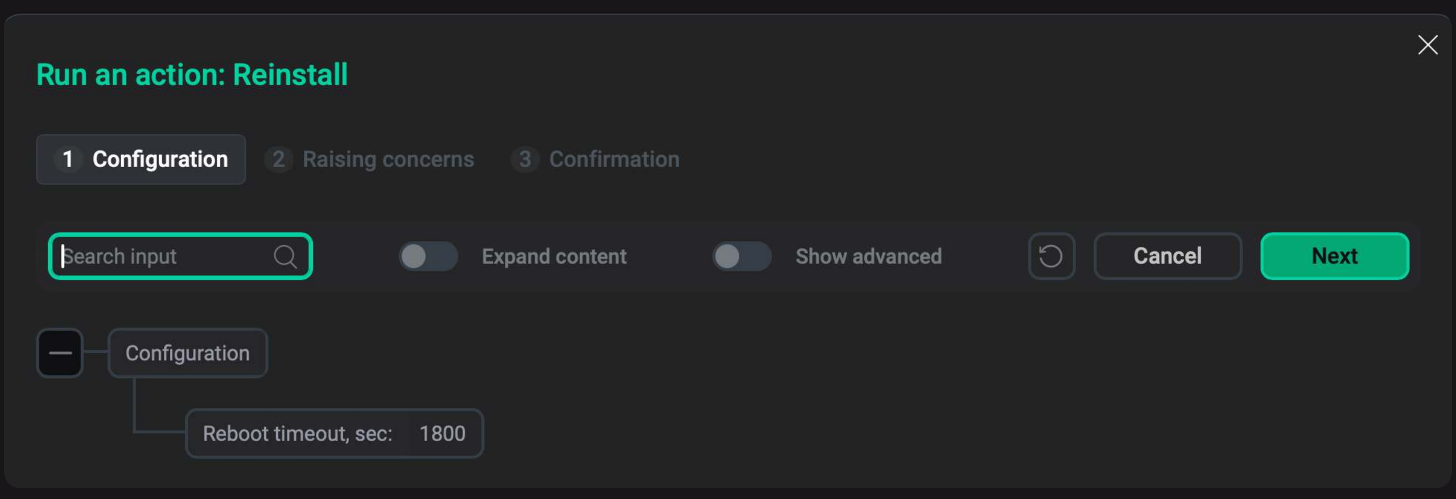
Install

Installs the Clickhouse service. This action is available only if a service has been added but not yet installed.

Reinstall

Reinstalls the Clickhouse service.

On the *Configuration* tab of the window that opens, in the ***Reboot timeout, sec*** field, enter the timeout to wait for hosts to restart (in seconds). The default value is `1800`. The value must be at least `600`.

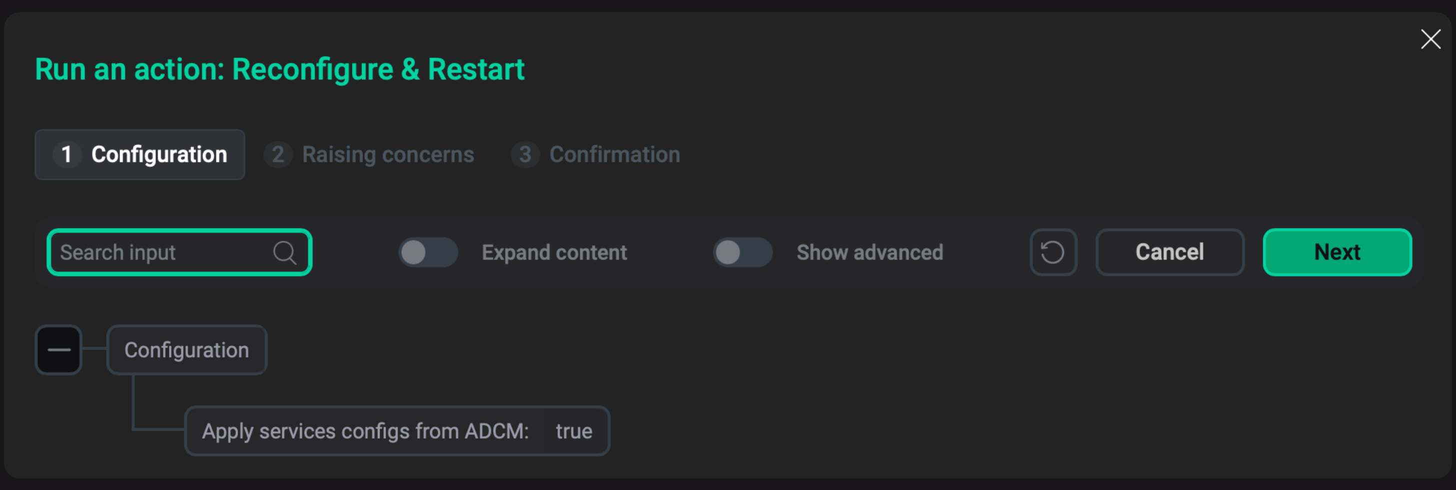


The Reinstall window

Reconfigure & Restart

Applies the current configuration to the service and restarts the service.

In the window that opens, use the ***Apply services configs from ADCM*** flag to determine whether to apply the current service configuration (`true`) or just restart the service (`false`).

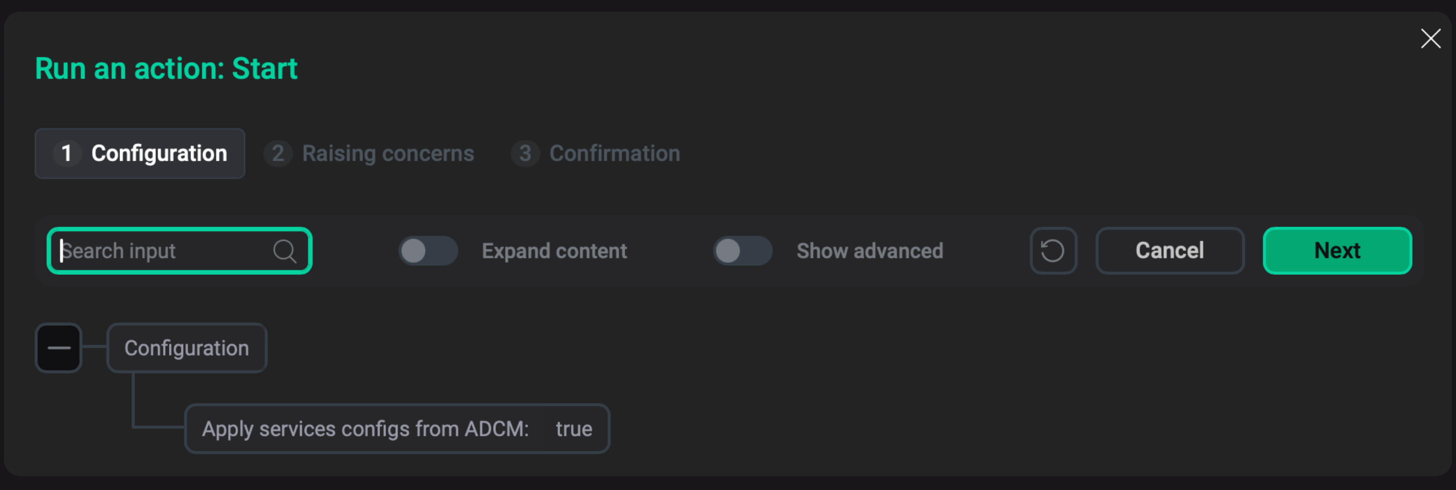


The Reconfigure & Restart window

Start

Starts the Clickhouse service.

In the window that opens, use the ***Apply services configs from ADCM*** flag to determine whether to apply the current service configuration (`true`) or just start the service (`false`).



The Start window

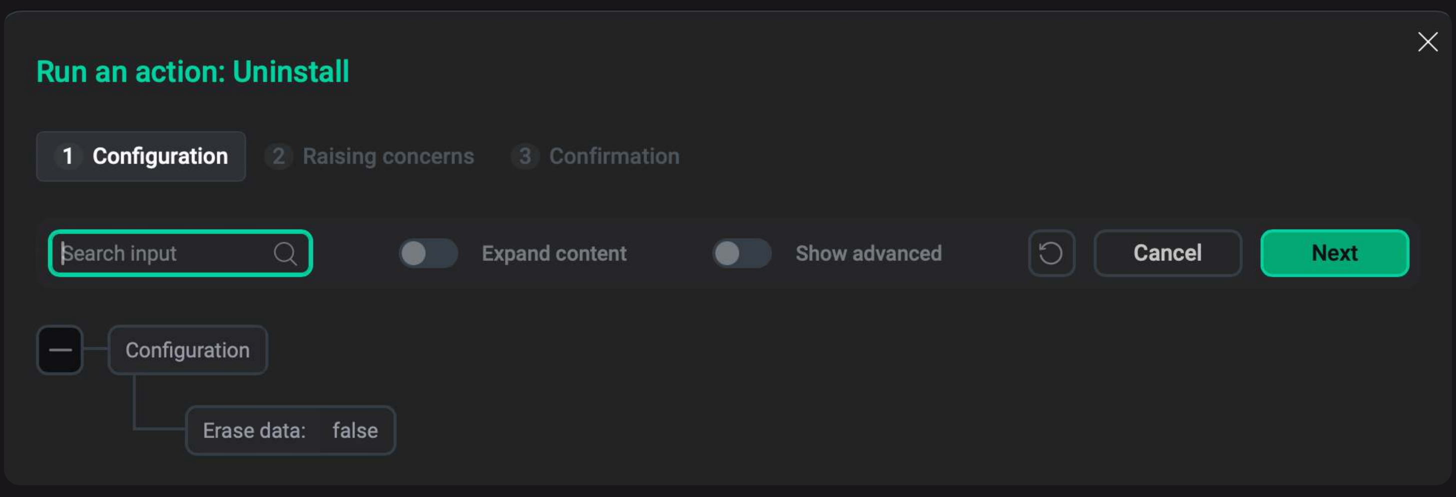
Stop

Stops the Clickhouse service.

Uninstall

Uninstalls the Clickhouse service.

In the window that opens, use the ***Erase data*** flag to select whether you need to delete files associated with a service.



The Uninstall window

Database service actions

Anton Vasilev

Contents

To Table of Contents

This article describes the actions available for the Database service in the ADCM web interface.

> IMPORTANT
>
> The full list of actions is available only when the service uses an internal PostgreSQL database (ADPG). When an external database is used, the following actions are available:
>
> - Install
> - Uninstall
>
> Note that after changing the database type, you need to reinstall the Database service. Until then, the wrong actions are shown (for the previous database type).

Install

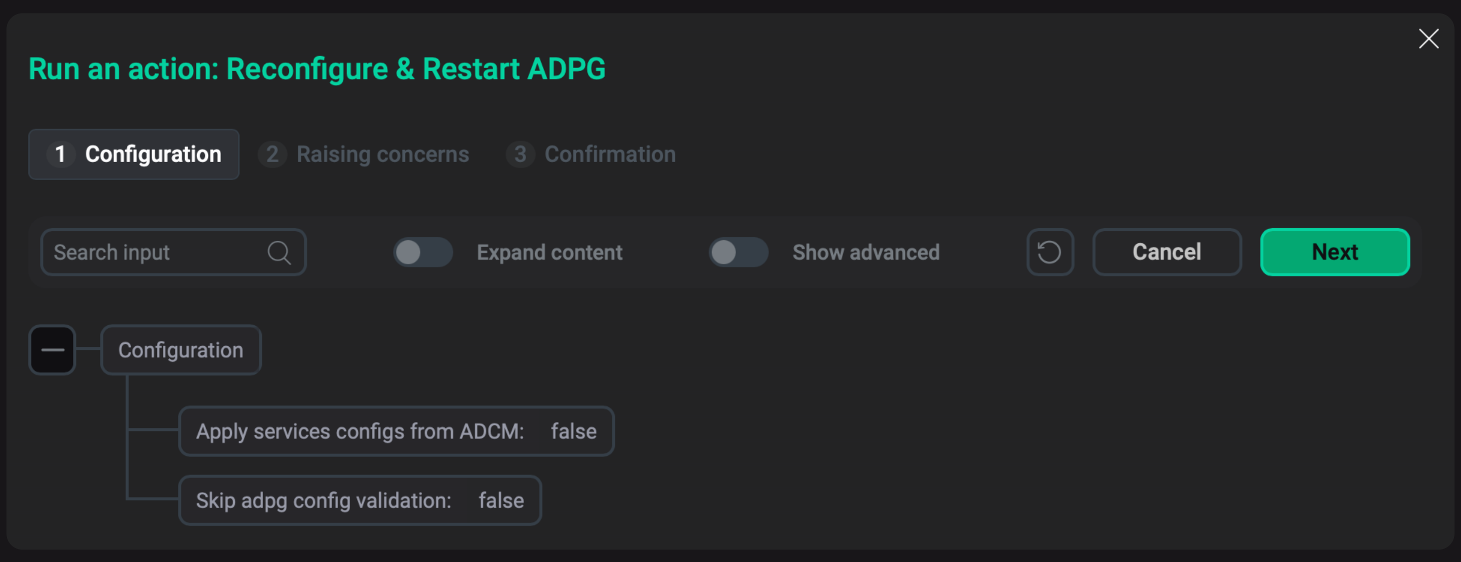
Installs the Database service. This action is available only if a service has been added but not yet installed.

Reconfigure & Restart ADPG

Applies the current configuration to the service and restarts the service.

In the window that opens, select additional options:

- ***Apply services configs from ADCM*** — use the flag to determine whether to apply the current service configuration (`true`) or just restart the service (`false`).
- ***Skip adpg config validation*** — use the flag to skip ADPG configuration validation before applying it. The default value is `false` meaning the validation is performed. If you use an external PostgreSQL instance, the validation is not performed.



The Reconfigure & Restart ADPG window

Reinstall

Reinstalls the Database service.

Reload

Rereads the ADPG configuration files without restarting the service. The action is available only for the internal type of Database. The type is selected on the Database configuration page.

Start

Starts the Database service.

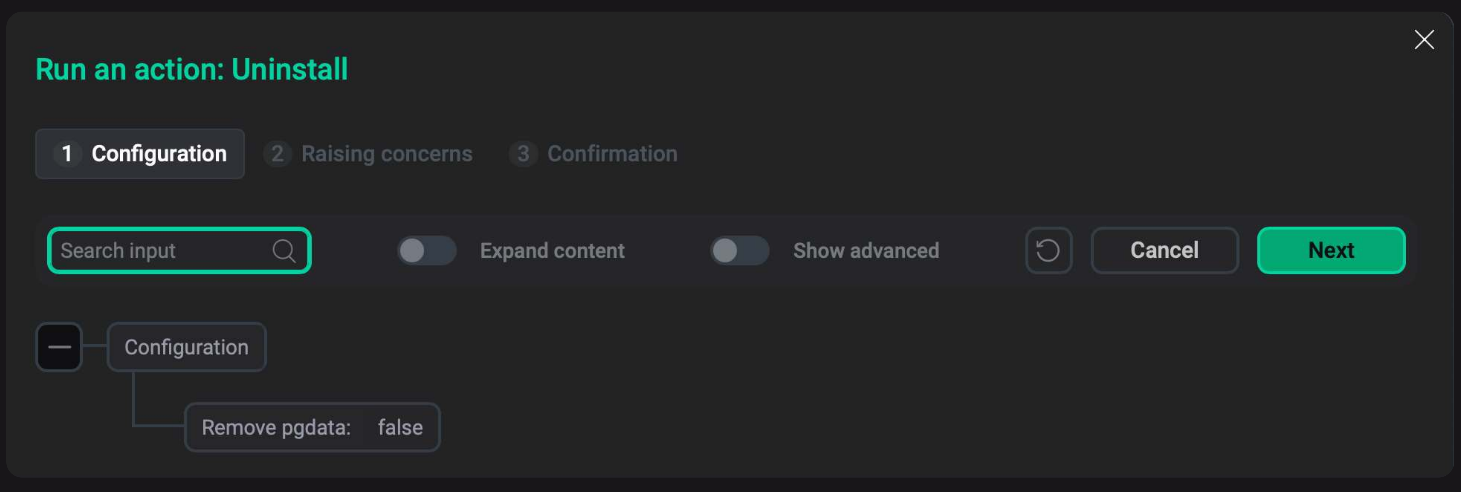
Stop

Stops the Database service.

Uninstall

Uninstalls the Database service.

In the window that opens, use the ***Remove pgdata*** flag to select whether you need to delete the PostgreSQL data catalog.



The Uninstall window

Monitoring service actions

Anton Vasilev

Contents

To Table of Contents

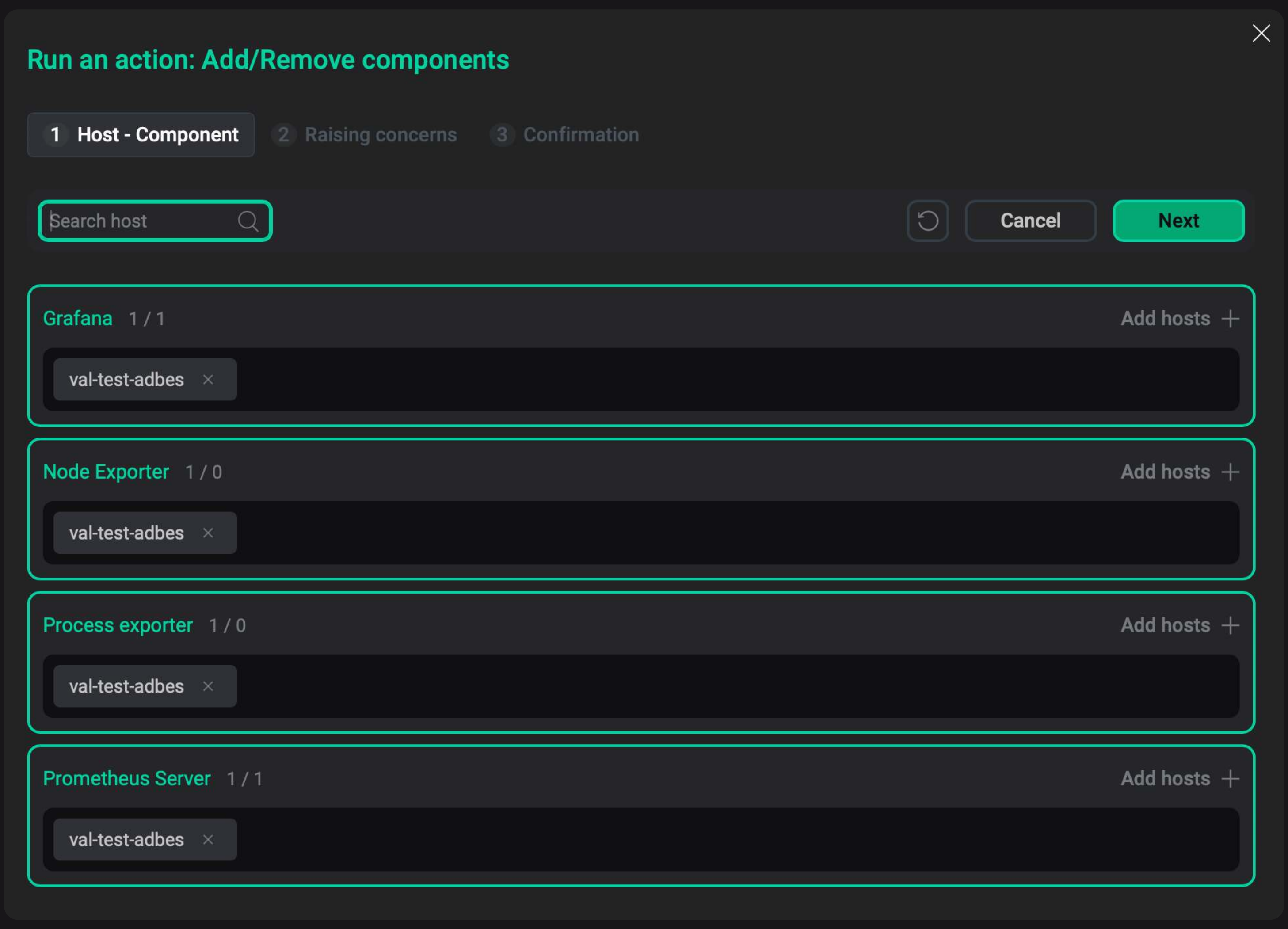
This article describes the actions available for the Monitoring service in the ADCM web interface.

Install

Installs the Monitoring service. This action is available only if a service has been added but not yet installed.

Add/Remove components

Opens the component-host mapping interface where you can add, remove, or redistribute service components.



The Add/Remove components window

Check

Checks the status of the Monitoring components: Prometheus, Grafana, and configured exporters.

Reconfigure and restart

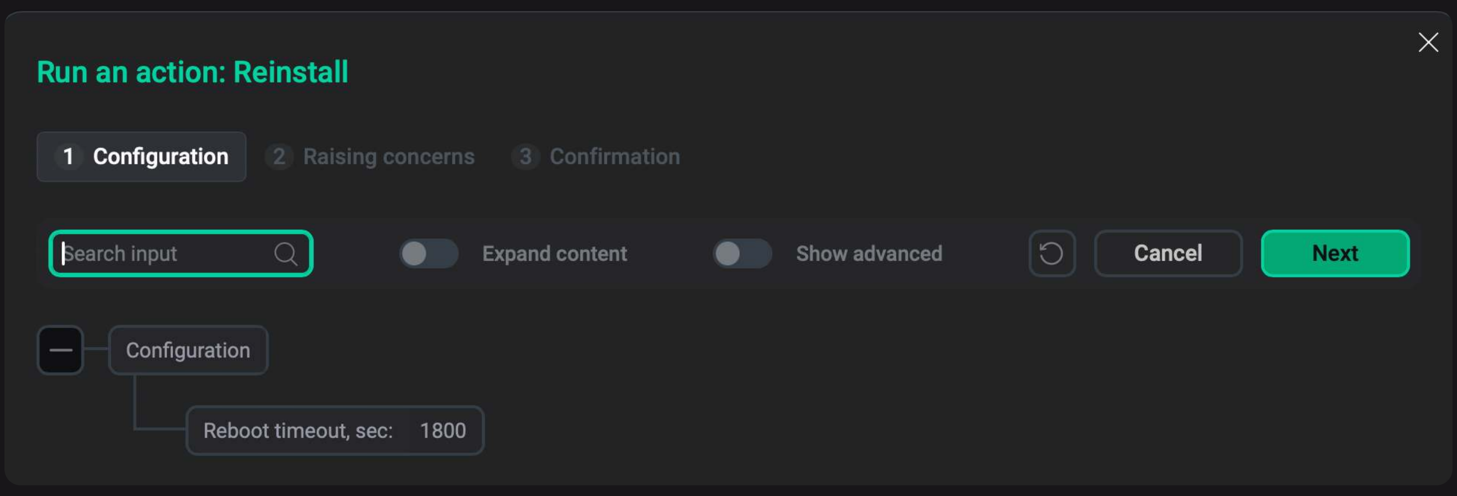
Applies the current configuration to the service and restarts the service.

On the ***Configuration*** tab of the window that opens, use the ***Grafana: apply password*** field to select whether the Grafana administrator password should be updated by the one from the current configuration.

Reinstall

Reinstalls the Monitoring service.

On the ***Configuration*** tab of the window that opens, in the ***Reboot timeout, sec*** field, enter the timeout to wait for hosts to restart (in seconds). The default value is `1800`. The value must be at least `600`.



The Reinstall window

Start

Starts the Monitoring service.

Stop

Stops the Monitoring service.

Configure external databases for ADB Control and ADBM

Daria Barysheva

Contents

To Table of Contents

For production environments, it is recommended to configure external PostgreSQL and ClickHouse databases to store data of ADB Control (see Query DB and Metrics DB in Arenadata DB Control overview) and ADBM (see PostgreSQL in Arenadata DB Backup Manager overview) for a number of reasons:

- You will be able to set up external storages on hosts of required size for desired performance and configure high availability for them.
- The ADB Control recovery process will be easier and consist mainly of the service configuration steps.

To configure external databases in ADB ES (starting with ADB Control 4.16.0), you need to follow the steps that are described below. For information on how to set up external databases for the previous ADB Control and ADBM versions (delivered as ADB services), refer to the corresponding versions of the ADB documentation:

- ADB Control
- ADBM

- For information on the initial ADB ES installation (including configuring external databases), refer to Migration of ADB Control and ADBM metadata during ADB ES installation.
- Note that actions for ADBM are needed only if you use that service in ADB ES.
- Since ADB ES provides one common service for storing PostgreSQL data in ADB Control and ADBM (namely Database), you should use one external PostgreSQL server for ADB Control and ADBM and specify its settings in the Database service (see ***External postgres parameters*** below). The owner of both databases should be the same.
- Names of the external PostgreSQL databases for ADB Control and ADBM services in the ADB ES cluster should be generated according to the following pattern:
  - ADB Control — `<prefix>_adcc`
  - ADBM — `<prefix>_adbm`

  The `<prefix>` value should be equal for both services. Subsequently, that value should be specified in the Database service parameter ***External postgres parameters → Database name prefix***. Examples of valid database names: `external_adcc` and `external_adbm` (where `external` is a prefix).
- If you are switching to external databases after the initial installation of the Database and/or Clickhouse services in ADB ES, it is important to apply the service action ***Reinstall*** to the Database and/or Clickhouse service after filling and saving the configuration settings described below.

PostgreSQL

Step 1. Run SQL queries for the ADB Control database

1. Perform the following queries in PostgreSQL. The queries should be run under a superuser:

```
CREATE ROLE <user_name> LOGIN PASSWORD '<user_password>';
ALTER ROLE <user_name> SET search_path TO public, adcc;
CREATE DATABASE <prefix>_adcc WITH owner <user_name>;
```

2. Connect to the created database under a superuser and run:

```
CREATE SCHEMA adcc;
ALTER SCHEMA adcc OWNER TO <user_name>;

CREATE EXTENSION IF NOT EXISTS pgcrypto WITH schema public;
CREATE EXTENSION IF NOT EXISTS btree_gin WITH schema public;
CREATE EXTENSION IF NOT EXISTS pg_trgm WITH schema public;

GRANT SELECT ON ALL TABLES IN SCHEMA pg_catalog TO <user_name>;

CREATE OPERATOR adcc.^| (PROCEDURE = jsonb_exists_any, LEFTARG = jsonb, RIGHTARG = _text, RESTRICT = contsel, JOIN = contjoinsel);
ALTER OPERATOR adcc.^|(jsonb, text[]) OWNER TO <user_name>;
CREATE OPERATOR CLASS adcc.jsonb_ops_custom
FOR TYPE jsonb USING gin AS
OPERATOR 7 pg_catalog.@>(jsonb, jsonb),
OPERATOR 10 adcc.^| (jsonb, _text),
FUNCTION 1  gin_compare_jsonb(text, text),
FUNCTION 2  gin_extract_jsonb(jsonb, internal, internal),
FUNCTION 3  gin_extract_jsonb_query(jsonb, internal, smallint, internal, internal, internal, internal),
FUNCTION 4  gin_consistent_jsonb(internal, smallint, jsonb, integer, internal, internal, internal, internal),
FUNCTION 6  gin_triconsistent_jsonb(internal, smallint, jsonb, integer, internal, internal, internal);
ALTER OPERATOR CLASS adcc.jsonb_ops_custom USING GIN OWNER TO <user_name>;
```

where `<user_name>` is a name of the previously created user.

If you can assign superuser rights to a PostgreSQL user (which will be used for connecting to an external database from ADB Control), it is enough to create a role and a database in PostgreSQL before installation or reconfiguration of ADB Control:

```
CREATE ROLE <user_name> LOGIN PASSWORD '<user_password>' SUPERUSER;
ALTER ROLE <user_name> SET search_path TO public, adcc;
CREATE DATABASE <prefix>_adcc WITH owner <user_name>;
```

3. Optionally, you can run the additional query to increase performance by disabling synchronous writes to WAL files:

```
ALTER DATABASE <prefix>_adcc SET synchronous_commit = OFF;
```

Step 2. Run SQL queries for the ADBM database

1. Perform the following queries in PostgreSQL. The queries should be run under the default user `postgres` with the `superuser` rights:

```
ALTER ROLE <user_name> SET search_path TO public, adcc, adbm; 1
CREATE DATABASE <prefix>_adbm WITH owner <user_name>; 2
```

1 Grant the `<user_name>` user (previously created for ADB Control) appropriate rights for connecting to the external ADBM database.

2 Create the database for ADBM data migration.

2. Connect to the created database under a superuser and run:

```
CREATE SCHEMA adbm; 1
GRANT USAGE, CREATE ON SCHEMA adbm TO <user_name>; 2
ALTER DEFAULT PRIVILEGES IN SCHEMA adbm GRANT UPDATE, INSERT, SELECT, DELETE ON TABLES TO <user_name>;
CREATE EXTENSION btree_gin WITH schema adbm;
CREATE EXTENSION pg_trgm WITH schema adbm;
GRANT SELECT ON pg_extension TO <user_name>;
```

1 Create the `adbm` schema in the database for ADBM data migration.

2 Create extensions in the `adbm` schema and grant their usage to the `<user_name>` user.

If you can assign superuser rights to a PostgreSQL user (which will be used for connecting to an external database from ADBM), it is enough to create a role and a database in PostgreSQL before installation or reconfiguration of ADBM:

```
ALTER ROLE <user_name> SET search_path TO public, adcc, adbm;
CREATE DATABASE <prefix>_adbm WITH owner <user_name>;
```

Step 3. Configure pg_hba.conf on the PostgreSQL server

Provide the ADB Control access to the external PostgreSQL databases by adding the following records to the *pg_hba.conf* file on the PostgreSQL server.

Without SSL

```
host    <prefix>_adcc  <user_name>       <adbc_address>      trust
host    <prefix>_adbm  <user_name>       <adbm_address>      trust
```

where:

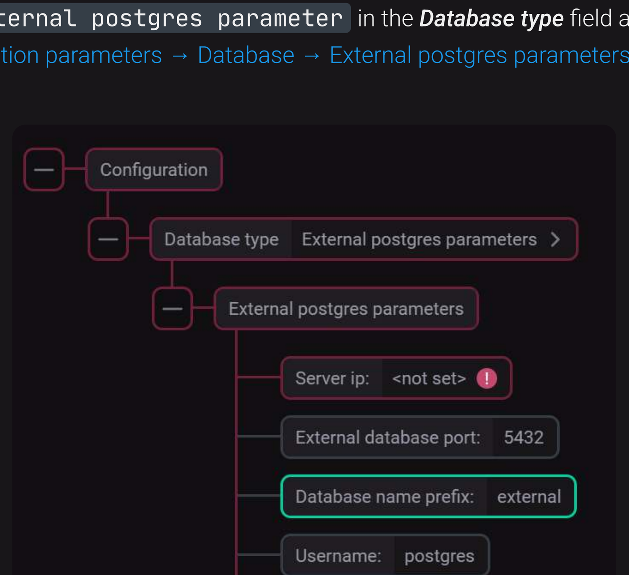
- `<prefix>_adcc`, `<prefix>_adbm` — names of the created databases.
- `<user_name>` — a name of the created user.
- `<adbc_address>` — an IP address of ADB Control with a subnet number.
- `<adbm_address>` — an IP address of ADBM with a subnet number.

With SSL

```
hostssl    all  all       0.0.0.0/0      md5
```

Step 4. Configure the Database service in ADB ES

1. On the Database service configuration page, select `External postgres parameter` in the ***Database type*** field and fill in all parameters in the list that opens. The assignment of parameters is described in the Configuration parameters → Database → External postgres parameters section.



Parameters for connecting to the external PostgreSQL DB

2. Click ***Save*** to save the Database configuration.
3. Install the Database service using the Install service action or apply the ***Reinstall*** action to the Database service if it is already installed.

ClickHouse

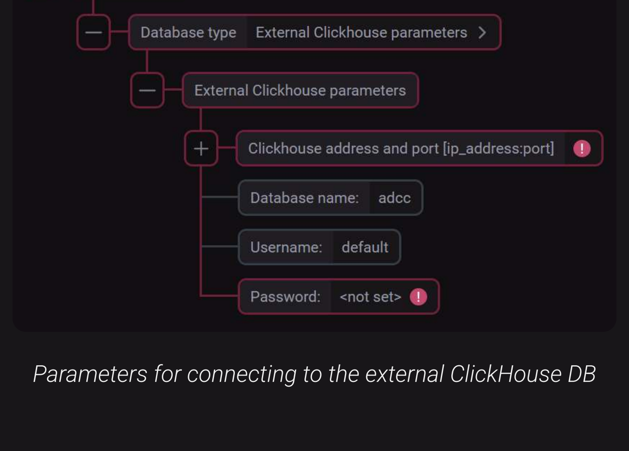
IMPORTANT

- To use clustered external ClickHouse as an external database (instead of a standalone node), you should first create a logical cluster called `adcc_cluster` in ClickHouse.
- If you use SSL encryption in ADB Control, you need to configure SSL on the ClickHouse side and define the `8443` port in the ClickHouse connection parameters (see ***External Clickhouse parameters*** below).

1. Run the following queries to create a user with appropriate permissions in ClickHouse:

```
CREATE USER <user_name> NOT IDENTIFIED;
GRANT SELECT, INSERT, ALTER, CREATE, DROP, TRUNCATE ON *.* TO <user_name>
```

2. On the Clickhouse service configuration page, select `External Clickhouse parameters` in the ***Database type*** field and fill in all parameters in the list that opens. The assignment of parameters is described in the Configuration parameters → Clickhouse → External Clickhouse parameters section.



Parameters for connecting to the external ClickHouse DB

3. Click ***Save*** to save the Clickhouse configuration.
4. Install the Clickhouse service using the Install service action or apply the ***Reinstall*** action to the Clickhouse service if it is already installed.

ADB ES monitoring metrics

Anton Vasilev

Contents

To Table of Contents

This article describes metrics for monitoring an ADB ES cluster. For information on how to install monitoring, refer to Install monitoring.

Overview

The Monitoring service consists of the following components:

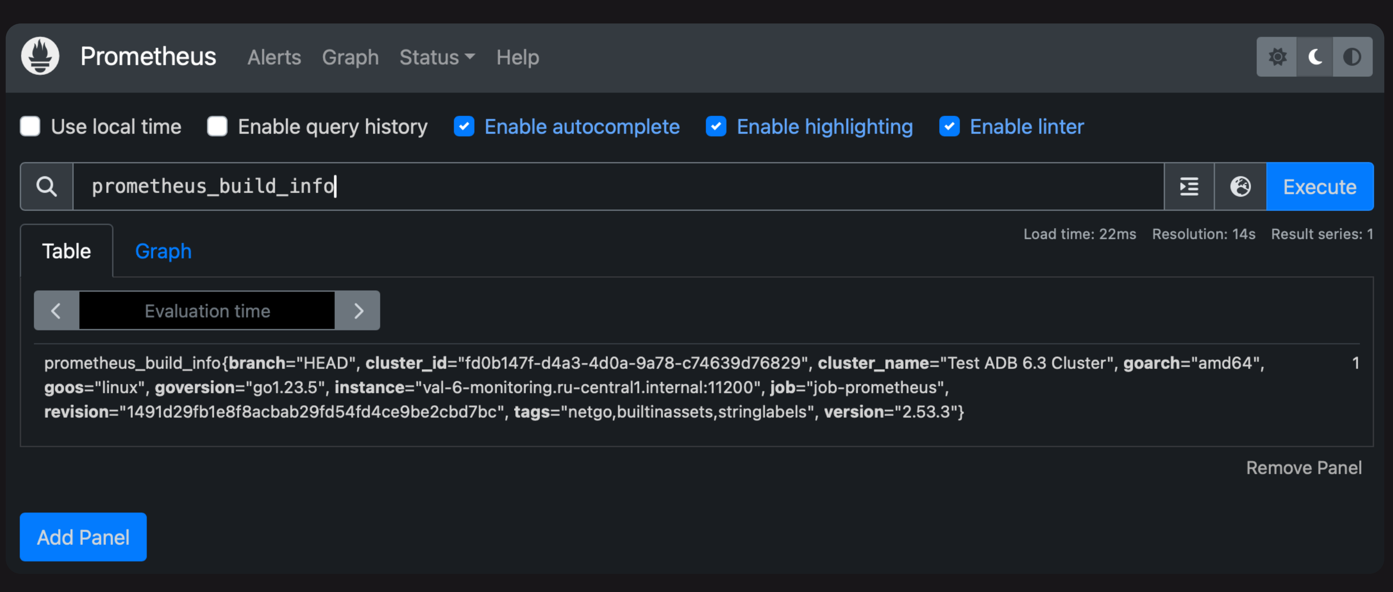
- Node Exporter — exposes hardware- and OS-related metrics, such as memory and CPU usage or filesystem space. These metrics are available on the port and endpoint specified in the ***Node Exporter settings*** section of the Monitoring service configuration (`11203/metrics` by default).
- Process Exporter — collects metrics for the specified processes. In ADB ES, these are processes related to ADBM and ADB Control. Note that processes related to ADBM and ADB Control agents are monitored by the Monitoring service deployed in ADB. The Process Exporter metrics are available on the port specified in the ***Process exporter settings*** section of the Monitoring service configuration (`9256` by default).
- Prometheus — scrapes and stores metrics from the configured data sources: Node Exporter and Process Exporter. These metrics are available in the Prometheus web interface on the port specified in the ***Prometheus settings*** section of the Monitoring service configuration (`11200` by default).
- Grafana — uses Prometheus as a data source and visualizes its metrics as graphs and charts organized into dashboards. These dashboards are available in the Grafana web interface on the port specified in the ***Grafana settings*** section of the Monitoring service configuration (`11210` by default).

View metrics in Prometheus

Prometheus is a monitoring and alerting toolkit. Prometheus collects metrics from exporters, and Grafana sends requests to Prometheus to collect data for its dashboards. If Grafana dashboards show empty panels or unexpected values, checking Prometheus helps determine whether the issue is related to metric collection or dashboard configuration:

1. In your browser, enter `<IP address of monitoring server>:<port>`. The default port is `11200`, and it can be changed in the ***Prometheus settings*** section in the Monitoring service configuration.
   The IP address, port, and hostname of Prometheus are also available on the ***Info*** tab of the Monitoring service.
2. In the window that opens, enter the user name and the password that you have configured in the ***Prometheus users to login/logout to Prometheus*** field in the Monitoring service configuration.

In the Prometheus web interface, you can check its configuration and the state of the exporters (on the ***Targets*** page). You can also use the Prometheus Query Language (PromQL) to check specific metrics.



Using Prometheus Query Language

Grafana dashboards

Grafana allows you to visualize metrics stored in Prometheus, create your own dashboards, or modify existing ones.

Open Grafana

1. In your browser, enter `<IP address of monitoring server>:<port>`. The default port is `11210`, and it can be changed in the ***Grafana settings*** section in the Monitoring service configuration.
   The IP address, port, and hostname of Grafana are also available on the ***Info*** tab of the Monitoring service.
2. In the window that opens, in the ***Email or username*** field, enter `admin` and in the ***Password*** field enter the password that you have configured in the ***Grafana administrator's password*** field in the Monitoring service configuration.

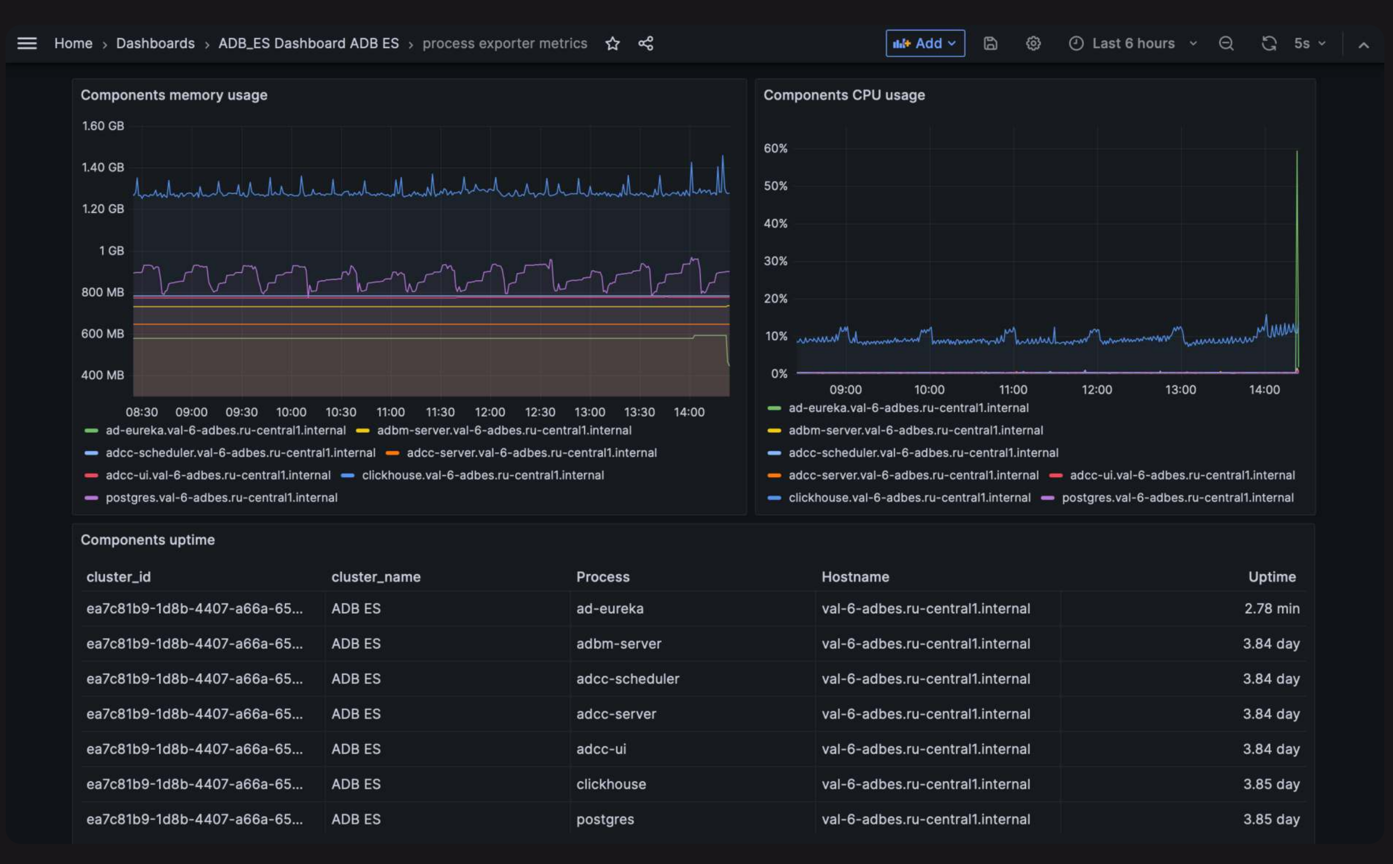
By default, the following dashboards are available in Grafana:

- Process exporter metrics
- System metrics

Process exporter metrics

This dashboard provides monitoring for essential ADBM and ADB Control components and database processes using metric provided by Process Exporter. The tracked processes are:

- `adbm-server` — ADBM Backend Server;
- `adcc-server` — ADB Control Backend Server;
- `adcc-scheduler` — ADB Control Scheduler;
- `adcc-ui` — ADB Control UI Server;
- `clickhouse` — ClickHouse DBMS used in ADB Control (tracked only if you use the internal Arenadata QuickMarts (ADQM) instance provided by the ADB ES bundle);
- `postgres` — PostgreSQL used by ADB Control and ADBM (tracked only if you use the internal Arenadata Postgres (ADPG) instance provided by the ADB ES bundle);
- `ad-eureka` — Arenadata Service Registry.



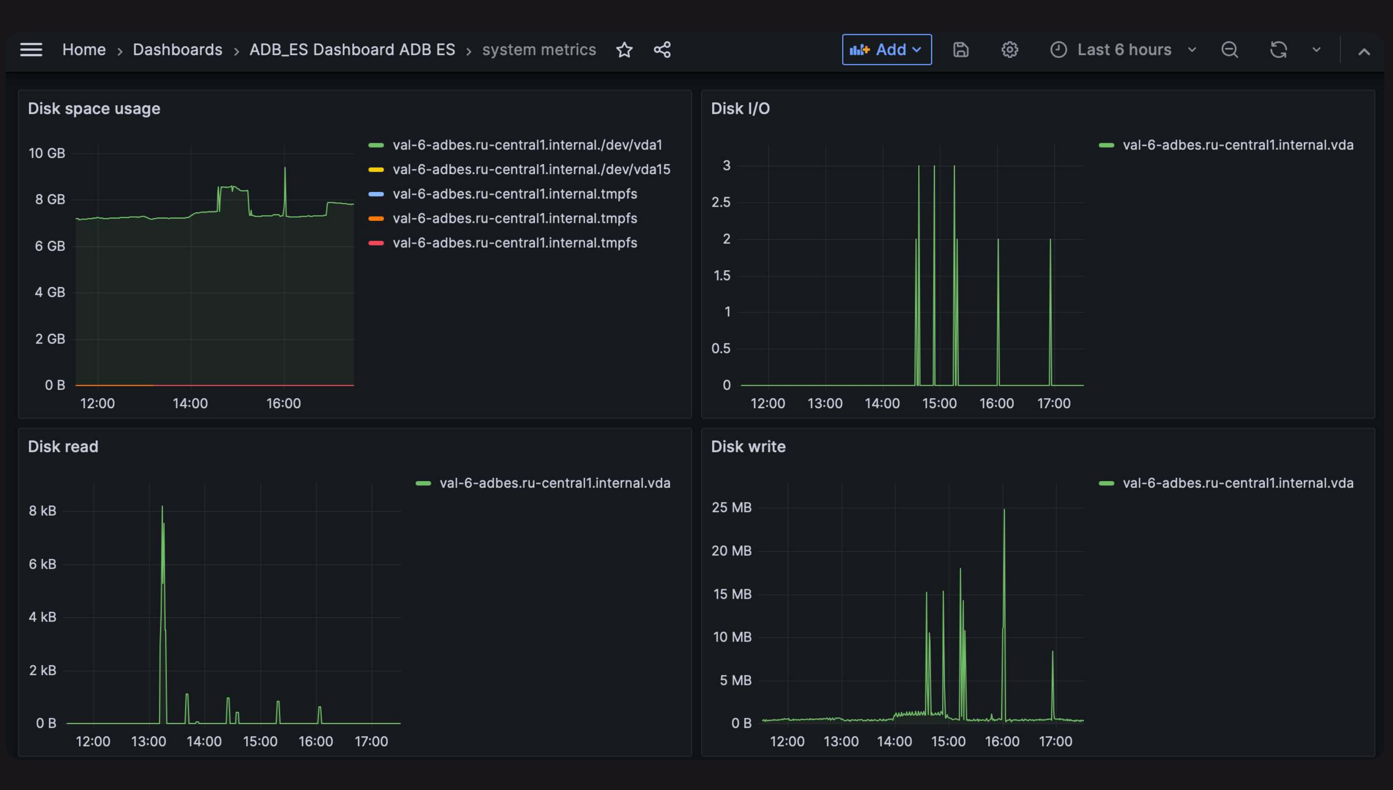
Process exporter metrics dashboard in Grafana

Process exporter metrics

Panel name	Description
Components uptime	Current uptime for each tracked process
Components memory usage	Memory usage of each process over time
Components CPU usage	CPU usage of each process over time

System metrics

This dashboard provides monitoring for host-level system metrics provided by Node Exporter.



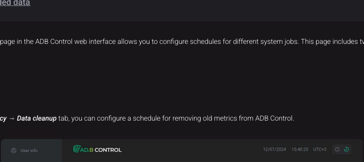
System metrics dashboard in Grafana

System metrics

Panel name	Description
Disk space usage	Used disk space on each mounted filesystem
Disk I/O	Number of disk I/O operations
Disk read	Rate of data read from disk
Disk write	Rate of data written to disk
Host CPU usage	CPU utilization percentage across all cores
Host memory usage	Memory used by processes on hosts
Network receive bytes	Inbound network traffic rate by interface and host
Network transmit bytes	Outbound network traffic rate by interface and host

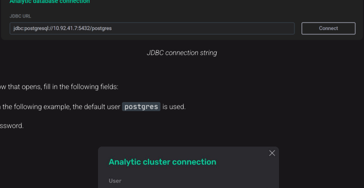
Configure job policies

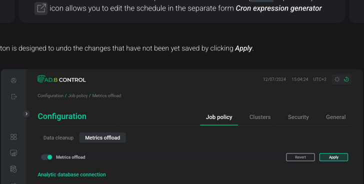
Data cleanup



Metrics offload

Configuration





Column	Description
object_type	A type of the object on which the operation was performed. Possible values: • `User` — user; • `User forbidden DB` — assigning advanced permissions to a specific database (actual for users with the `Advanced User` role); • `LDAP mapping` — role mapping with LDAP group; • `Transaction` — transaction; • `Resource group` — resource group; • `Cluster` — cluster; • `Cluster export databases` — databases, whose metrics will be offloaded; • `Job` — ADB Control job; • `System settings` — system configuration parameters; • `System user` — system user; • `Session` — ADB session; • `User session` — ADB Control user session; • `Cluster guc` — cluster GUC; • `Cluster databases` — databases that should be monitored in ADB Control.
operation_type	A type of the operation that was applied to the `object_name` object. Possible values: • `Cancel` • `Create` • `Delete` • `Resource group change` • `Terminate` • `Update` • `Archive` • `Activate`
operation_result	An operation result. Possible values: • `Success` • `Failed` • `Denied`
operation_time	An operation end timestamp
username	A name of the user who performed the operation
session	An identifier of the ADB Control user session
hostname	IP address of the host from which the operation was initiated
diffs	Information about changed attributes of the `object_name` object in the JSON format. For each changed attribute, the following data is stored: • `name` — an attribute name; • `oldValue` — a previous value; • `newValue` — a new value. The example: `[ { "name":"CLUSTER_ARCHIVED", "oldValue":true, "newValue":false } ]`

cluster

This table contains information on ADB clusters selected in ADB Control for export of metrics and events to the external database.

Column	Description
id	A cluster unique identifier
name	A cluster name

cluster_query_metric

This table stores statistics of system resource consumption by commands at the cluster level. For more information, see the ***System metrics collected for commands*** section of the Monitoring of commands article.

Column	Description
query_id	A command identifier. Refers to `query.id` (see query)
cpu_usage_total	The total CPU consumption during the command execution (in seconds)
cpu_usage_percent_sum	The CPU amount consumed at the end of the command execution (percentage)
cpu_usage_percent_avg	The average amount of CPU consumed across cluster segments at the end of the command execution (percentage)
cpu_usage_percent_skew	The indicator that displays the CPU consumption skew across cluster segments at the end of the command execution (percentage)
mem_sum	The RAM amount consumed at the end of the command execution (in bytes)
mem_max	The maximum amount of RAM consumed at the end of the command execution (in bytes)
mem_avg	The average amount of RAM consumed across cluster segments at the end of the command execution (in bytes)
mem_skew	The indicator that displays the RAM consumption skew across cluster segments at the end of the command execution (percentage)
vmem_sum	The virtual memory amount allocated at the end of the command execution (in bytes)
vmem_max	The maximum amount of virtual memory allocated at the end of the command execution (in bytes)
vmem_avg	The average amount of virtual memory allocated across cluster segments at the end of the command execution (in bytes)
vmem_skew	The indicator that displays the virtual memory consumption skew across cluster segments at the end of the command execution (percentage)
spill	The amount of spill files on cluster segment hosts (excluding master) at the end of the command execution (in bytes)
spill_hosts	The amount of spill files on cluster hosts (including master) at the end of the command execution (in bytes)
spill_max	The maximum amount of spill files produced on cluster segments at the end of the command execution (in bytes)
spill_skew	The indicator that displays the spill file amount skew at the end of the command execution (percentage)
read_bytes_total	The total amount of data read during the command execution (in bytes)
read_bytes_sum	The amount of data read at the end of the command execution (in bytes)
read_bytes_max	The maximum amount of data read at the end of the command execution (in bytes)
read_bytes_avg	The average amount of data read across cluster segments per second at the end of the command execution (in bytes)
read_bytes_skew	The indicator that displays the data reading skew across cluster segments at the end of the command execution (percentage)
write_bytes_total	The total amount of data written during the command execution (in bytes)
write_bytes_sum	The amount of data written at the end of the command execution (in bytes)
write_bytes_max	The maximum amount of data written at the end of the command execution (in bytes)
write_bytes_avg	The average amount of data written across cluster segments per second at the end of the command execution (in bytes)
write_bytes_skew	The indicator that displays the data writing skew across cluster segments at the end of the command execution (percentage)

query

This table contains information on completed SQL commands that were registered in ADB Control.

Column	Description
query_id	A unique command identifier
status	A command status. Possible values: • `Done` — successfully completed commands. • `Cancelled` — cancelled commands (as a part of the transaction cancellation or termination). • `Error` — the commands during which some errors occurred. • `Unknown` — the commands, the final status of which is undefined. It is set on a timeout for queries in the `Queued`, `Running`, `Cancelling` statuses, for which there are no corresponding rows in the `pg_stat_activity` view.
pid	An identifier of the process, within which the command was started
submitted	The timestamp when the user submitted the command
run_at	A command start timestamp
finish_at	A command end timestamp
username	A name of the user who started the command
database	A name of the database where the command was started
resource_group	A name of the resource group that was used for the command
planner	A name of the query optimizer that was used to produce the query execution plan. Possible value: • `GPORCA` — GPORCA is used (`optimizer = on`). • `Fallbacked` — after the attempt to produce a plan via GPORCA (`optimizer = on`) failed, Postgres query optimizer is used. • `Legacy` — Postgres query optimizer is used (`optimizer = off`).
querytext	A text of the SQL query
plantext	A text of the command execution plan
session_id	An identifier of the session, within which the command was started
cluster_id	An identifier of the ADB cluster where the command was started. Refers to `cluster.id` (see cluster)
queued_time	The total time that the command was queued (in hours, minutes, seconds)
run_time	The total command duration (in hours, minutes, seconds)
sql_id	A common identifier for SQL commands with the same structure
tags	The query tags

query_error

This table contains information about errors that occurred during execution of SQL commands.

Column	Description
id	A unique identifier of the error record
query_id	A command identifier. Refers to `query.id` (see query)
seg_id	A content identifier for a segment instance. Corresponds to `gp_segment_configuration.content`. For more information, see the ***System catalog tables → gp_segment_configuration*** section of the Tables overview article
slice_id	An identifier of the slice during which execution the current error occurred
message	An error message text

resgroup_config_audit

This table stores information on configuration changes of resource groups.

Column	Description
cluster_id	An identifier of the ADB cluster. Refers to `cluster.id` (see cluster)
group_id	A resource group identifier
datetime	A timestamp of the resource group configuration modification
group_name	A resource group name
limit_type	A type of the modified resource group parameter. Possible values: • `CONCURRENCY` • `CPU_RATE_LIMIT` • `MEMORY_LIMIT` • `MEMORY_SHARED_QUOTA` • `MEMORY_SPILL_RATIO` • `CPUSET` For more details on resource group parameters, see the ***Configuration*** section of the Work with resource groups article
value_from	An original value of the modified resource group parameter
value_to	A new value of the modified resource group parameter
username	A name of the user who changed the resource group configuration

relation_statistic

The table contains the usage statistics of ADB relations

Column	Description
date	A date of the user access to the relation
cluster	A cluster name
database	A database name
schema	A schema name
relation	A relation name
username	A name of the user who accessed the relation during the selected day
total_count_per_day	A total number of user accesses to the relation during the selected day
last_access_at	A timestamp of the last user access to the relation during the selected day

resgroup_status

This table stores the statistics of the resource group usage. For more information, see the ***Statistic*** section of the Work with resource groups article.

Column	Description
cluster_id	An identifier of the ADB cluster. Refers to `cluster.id` (see cluster)
group_id	A resource group identifier
datetime	The timestamp when the measurements were taken
group_name	A resource group name
num_running	A number of transactions running at the `datetime` moment within the current resource group
num_queueing	A number of transactions queued at the `datetime` moment within the current resource group
num_queued	A total number of transactions queued within the current resource group over the entire observation period
num_executed	A total number of transactions executed within the current resource group over the entire observation period
total_queue_duration	The total time that all transactions were queued within the current resource group over the entire observation period (in hours, minutes, seconds)

resgroup_status_per_segment

This table contains statistics on the consumption of system resources by resource groups on each cluster segment. For more information, see the ***Resource group details → Usage*** section of the Work with resource groups article.

Column	Description
cluster_id	An identifier of the ADB cluster. Refers to `cluster.id` (see cluster)
group_id	A resource group identifier
hostname	A name of the host for which the measurements were taken
segment_id	A content identifier for a segment instance. Corresponds to `gp_segment_configuration.content`. For more information, see the ***System catalog tables → gp_segment_configuration*** section of the Tables overview article
datetime	The timestamp when the measurements were taken
cpu	The percentage of CPU usage by the resource group on the current segment
memory_used	The total RAM value used by the resource group on the current segment (in bytes). This value includes the resource group fixed and shared memory
memory_available	The total RAM value available to the resource group on the current segment (in bytes). This value includes the resource group fixed and shared memory
memory_quota_used	The RAM value used by the resource group on the current segment — from the amount that is guaranteed fixed for resource group transactions (in bytes)
memory_quota_available	The RAM value available to the resource group on the current segment — from the amount that is guaranteed fixed for resource group transactions (in bytes)
memory_shared_used	The RAM value used by the resource group on the current segment — from the amount that is commonly used by all resource group transactions (in bytes)
memory_shared_available	The RAM value available to the resource group on the current segment — from the amount that is commonly used by all resource group transactions (in bytes)

session

This table contains information on sessions that were registered in ADB Control.

Column	Description
cluster_id	An identifier of the ADB cluster. Refers to `cluster.id` (see cluster)
datetime	The timestamp when the data was received
datid	A database OID
datname	A database name
pid	A backend process identifier
sess_id	A session identifier
usesysid	An OID of the user who connected to the backend
usename	A name of the user who connected to the backend
application_name	A name of the application that is connected to the backend
client_addr	A client IP address. The `null` value indicates the use of a Unix socket on the server or the execution of an internal process (e.g. `autovacuum`)
client_hostname	A host name of the connected client, as reported by a reverse DNS lookup of the `client_addr` value. The field is not `null` only for IP connections, and only when the server configuration parameter (GUC) log_hostname is set to `on`
client_port	The TCP port number that is used by the client to communicate with the backend, or `-1` if a Unix socket is used
backend_start	The time when the backend process was started
xact_start	A transaction start timestamp
query_start	A command execution start timestamp
state_change	The last time the value of the `state` field was changed
waiting	`true` in case of waiting on a lock, otherwise `false`
state	The current session state. Possible values: • `active` — the backend is running a query. • `idle` — the backend is waiting for a new client command. • `idle in transaction` — the backend is in a transaction, but there are no currently running queries.
query	A text of the most recent SQL query that was run at the backend. If `state` is `active`, the field shows the currently running query
waiting_reason	The reason why the server process is waiting. Possible values: • `lock`; • `replication`; • `resgroup` — resource group queue.
rsgid	A resource group OID or `0` if definition is not possible (for more information, see pg_stat_activity in the Greenplum documentation)
rsgname	A resource group name or `unknown` if definition is not possible (for more information, see pg_stat_activity in the Greenplum documentation)
rsgqueueduration	The total time the query has been queued (if applicable)

transaction

This table contains information on completed transactions that were registered in ADB Control.

Column	Description
txn_id	A unique transaction identifier in ADB Control
status	A transaction status. Possible values: • `Committed` — successfully completed transactions. • `Aborted` — cancelled/terminated transactions. • `Unknown` — the transactions, the final status of which is undefined. It is set on a timeout for transactions in the `Active` status, for which there are no corresponding rows in the `pg_stat_activity` view.
pid	An identifier of the process, within which the transaction was started
start_time	A transaction start timestamp
end_time	A transaction end timestamp
username	A name of the user who started the transaction
database	A name of the database where the transaction was started
resource_group	A name of the resource group that was used for the transaction
session_id	An identifier of the session, within which the transaction was started
cluster_id	An identifier of the ADB cluster where the transaction was started. Refers to `cluster.id` (see cluster)
tags	The transaction tags

Cluster management

Daria Barysheva

Contents

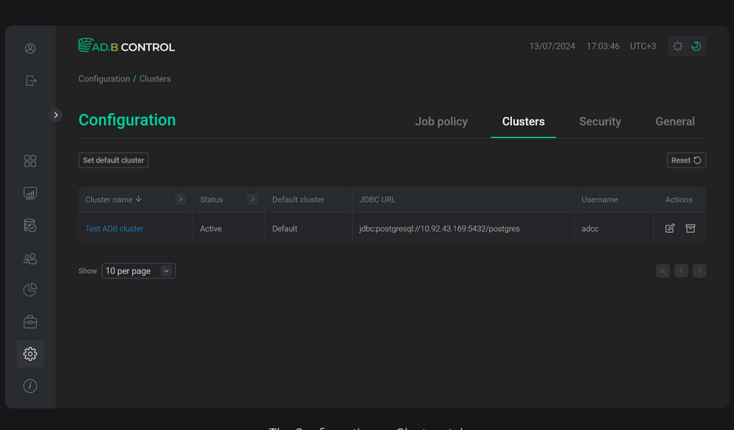
To Table of Contents

ADB Control allows you to monitor multiple ADB clusters simultaneously. In this case, clusters do not have to use the same ADCM.

To manage clusters, you can use the ***Clusters*** tab on the ***Configuration*** page in the ADB Control web interface.

View a list of clusters

The ***Configuration*** → ***Clusters*** tab displays the following information on clusters.



The Configuration → Clusters tab

The "Configuration → Clusters" tab fields

Field	Description
Cluster name	A cluster name
Status	A cluster status: ▪ `Active` — the cluster is currently monitored. ▪ `Archived` — the cluster monitoring is suspended.
Default cluster	For a default cluster, the `Default` label is shown. Otherwise, `No`
JDBC URL	A URL of the JDBC connection
Username	A user name that is used to connect to a cluster

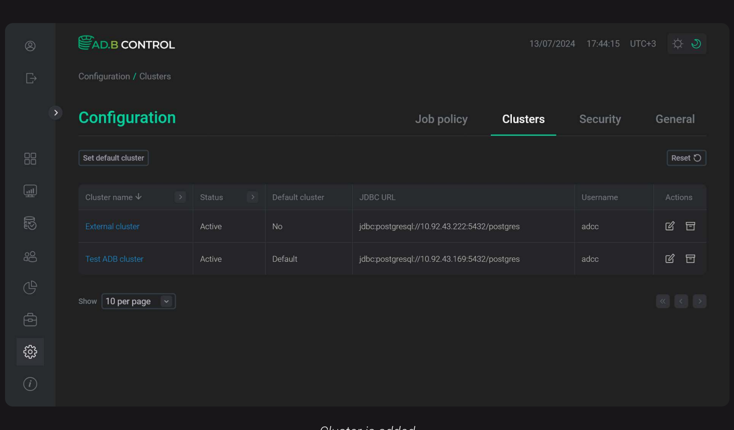
In the column headers of the table with a list of clusters, there are filters that you can use to select specific data. To open a filter, click the › icon. For those columns where the set of possible values is limited (e.g. ***Status***), you can select a value from the drop-down list. For some columns (e.g. ***Cluster name***), the search value should be entered.

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

Add a cluster

To add a new ADB cluster to be monitored in ADB Control, follow the steps:

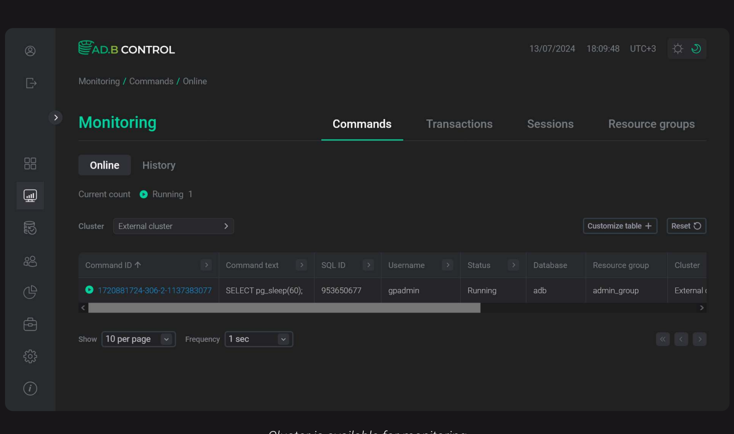
1. Using ADCM, in the ADB cluster that should be added to the existing monitoring system — add ADBC and ADBM agents, import ADB ES configuration, and install agents.
2. In the ADB Control web interface, open the ***Configuration*** → ***Clusters*** tab. The new cluster should be added automatically.



Cluster is added

3. To ensure that the cluster connection is successful, you can run an SQL query in one of the cluster databases that were previously selected for monitoring, and verify that the query information appears on the ***Monitoring*** page. Note that the added cluster must be pre-selected in the filters on the page (if the cluster is not marked as default). You can use a test query as follows:

```
SELECT pg_sleep(60);
```

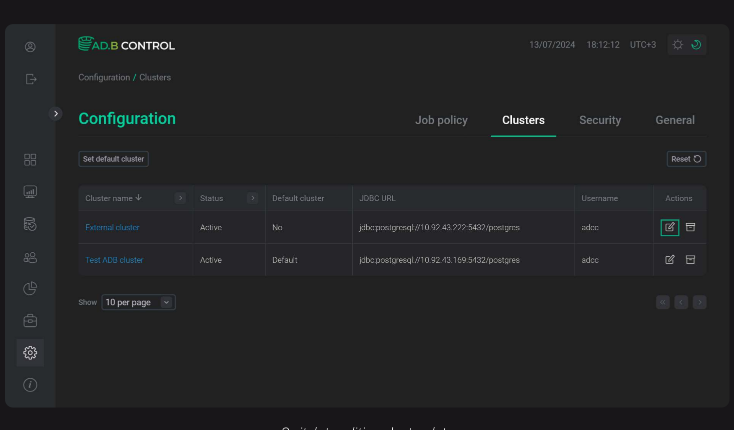


Cluster is available for monitoring

Edit a cluster connection

To edit cluster data, follow the steps:

1. Click the icon in the ***Actions*** column on the ***Configuration*** → ***Clusters*** tab.



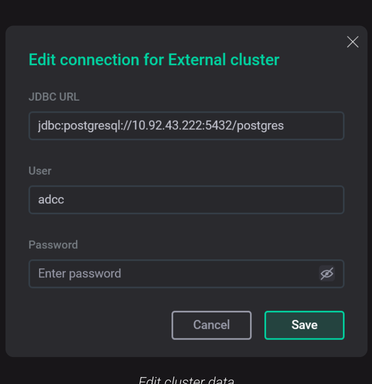
Switch to editing cluster data

2. In the window that opens, edit necessary fields (the ***Cluster name*** field value cannot be changed):

   - ***JDBC URL*** — a URL of the JDBC connection to the master host of the ADB cluster. Use the following format:

```
jdbc:postgresql://<ADB master host IP address>:5432/postgres
```

   - ***User*** — a user name. Use `adcc` by default.
   - ***Password*** — a user password. The default user `adcc` does not require a password.



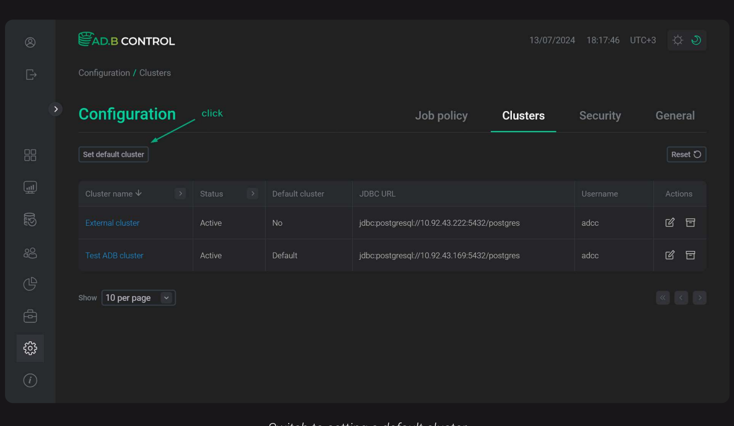
Edit cluster data

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

Set a default cluster

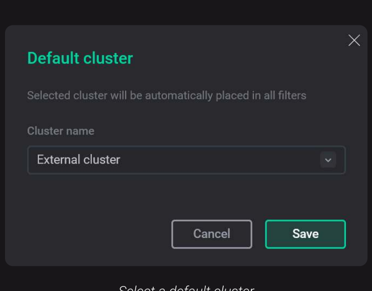
Many pages of the ADB Control web interface have cluster name filters. To assign the cluster to be selected in these filters by default, follow the steps:

1. Click ***Set default cluster*** on the ***Configuration*** → ***Clusters*** tab.



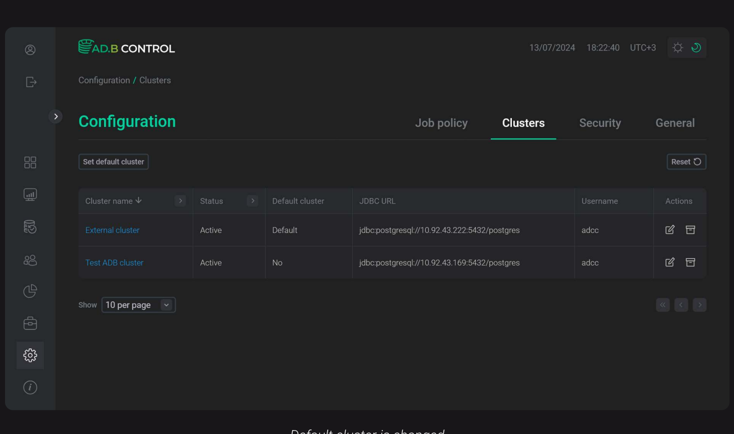
Switch to setting a default cluster

2. In the window that opens, select a default cluster from the ***Cluster name*** drop-down list.



Select a default cluster

3. Click ***Save***. As a result, the ***Default cluster*** column in the table of clusters displays `Default` for a new default cluster and `No` — for a previous default cluster.



Default cluster is changed

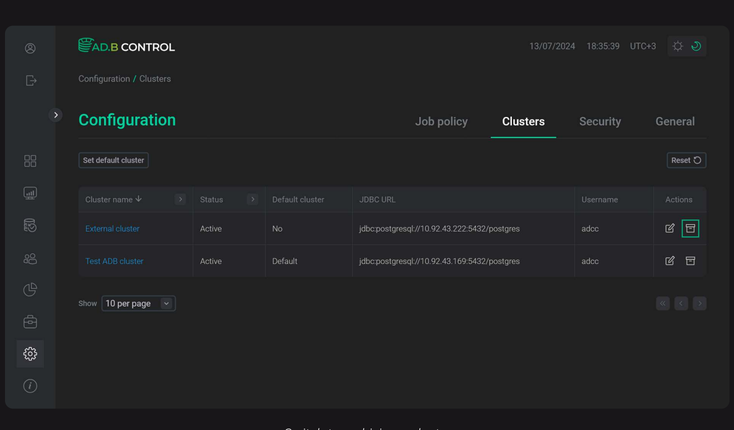
Archive a cluster

In some cases, it is necessary to suspend the processing of monitoring metrics in ADB Control for a specific cluster. For this purpose, you can **archive** a cluster, which means:

- ADB Control does not collect audit events for the cluster.
- The export job skips the data related to the archived cluster.
- Resource group statistics is not collected for the cluster.
- On the ***Information*** page, there are no cluster details.

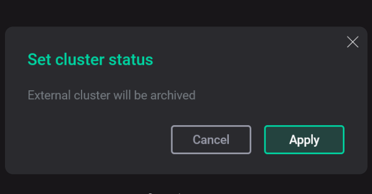
To archive a cluster, follow the steps:

1. Click the icon in the ***Actions*** column on the ***Configuration*** → ***Clusters*** tab.



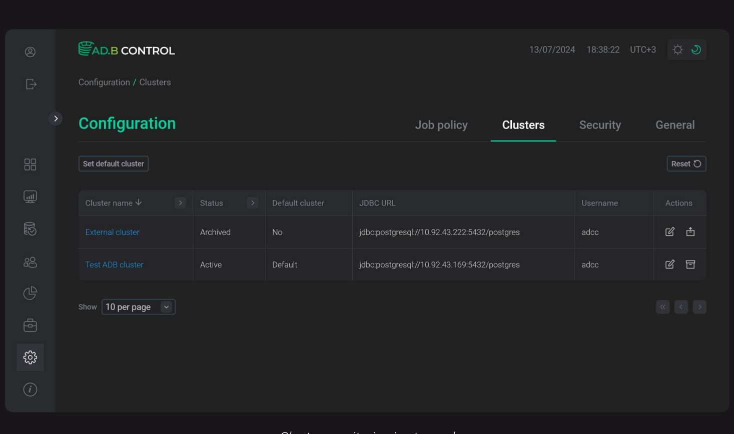
Switch to archiving a cluster

2. In the window that opens, confirm the operation by clicking ***Apply***.



Confirm the operation

As a result, the cluster status is changed to `Archived` on the ***Configuration*** → ***Clusters*** tab.

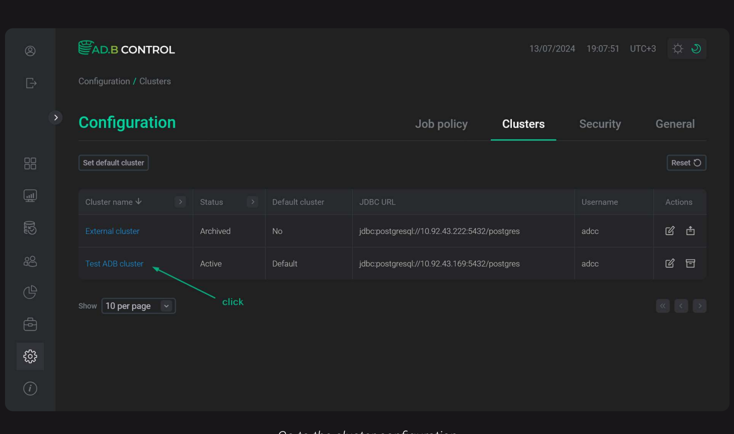


Cluster monitoring is stopped

> **NOTE**
>
> If you later need to resume the cluster monitoring, click the icon in the ***Actions*** column for that cluster on the ***Configuration*** → ***Clusters*** tab and confirm the action in the window that opens.

Configure monitoring of the selected cluster

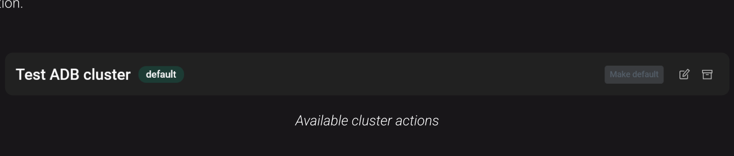
To navigate to the cluster monitoring configuration, click a cluster name on the ***Configuration*** → ***Clusters*** tab.



Go to the cluster configuration

The top of the opened page displays a cluster name, the `default` label (if a cluster is selected by default), and a number of control elements for a quick transition to the actions described above:

- ***Make default*** — make a cluster default.
- — archive a cluster.
- — activate an archived cluster.
- — edit a cluster connection.

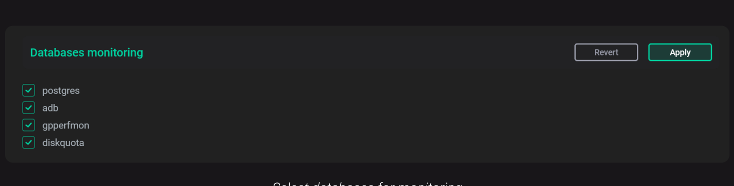


Available cluster actions

The main part of the opened page contains several sections that are described below. To apply any changes to the parameters in these sections, click ***Apply*** in the appropriate section, to undo the changes that have not been yet saved — click ***Revert***.

Databases monitoring

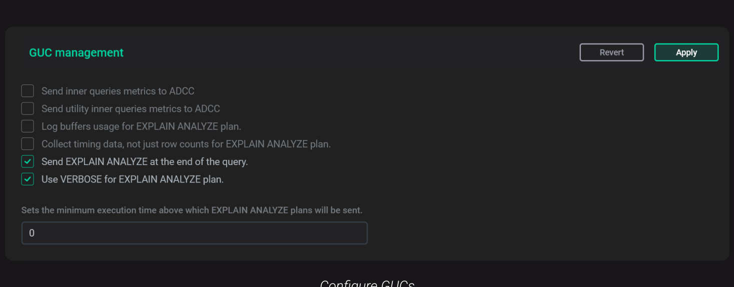
In the ***Databases monitoring*** section on the cluster page, you can select databases that should be monitored in ADB Control. By default, all databases are selected.



Select databases for monitoring

GUC management

In the ***GUC management*** section on the cluster page, you can configure values of the GUCs that control ADB Control behavior.



Configure GUCs

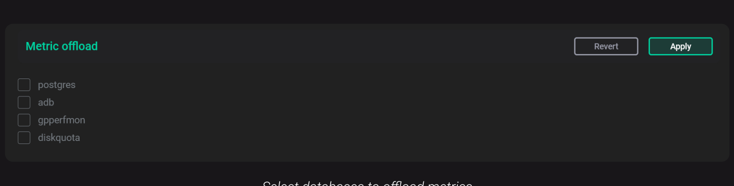
Available GUCs are shown in the table below.

GUCs that manage ADB Control behavior

GUC name	UI description
adcc.monitor_inner_queries	Send inner queries metrics to ADCC
adcc.monitor_utility_inner_queries	Send utility inner queries metrics to ADCC
adcc.explain_log_buffers	Log buffers usage for EXPLAIN ANALYZE plan
adcc.explain_log_timing	Collect timing data, not just row counts for EXPLAIN ANALYZE plan
adcc.explain_log_analyze	Send EXPLAIN ANALYZE at the end of the query
adcc.explain_log_verbose	Use VERBOSE for EXPLAIN ANALYZE plan
adcc.explain_log_min_duration	Sets the minimum execution time above which EXPLAIN ANALYZE plans will be sent (in milliseconds)

Metric offload

The ***Metric offload*** section of the cluster page allows you to select databases, whose metrics will be offloaded to the external analytical database (according to the schedule that is configured on the Configuration → Job policy → Metrics offload page). By default, no database is selected.



Select databases to offload metrics

Configure security policies

Daria Barysheva

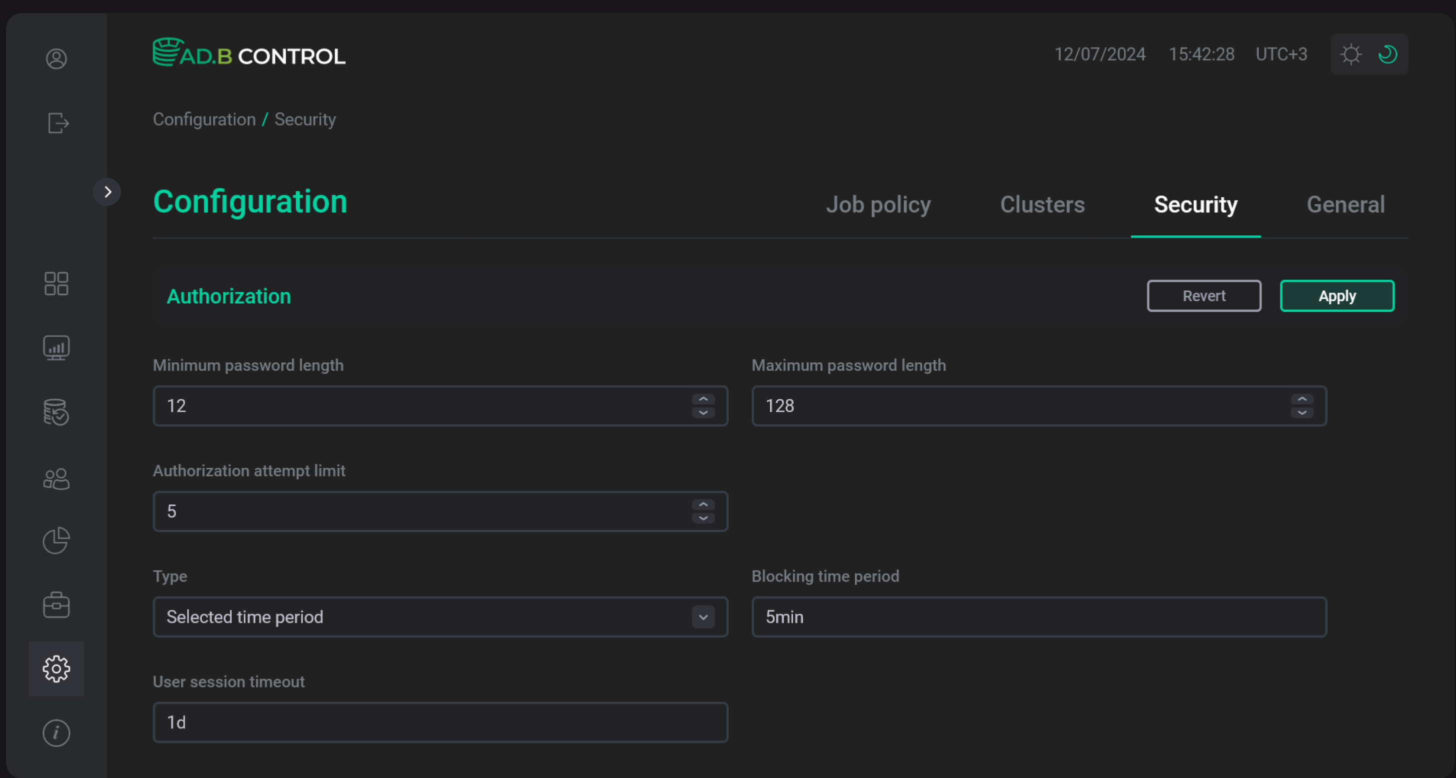
To Table of Contents

The ***Security*** tab on the ***Configuration*** page in the ADB Control web interface allows you to configure security policies. On this tab, you can define the following restrictions, which will be checked during user creation and authorization.

The "Configuration → Security" tab fields

Field	Description	Default value
Minimum password length	The minimal password length for user creation	12
Maximum password length	The maximum password length for user creation	128
Authorization attempt limit	The maximum number of failed login attempts that permanently or temporarily blocks the user (depending on the ***Type*** value)	5
Type	The blocking type that is to be used when the maximum number of failed logon attempts (***Authorization attempt limit***) is reached. Possible values: ■ `Permanently` — block permanently until the system owner performs unblocking. ■ `Selected time period` — block temporarily for the time specified in the ***Blocking time period*** field (or until the system owner performs unblocking).	Selected time period
Blocking time period	The temporary user blocking period in the `<value><unit>` format (without spaces), where `<unit>` has the following possible values: ■ `min` — minutes; ■ `hr` — hours; ■ `d` — days. Examples: `5min`, `1hr`, `1d`	5min
User session timeout	The timeout after which ADB Control user sessions should be terminated. Requires the same format as the ***Blocking time period*** field	1d

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



The Configuration → Security tab

General settings

Daria Barysheva

Contents

To Table of Contents

The ***General*** tab on the ***Configuration*** page in the ADB Control web interface allows you to configure general-purpose settings. This page consists of three tabs, each of which is described in detail below.

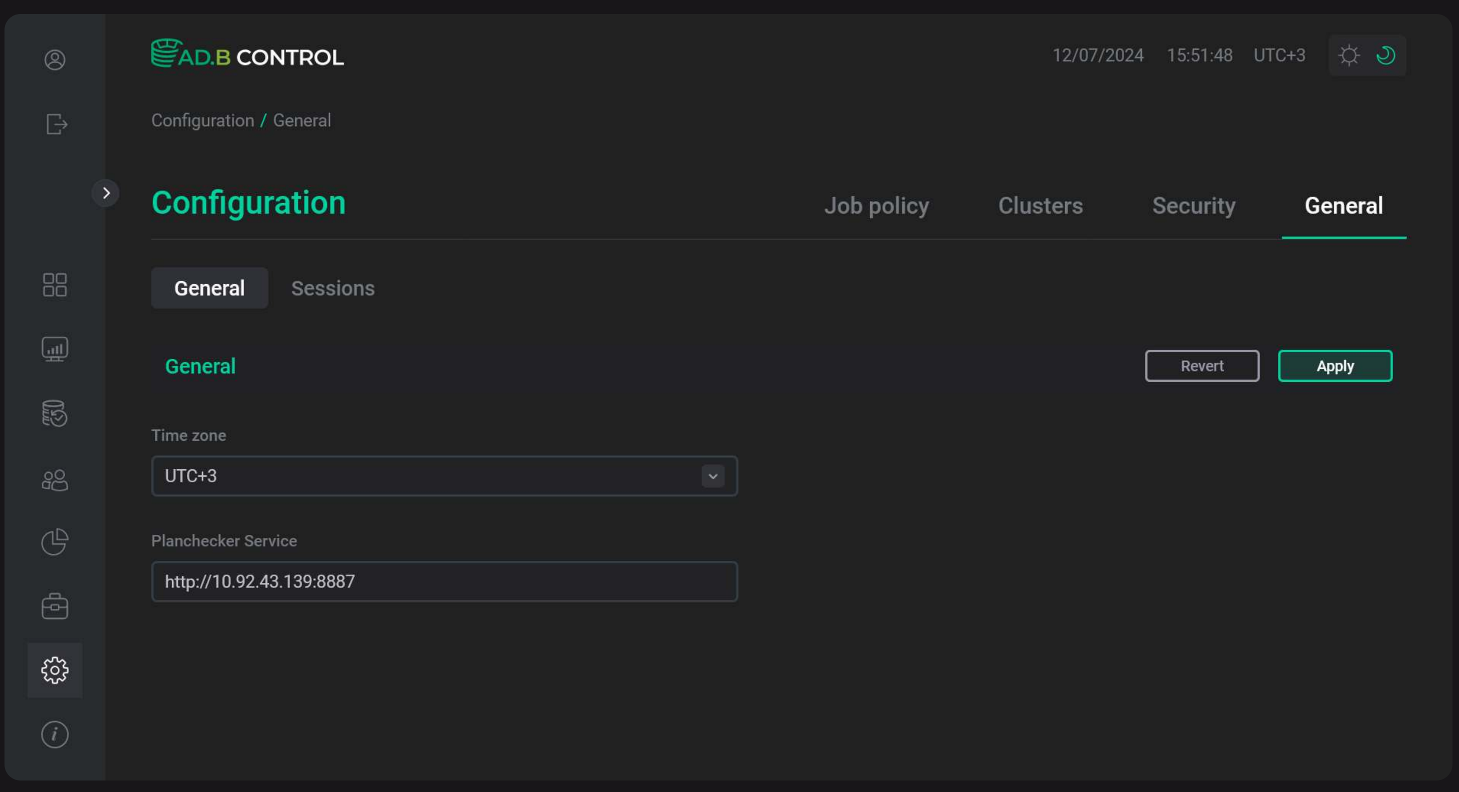
To change the default settings, edit the fields and click ***Apply***. Click ***Revert*** if you need to undo the changes that have not been yet saved by clicking ***Apply***.

General

The ***General*** tab allows you to set the general parameters.

The "Configuration → General → General" tab fields

Field	Description	Default value
Time zone	A user time zone	UTC+3
Planchecker service	Planchecker URL. Requires the following format: *http://<ADB Control IP>:8887*. This parameter is deprecated starting with ADB Control 4.16.0 and will be removed in the future releases	—



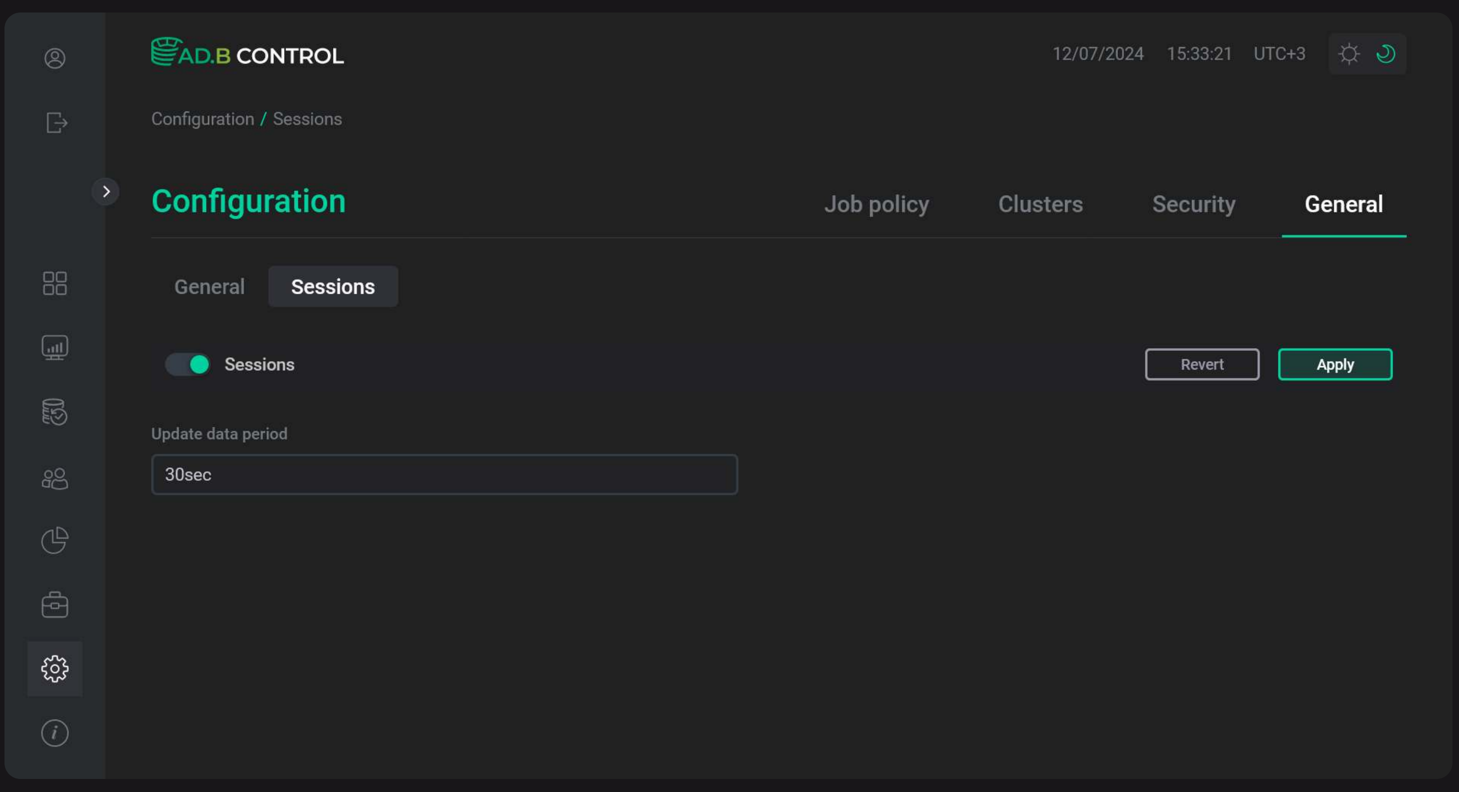
The Configuration → General → General tab

Sessions

The ***Sessions*** tab is designed to configure session monitoring.

The "Configuration → General → Sessions" tab fields

Field	Description	Default value
Sessions	A switcher that indicates whether to monitor sessions in ADB Control	True
Update data period	Indicates how often to receive information on sessions	30sec



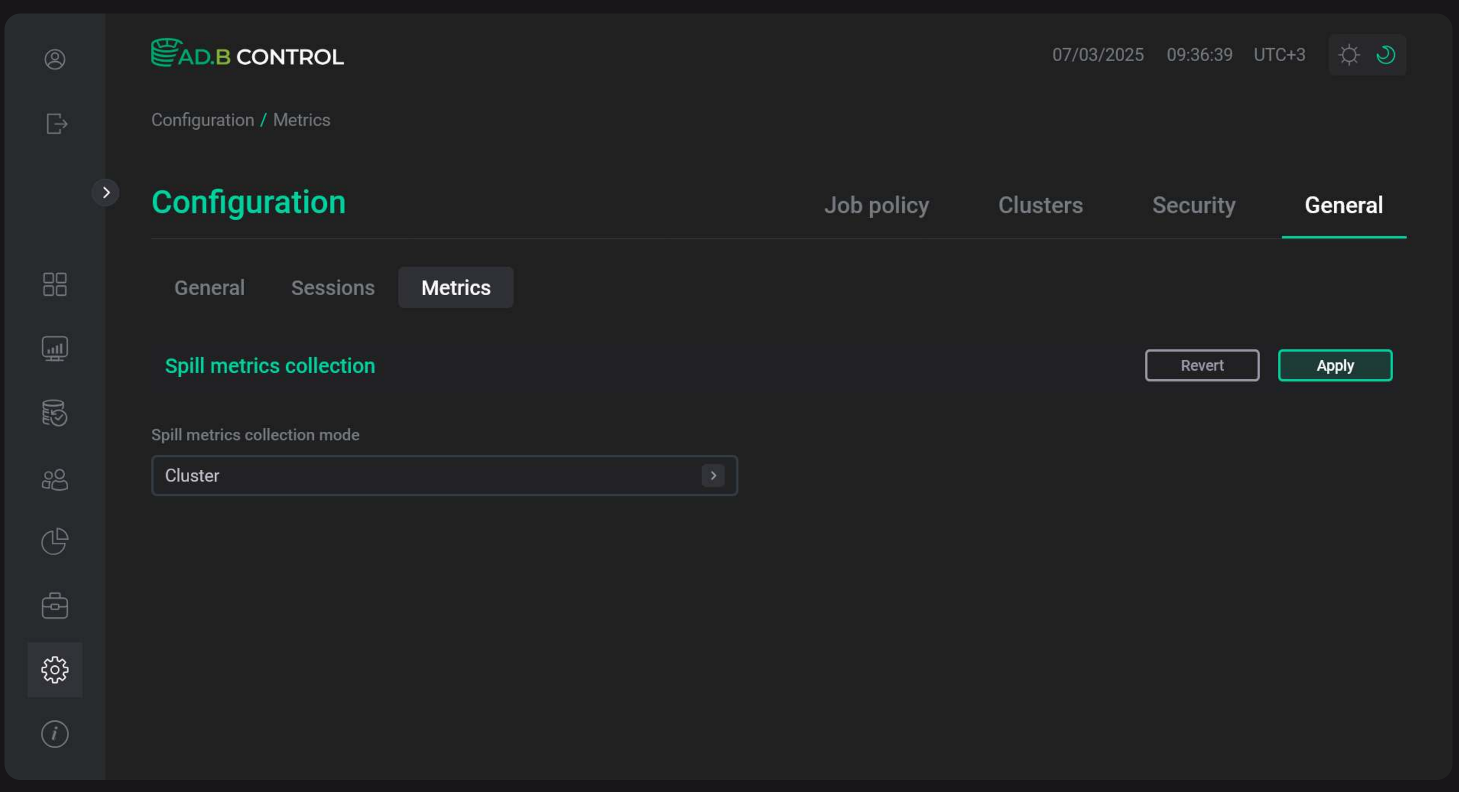
The Configuration → General → Sessions tab

Metrics

The ***Metrics*** tab allows you to configure collecting various metrics. This tab is available starting with ADB Control 4.12.0.

The "Configuration → General → Metrics" tab fields

Field	Description	Default value
Spill metrics collection mode	Mode of collecting spill file metrics: ▪ `Cluster` — only for cluster; ▪ `Cluster and segment` — for whole cluster and each segment. If the `Cluster and segment` value is selected, the ***Spill size*** and ***Spill files number*** metrics will be collected and shown for each cluster segment in the Performance section on the page with command details	Cluster



The Configuration → General → Metrics tab

Dashboard

Daria Barysheva

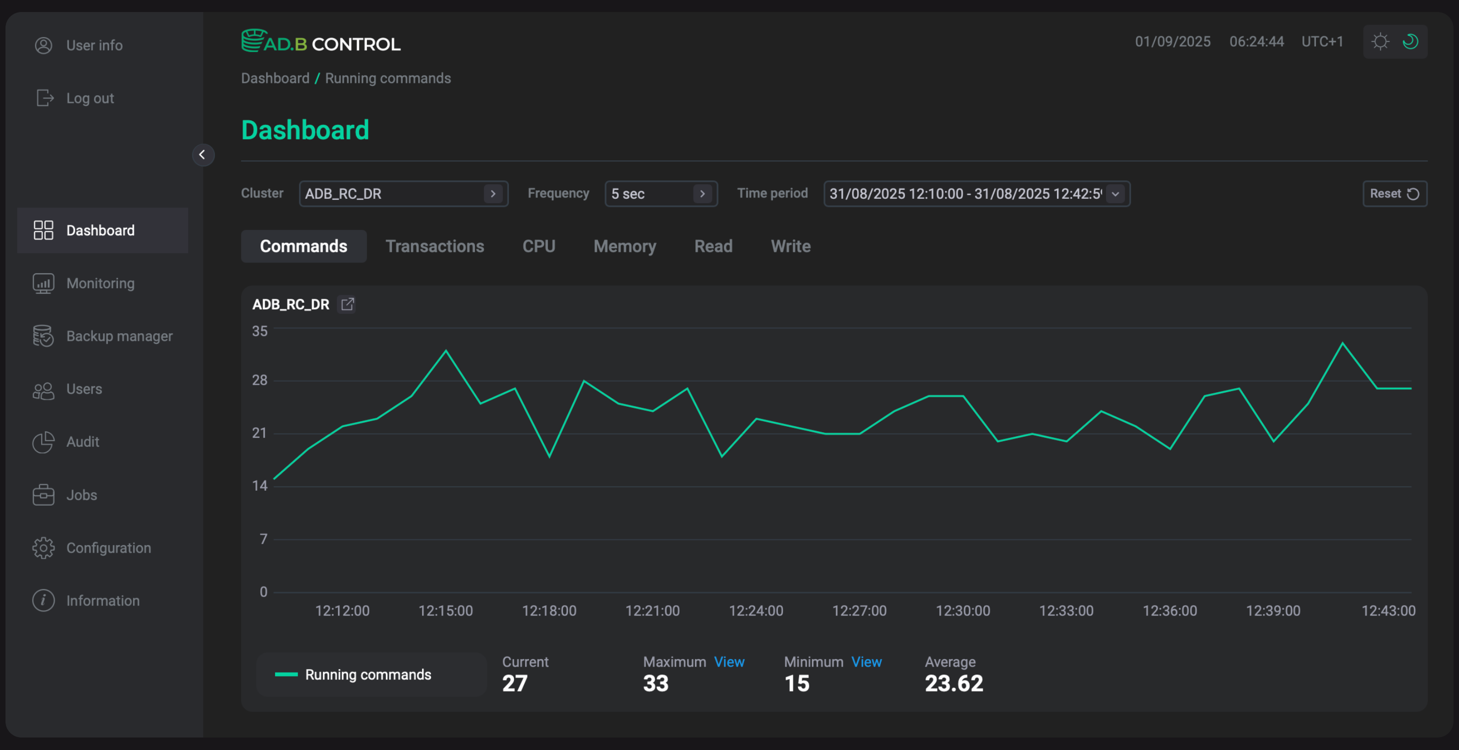
To Table of Contents

The **Dashboard** page in the ADB Control web interface allows you to display charts for tracking the dynamic changes of various metrics over the selected time period in the ADB clusters connected for monitoring.

Each tab on the page corresponds to a separate chart:

- ***Commands*** — a number of SQL commands in the `Running` status.
- ***Transactions*** — a number of transactions in the `Active` status.
- ***CPU*** — a total CPU consumption in the ADB cluster (percentage).
- ***Memory*** — a total virtual memory allocation in the ADB cluster (percentage).
- ***Read*** — a total data read amount in the ADB cluster (in bytes).
- ***Write*** — a total data write amount in the ADB cluster (in bytes).

To open a full screen view of any of the charts, click the icon to the right of the cluster name in the top part of the chart.

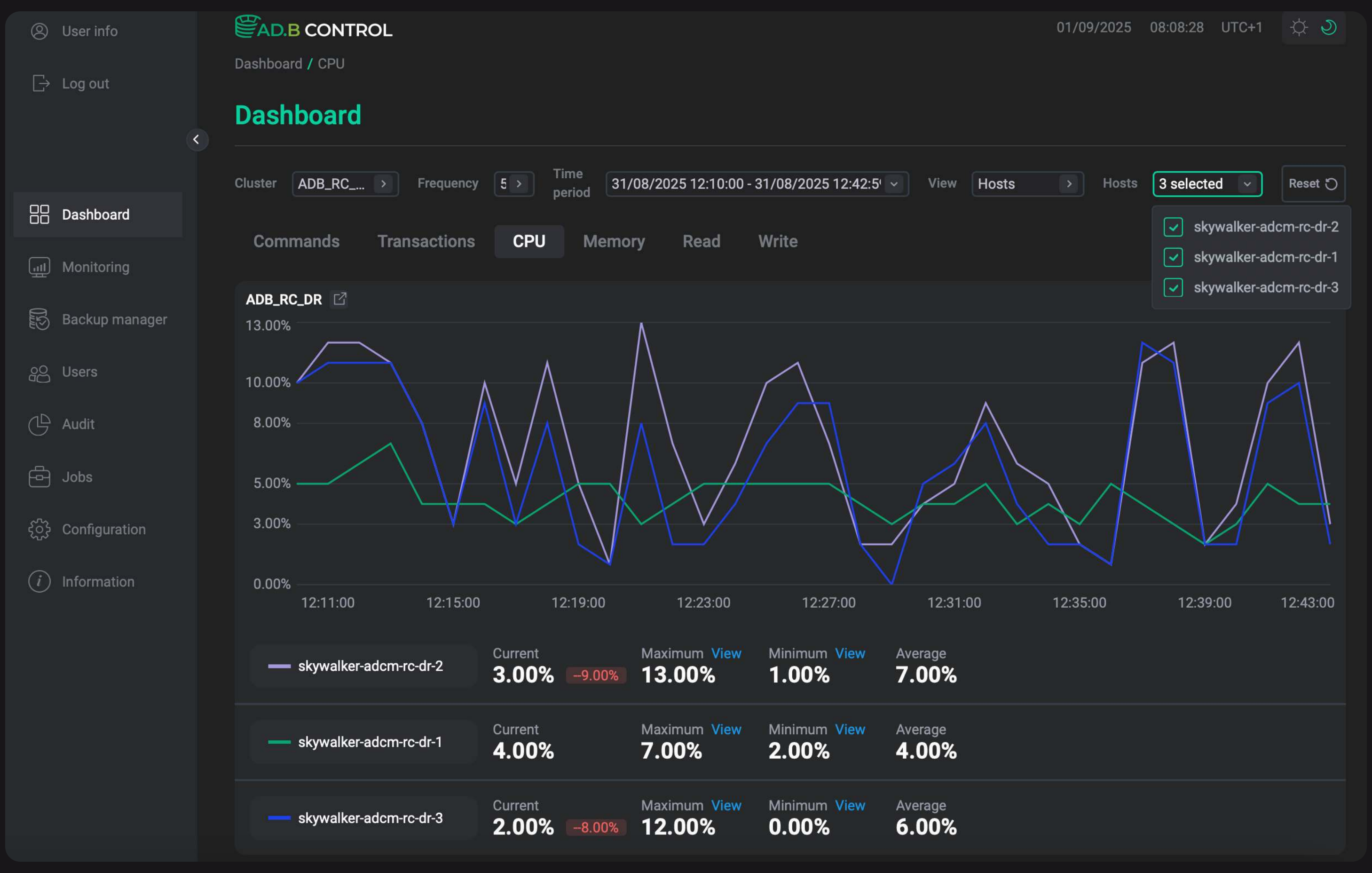


The Dashboard page

To configure charts, you can use the following fields in the top part of the page:

- ***Cluster*** — the ADB cluster for which all charts should be drawn. The default cluster can be selected in the ADB Control settings.
- ***Frequency*** — defines how often to update charts.
- ***Time period*** — the time period for which a chart is required. In the window that opens after clicking the field, you can select an interval from the offered options or set the custom time range.
- ***View*** — defines how to display charts. The field is hidden on the ***Commands*** and ***Transactions*** tabs. Possible values:
  - `Summary` — display a summary chart for the whole ADB cluster.
  - `Hosts` — display information for each cluster host in the form of separate charts.
- ***Hosts*** — the drop-down list that is used to select cluster hosts. Becomes enabled if the `Hosts` value is selected in the ***View*** field.

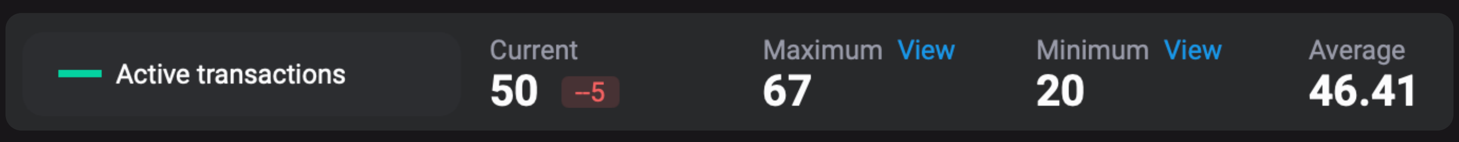
The figure below shows the option of displaying chart information in the context of cluster hosts.



Display a chart in the context of hosts

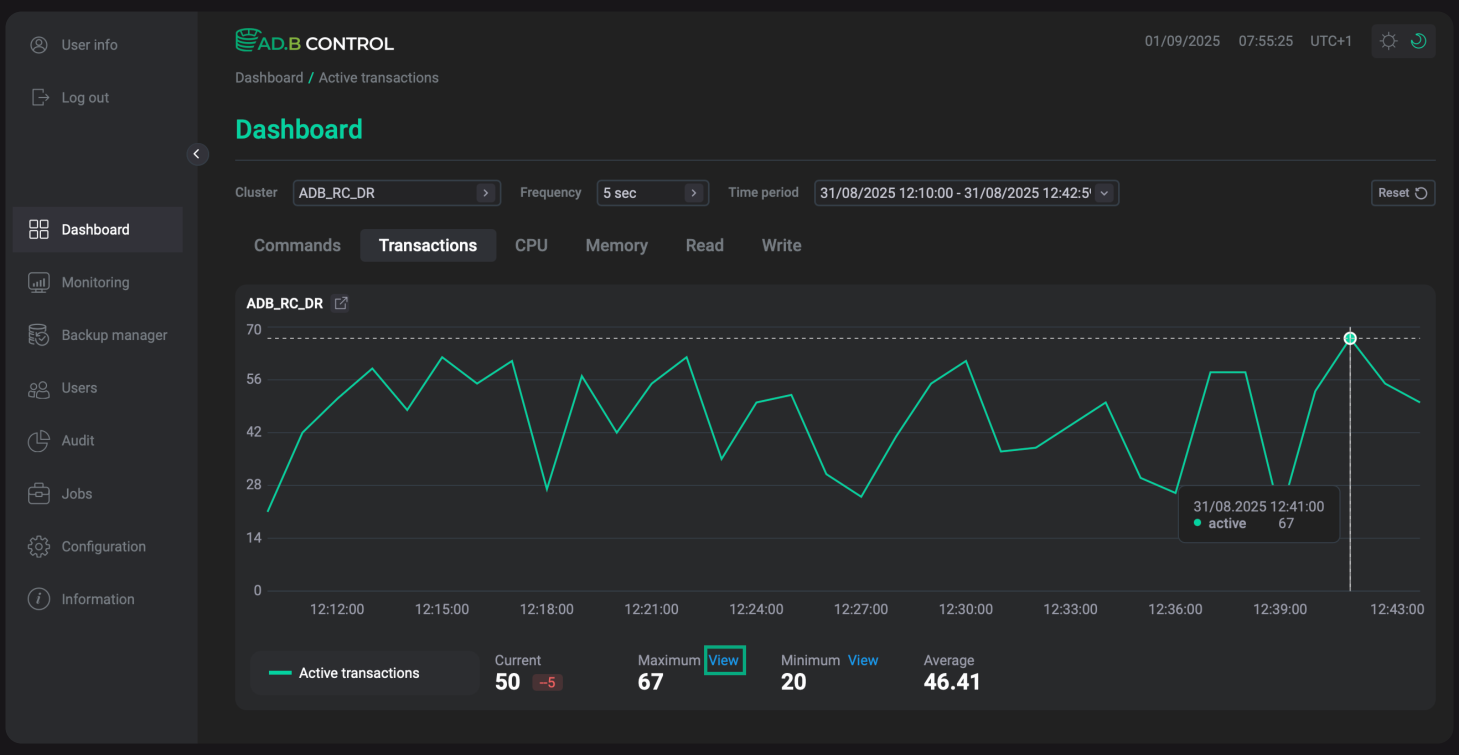
On each tab, you can see the following information under charts:

- ***Current*** — the last metric value received over the selected time period. To the right of the current value, you can see a special indicator that shows the difference between the previous and current metric values (if any).
- ***Maximum*** — the maximum metric value received over the selected time period. By clicking ***View***, you can head to the maximum value on the chart.
- ***Minimum*** — the minimum metric value received over the selected time period. By clicking ***View***, you can head to the minimum value on the chart.
- ***Average*** — the average value, which is calculated based on all metric values received over the selected time period.



Aggregated metric information

Below, you can see how to highlight the maximum or minimum metric value on the chart by clicking ***View***.



View the maximum metric value on the chart

You can also see a metric value for any other time point by hovering the mouse over the chart.

Monitoring of commands

Daria Barysheva

Contents

To Table of Contents

The ***Monitoring → Commands*** page in the ADB Control web interface displays the SQL commands that are performed in the ADB clusters connected to the monitoring system. This page includes two tabs:

- Queries shows individual SQL commands.
- Groups shows SQL commands grouped by SQL ID, node number, and node type. The tab is available starting with ADB Control 4.14.2.

Queries

The ***Queries*** tab displays SQL commands. It includes two tabs, ***Online*** and ***History***, described in detail below.

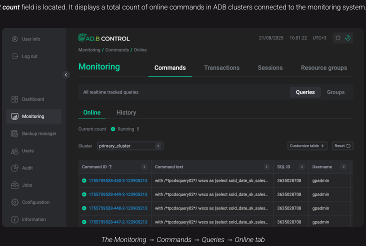
> - By default, ADB Control monitors (and displays on the ***Monitoring → Commands*** page) SQL commands longer than 5 seconds. You can configure this threshold in configuration parameters of the ADBC agents service in ADB, under ***ADB control agent parameters → Environment key-value*** — by clicking ***Add property*** and sequentially specifying the `ADCC_QUERY_ID_CACHE_MOVE_TO_ACTIVE_TIMEOUT_SECONDS` variable name and its value.
> - Starting with ADB Control 4.1.0, only top-level queries are fixed in the monitoring system by default. This excludes all queries that are called during the processing of the top-level query within functions, SPI (Server Programming Interface), etc. If you need to enable monitoring for inner queries, set the `true` value for the following GUCs (starting with 4.7.5 version, you can edit them in the ADB Control web UI):
>   - `adcc.monitor_utility_inner_queries` — defines whether to monitor inner queries for utility commands like `COPY`, `VACUUM`, DDL, and so on.
>   - `adcc.monitor_inner_queries` — defines whether to monitor inner queries for other commands.

Online

The ***Monitoring → Commands → Queries → Online*** tab displays commands in the following statuses:

- `Queued` — the commands on which information about the actual start of execution has not yet received.
- `Running` — the commands that are currently running.
- `Cancelling` — the commands that are currently cancelling (as a part of the transaction cancellation or termination).

At the top of the tab, the ***Current count*** field is located. It displays a total count of online commands in ADB clusters connected to the monitoring system.

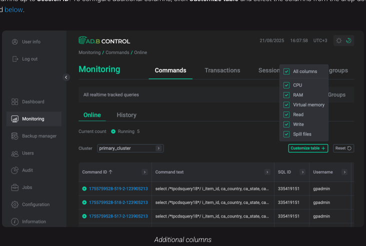


The Monitoring → Commands → Queries → Online tab

On the ***Monitoring → Commands → Queries → Online*** tab, there is a table with the following information on commands.

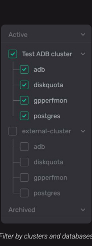
Field	Description
Command ID	A unique command identifier, which includes: - cluster start timestamp; - session number; - command number within the current session; - hash code based on the cluster name (can be negative).
Command text	The first symbols of a command text. To view the full text (in case of long queries), hover the mouse over the field value
Cluster	The name of the ADB cluster where the command is launched
Submitted	The timestamp when the user submitted the command (in the `DD/MM/YYYY HH:mm:ss` format)
Run time	A total command duration in hours, minutes, seconds
Session ID	A unique session identifier assigned by the ADB master when a client connection is established
SQL ID	A common identifier for SQL commands with the same structure
Username	A name of the user who started the command
Status	A command status. Possible values are listed above
Database	A name of the database where the command is launched
Resource group	A name of the resource group that is used for the command
Planner	A name of the query optimizer that is used to produce the query execution plan. Possible values: - `GPORCA` — GPORCA is used (`optimizer = on`). - `Fallbacked` — after the attempt to produce a plan via GPORCA (`optimizer = on`) failed, Postgres query optimizer is used. - `Legacy` — Postgres query optimizer is used (`optimizer = off`).
Tags	The command tags

By default, the table displays columns up to ***Session ID***. To configure additional columns, click ***Customize table*** and select the columns from the drop-down list. In that list, you can also select metrics that are described below.



Additional columns

Above the table with a list of commands, the ***Cluster*** filter is located. You can use this filter to select the ADB cluster and its databases for which you want to display data in the table. Initially all databases of the default cluster are selected.



Filter by clusters and databases

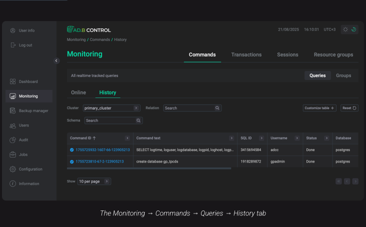
In the column headers of the table with a list of commands, there are filters that you can use to select specific data. To open a filter, click the icon. For those columns where the set of possible values is limited (e.g. ***Status***), you can select a value from the drop-down list. For some columns (e.g. ***Username***), the search value should be entered. For columns that show date and time (e.g. ***Submitted***), the time range can be selected from the calendar.

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

History

The ***Monitoring → Commands → Queries → History*** tab displays commands in the following statuses:

- `Done` — successfully completed commands.
- `Cancelled` — cancelled commands (as a part of the transaction cancellation or termination).
- `Error` — the commands during which some errors occurred.
- `Unknown` — the commands, the final status of which is undefined. It is set on a timeout for queries in the `Queued`, `Running`, `Cancelling` statuses, for which there are no corresponding rows in the `pg_stat_activity` view.



The Monitoring → Commands → Queries → History tab

Most of the fields in the table with the list of commands on the ***History*** tab match the fields that are described above for the ***Online*** tab. Additionally, the following field is available:

- ***Ended*** — a command end timestamp in the `DD/MM/YYYY HH:mm:ss` format. The field is filled in both as a result of successful execution and in case of error or cancellation.

Like on the ***Online*** tab, you have the ability to add additional columns to the table by clicking ***Customize table***.

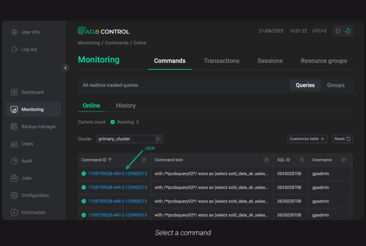
Above the table with a list of commands, there are filters that you can use to select specific data. In addition to the ***Cluster*** filter described above, the ***History*** tab contains the following filters:

- ***Relation*** — filter by the relations that are used in commands. Enter a full value.
- ***Schema*** — filter by the database schema, to which the relations used in commands belong. Enter a full value.

Filters on the Monitoring → Commands → Queries → History tab

Command details

To view the command details, click the command identifier (***Command ID***) in the table on the Monitoring → Commands → Queries → Online or Monitoring → Commands → Queries → History tab.



Select a command

The following page contains multiple sections that are described below.

Header

At the top of the page, the following information is displayed:

- The command unique identifier (see ***Command ID*** above). In the example below — `1695219736-11866-3-209077013`.
- The command status (see ***Status*** above). In the example below — `Running`.
- The command sequence number in ADB Control (starting with `1`). The continuous numbering is used across all clusters. In the example below — `2683`.
- ***Transaction ID*** — the transaction identifier. To view transaction details, click the identifier.
- ***Run time*** — the command duration in hours, minutes, seconds.

1695219736-11866-3--209077013
Running command ID: 2683. Transaction ID: 696094
Run time:
00:34:46

Header of the page with command details

Overview

The ***Overview*** section displays the general command information. In addition to the fields mentioned above, the following information is available:

- ***Run time*** — the command duration in hours, minutes, seconds.
- ***Est. progress*** — the progress of the command execution, which is calculated as a percentage of the time predicted by the query planner. The following formula is used:

$$Est.\,progress = \frac{\sum(NodeProgress(x) \cdot Est.\,Cost(x))}{\sum Est.\,Cost(x)}$$

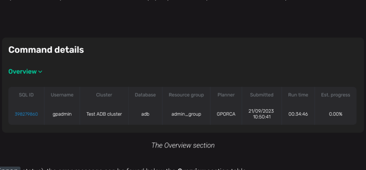
where:

- `Est.Cost(x)` — the estimate of the plan node cost made by the planner.
- `NodeProgress(x)` — the progress for each node of the query plan, which is calculated by the following formula:

$$NodeProgress = \frac{ActualRows}{EstimatedRows}$$

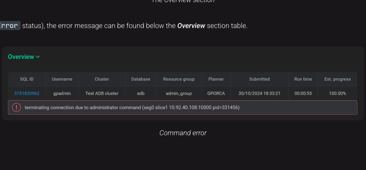
where:

- `ActualRows` — an actual number of processed table rows (tuples).
- `EstimatedRows` — the planner's assumption of the number of rows (tuples) that will be processed for the current plan node.



The Overview section

If the command fails (has the `Error` status), the error message can be found below the ***Overview*** section table.



Command error

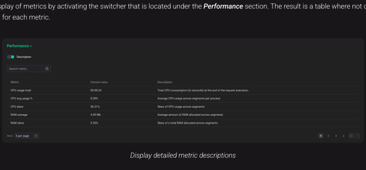
Performance

The ***Performance*** section displays the statistics of system resource consumption by the selected command. The metrics available for monitoring are listed in System metrics collected for commands.

At the top of the ***Performance*** section, the average metric values with skews for all cluster segments are displayed.

Display average metric values for a cluster

You can select an alternative display of metrics by activating the switcher that is located under the ***Performance*** section. The result is a table where not only the current value but also a detailed description is displayed for each metric.



Display detailed metric descriptions

In the bottom part of the ***Performance*** section, the metric values are shown separately for each cluster segment, including the master segment `seg-1` (where ***Segment ID*** equals to `-1`).

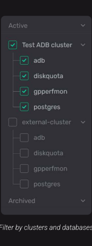
Fields of the table with segment metrics

Field	Description
Segment ID	A content identifier for a segment instance. Corresponds to `gp_segment_configuration.content`. For more information, see the ***System catalog tables → gp_segment_configuration*** section of the Tables overview article

[illegible]

Field	Description
Cluster	A name of the ADB cluster where the group queries are launched
Database	A name of the database where the group queries are launched
Last submitted	The time of the last query in the group
Command text	The text of the first SQL query in the group
Type	The type of the group. Currently, only the `Short` type is implemented

Above the table with a list of groups, the ***Cluster*** filter is located. You can use this filter to select the ADB cluster and its databases for which you want to display data in the table. Initially all databases of the default cluster are selected.



Filter by clusters and databases

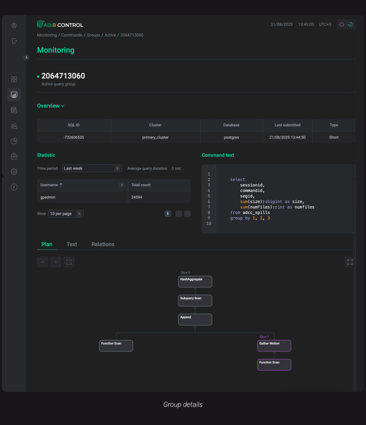
Group details

To view the group details, click the group identifier in the ***Query group ID*** column.

The page that opens contains the following sections:

- ***Overview*** displays the main parameters of the group described in the table above.
- ***Statistic*** shows the average query duration and the number of queries per user in the group. Use the ***Time period*** filter to show data for the last hour, last day, or last week.
- ***Command text*** shows the text of the first SQL query in the group.
- ***Plan***, ***Text***, and ***Relations*** — refer to Command details → Query plan sections for description of these tabs.

> **NOTE**
>
> Unlike the ***Queries*** tab, query groups don't show the progress and the statuses of nodes.



Group details

SQL ID

An SQL ID is an identifier of an SQL command structure.

ADB Control uses SQL ID to identify commands with the same structure and to categorize them into groups. In ADB Control, SQL ID corresponds to the `queryid` value from the `pg_stat_statements` table in ADB.

Semantically equivalent SQL commands share the same SQL ID. In practice, this means the command properties described below will match.

Subqueries are analyzed recursively according to these same principles.

Factors affecting SQL ID

When the following SQL command properties differ, the commands will have separate SQL IDs.

<table>
<tr><th>Property</th><th>Examples of commands with different SQL IDs</th></tr>
<tr><td>Operation type: <code>SELECT</code>, <code>INSERT</code>, <code>UPDATE</code>, and <code>DELETE</code></td><td><pre>INSERT INTO employees (id, name) VALUES (1, 'John Doe');
SELECT id, name FROM employees;</pre></td></tr>
<tr><td>Target relations</td><td><pre>SELECT id, name FROM employees;
SELECT id, name FROM users;</pre><pre>INSERT INTO employees (id, name) VALUES (1, 'John Doe');
INSERT INTO people (id, name) VALUES (1, 'John Doe');</pre></td></tr>
<tr><td>Affected columns including their order</td><td><pre>SELECT id,name FROM users;
SELECT name,id FROM users;</pre></td></tr>
<tr><td>Functions. Different functions result in different SQL IDs regardless of their arguments</td><td><pre>SELECT DISTINCT UPPER (country) FROM users;
SELECT DISTINCT LOWER (country) FROM users;</pre></td></tr>
<tr><td>Common table expressions. The structure and the order of CTEs if there are many. Different names result in different SQL IDs even if the content is identical</td><td><pre>WITH my_filter AS (SELECT * FROM users WHERE active = true)
    SELECT * FROM my_filter;
WITH my_cte AS (SELECT * FROM users WHERE active = true)
    SELECT * FROM my_cte;</pre></td></tr>
<tr><td>The <code>WHERE</code> clause. Separate SQL IDs are assigned when the columns and operators differ. The constants are ignored</td><td><pre>SELECT * FROM users WHERE id > 5;
SELECT * FROM users WHERE id < 5;
SELECT * FROM users WHERE id BETWEEN 5 AND 10;
SELECT * FROM users WHERE age > 5;</pre></td></tr>
<tr><td><code>ORDER BY</code>, <code>GROUP BY</code>, <code>DISTINCT</code>, <code>DISTINCT ON</code>. Separate SQL IDs are assigned when the columns and comparison and sorting operators differ. The constants are ignored</td><td><pre>SELECT * FROM users ORDER BY age;
SELECT * FROM users ORDER BY name;
SELECT * FROM users ORDER BY name DESC;</pre></td></tr>
<tr><td><code>JOIN</code> — the presence and the type of <code>JOIN</code> and the structure of the <code>JOIN</code> queries</td><td><pre>SELECT * FROM orders o LEFT JOIN users u ON o.user_id = u.id;
SELECT * FROM orders o INNER JOIN users u ON o.user_id = u.id;
SELECT * FROM orders NATURAL JOIN users;</pre></td></tr>
<tr><td><code>HAVING</code> — operators, functions</td><td><pre>SELECT user_id, COUNT(*) as order_count
FROM orders
GROUP BY user_id
HAVING COUNT(*) > 2;

SELECT user_id, SUM(amount) as total
FROM orders
GROUP BY user_id
HAVING SUM(amount) > 200;</pre></td></tr>
<tr><td>Window functions — columns, functions, operators</td><td><pre>SELECT id, user_id, amount,
       ROW_NUMBER() OVER (PARTITION BY user_id ORDER BY amount ASC) as rn_asc
FROM orders;
SELECT id, user_id, amount,
       ROW_NUMBER() OVER (PARTITION BY user_id ORDER BY amount DESC) as
rn_desc
FROM orders;</pre></td></tr>
<tr><td><code>UNION</code>, <code>INTERSECT</code>, and <code>EXCEPT</code> and the presence of <code>ALL</code> lead to separate SQL IDs, but <code>DISTINCT</code> is ignored</td><td><pre>SELECT user_id FROM orders WHERE amount > 50
UNION
SELECT user_id FROM orders WHERE status = 'completed';

SELECT user_id FROM orders WHERE amount > 50
INTERSECT
SELECT user_id FROM orders WHERE status = 'completed';

SELECT user_id FROM orders WHERE amount > 50
EXCEPT
SELECT user_id FROM orders WHERE status = 'completed';</pre></td></tr>
</table>

Factors not affecting SQL ID

If only the following properties differ in SQL command text, the commands will have the same SQL ID.

<table>
<tr><th>Property</th><th>Examples of commands having the same SQL ID</th></tr>
<tr><td>Aliases</td><td><pre>SELECT id AS user_id, name AS full_name FROM users;
SELECT id AS client_id, name AS person_name FROM users;</pre></td></tr>
<tr><td>Constants (literal values) in operators and functions</td><td><pre>SELECT * FROM users WHERE id > 5;
SELECT * FROM users WHERE id > 10;</pre><pre>SELECT * FROM users WHERE active = true;
SELECT * FROM users WHERE active = false;</pre><pre>SELECT * FROM users LIMIT 5;
SELECT * FROM users LIMIT 10;</pre></td></tr>
<tr><td>Line breaks, spaces, and tabs</td><td><pre>SELECT id, name FROM users WHERE active = true;
SELECT id, name
FROM users
WHERE active = true;</pre></td></tr>
<tr><td><code>FOR [KEY] UPDATE/SHARE</code> locks</td><td><pre>SELECT * FROM orders WHERE amount > 100 FOR UPDATE;
SELECT * FROM orders WHERE amount > 100 FOR SHARE;
SELECT * FROM orders WHERE amount > 100 FOR KEY SHARE;</pre></td></tr>
<tr><td>All utility commands (those other than <code>SELECT</code>, <code>INSERT</code>, <code>UPDATE</code>, and <code>DELETE</code>) have the SQL ID of <code>0</code>.
If a utility command has a subquery that is not a utility command, that subquery will have a separate SQL ID</td><td><pre>CREATE TABLE employees () INHERITS (users);
ALTER TABLE employees DROP COLUMN name;
VACUUM employees;
DROP TABLE employees;</pre></td></tr>
</table>

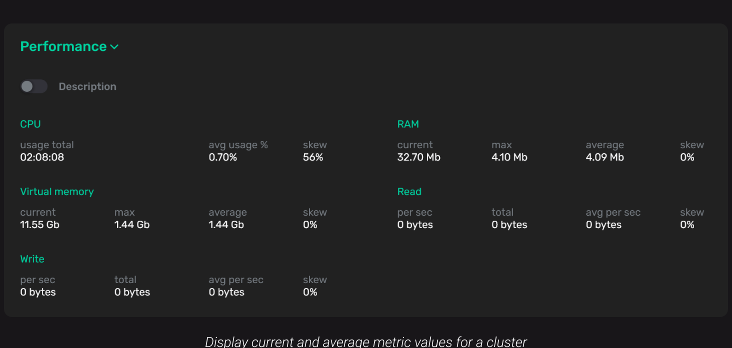
Monitoring of transactions

Contents

Performance

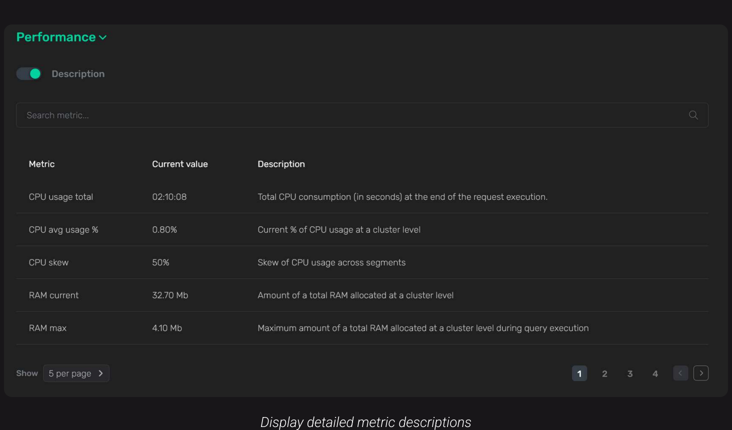
The **Performance** section displays the statistics of system resource consumption by the selected transaction. The metrics available for monitoring are listed in System metrics collected for transactions.

At the top of the **Performance** section, the current and average metric values with skews for all cluster segments are displayed.



Display current and average metric values for a cluster

You can select an alternative display of metrics by activating the switcher that is located under the **Performance** section. The result is a table where not only the current value but also a detailed description is displayed for each metric.

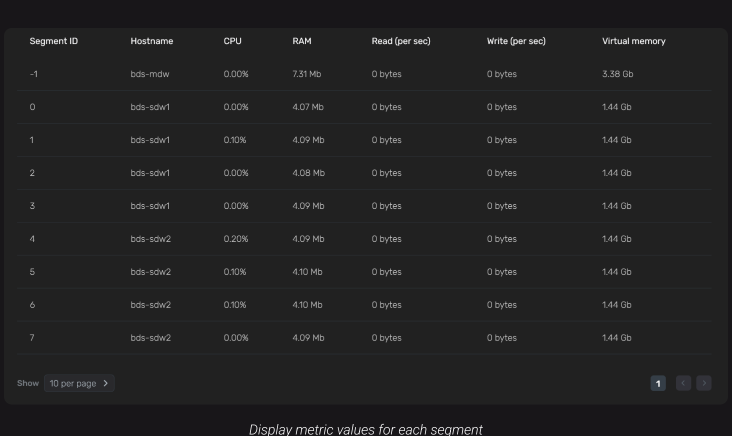


Display detailed metric descriptions

In the bottom part of the **Performance** section, the metric values are shown separately for each cluster segment, including the master segment `seg-1` (where **Segment ID** equals to `-1`).

Fields of the table with segment metrics

Field	Description
Segment ID	A content identifier for a segment instance. Corresponds to `gp_segment_configuration.content`. For more information, see the **System catalog tables → gp_segment_configuration** section of the Tables overview article
Hostname	A host name
CPU	The amount of CPU consumed by the segment (percentage). Based on the metric values for all segments, the average value CPU avg usage % is calculated
RAM	The amount of RAM consumed by the segment (in bytes). Based on the metric values for all segments, the average value RAM average is calculated
Read (per sec)	The amount of data read per second by the segment (in bytes). Based on the metric values for all segments, the average value Read avg per sec is calculated
Write (per sec)	The amount of data written per second by the segment (in bytes). Based on the metric values for all segments, the average value Write avg per sec is calculated
Virtual memory	The amount of virtual memory allocated by the segment (in bytes). Based on the metric values for all segments, the average value Virtual memory average is calculated



Segment ID	Hostname	CPU	RAM	Read (per sec)	Write (per sec)	Virtual memory
-1	bds-mdw	0.00%	7.31 Mb	0 bytes	0 bytes	3.38 Gb
0	bds-sdw1	0.00%	4.07 Mb	0 bytes	0 bytes	1.44 Gb
1	bds-sdw1	0.10%	4.09 Mb	0 bytes	0 bytes	1.44 Gb
2	bds-sdw1	0.00%	4.08 Mb	0 bytes	0 bytes	1.44 Gb
3	bds-sdw1	0.00%	4.09 Mb	0 bytes	0 bytes	1.44 Gb
4	bds-sdw2	0.20%	4.09 Mb	0 bytes	0 bytes	1.44 Gb
5	bds-sdw2	0.10%	4.10 Mb	0 bytes	0 bytes	1.44 Gb
6	bds-sdw2	0.10%	4.10 Mb	0 bytes	0 bytes	1.44 Gb
7	bds-sdw2	0.00%	4.09 Mb	0 bytes	0 bytes	1.44 Gb

Show 10 per page

Display metric values for each segment

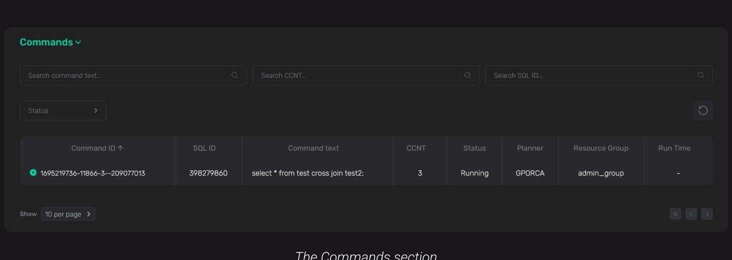
> **IMPORTANT**
>
> System metrics are collected from ADB segments by ADB Control agents only for transactions longer than 15 seconds. Due to this fact, the section with metrics will be empty for the transactions that complete faster.

Commands

The **Commands** section lists the commands that are performed within the current transaction.

The "Commands" tab fields

Field	Description
Command ID	A unique command identifier, which includes: ▪ cluster start timestamp; ▪ session number; ▪ command number within the current session; ▪ hash code based on the cluster name (can be negative). To open the page with command details, click the command identifier
SQL ID	A common identifier for SQL commands with the same structure
Command text	The first symbols of a command text. To view the full text (in case of long queries), hover the mouse over the field value
CCNT	A command number within the current session
Status	A command status. Possible values: ▪ `Queued` — the commands on which information about the actual start of execution has not yet received. ▪ `Running` — the commands that are currently running. ▪ `Cancelling` — the commands that are currently cancelling (as a part of the transaction cancellation or termination). ▪ `Done` — successfully completed commands. ▪ `Cancelled` — cancelled commands (as a part of the transaction cancellation or termination). ▪ `Error` — the commands during which some errors occurred. ▪ `Unknown` — the commands, the final status of which is undefined. It is set on a timeout for queries in the `Queued`, `Running`, `Cancelling` statuses, for which there are no corresponding rows in the `pg_stat_activity` view.
Planner	A name of the query optimizer that is used to produce the query execution plan. Possible values: ▪ `GPORCA` — GPORCA is used (`optimizer = on`). ▪ `Fallbacked` — after the attempt to produce a plan via GPORCA (`optimizer = on`) failed, Postgres query optimizer is used. ▪ `Legacy` — Postgres query optimizer is used (`optimizer = off`).
Resource group	A name of the resource group that is used for the command
Run time	A command duration in hours, minutes, seconds



The Commands section

Above the table with commands, there are filters that you can use to select specific data:

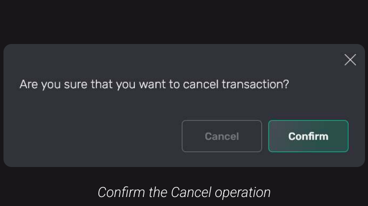
- **Search command text...** — filter by the command text (see **Command text** above). You can search by any substring of the SQL command.
- **Search CCNT...** — filter by the command number within a session (see **CCNT** above). Enter a full value.
- **Search SQL ID...** — filter by the identifier that is common for SQL commands with the same structure (see **SQL ID** above). Enter a full value.
- **Status** — filter by the command status (see **Status** above). Select a value from the drop-down list.

Cancel or terminate the transaction

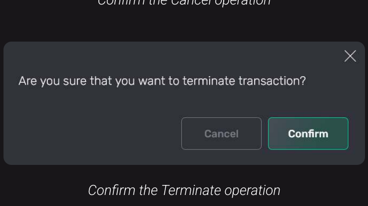
On the page with transaction details, you can cancel a transaction by clicking one of the following buttons:

- **Cancel** — calls the `pg_cancel_backend` function, which cancels a transaction in the process.
- **Terminate** — calls the `pg_terminate_backend` function, which terminates the process within which the transaction is running.

After clicking the button, confirm the operation.



Confirm the Cancel operation



Confirm the Terminate operation

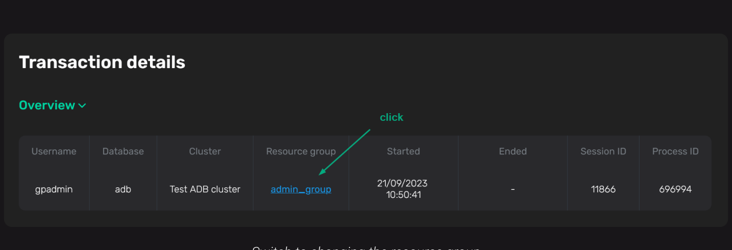
After a transaction is cancelled or terminated, it acquires the `Aborted` status, whereas all transaction commands get the `Cancelled` status.

> **IMPORTANT**
>
> - Cancellation and termination of transactions is available for ADB Control users with appropriate permissions (see `Kill self queries`, `Kill some queries`, `Kill all queries` in the Authorization article).
> - You can cancel and terminate only active transactions. For transactions that are opened on the History tab, **Cancel** and **Terminate** buttons are disabled.
> - Cancellation and termination of transactions are recorded as a part of the operation audit under the operation names `Cancel` and `Terminate`, respectively (for the `Transaction` object type). You can view audit events on the Audit → Operations tab.

Change a resource group

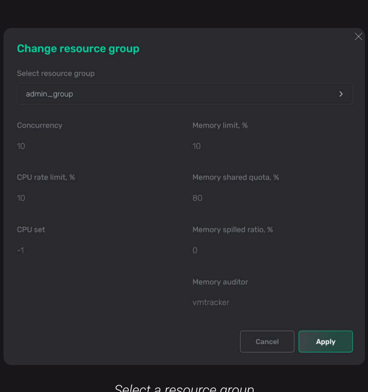
To change a resource group for the selected transaction, follow the steps:

1. Open the transaction details as described above.
2. Click the resource group name in the **Overview** section.



Switch to changing the resource group

3. In the window that opens, select a resource group from the **Select resource group** drop-down list. After the group is selected, its configuration parameters are displayed in the main window part.
4. Click **Apply**.



Select a resource group

> **NOTE**
>
> The resource group change is recorded as a part of the operation audit under the operation name `Resource group change` (for the `Transaction` object type). You can view audit events on the Audit → Operations tab.

System metrics collected for transactions

Both Online and History tabs allow you to view the statistics of the system resource consumption by transactions (on the Performance tab in the transaction details). The metrics available for transactions are listed below.

> **NOTE**
>
> The **Online** tab displays current metric values, whereas the **History** tab shows metric values at the end of the transaction execution.

Transaction metrics

Group	Metric	Description
CPU	CPU usage total	The total CPU consumption across cluster segments (in seconds)
	CPU avg usage %	The CPU percent average for all segment processes executing the query. Calculated as the result of dividing the total CPU consumption on all cluster segments (excluding the master and standby) by the total number of processes processing the query on segment hosts (excluding the master and standby)
	CPU skew	The indicator that displays the CPU consumption skew across cluster segments (percentage). A value other than zero indicates that one of the segments uses more CPU than others
RAM	RAM current	The current amount of RAM consumed at the cluster level (in bytes)
	RAM max	The maximum amount of RAM consumed at the cluster level during the transaction execution (in bytes)
	RAM average	The average amount of RAM consumed across cluster segments (in bytes). The RAM consumption at any given moment is calculated based on the operating system metric **rss (Resident Set Size)**
	RAM skew	The indicator that displays the RAM consumption skew across cluster segments (percentage). A value other than zero indicates that one of the segments uses more RAM than others
Virtual memory	Virtual memory current	The current amount of virtual memory allocated at the cluster level (in bytes)
	Virtual memory max	The maximum amount of virtual memory allocated at the cluster level during the transaction execution (in bytes)
	Virtual memory average	The average amount of virtual memory allocated across cluster segments (in bytes). The virtual memory consumption at any given moment is calculated based on the operating system metric **vsize (Virtual Memory Size)**
	Virtual memory skew	The indicator that displays the virtual memory skew across cluster segments (percentage). A value other than zero indicates that one of the segments uses more virtual memory than others
Read	Read per sec	The current amount of data read per second at the cluster level (in bytes)
	Read total	The total amount of data read across cluster segments (in bytes). The following formula is used for calculation: $ReadTotal = TotalReadSpeed \cdot QueryTime$
	Read avg per sec	The average amount of data read across cluster segments per second (in bytes). The data read rate at any given moment is calculated based on the I/O Read information from */proc/[pid]/stat* as a delta between the current and previous values
	Read skew	The indicator that displays the data reading skew across cluster segments (percentage). A value other than zero indicates that one of the segments reads more data from a disk than others
Write	Write per sec	The current amount of data written per second at the cluster level (in bytes)
	Write total	The total amount of data written across cluster segments (in bytes). The following formula is used for calculation: $WriteTotal = TotalWriteSpeed \cdot QueryTime$
	Write avg per sec	The average amount of data written across cluster segments per second (in bytes). The data write rate at any given moment is calculated based on the I/O Write information from */proc/[pid]/stat* as a delta between the current and previous values
	Write skew	The indicator that displays the data writing skew across cluster segments (percentage). A value other than zero indicates that one of the segments writes more data to a disk than others

Monitoring of sessions

Daria Barysheva

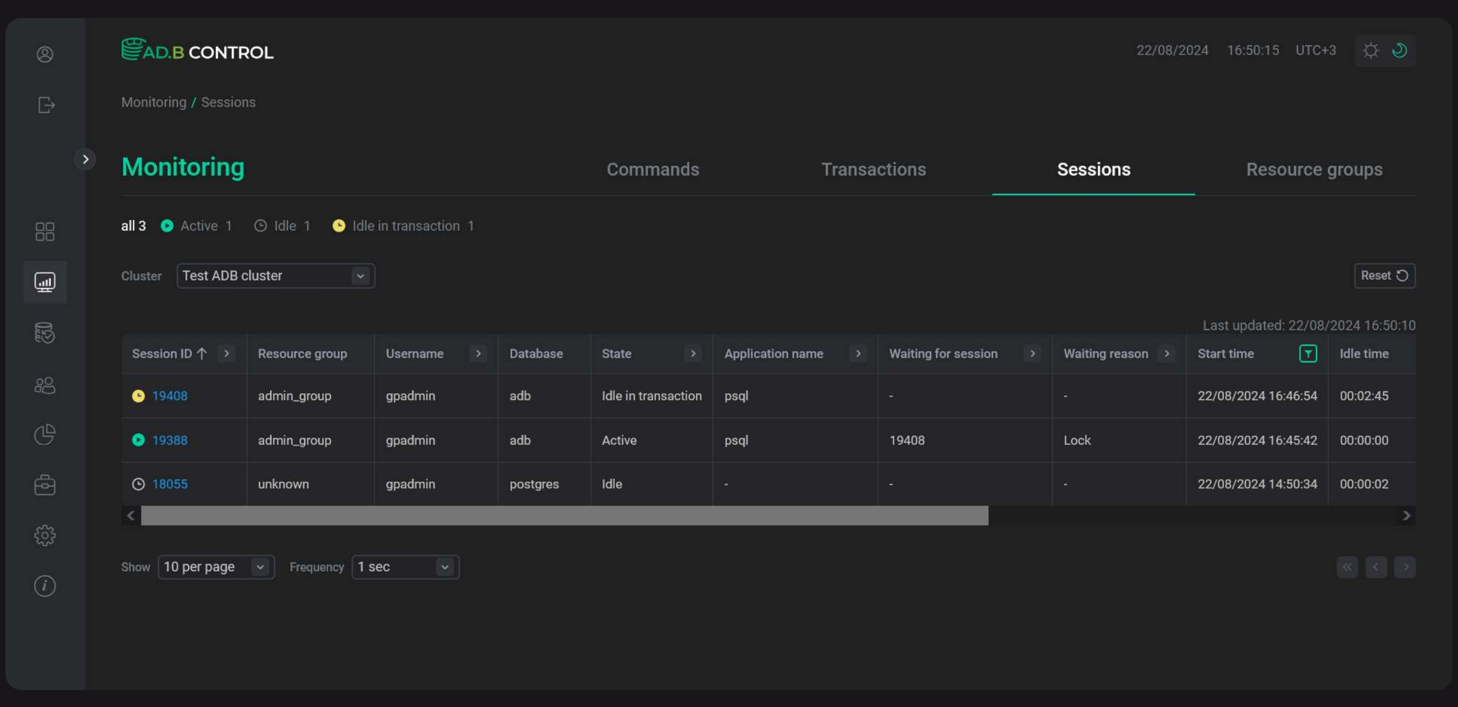
Contents

To Table of Contents

The ***Monitoring → Sessions*** tab is designed to monitor current ADB sessions, both active and idle. This page also allows you to terminate specified sessions in order to release the connection pool for further connections.

View a list of sessions

On the ***Monitoring → Sessions*** tab, there is a table with the following information on sessions.



List of sessions

The "Monitoring → Sessions" tab fields

Field	Description
Session ID	A session identifier
Resource group	A resource group name
Username	A name of the user who connected to the backend
Database	A database name
State	A session state. Possible values: • `Active` — the backend is running a query. • `Idle` — the backend is waiting for a new client command. • `Idle in transaction` — the backend is in a transaction, but there are no currently running queries.
Application name	A name of the application that started a session
Waiting for session	An identifier of the session that blocks the current session
Waiting reason	A type of session waiting. Possible values: • `Lock` — caused by waiting for the ability to obtain the lock. • `Resgroup` — caused by exceeding the resource group limits (see the ***Resource group*** column). To see all queueing commands, head to the ***Commands → Queueing*** tab of the resource group page. • `Replication` — caused by replication.
Start time	The time when the backend process was started
Idle time	An idle time (in seconds)
Command text	A text of the most recent SQL query that was run at the backend (within the current session). If `state` is `active`, the field shows the currently running query
Tags	The session tags

NOTE

To view the details of the most recent command within the current session, click the session identifier (***Session ID***). Note that you can view the command details if this command is registered in ADB Control. For more information on necessary requirements, see Monitoring of commands.

At the top of the tab, the ***Active***, ***Idle***, and ***Idle in transaction*** fields are located. They display a total count of sessions in the specified statuses.

Above the table with a list of sessions, the ***Cluster*** filter is located. You can use this filter to select the ADB cluster for which you want to display data in the table. The default cluster for all such filters is defined in the ADB Control settings.

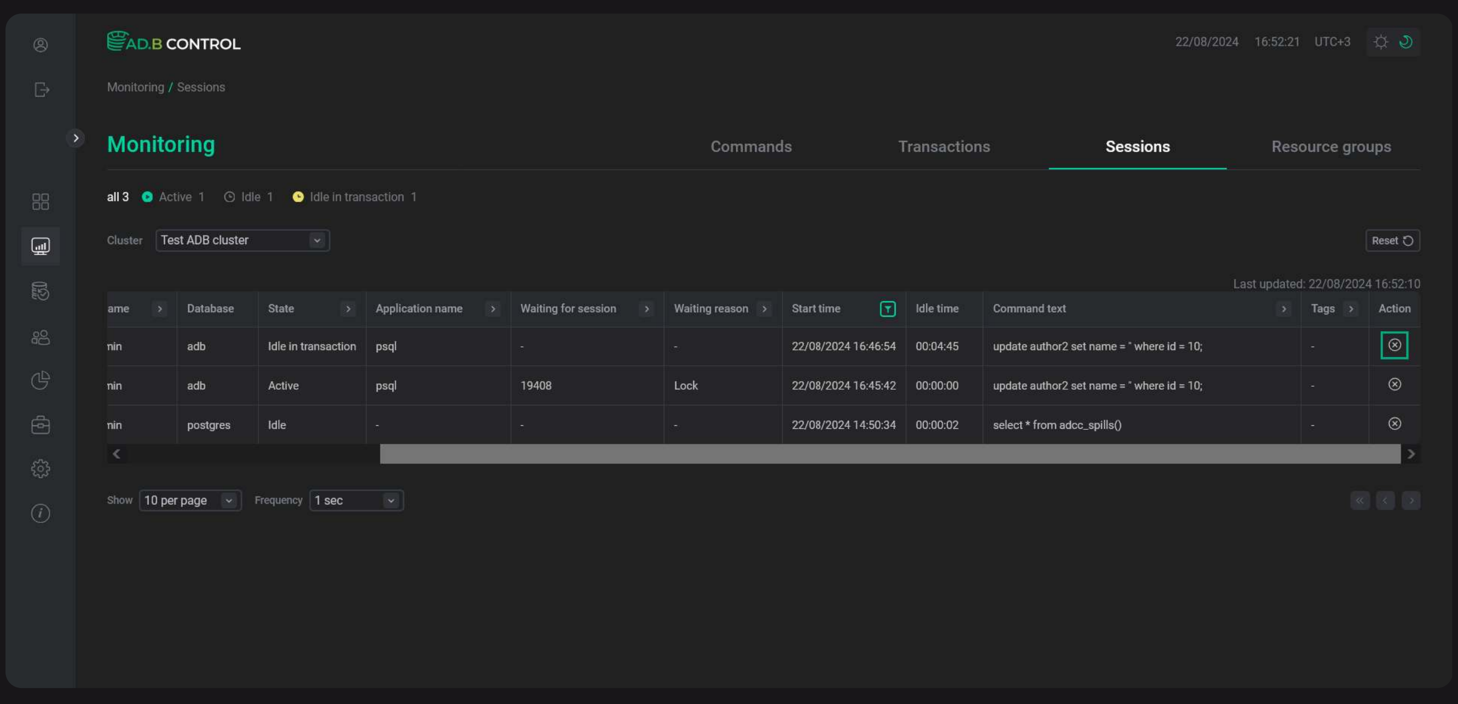
In the column headers of the table with a list of sessions, there are filters that you can use to select specific data. To open a filter, click the icon. For those columns where the set of possible values is limited (e.g. ***State***), you can select a value from the drop-down list. For some columns (e.g. ***Session ID***), the search value should be entered. For columns that show date and time (e.g. ***Start time***), the time range can be selected from the calendar.

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

Terminate a session

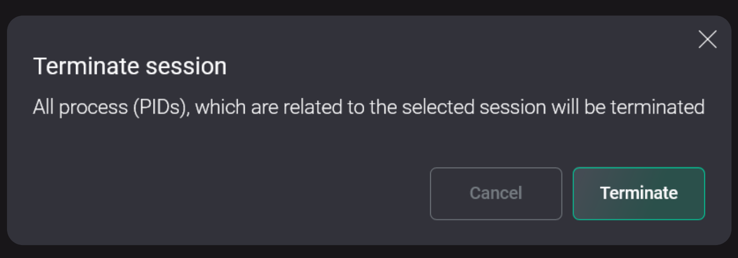
To terminate a session, follow the steps:

1. Click the icon in the ***Actions*** column on the ***Monitoring → Sessions*** tab.



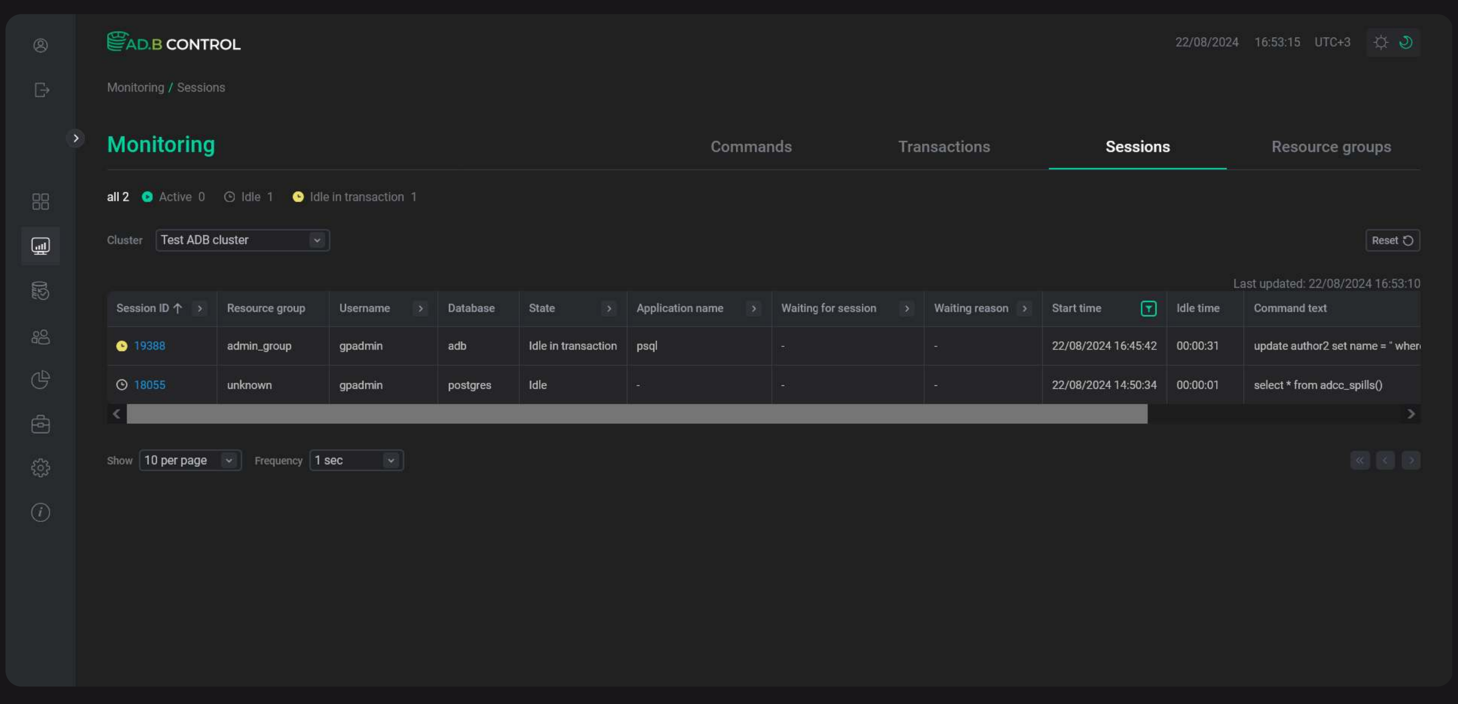
Switch to terminating a session

2. In the window that opens, confirm the operation by clicking ***Terminate***.



Confirm the operation

As a result, the session is terminated and no longer displayed on the ***Monitoring → Sessions*** tab.



Session data is removed

Work with resource groups

Daria Barysheva

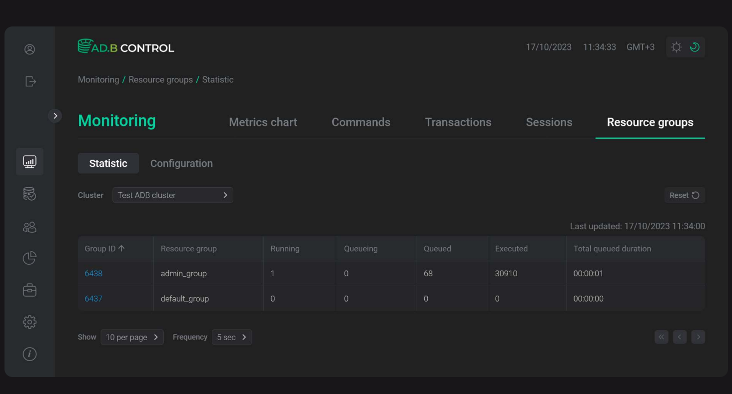
Contents

To Table of Contents

The ***Monitoring*** → ***Resource groups*** page in the ADB Control web interface displays the resource groups that are used in ADB clusters connected to the monitoring system. This page includes two tabs ***Statistic*** and ***Configuration***, each of which is described in detail below.

Statistic

The ***Monitoring*** → ***Resource groups*** → ***Statistic*** tab displays the statistics of the resource group usage in the selected ADB cluster. The ***Last updated*** field contains date and time of the last data update on the page in the `DD/MM/YYYY HH:mm:ss` format.



The Monitoring → Resource groups → Statistic tab

On the tab, there is a table with the following information on resource groups.

The "Monitoring → Resource groups → Statistic" tab fields

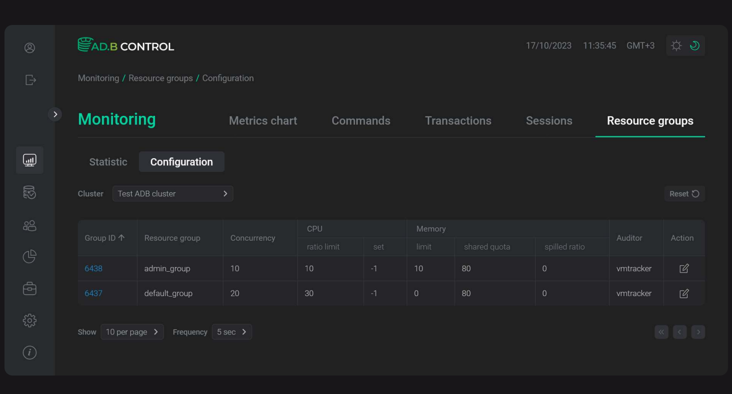
Field	Description
Group ID	A unique resource group identifier
Resource group	A resource group name
Running	A number of transactions currently running within the current resource group
Queueing	A number of transactions currently queued within the current resource group
Queued	A total number of transactions queued within the current resource group over the entire observation period
Executed	A total number of transactions executed within the current resource group over the entire observation period
Total queued duration	The total time that all transactions were queued within the current resource group over the entire observation period (in hours, minutes, seconds)

Above the table with a list of resource groups, the ***Cluster*** filter is located. You can use this filter to select the ADB cluster for which you want to display data in the table. The default cluster for all such filters is defined in the ADB Control settings.

Configuration

List of resource groups

The ***Monitoring*** → ***Resource groups*** → ***Configuration*** tab displays the current configurations of existing resource groups.



The Monitoring → Resource groups → Configuration tab

On the tab, there is a table with the following information on resource groups.

The "Monitoring → Resource groups → Configuration" tab fields

Field	Description
Group ID	A unique resource group identifier
Resource group	A resource group name
Concurrency	The maximum number of concurrent transactions that are permitted within the resource group. If the limit is reached, new commands will be queued
CPU ratio limit	The percentage of the reserved CPU resources of the entire cluster (specified in the `gp_resource_group_cpu_limit` parameter) that is available to the resource group. The sum of this parameter values for all resource groups should be <= `100`. The parameter is not compatible with ***CPU set*** — fill only one of two parameters
CPU set	The CPU cores that are reserved for the resource group on the master and segment hosts. If the parameter is set, all SQL commands within the resource group will only use the specified cores. If the resource group does not use cores, they will be idle. The parameter is not compatible with ***CPU ratio limit*** — fill only one of two parameters. The `-1` value means that ***CPU ratio limit*** is used. Example: `1,3-4`
Memory limit	The percentage of the reserved RAM resources of the entire cluster (specified in the `gp_resource_group_memory_limit` parameter) that is available to the resource group. The sum of this parameter values for all resource groups should be <= `100`
Memory shared quota	The percentage of the reserved RAM resources of the resource group to be shared between all transactions running within that group. The amount of memory equal to `(100-<Memory shared quota>)/<Concurrency>` will be guaranteed to be available for each transaction in the group
Memory spilled ratio	The memory usage threshold for memory-intensive transactions. When a transaction reaches the specified threshold, spill files are created
Auditor	The memory auditor that is used for the resource group. The default value is `vmtracker`

Above the table with a list of resource groups, the ***Cluster*** filter is located. You can use this filter to select the ADB cluster for which you want to display data in the table. The default cluster for all such filters is defined in the ADB Control settings.

> **NOTE**
>
> For more information on resource group configuration parameters, see Greenplum documentation.

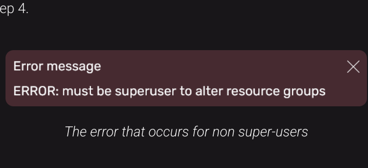
Edit a resource group configuration

To edit the resource group configuration, follow the steps:

1. In the ADB cluster for which you plan to modify the resource group settings, assign the super-user rights to the role that is used in the JDBC connection to that cluster. The user is defined on the Configuration → Clusters page (by default `adcc`). To set super-user rights, run the following query:

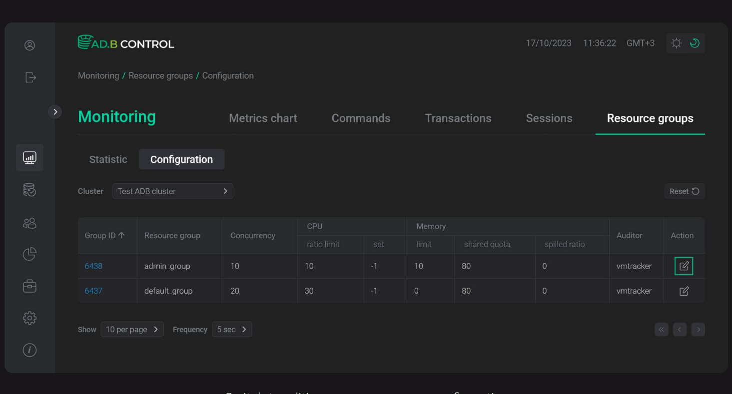
```
ALTER ROLE adcc SUPERUSER;
```

   Skipping the first step will cause the following error in step 4.



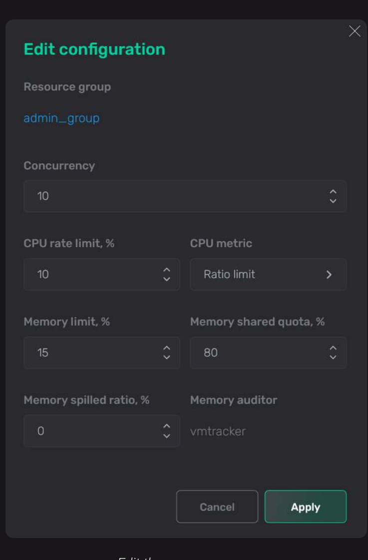
The error that occurs for non super-users

2. Click the icon in the ***Action*** column on the ***Monitoring*** → ***Resource groups*** → ***Configuration*** tab.



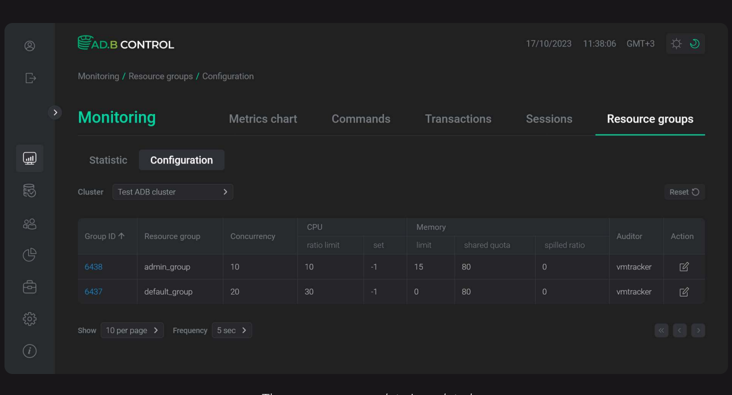
Switch to editing a resource group configuration

3. In the window that opens, edit necessary data. Descriptions of all fields are available above. A controller name in the ***Memory auditor*** field value cannot be changed.



Edit the resource group

4. Click ***Apply***. As a result, the resource group data is updated on the ***Monitoring*** → ***Resource groups*** → ***Configuration*** tab.

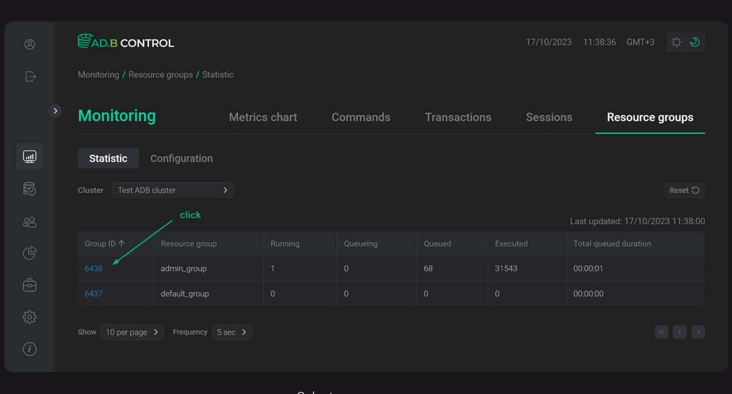


The resource group data is updated

> **IMPORTANT**
>
> - The resource group update is available only to ADB Control users with appropriate permissions (see `Change resource group configuration` in the Authorization article).
> - The resource group update is recorded as a part of the operation audit under the operation name `Update` (for the `Resource group` object type). You can view audit events on the Audit → Operations tab.

Resource group details

To view the resource group details, click the group identifier (***Group ID***) in the table on the Monitoring → Resource groups → Statistic or Monitoring → Resource groups → Configuration tab.



Select a resource group

The following page contains multiple sections that are described below.

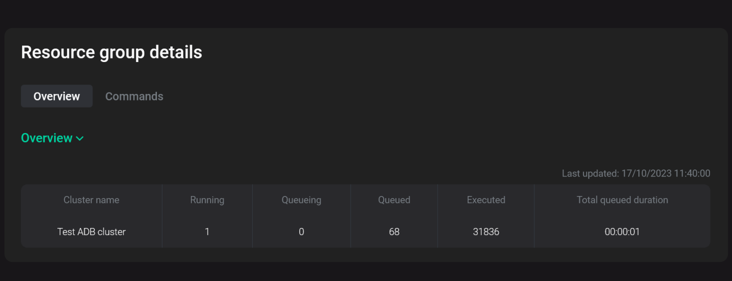
Header

At the top of the page, the resource group identifier and name are displayed.

Header of the page with resource group details

Overview

The ***Overview*** section displays the general resource group information. All fields are described in detail above.



The Overview section

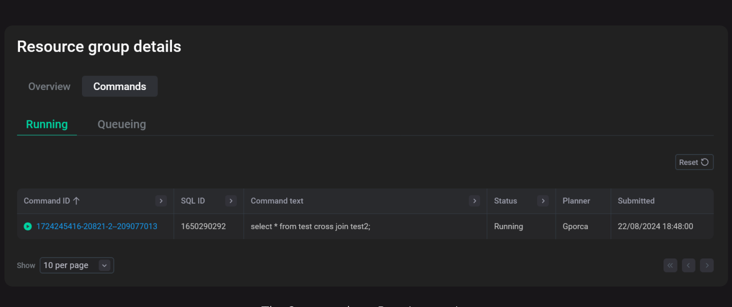
Commands → Running

The ***Commands*** → ***Running*** section lists the commands that were started within the current resource group and currently have the following statuses:

- `Queued` — the commands on which information about the actual start of execution has not yet received.
- `Running` — the commands that are currently running.
- `Cancelling` — the commands that are currently cancelling (as a part of the transaction cancellation or termination).

The "Commands → Running" tab fields

Field	Description
Command ID	A unique command identifier, which includes: - cluster start timestamp; - session number; - command number within the current session; - hash code based on the cluster name (can be negative). To open the page with command details, click the command identifier
SQL ID	[illegible]
Command text	[illegible]
Status	[illegible]
Planner	[illegible]
Submitted	[illegible]



The Commands → Running section

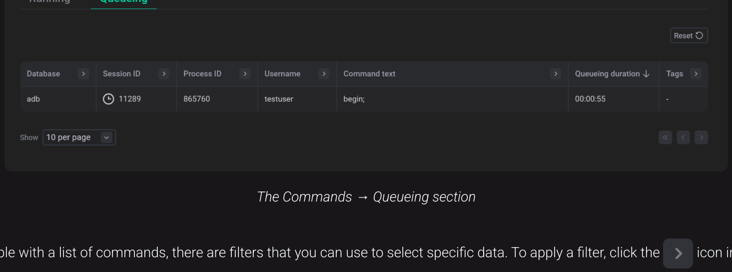
[illegible]

Commands → Queueing

[illegible]

The "Commands → Queueing" tab fields

Field	Description
Database	[illegible]
Session ID	[illegible]
Process ID	[illegible]
Username	[illegible]
Command text	[illegible]
Queueing duration	[illegible]
Tags	[illegible]



The Commands → Queueing section

[illegible]

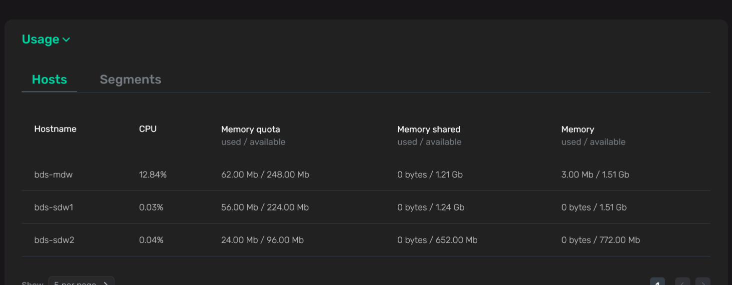
Usage

[illegible]

- Hosts
- Segments

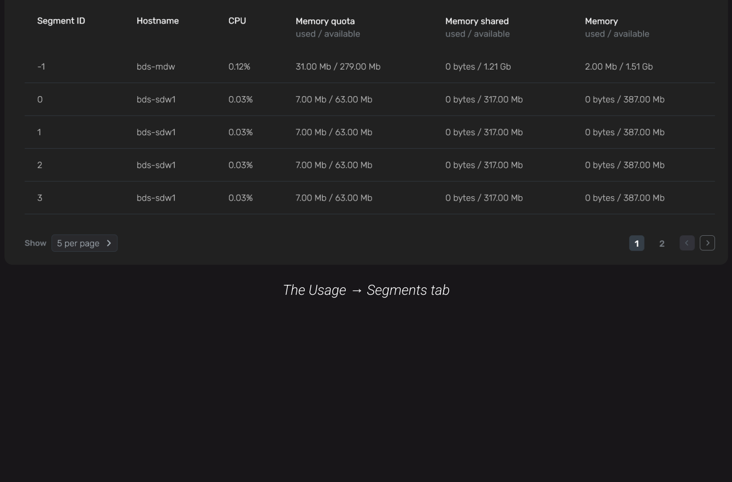
The "Usage → Hosts" tab fields

Field	Description
Hostname	[illegible]
CPU	[illegible]
Memory quota	[illegible]
Memory shared	[illegible]
Memory	[illegible]



The Usage → Hosts tab

[illegible]



The Usage → Segments tab

Work with query tags

Daria Barysheva

To Table of Contents

Starting with 4.7.5, ADB Control supports **query tagging** — assigning custom metadata to SQL queries in the form of `<key>: <values>` pairs. You can set tags for a session or transaction in ADB and then view and filter these tags on the *Monitoring → Commands*, *Monitoring → Transactions*, and *Monitoring → Sessions* pages in ADB Control. Tags are assigned to ADB queries via the `query_band` GUC. This functionality can be useful for monitoring query groups, for example, as part of the common ETL process.

Below is an example of working with query tags:

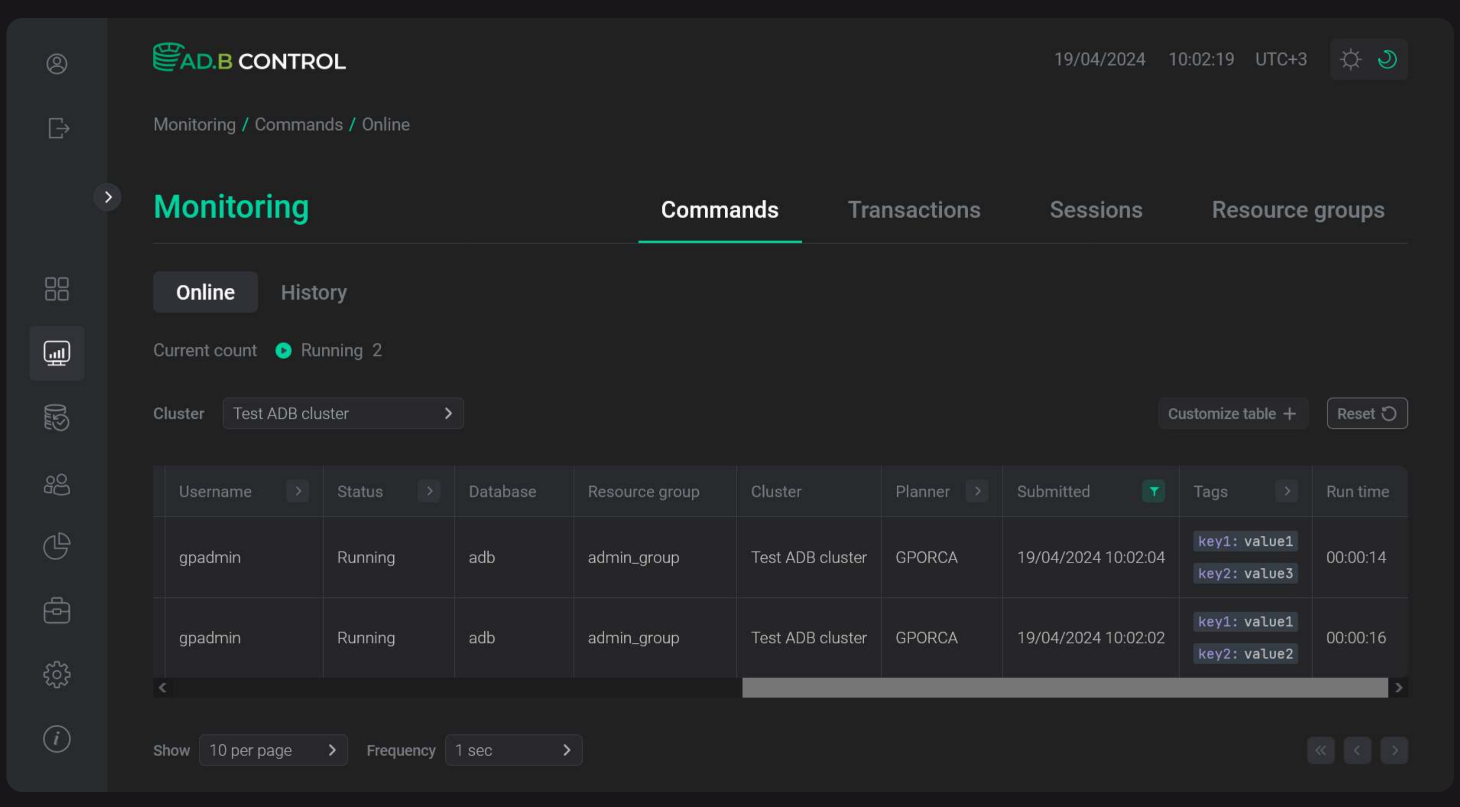
1. Open the first session in ADB (e.g. by connecting to the `adb` database on the master host via `psql`).
2. Set values for two tags at the session level:

```
SET query_band = 'key1: value1,key2: value2';
```

3. Run any SQL query that lasts long enough to appear in the ADB Control web interface.
4. Within the second ADB session, set a different value for the `key2` tag:

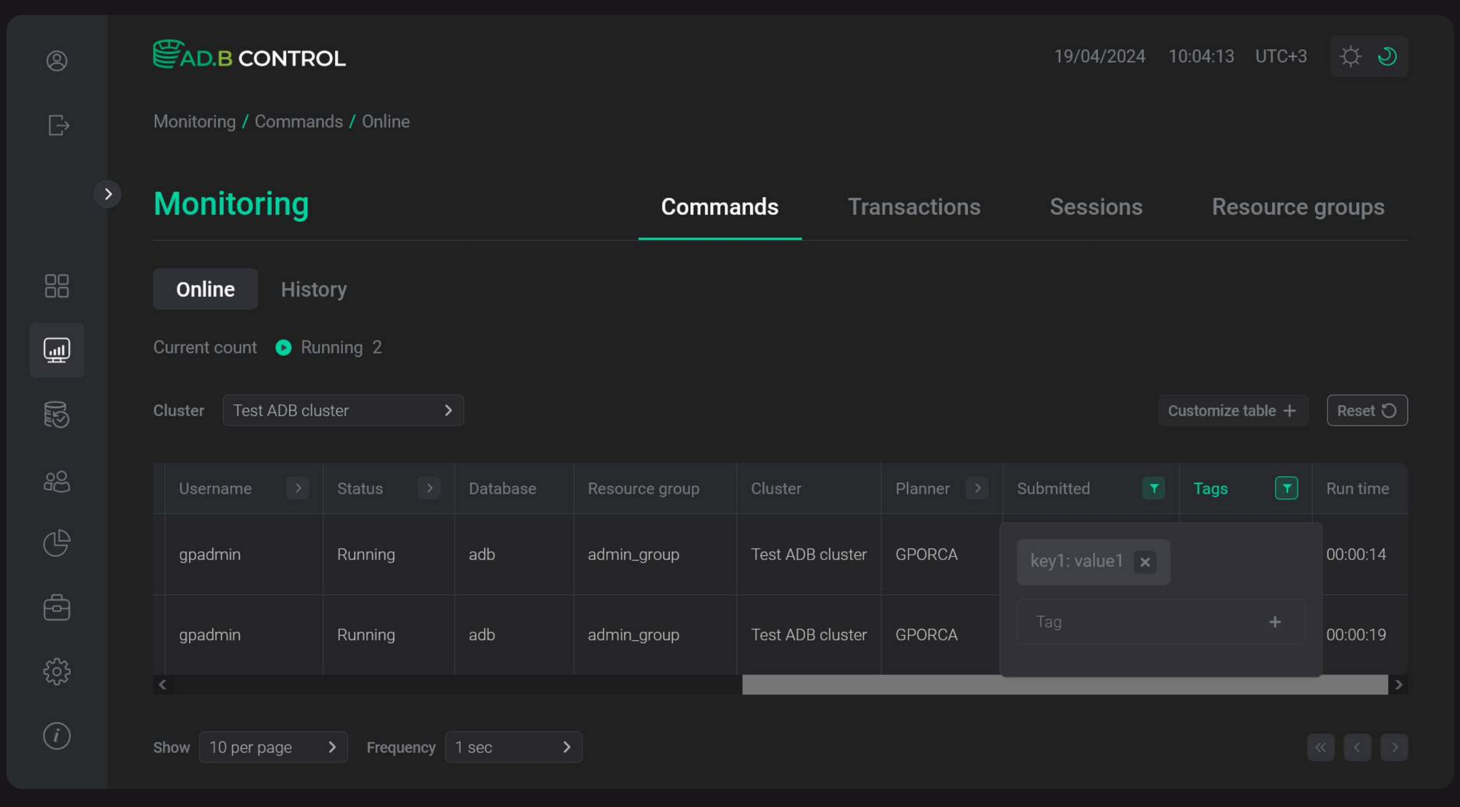
```
SET query_band = 'key1: value1,key2: value3';
```

5. Run another SQL query within the second session.
6. Go to the *Monitoring → Commands → Queries → Online* page in the ADB Control web UI. Tag values for both running queries are shown in the *Tags* column. The same column is available on the *Monitoring → Commands → Queries → History*, *Monitoring → Transactions* (*Online* and *History*), *Monitoring → Sessions*, and *Monitoring → Resource groups → Statistics → <resource group> → Commands → Queueing* pages.



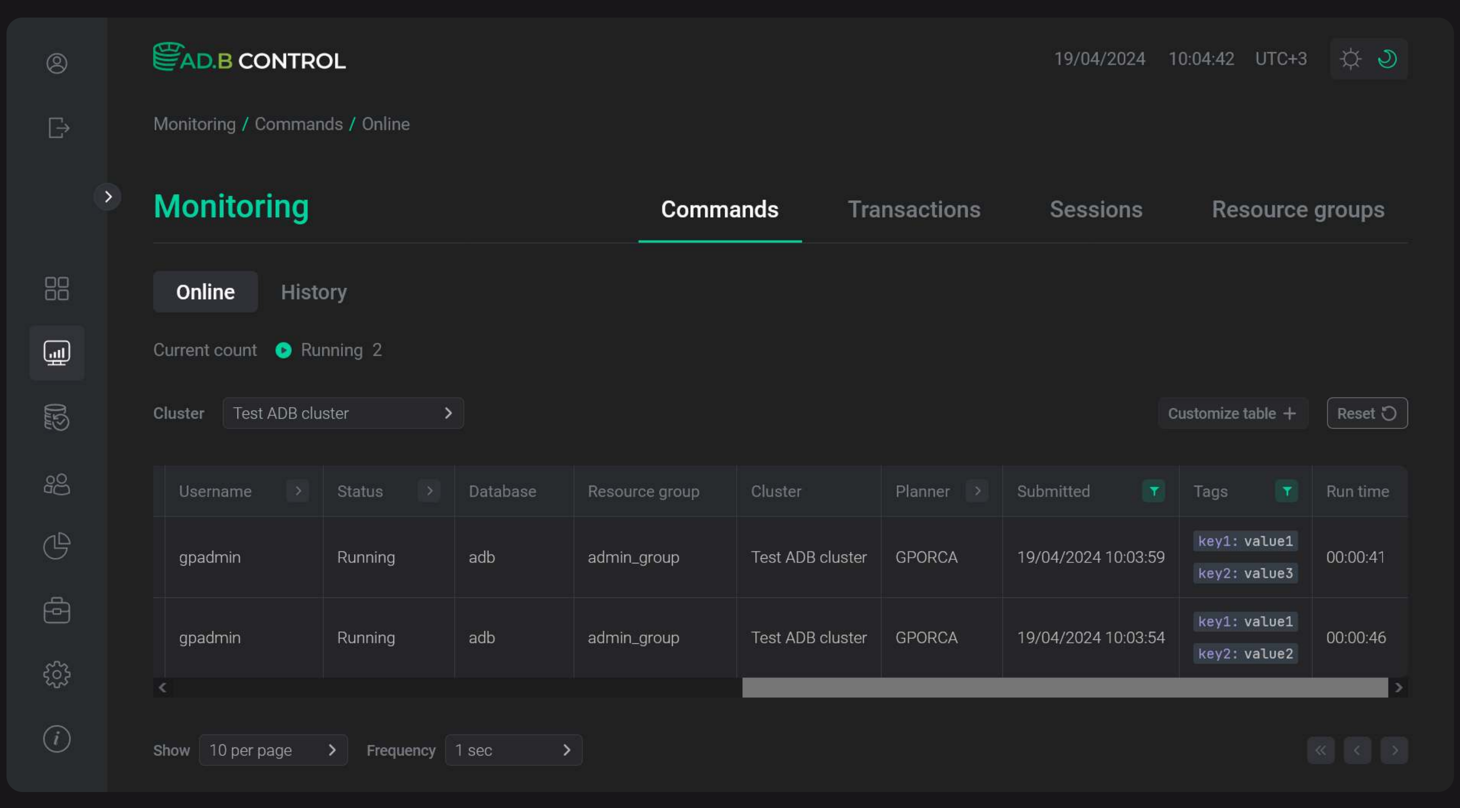
Tags in ADB Control

7. To select the queries with the specific tag value, open a drop-down list in the *Tags* column header, set the tag value in the *Tag* column as `<key>: <value>`, and click the + icon. To remove any tag from the filter, click ✕ .



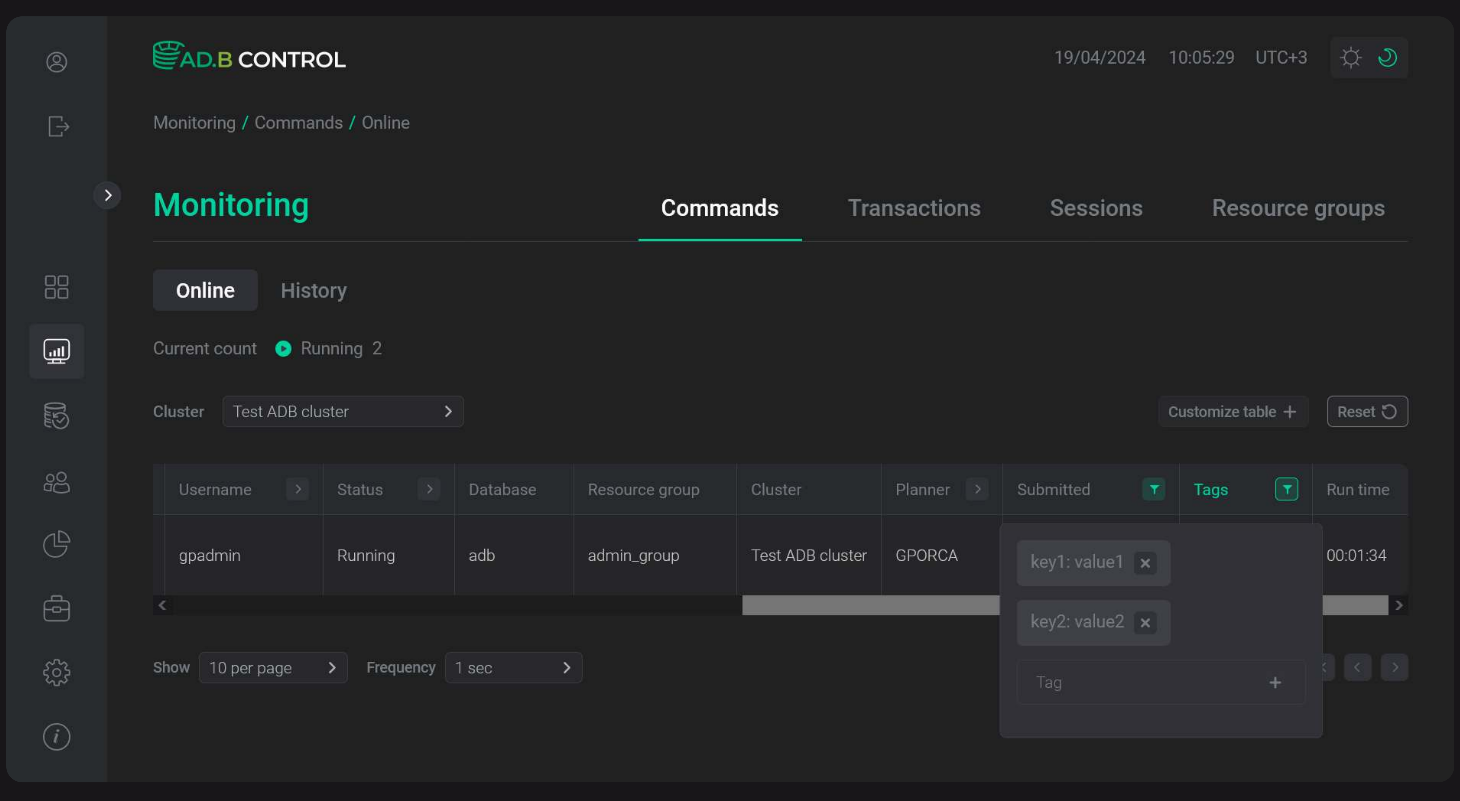
Filter by the first tag value

As a result of filtering, both queries are left on the page, because for both of them `value1` is set as the `key1` value.



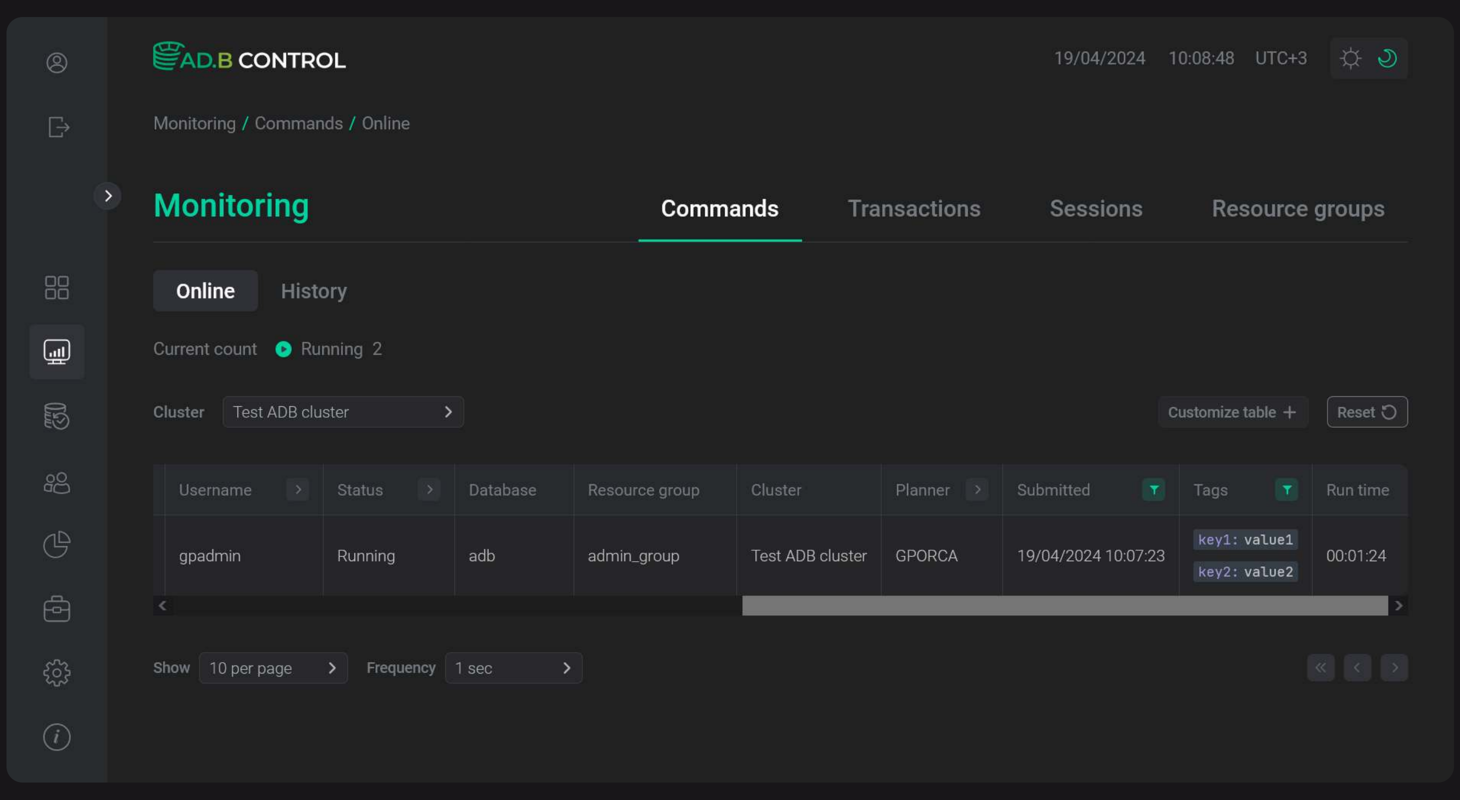
Filtering result for one tag

8. You can add a filter by the second tag value as shown below. Each tag value should be entered separately.



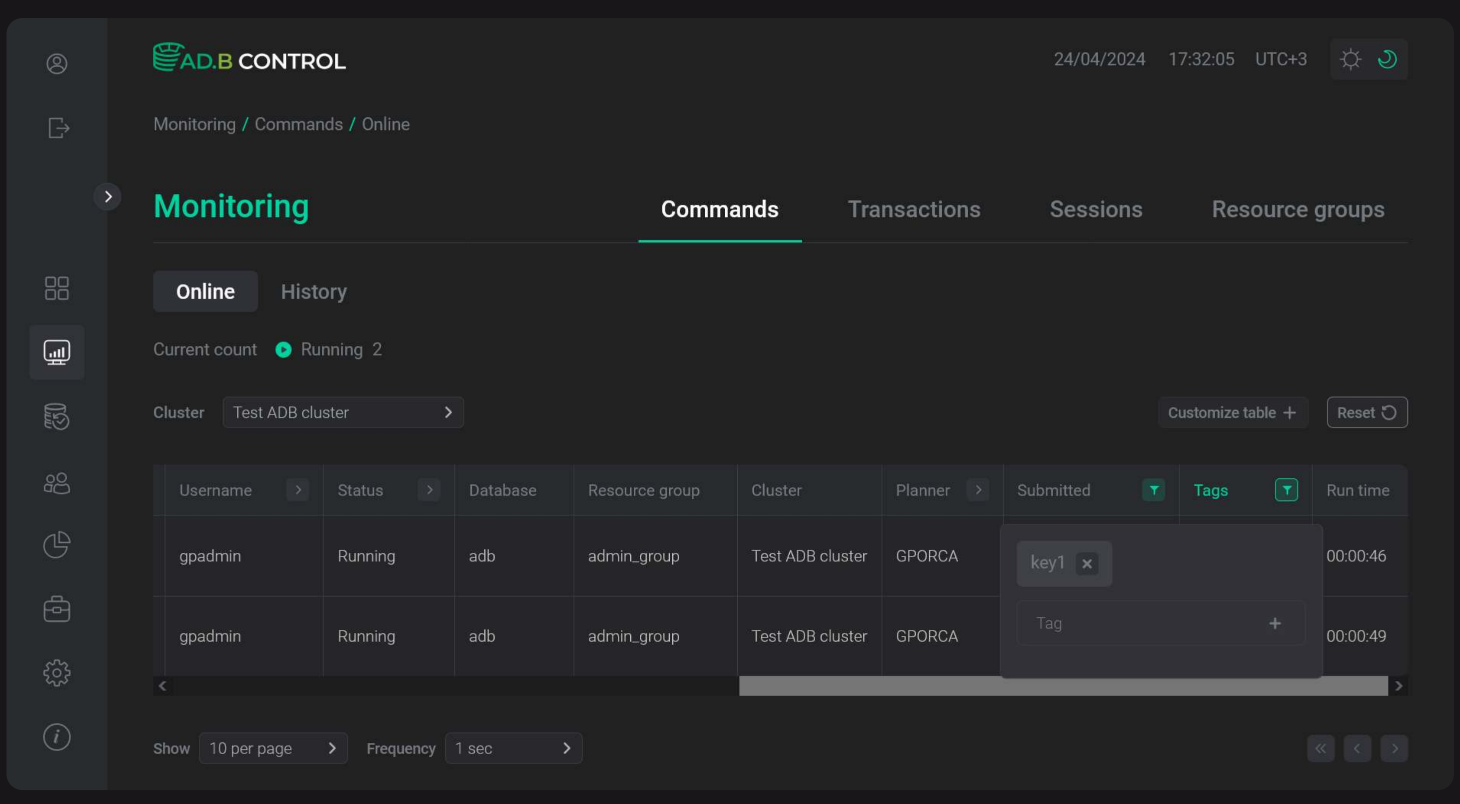
Add a filter by the second tag value

After you apply a filter, only the first session query is left on the page.



Filtering result for two tags

9. Also, you can filter data by tag names without tag values. As a result, all queries, for which the selected tag is specified, are shown, despite the tag value.



Filtering by the tag name

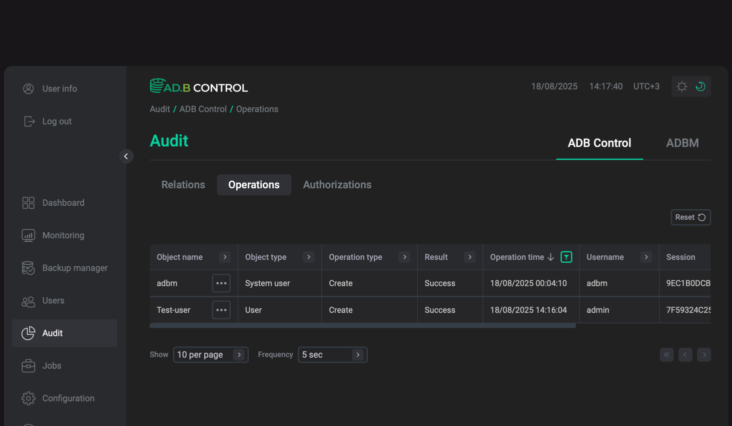
Audit

Contents

[illegible]

ADB Control

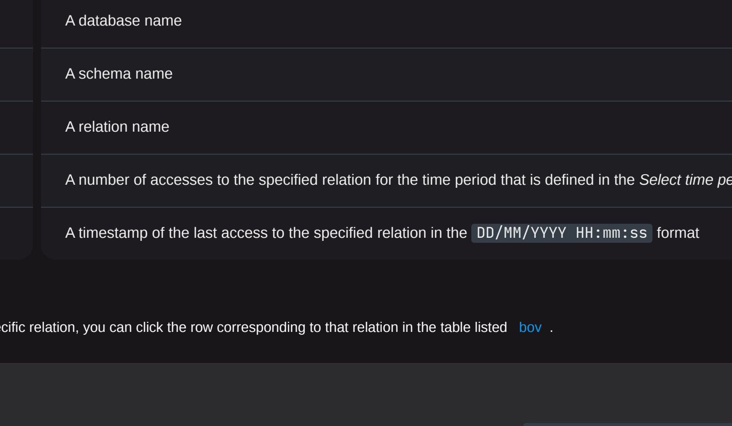
[illegible]



The Audit page

Relations

[illegible]



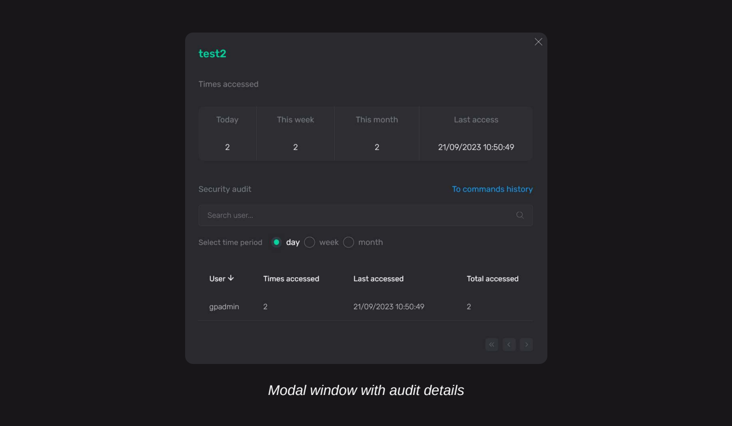
The Relations tab

On the tab, there is a table with the following information.

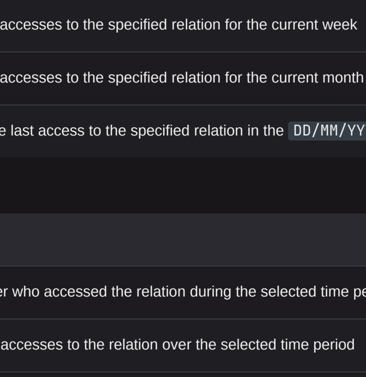
The Relations tab fields

Field	Description
Database	[illegible]
Schema	[illegible]
Relation	[illegible]
Times accessed	[illegible]
Last accessed	[illegible]

[illegible]



[illegible]



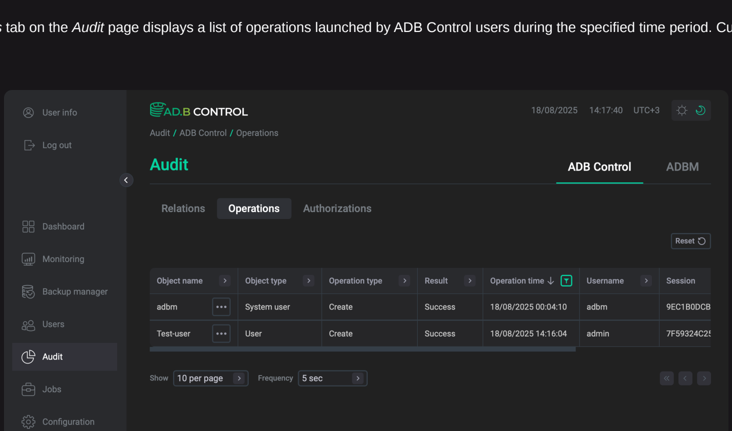
[illegible]

The Times accessed section fields

Field	Description
Total	[illegible]
This week	[illegible]
This month	[illegible]
Last accessed	[illegible]

The Security audit tab

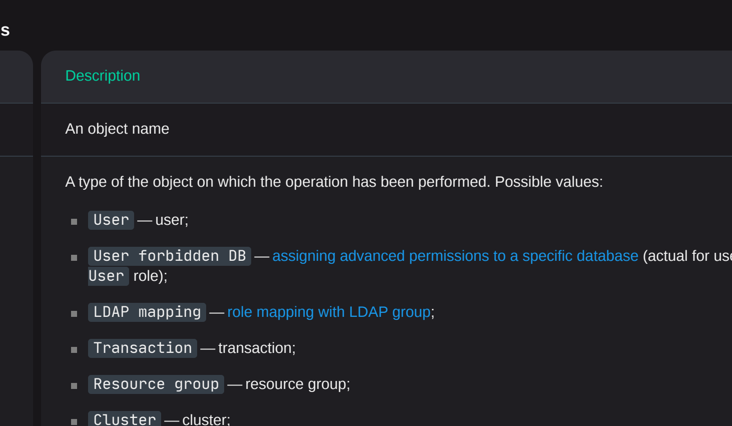
[illegible]



[illegible]

Operations

[illegible]



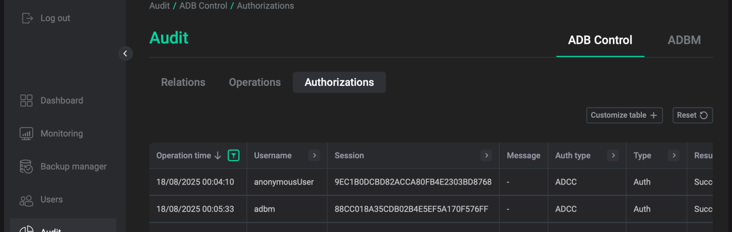
The Operations tab

On the tab, there is a table with the following information.

The Operations tab fields

Field	Description
Object name	[illegible]
Object type	A type of the object on which the operation has been performed. Possible values: [illegible] ■ `LDAP mapping` — role mapping with LDAP group; ■ `Transaction` — transaction; ■ `Resource group` — resource group; ■ `Cluster` — cluster; ■ `Cluster export databases` — databases, whose metrics will be offloaded; ■ `Job` — ADB Control job; ■ `System settings` — system configuration parameters; ■ `System user` — system user; ■ `Session` — ADB session; ■ `User session` — ADB Control user session; ■ `Cluster guc` — cluster GUC; ■ `Cluster databases` — databases that should be monitored in ADB Control.
Operation type	A type of the operation that has been applied to the *Object name* object. Possible values: ■ `Cancel` ■ `Create` ■ `Delete` ■ `Resource group change` ■ `Terminate` ■ `Update` ■ `Archive` ■ `Activate`
Result	An operation result. Possible values: ■ `Success` ■ `Failed` ■ `Denied`
Operation time	An operation end timestamp in the `DD/MM/YYYY HH:mm:ss` format
Username	A name of the user who performed the operation
Session	An identifier of the ADB Control user session where the operation was initiated
Host	IP address of the host from which the operation was initiated

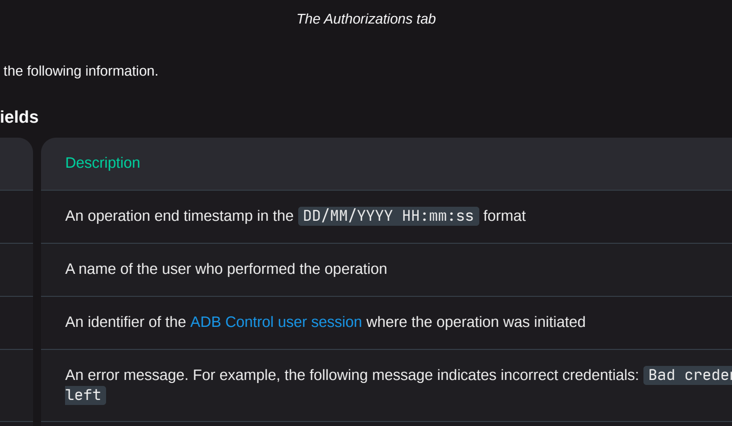
The first table column *Object name* contains the ••• button. When you click this button, a block appears under the current table row showing a list of object parameters that were modified during the current operation.



List of modified parameters

Authorizations

The *ADB Control → Authorizations* tab on the *Audit* page displays a list of ADB Control user authorizations for the selected time period. It displays both successful and failed login attempts as well as failed authorizations in ADB. Current day data is displayed by default.



The Authorizations tab

On the tab, there is a table with the following information.

The Authorizations tab fields

Field	Description
Operation time	An operation end timestamp in the `DD/MM/YYYY HH:mm:ss` format
Username	A name of the user who performed the operation
Session	An identifier of the ADB Control user session where the operation was initiated
Message	An error message. For example, the following message indicates incorrect credentials: `Bad credentials, 4 attempts left`
Auth type	An authorization type: ■ `ADCC` — ADB Control login attempt via basic authentication; ■ `ADCC LDAP` — ADB Control login attempt via LDAP authentication; ■ `ADB` — failed ADB login attempt; ■ `ADB LDAP` — failed ADB login attempt via LDAP authentication.
Type	An event type. Possible values: ■ `Auth` ■ `Logout` ■ `Session expired`
Result	An event result. Possible values: ■ `Success` ■ `Failed`
Host	IP address of the host from which the event was initiated. This column becomes visible after table customization
Session Time	A start timestamp of the session (within which the current event took place) in the `DD/MM/YYYY HH:mm:ss` format. This column becomes visible after table customization

If necessary, you can add columns to the table. To do this, click *Customize table* and select column names in the section that opens. Currently you can add the *Host* and *Session Time* columns (see column descriptions above).

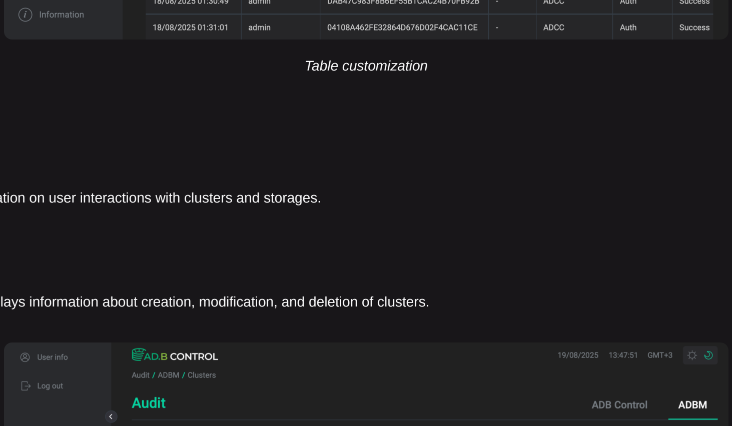


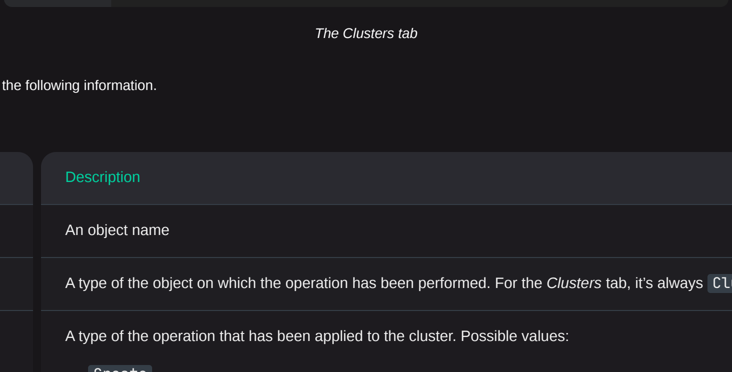
Table customization

ADBM

The *ADBM* tab displays information on user interactions with clusters and storages.

Clusters

The *ADBM → Clusters* tab displays information about creation, modification, and deletion of clusters.



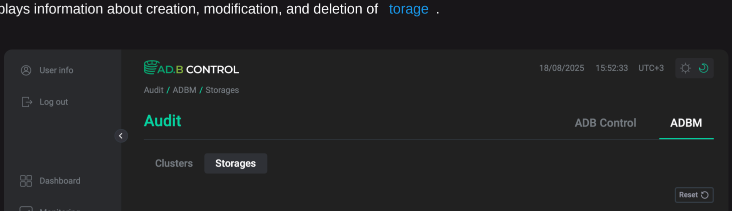
The Clusters tab

The tab consists of a table with the following information.

The Clusters tab fields

Field	Description
Object name	An object name
Object type	A type of the object on which the operation has been performed. For the *Clusters* tab, it's always `Cluster`
Operation type	A type of the operation that has been applied to the cluster. Possible values: ■ `Create` ■ `Update` ■ `Delete`
Result	An operation result. Possible values: ■ `Success` ■ `Failed` ■ `Denied`
Operation time	An operation end timestamp in the `DD/MM/YYYY HH:mm:ss` format
Username	A name of the user who performed the operation
Session	An identifier of the ADB Control user session where the operation was initiated
Host	IP address of the host from which the operation was initiated

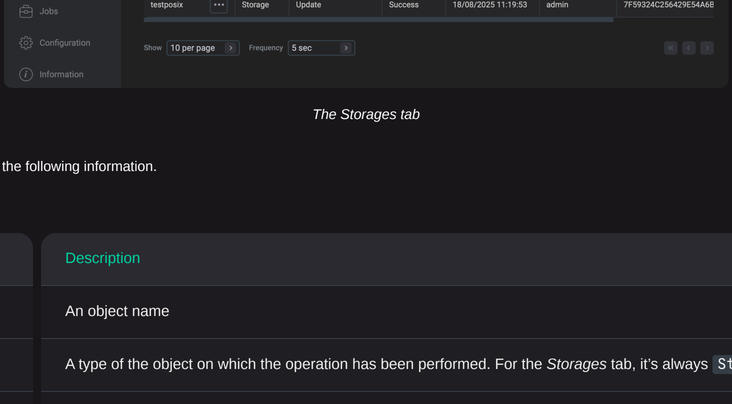
The first table column *Object name* contains the ••• button. When you click this button, a block appears under the current table row showing a list of the cluster parameters that were modified during the current operation.



List of modified cluster parameters

Storages

The *ADBM → Storages* tab displays information about creation, modification, and deletion of torage .



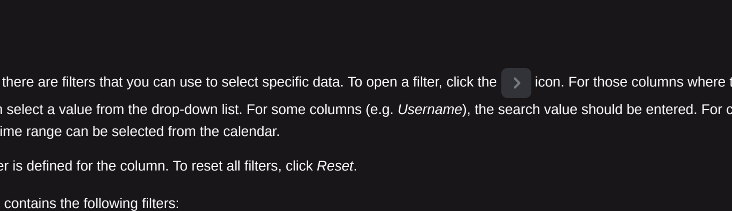
The Storages tab

The tab consists of a table with the following information.

The Storages tab fields

Field	Description
Object name	An object name
Object type	A type of the object on which the operation has been performed. For the *Storages* tab, it's always `Storage`
Operation type	A type of the operation that has been applied to the storage. Possible values: ■ `Create` ■ `Update` ■ `Delete`
Result	An operation result. Possible values: ■ `Success` ■ `Failed` ■ `Denied`
Operation time	An operation end timestamp in the `DD/MM/YYYY HH:mm:ss` format
Username	A name of the user who performed the operation
Session	An identifier of the ADB Control user session where the operation was initiated
Host	IP address of the host from which the operation was initiated

The first table column *Object name* contains the ••• button. When you click this button, a block appears under the current table row showing a list of the storage parameters that were modified during the current operation.



List of modified storage parameters

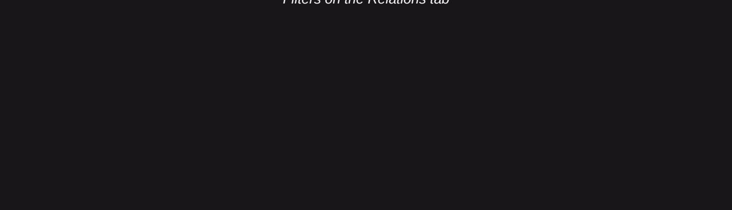
Table filters

In most of the column headers, there are filters that you can use to select specific data. To open a filter, click the > icon. For those columns where the set of possible values is limited (e.g. *Auth type*), you can select a value from the drop-down list. For some columns (e.g. *Username*), the search value should be entered. For columns that show date and time (e.g. *Operation time*), the time range can be selected from the calendar.

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

Additionally, the elations tabl contains the following filters:

- *Cluster* — filter by the cluster name. Select a value from the drop-down list.
- *Time range* — filter by the last access time (see *Last accessed* above). Select one of the following values:
  - `Last day` — current day;
  - `Last week` — current week;
  - `Last month` — current month.



Filters on the Relations tab

View jobs

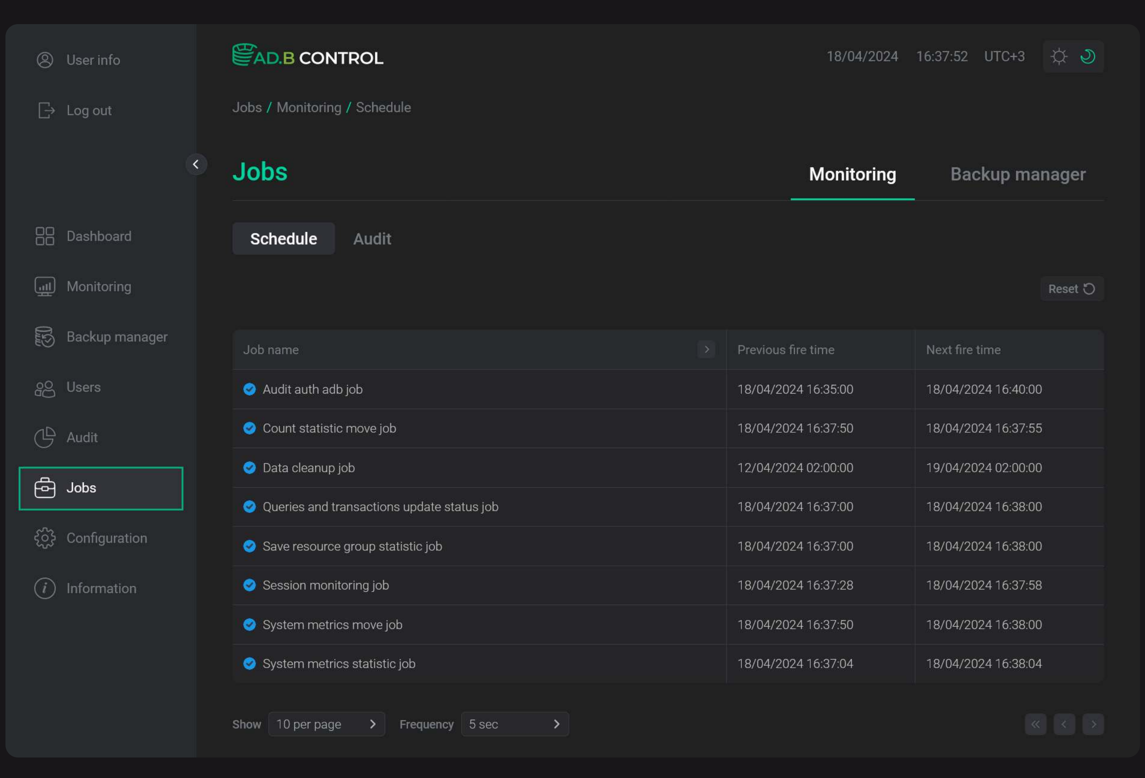
Daria Barysheva

AD.B ES

Contents

To Table of Contents

The ***Jobs*** page in the ADB Control web interface displays information on planned and completed system jobs in ADB Control/ADB Backup Manager. This page includes multiple tabs, each of which is described in detail below.

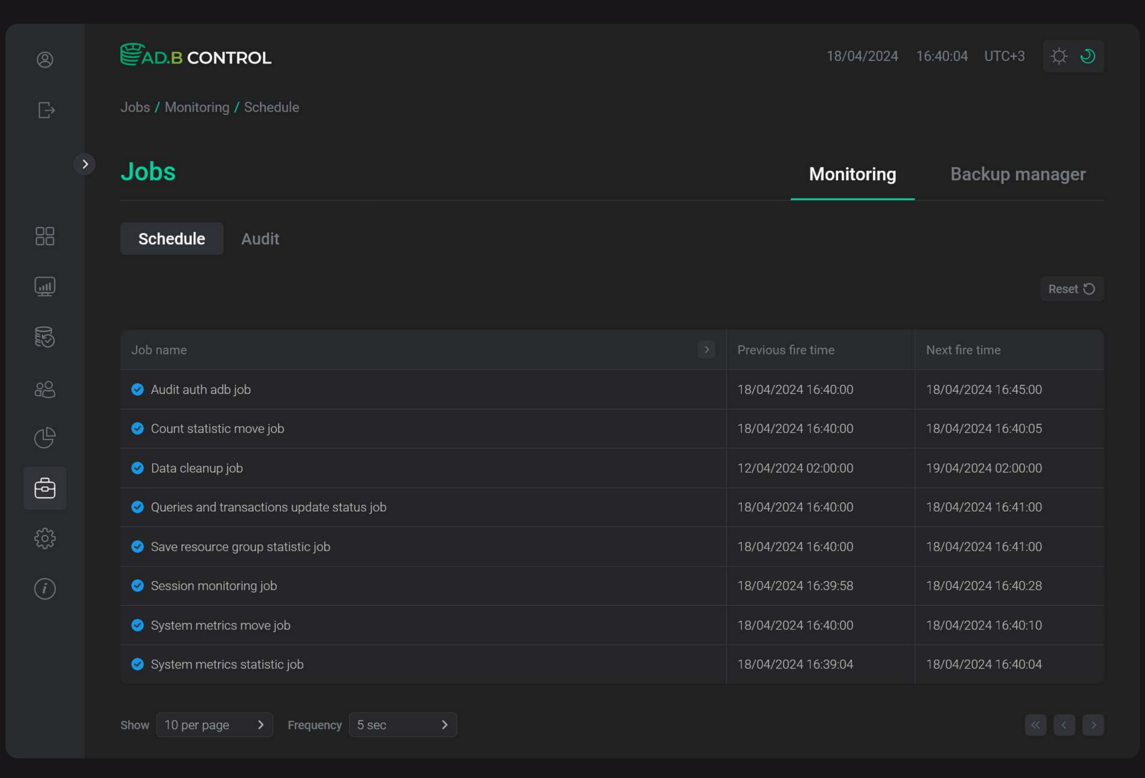


The Jobs page

Monitoring

Schedule

The ***Monitoring*** → ***Schedule*** tab on the ***Jobs*** page displays scheduled system jobs in ADB Control.



The Monitoring → Schedule tab

On the tab, there is a table with the following information.

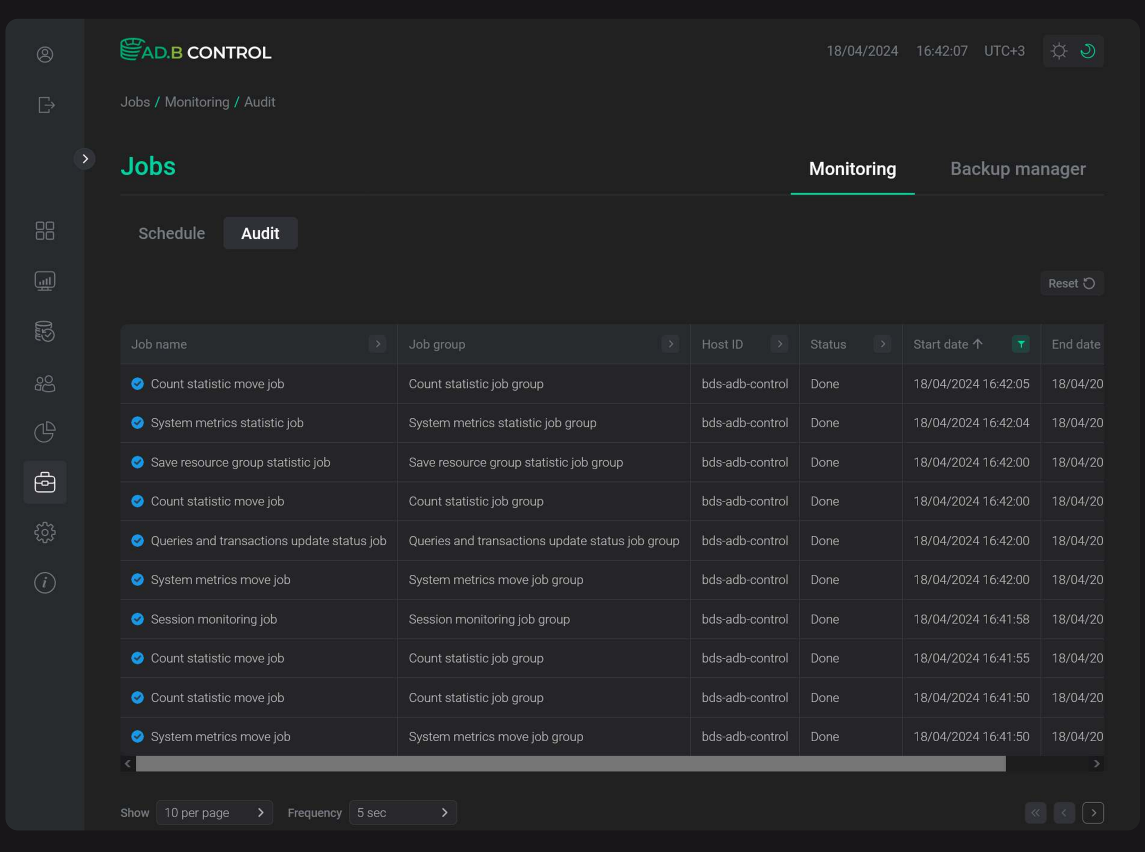
The "Monitoring → Schedule" tab fields

Field	Description
Job name	A system job name. You can get a full job description by pointing a cursor at its name
Previous fire time	A timestamp of the previous job launch in the `DD/MM/YYYY HH:mm:ss` format
Next fire time	A timestamp of the next scheduled job launch in the `DD/MM/YYYY HH:mm:ss` format

In the ***Job name*** column header of the table with a list of scheduled jobs, there is a filter that you can use to select data by the job name. You can search by the first characters entered. To open a filter, click the > icon. The icon means that a filter is defined for the column. To reset all filters, click ***Reset***.

Audit

The ***Monitoring*** → ***Audit*** tab on the ***Jobs*** page displays all jobs launched in ADB Control during the specified time period. Current day data is displayed by default.



The Monitoring → Audit tab

On the tab, there is a table with the following information.

The "Monitoring → Audit" tab fields

Field	Description
Job name	A job name. If the job has the `Error` status, the ***Job name*** table column contains ••• . When you click this button, under the current table row a block appears with a message on the source error, which can be copied by clicking Copy in the top right corner of the block
Job group	A job group name
Host ID	An identifier of the host where the job was launched (in the `<IP address>:<port>` format)
Status	A job status. Possible values: • `Done` • `Error` • `Unknown` • `Running`
Start date	A job start timestamp in the `DD/MM/YYYY HH:mm:ss` format
End date	A job end timestamp in the `DD/MM/YYYY HH:mm:ss` format

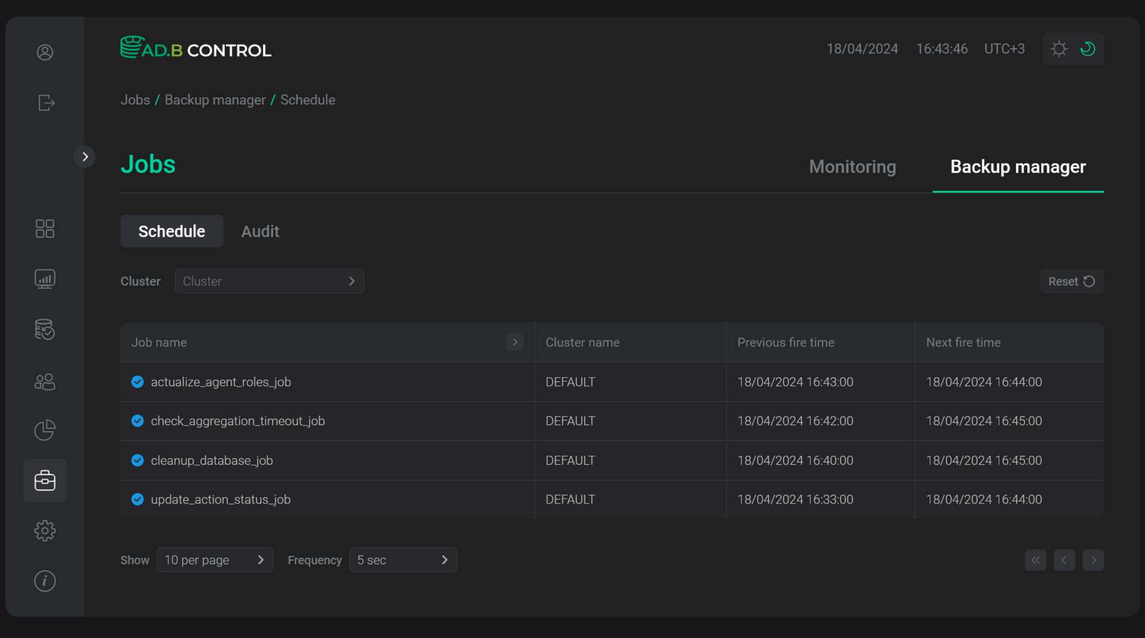
In the column headers of the table with a list of launched jobs, there are filters that you can use to select specific data. To open a filter, click the > icon. For those columns where the set of possible values is limited (e.g. ***Job name***), you can select a value from the drop-down list. For some columns (e.g. ***Host ID***), the search value should be entered. For columns that show date and time (e.g. ***Start date***), the time range can be selected from the calendar.

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

Backup manager

Schedule

The ***Backup manager*** → ***Schedule*** tab on the ***Jobs*** page displays scheduled system jobs in ADB Backup Manager.



The Backup manager → Schedule tab

On the tab, there is a table with the following information.

The "Backup manager → Schedule" tab fields

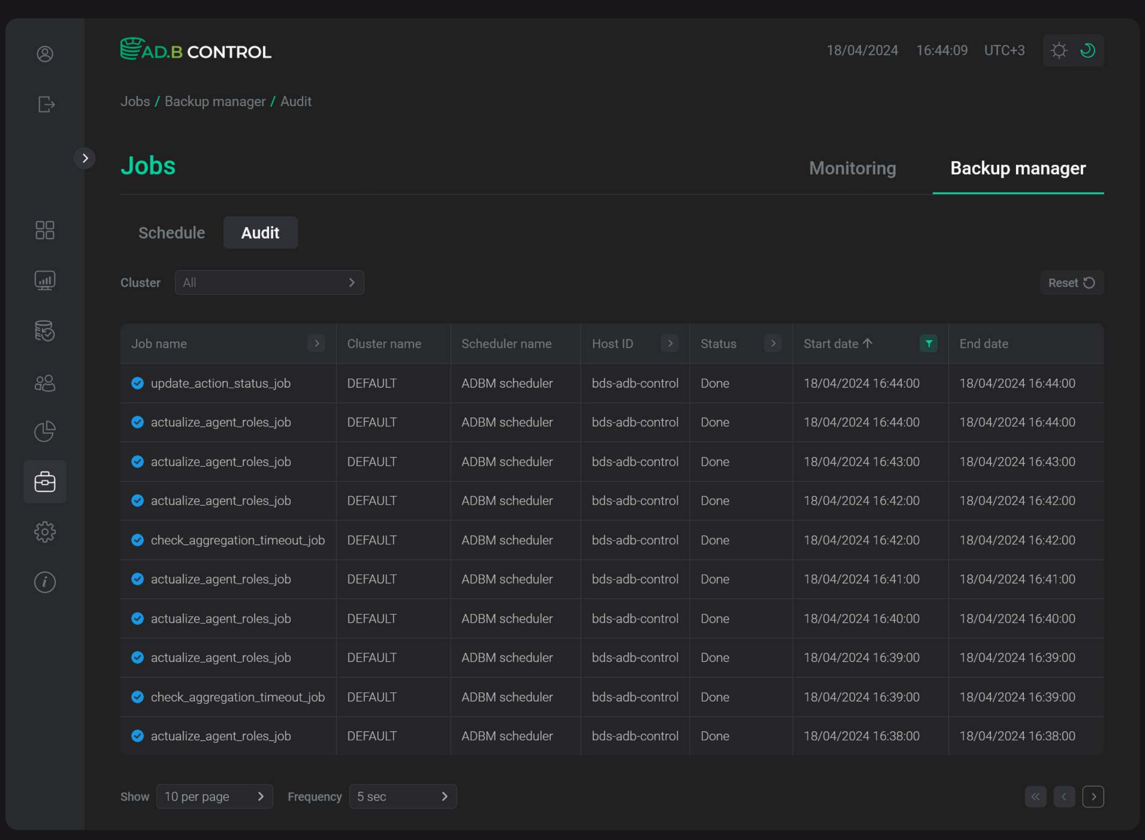
Field	Description
Job name	A system job name. You can get a full job description by pointing a cursor at its name
Cluster name	A name of the cluster for which the job is scheduled. The `DEFAULT` value is used for common jobs that are not related to any specific cluster
Previous fire time	A timestamp of the previous job launch in the `DD/MM/YYYY HH:mm:ss` format
Next fire time	A timestamp of the next scheduled job launch in the `DD/MM/YYYY HH:mm:ss` format

Above the table with a list of scheduled jobs, the ***Cluster*** filter is located. You can use this filter to select the ADB cluster for which you want to display data in the table. By default, all clusters are selected.

In the ***Job name*** column header of the table with a list of scheduled jobs, there is a filter that you can use to select data by the job name. You can search by the first characters entered. To open a filter, click the > icon. The icon means that a filter is defined for the column. To reset all filters, click ***Reset***.

Audit

The ***Backup manager*** → ***Audit*** tab on the ***Jobs*** page displays all jobs launched in ADB Backup Manager during the specified time period. Current day data is displayed by default.



The Backup manager → Audit tab

On the tab, there is a table with the following information.

The "Backup manager → Audit" fields

Field	Description
Job name	A job name. If the job has the `Error` status, the ***Job name*** table column contains ••• . When you click this button, under the current table row a block appears with a message on the source error, which can be copied by clicking Copy in the top right corner of the block
Cluster name	A name of the cluster where the job was launched. The `DEFAULT` value is used for common jobs that are not related to any specific cluster
Scheduler name	A scheduler name
Host ID	A name of the host where the job was launched
Status	A job status. Possible values: • `Done` • `Error` • `Unknown` • `Running`
Start date	A job start timestamp in the `DD/MM/YYYY HH:mm:ss` format
End date	A job end timestamp in the `DD/MM/YYYY HH:mm:ss` format

Above the table with a list of launched jobs, the ***Cluster*** filter is located. You can use this filter to select the ADB cluster for which you want to display data in the table. By default, all clusters are selected.

In the column headers of the table with a list of launched jobs, there are filters that you can use to select specific data. To open a filter, click the > icon. For those columns where the set of possible values is limited (e.g. ***Job name***), you can select a value from the drop-down list. For some columns (e.g. ***Host ID***), the search value should be entered. For columns that show date and time (e.g. ***Start date***), the time range can be selected from the calendar.

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

View information on components

Daria Barysheva

Contents

To Table of Contents

The ***Information*** page in the ADB Control web interface displays the state of the ADB Control and ADBM components. This page includes multiple tabs, each of which is described in detail below.

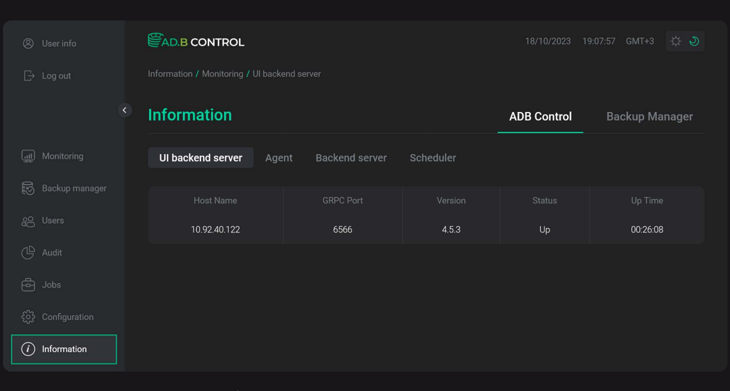
ADB Control

On the ***Information → ADB Control*** page, the status of ADB Control components is displayed.

> **NOTE**
>
> For more information on the ADB Control architecture, see Arenadata DB Control overview.

UI backend server

The ***UI backend server*** tab on the ***Information → ADB Control*** page contains information on the backend server of the ADB Control web interface (UI backend server).



The Information → ADB Control → UI backend server tab

Fields of the "Information → ADB Control → UI backend server" tab

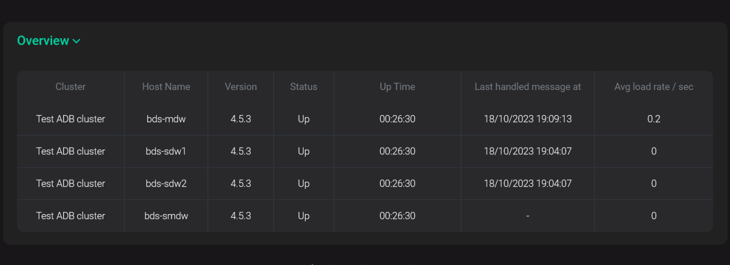
Field	Description
Host Name	An IP address of the host where UI backend server is deployed
GRPC Port	The gRPC port that is used by UI backend server
Version	A UI backend server version
Status	A UI backend server status. The possible value is `UP`, which means that UI backend server works
Up Time	The time since the last UI backend server startup (in hours, minutes, seconds)

Agent

The ***Agent*** tab on the ***Information → ADB Control*** page displays information on ADB Control agents. The tab consists of the following sections:

- ***Overview***
- ***Aggregated load ratio info***

The ***Overview*** section displays the general information on all ADB Control agents installed on hosts of the ADB clusters that are connected to the monitoring system.

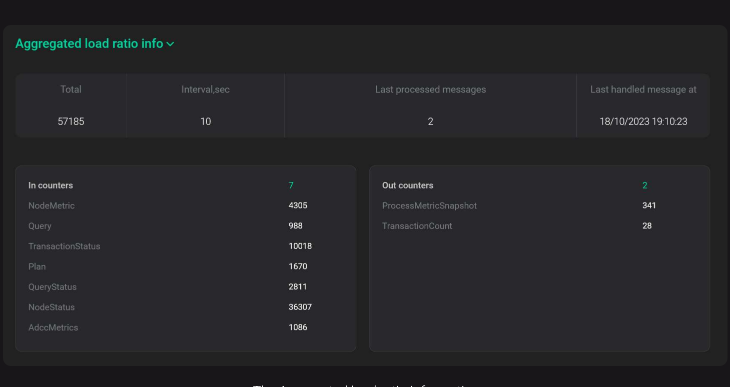


The Overview section

Fields of the Overview section

Field	Description
Cluster	A name of the ADB cluster
Host Name	A name of the host where the ADB Control agent is deployed
Version	A version of the ADB Control agent
Status	A status of the ADB Control agent. The possible value is `Up`, which means that the agent works. Stopped agents are not displayed in the table (for example, when a host is disabled)
Up Time	The time since the last agent startup in hours, minutes, seconds
Last handled message at	A timestamp of the last processed protobuf message (received from segments) in the `DD/MM/YYYY HH:mm:ss` format
Avg load rate / sec	An average number of incoming protobuf messages per second. It is calculated as a ratio of the number of messages received by the agent during the time that is defined in the Interval,sec field to the value of that period

The ***Aggregated load ratio info*** displays the aggregated statistics on the processing of protobuf messages (received from segments) by all ADB Control agents.

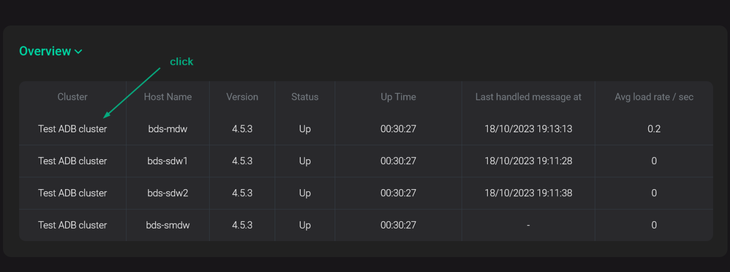


The Aggregated load ratio info section

Fields of the "Aggregated load ratio info" section

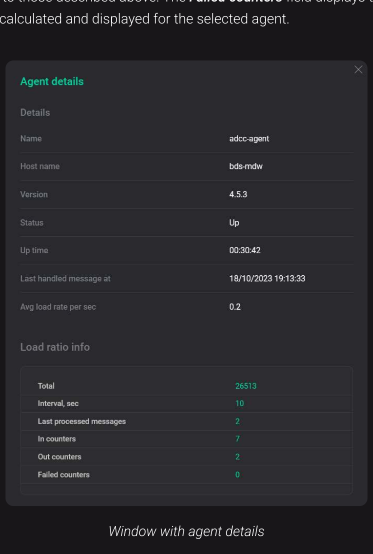
Field	Description
Total	A total number of received protobuf messages (since agent startup)
Interval,sec	A time interval between counters sampling (in seconds). By default, `10`. You can change the limit via the `eureka.update.counter.interval` configuration parameter
Last processed messages	A number of processed messages during the last ***Interval,sec*** time period
Last handled message at	A timestamp of the last processed protobuf message (in the `DD/MM/YYYY HH:mm:ss` format)
In counters	A total number of protobuf messages received from segments by ADB Control agents (since agent startup). Messages are grouped by query type (counter). The number of types for incoming messages is displayed at the top of the list
Out counters	A total number of protobuf messages sent by ADB Control agents to Backend server (since agent startup). Messages are grouped by query type (counter). The number of types for outcoming messages is displayed at the top of the list

The information that is described above for the Overview and Aggregated load ratio info sections can be obtained for a specific ADB Control agent. To view the agent details, click the corresponding row in the ***Overview*** section.



Switch to the detailed agent information

All fields displayed for the selected agent are almost identical to those described above. The ***Failed counters*** field displays the number of out counters, which messages failed to be sent. Also note that an average load rate and all counters are calculated and displayed for the selected agent.



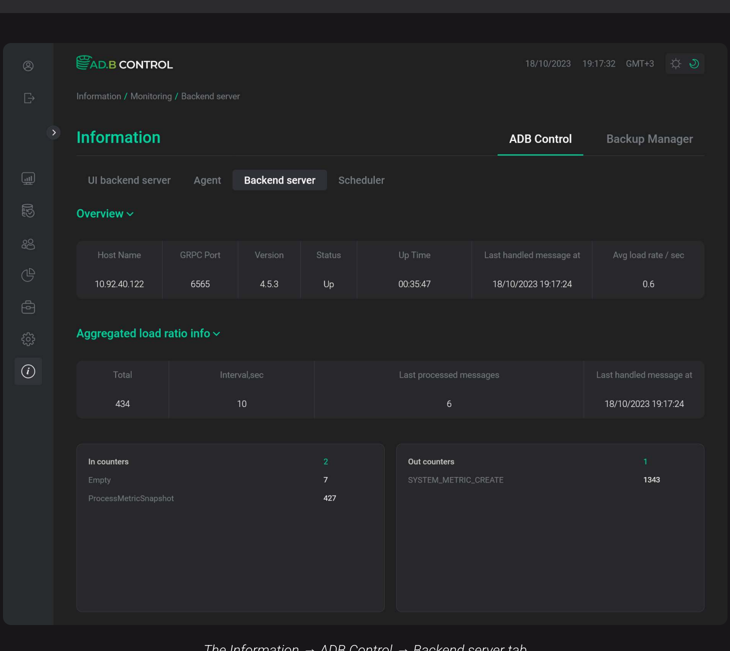
Window with agent details

Backend server

The ***Backend server*** tab on the ***Information → ADB Control*** page displays information on the ADB Control Backend server. The tab functionality is completely identical to the Agent tab described above. The only difference is that for Backend server ADB Control agents are the source of incoming messages, and PostgreSQL is the receiver of outcoming requests (***Query Database*** on the ADB Control architecture scheme).

You can also view the Backend server details in a separate window by clicking the row in the ***Overview*** section.

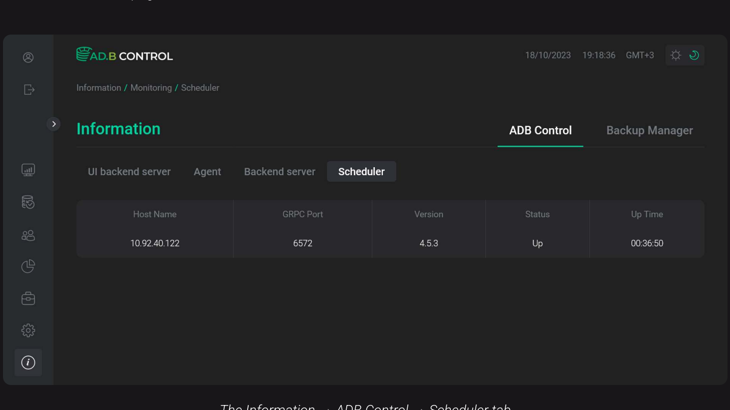
> **NOTE**
>
> The counter names in the ***In counters*** and ***Out counters*** sections may not be the same because Backend server aggregates and otherwise processes incoming messages and then creates new objects.



The Information → ADB Control → Backend server tab

Scheduler

The ***Scheduler*** tab on the ***Information → ADB Control*** page contains information on the ADB Control Scheduler.



The Information → ADB Control → Scheduler tab

Fields of the Information → ADB Control → Scheduler tab

Field	Description
Host Name	An IP address of the host where Scheduler is deployed
GRPC Port	The gRPC port that is used by Scheduler
Version	A Scheduler version
Status	A Scheduler status. The possible value is `UP`, which means that Scheduler works
Up Time	The time since the last Scheduler startup (in hours, minutes, seconds)

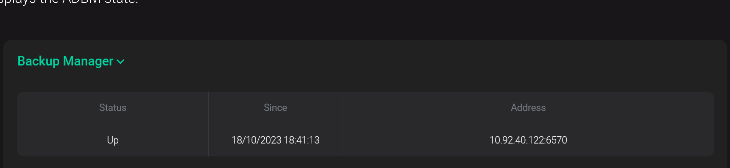
Backup manager

> **NOTE**
>
> For more information on the ADBM architecture, see Arenadata DB Backup Manager overview.

The ***Information → Backup manager*** page displays the state of ADBM and its agents. The page consists of two sections:

- ***Backup manager***
- ***Database & Agents***

The ***Backup manager*** section displays the ADBM state.

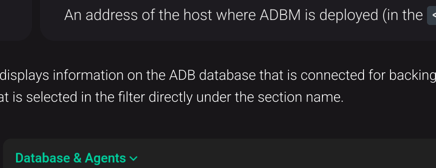


The Backup manager section

Fields of the "Backup manager" section

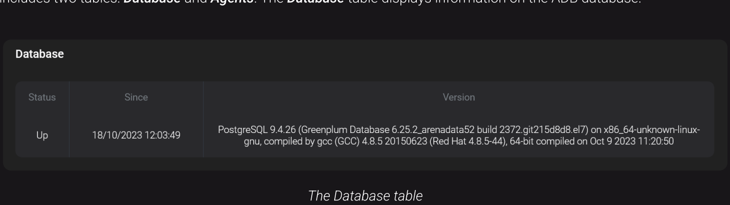
Field	Description
Status	An ADBM status. The possible value is `Up`, which means that ADBM works. If ADBM is stopped, the ***Backup manager*** page will be empty
Since	A timestamp of the last ADBM startup in the `DD/MM/YYYY HH:mm:ss` format
Address	An address of the host where ADBM is deployed (in the `<IP address>:<port>` format)

The ***Database & Agents*** section displays information on the ADB database that is connected for backing up and the ADBM agents that are installed on cluster hosts. The list of clusters that is displayed can be selected in the filter directly under the section name.



Filter by clusters

The ***Database & Agents*** section includes two tables: ***Database*** and ***Agents***. The ***Database*** table displays information on the ADB database.

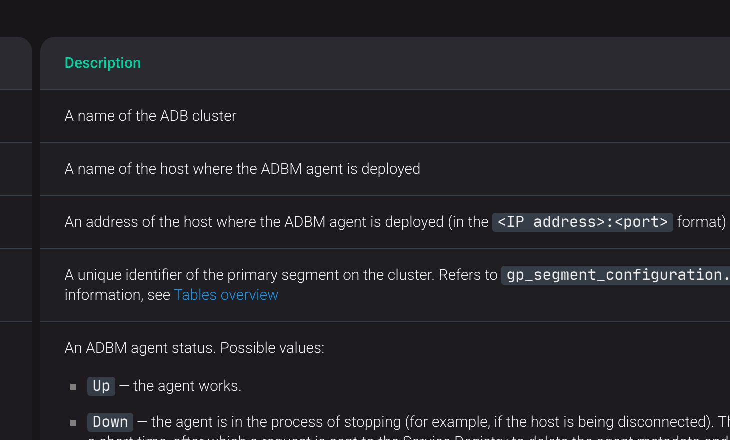


The Database table

Fields of the Database table

Field	Description
Status	A database status. Possible values: ■ `Up` — the database works. ■ `Down` — the database is stopped. In this case, the Agents table will be empty.
Since	A timestamp of the last database startup in the `DD/MM/YYYY HH:mm:ss` format
Version	A detailed description of the ADB build including PostgreSQL and Greenplum Database versions

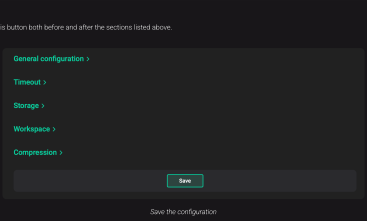
The ***Agents*** table displays information on ADBM agents.



The Agents table

Fields of the Agents table

Field	Description
Cluster	A name of the ADB cluster
Host name	A name of the host where the ADBM agent is deployed
Agent address	An address of the host where the ADBM agent is deployed (in the `<IP address>:<port>` format)
Segment number	A unique identifier of the primary segment on the cluster. Refers to `gp_segment_configuration.content`. For more information, see Tables overview
Status	An ADBM agent status. Possible values: ■ `Up` — the agent works. ■ `Down` — the agent is in the process of stopping (for example, if the host is being disconnected). The status is displayed for a short time, after which a request is sent to the Service Registry to delete the agent metadata and the agent entry is removed from the table.
Version	A version of the ADBM agent

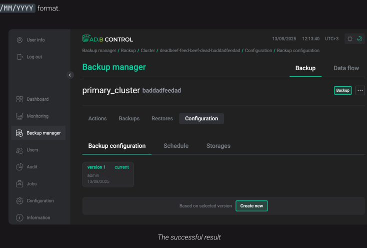


Save the configuration

4. As a result, the ***Apply backup configuration*** action runs. This action, in turn, generates several subactions. You can see all of them on the ***Actions*** tab (for more details, see View actions in ADBM).
   Starting with ADBM 2.6.5, you can open the ***Actions*** tab and view the action details by clicking the ***Apply backup configuration*** link, which is available in the ***Action created*** field after you save the configuration.

Until all necessary actions are completed successfully, the configuration will have the draft status. . As a result successful result of all actions, the configuration acquires the `current` status. Along with status, the following information about the configuration is available:

- The version number (starting with `1`).
- The user who created a configuration version.
- The creation date in the `DD/MM/YYYY` format.



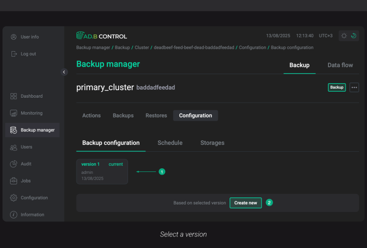
The successful result

Edit a configuration

After you created the configuration, you cannot edit it. But it is possible to add a new configuration version based on one of the previously created. To do this, perform the following steps:

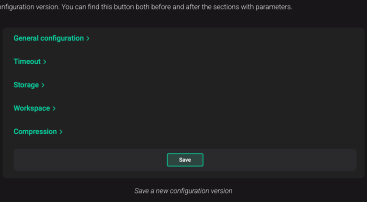
1. Open the ***Configuration*** → ***Backup configuration*** tab as described above.
2. In the list of available configuration versions, click the version on base of which you want to create a new one. All parameters of the selected version are displayed under the list with available versions (in the read-only mode).

> **NOTE**
>
> You can create a configuration based on any previous version, not just the latest.



Select a version

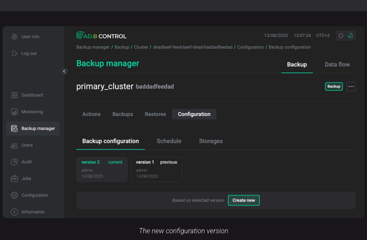
3. Click ***Create new***.
4. Apply necessary changes to the parameters that are located in the ***General configuration***, ***Timeout***, ***Repository***, and ***Compression*** sections. The parameters are the same as described above.
5. Click ***Save*** to add a new configuration version. You can find this button both before and after the sections with parameters.



Save a new configuration version

6. As a result, the ***Apply backup configuration*** action runs. This action, in turn, generates several subactions. You can see all of them on the ***Actions*** tab (for more details, see View actions in ADBM).
7. After all necessary actions are completed successfully, the new configuration version becomes available on the ***Configuration*** tab and has the `current` status. The previous configuration acquires the `previous` status.

> **NOTE**
>
> - ADBM uses the latest configuration version you added.
> - Inactive configuration versions that have no related backups will be removed during the next automatic or manual Cleanup action.

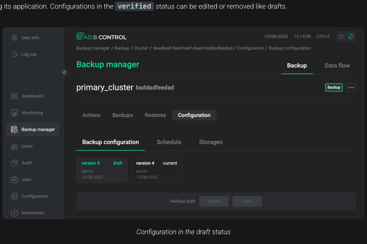


The new configuration version

Draft configurations

Starting with the version 1.6.3, ADBM supports an initial status `draft` for backup configurations. This status is assigned to all configurations when attempting to save them until all actions are successfully applied and the status is changed to `current`. However, if a new configuration cannot be validated at the moment of its creation, it is saved as a draft. Thus, you can avoid re-entering all information during the next applying attempt. Configuration drafts can be modified (and re-applied) or removed by clicking ***Edit*** and ***Delete***, respectively.

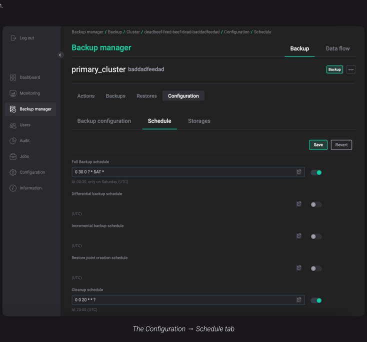
Starting with ADBM 1.7.3, the `verified` status is also available for backup configurations. This status means that validation of the backup configuration is successfully completed, but some errors occurred during its application. Configurations in the `verified` status can be edited or removed like drafts.



Configuration in the draft status

Schedule

The ***Configuration*** → ***Schedule*** tab allows you to select a schedule for various operations related to backups, such as creating various types of backups and restore points, data cleanups, etc. The tab is available starting with ADBM 2.5.0, before that schedules were part of a backup configuration and can be filled in along with other settings in the ***Configuration*** → ***General configuration*** section.



The Configuration → Schedule tab

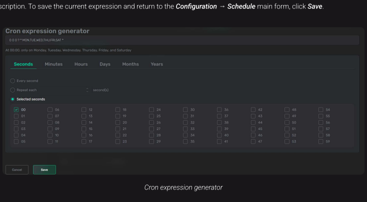
Below is the description of schedules that can be filled in on the ***Configuration*** → ***Schedule*** tab.

The "Configuration → Schedule" tab fields

Parameter	Description
Full Backup schedule	A schedule for automatic creation of full backups. If you are going to create backups only manually, do not activate the switcher that is located next to the field
Differential backup schedule	A schedule for automatic creation of differential backups. If you are going to create backups only manually, do not activate the switcher that is located next to the field
Incremental backup schedule	A schedule for automatic creation of incremental backups. If you are going to create backups only manually, do not activate the switcher that is located next to the field
Restore point creation schedule	A schedule for automatic creation of restore points. If you are going to create restore points only manually, do not activate the switcher that is located next to the field
Cleanup schedule	A schedule for deleting extra backups (see ***Number of full backups*** and ***Number of differential backups*** above) and backups with the `failed` and `invalid` statuses. If you are going to run the cleanup action only manually, do not activate the switcher that is located next to the field

To fill in any of the schedules mentioned above, activate the corresponding switch and enter the `cron` expression in the field that becomes available.

To switch to the alternative schedule fill mode, click the icon. As a result, the ***Cron expression generator*** window opens where you can set a schedule by selecting seconds, minutes, hours, days, months, and years. The selected values are automatically transformed into the `cron` expression that will be used as the field value. You can see it in the top part of the window along with its description. To save the current expression and return to the ***Configuration*** → ***Schedule*** main form, click ***Save***.



Cron expression generator

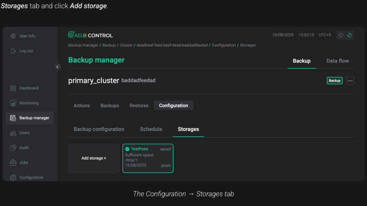
To save changes made on the ***Configuration*** → ***Schedule*** tab, click ***Save***. To undo changes, click ***Revert***.

Storages

In ADBM, a backup **storage** is a location where of your data backup is stored. Starting with ADBM 2.6.5, it's a separate entity that you can configure, edit, and share among different backup configurations. To use a storage for backup, you can select it in the ***Storage*** section of your backup configuration.

Add a new storage

1. Open the ***Configuration*** → ***Storages*** tab and click ***Add storage***.



The Configuration → Storages tab

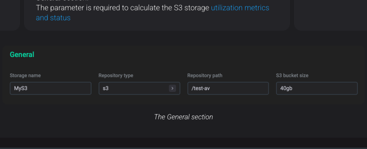
2. Fill in the parameters that are located in the ***Alert settings for storage*** and ***General*** sections described below. For S3 storages, you also need to fill in the parameters of the ***s3 configuration*** section.
3. Click ***Save***.

Alert settings for storage

Parameter	Description	Default value
Almost Full	The storage usage threshold (in percent) at which the storage configuration will have a yellow indicator and the `Almost Full` status	80
Critical	The storage usage threshold (in percent) at which the storage configuration will have a red indicator and the `Critical` status	90

General

Parameter	Description	Default value
Storage name	A unique name for the storage by which you can identify it when selecting a storage in a backup configuration	—
Repository type	A repository type. Possible values: - `posix` - `s3` - `cifs`	posix
Repository path	The repository where backups and archived WAL segments are to be stored. The repository directory should be pre-created and available from master and segment hosts. You can use the NFS shared directory, which is connected to all cluster segments. Do not use local directories on segments if mirroring is switched on — in that case, the hosts with mirror segments won't have the required stanzas. Ensure that you set up the same repository path for complementary primary and mirror segments	—
S3 bucket size	The amount of storage space that you want to allocate for backups in the bucket. Displayed if `s3` is selected in the ***Repository type*** field of the ***General*** section. The parameter is required to calculate the S3 storage utilization metrics and status	—



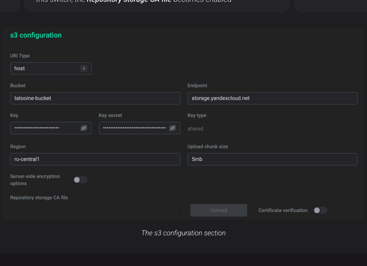
The General section

s3 configuration

The following parameters are available if `s3` is selected in the ***Repository type*** field.

> **NOTE**
>
> SSL is always used for connections of ADBM agents to S3.

Parameter	Description	Default value
URI Type	An endpoint type (see ***Endpoint*** below). Displayed if `s3` is selected in the ***Repository type*** field. Possible values: - `host` - `path`	`host`
Bucket	A bucket name in the S3 storage. A bucket name cannot contain dots	—
Endpoint	An endpoint of the S3 storage	—
Key	A key to access the S3 storage	—
Key secret	A secret key to access the S3 storage	—
Region	A region of the S3 storage	—
Upload chunk size	Size of the uploaded data chunk (**part size** in S3 terms). Limitations shown in the tooltip (`64kb-1tb`) come from `pgbackrest`. Note that S3 storages may have their own restrictions	5mb
Server-side encryption options	Indicates whether to perform data encryption on the server side. After this switch is activated, the ***Type*** and ***Key*** fields become visible and mandatory (see below)	false
Type	A type of the server-side data encryption: - `kms-key-id` — using the specified AWS key management service key. - `sse-customer-key` — using the specified customer key. This field is mandatory if the ***Server-side encryption options*** switch is activated. You should also specify a key value in the ***Key*** field (see below)	—
Key	A key that is used for data encryption on the server side. The key type is specified in the ***Type*** field (see above). This field is mandatory if the ***Server-side encryption options*** switch is activated. The field becomes enabled after you select a value from the ***Type*** drop-down list	—
Repository storage CA file	Certificate Authority (CA) PEM file to access the S3 storage. To select the file, click ***Upload***	—
Certificate verification	Defines whether to verify a repository storage certificate. If you activate this switch, the ***Repository storage CA file*** becomes enabled	true



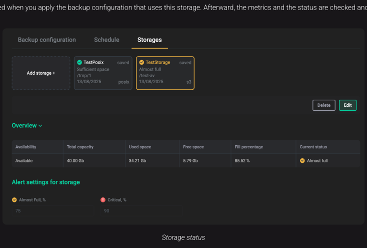
The s3 configuration section

Monitor storages

To monitor the storage status and space utilization, use the ***Overview*** section, which is available after applying the configuration with this storage. The ***Overview*** section allows you to monitor the following parameters:

- ***Availability***. Shows whether the storage is available. If ADBM cannot access the storage, it won't perform a backup.
- Space utilization: ***Total capacity***, ***Used space***, ***Free space***, and ***Fill percentage***.
  The fill percentage shows the ratio between the total capacity and used space. For POSIX and CIFS storages, the total capacity is calculated automatically. For S3 storages, you should manually specify the total capacity using the ***S3 bucket size*** field. If no value is set, the S3 storage utilization metrics and status won't be available.
- ***Current status***. Depending on the fill percentage value, each storage may have one of the following statuses:
  - `Sufficient space` — the fill percentage value is below the percentage configured under Alert settings for storage, in the ***Almost Full*** limit.
  - `Almost Full` — the fill percentage value is above the ***Almost Full*** limit, but below the ***Critical*** limit.
  - `Critical` — the fill percentage value is above the ***Critical*** limit. To avoid storage overflow during a backup, you can use the Fail on low space flag, which ensures the backup won't proceed if the storage is in the `Critical` state.

These metrics are first calculated when you apply the backup configuration that uses this storage. Afterward, the metrics and the status are checked and updated every minute.



Storage status

Switch to the "Create restore point" action

The window that opens contains the following fields:

- ***Restore point*** — an automatically generated name for a new restore point.
- ***Configuration*** — the configuration version that will be applied. You can click it to check or edit the configuration parameters in the list of configurations.
- ***Date*** — a current timestamp.

The "Create restore point" action form

2. Change the restore point name in the ***Restore point*** field if necessary.
3. Click ***Run***.
4. As a result, the ***Create restore point*** action runs. This action, in turn, generates several subactions. You can see all of them on the ***Actions*** tab (for more details, see View actions in ADBM).
5. As a successful result of all actions, a new restore point becomes available. You can check it by running the Common restore, Partial restore, or Cleanup action.

The new restore point is available

Common restore

The ***Common restore*** action allows you to restore databases at the moment of one of existing restore points. This is the only action that can be applied to both a running and a stopped ADB cluster. The action logic depends on the cluster status:

- **ADB cluster is running (marked as Up in the ADBM UI)**. In this case, before the action is started, additional checks are automatically performed to ensure that the selected backup can be applied to the cluster:
  - The topology that is saved in the backup is compared with the topology obtained from the ADB cluster (from the system table `gp_segment_configuration`).
  - The backup metadata is also checked (see Verify below).

  If the validation is successful, the cluster stops automatically. In addition, if any error is fixed during the ***Common restore*** action, you can run Restore retry — this is a restart of the last restore process without any additional checks.
- **ADB cluster is stopped (marked as Down in the ADBM UI)**. In this case, any validation is skipped. However, it is still possible to start the restore process. This method is recommended only in case of fatal problems when the cluster does not even start.

> **NOTE**
>
> Starting with ADBM 2.5.0, you can also restore data in a stopped cluster using a data flow with the `Copy` type.

To restore databases, follow the steps:

1. Select the ***Common restore*** action on the cluster page or in the Clusters section.

Switch to the Common restore action

The window that opens contains the following fields:

- ***Time period*** — the timestamp range that is used to search restore points. The default value is the current date.
- ***Select restore point*** — the restore point at the moment of which you want to restore databases.
- ***Databases*** — the databases that you can restore for the selected restore point.
- ***Processes*** — a maximum number of processes to restore one segment.
- ***Restore mirrors*** — the drop-down list, which value indicates whether to restore data for mirrors and standby if they exist in the cluster:
  - `No` — do not restore mirrors.
  - `Parallel` — restore mirrors concurrently with primary segments.
  - `Sequential` — restore mirrors after all primary segments are restored.
- ***Flags***:
  - ***Delta*** — a flag that indicates the need to use checksums during the data restore from the `Incr` or `Diff` backup. It does not affect `Full` backups.
  - ***Force*** — a flag that indicates the need to completely overwrite data and tablespace paths in PostgreSQL.
  - ***Verify*** — a flag that indicates the need to check backup data before data restore.

> **IMPORTANT**
>
> - If the ***Delta*** flag is set and the ***Force*** flag is unset, checksums will be verified when restoring files with data, and only files that have been updated since the backup creation will be overwritten.
> - If both flags ***Delta*** and ***Force*** are set, data files will be compared based on their sizes and modification dates. Although the restore process in such case is quicker, it provides fewer assurances of integrity than the checksum check.

After you click ***Details***, you can also see additional fields:

- ***Date created*** — a creation timestamp of the selected restore point.
- ***Backup date*** — a creation timestamp of the latest backup that was completed before the selected restore point.
- ***Backup type*** — a type of the latest backup that was created before the selected restore point.
- ***Configuration*** — the configuration version that was applied to the selected restore point. You can click it to view the configuration parameters in the list of configurations.

The Common restore action form

2. Fill in the necessary fields:
   - Select a restore point in the ***Select restore point*** drop-down-list. Use the ***Time period*** filter if necessary.
   - In the ***Databases*** drop-down-list, select the databases that you want to restore.
   - Select the ***Restore mirrors*** list value.
   - If necessary, set the ***Delta*** and ***Force*** flags.
3. Click ***Run***.
4. As a result, the ***Restore*** action runs. This action, in turn, generates several subactions. You can see all of them on the ***Actions*** tab (for more details, see View actions in ADBM).
5. As a successful result of all actions, the selected databases should be restored at the moment of the specified restore point.

For running ADB clusters (with the `Up` status), if any error occurs when running the ***Common restore*** action — you can retry the latest restore process. To do this, select the ***Common restore*** action again (see step 1 above) and click ***Retry*** in the window that opens. If you want to run a new ***Common restore*** action from scratch (with all checks), click ***New restore***.

Retry the last Common restore action

Partial restore

> **NOTE**
>
> The ***Partial restore*** action is available starting with ADBM 2.2.2.

The ***Partial restore*** action allows you to restore specific tables at the moment of one of existing restore points. To restore tables, follow the steps:

1. Select the ***Partial restore*** action on the cluster page or in the Clusters section.

Switch to the Partial restore action

The window that opens contains the following fields:

- ***Time period*** — the timestamp range that is used to search restore points. The default value is the current date.
- ***Select restore point*** — the restore point at the moment of which you want to restore tables.
- ***Source database*** — the source database whose tables you can restore for the selected restore point.
- ***Target database*** — the target database where tables should be restored. If not specified, all tables will be restored in the source database.
- ***Table suffix*** — the suffix that will be added to all table names in the target database during the restore process. Maximum length is 32 symbols. If not specified, table names from the source database will be used. In the latter case, for successful data restore, it is important that the selected tables do not exist in the target database.
- ***Table name masks*** — an array of name masks for tables that should be restored. Each array element requires the following format: `<schema>/<table>`, where `<schema>` is a schema name template and `<table>` is a table name template. The `<table>` template may contain the `*` wildcards. Thus, each array element should match the following regular expression: `\w+(/[a-zA-Z_*]+)`. If `<schema>/` is absent, the `public` schema is used. To enter each mask, fill in the ***Add*** field and click `+`. To delete a mask, click `x`.
- ***Processes*** — a maximum number of parallel processes per host during the restore process. Values from the following range are supported: `[1,100]`. The default value comes from the ***Max processes per host*** field in the backup configuration (see the ***Workspace*** section in Manage configurations in ADBM).
- ***Verify*** — a flag that indicates the need to check backup data before data restore.

> **TIP**
>
> Since one process restores data of one segment on a host, it is recommended to set the maximum value equal to the number of primary segments on the host. If the value is higher, the rest of the processes will not be used, if the value is lower — the data restore process will take longer, yet consume less resources (due to the limit).

After you click ***Details***, you can also see additional fields:

- ***Date created*** — a creation timestamp of the selected restore point.
- ***Backup date*** — a creation timestamp of the latest backup that was completed before the selected restore point.
- ***Backup type*** — a type of the latest backup that was created before the selected restore point.
- ***Configuration*** — the configuration version that was applied to the selected restore point. You can click it to view the configuration parameters in the list of configurations.

The Partial restore action form

2. Fill in the necessary fields:
   - Select a restore point in the ***Select restore point*** drop-down-list. Use the ***Time period*** filter if necessary.
   - In the ***Source databases*** drop-down-list, select the database whose tables you want to restore.
   - If necessary, fill in the ***Target database*** and ***Table suffix*** fields.
   - Add an array of table name templates in the ***Table name masks*** field.
   - Change the default value of the ***Processes*** field if needed.

> **CAUTION**
>
> - If the ***Table suffix*** value is empty and a table with the source name already exists in the target database, the following error occurs in one of subactions: `Couldn't create table <table_name>`. In that case, the table data restore fails (the process of restoring other tables still continues). To avoid such a problem, use a suffix in case of existence of the table being restored.
> - When choosing the ***Table suffix*** value, remember that the maximum length of any DB relation name (including partition names) in ADB is `64` characters. If the overall length of the name with the suffix exceeds the specified limit, the name will be trimmed, which could potentially lead to duplication of table names.
> - If a source database does not contain tables with names matching the ***Table name masks*** array elements (for example, if a non-existing schema is specified), the following error occurs in one of subactions: `No tables were found by provided filter`, and the restore status is changed to `Failed`. If at least one mask specified in the ***Table name masks*** field is valid, the tables corresponding to this mask are successfully restored and invalid masks are ignored. To avoid such situations, ensure that all schemas and tables are present in the source database before launching the ***Partial restore*** action.

3. Click ***Run***.
4. As a result, the ***Table restore*** action runs. This action, in turn, generates several subactions. You can see all of them on the ***Actions*** tab (for more details, see View actions in ADBM).
5. As a successful result of all actions, the selected tables should be restored at the moment of the specified restore point. If at least one table is restored with a failure, the ***Table restore*** action status changes to `Warning`, and the restore status of each table can be seen on the ***Restores*** page — in details of the selected data restore on the Tables tab.

Cleanup

The ***Cleanup*** action is automatically executed according to the schedule that is defined by the ***Cleanup schedule*** parameter of the current configuration version (see the Configuration → Schedule tab). This action removes:

- The oldest extra full backups — if the number of full backups exceeds the ***Number of full backups*** parameter defined for the current configuration.
- The oldest extra differential backups — if the number of differential backups exceeds the ***Number of differential backups*** parameter defined for the current configuration.
- All backups in the following statuses: `Failed`, `Invalid`.
- Inactive configuration versions that have no related backups.

When you run the ***Cleanup*** action manually, it deletes all objects mentioned above but also allows you to remove restore points and backups related to the specified restore point. You define the restore point, and ADBM finds the latest full or differential backup that was completed before the selected point created. Then, all backups and restore points related to that backup are removed.

To run the ***Cleanup*** action manually, follow the steps:

1. Select the ***Cleanup*** action on the cluster page or in the Clusters section.

Switch to the Cleanup action

The window that opens contains the following fields:

- ***Time period*** — the timestamp range that is used to search restore points. The default value is the current date.
- ***Select restore point*** — the restore point at the moment of which you want to remove a set of related backups and restore points.
- ***Configuration*** — the configuration version that was applied to the selected restore point. You can click it to view the configuration parameters in the list of configurations.
- ***Date created*** — a creation timestamp of the selected restore point.

The Cleanup action form

2. Select a restore point in the ***Select restore point*** drop-down-list. Use the ***Time period*** filter if necessary.
3. Click ***Run***.
4. As a result, the ***Start cleanup job for Verify backup*** action runs. This action, in turn, generates several subactions. You can see all of them on the ***Actions*** tab (for more details, see View actions in ADBM).
5. As a successful result of all actions, backups and restore points are removed according to the rules described above. You can check it in the list of backups on the ***Backups*** tab. For example, if you have two backups, full and incremental, and select the restore point that was created after the incremental backup — both backups will be deleted.

List of backups before the Cleanup action

List of backups after the Cleanup action

To better understand the logic of the ***Cleanup*** manual action, you can view the example listed below. The figure shows a timeline with a list of backups of different types (called `full*`, `diff*`, `incr*`) and restore points (called `rp*`). The table explains which objects will be deleted when any of restore points is selected in the Cleanup action form.

Example with a list of backups and restore points

Logic of the manual Cleanup action

Selected restore point	Objects to be removed
rp1	All backups and restore points for the specified period
rp2	All backups and restore points for the specified period
rp3	All backups and restore points for the specified period
rp4	All backups and restore points for the specified period
rp5	diff1, incr2, rp5 — rp8
rp6	diff1, incr2, rp5 — rp8
rp7	diff1, incr2, rp5 — rp8
rp8	diff1, incr2, rp5 — rp8

Verify

The ***Verify*** action performs the following:

- Compares the backup metadata that is stored in ADBM with the backup metadata that is being received from `pgbackrest` (i.e. actual backups).
- Checks whether the backups and archives in a repository are valid on the moment of the selected restore point. The following checks are made:
  - Existence of backup files.
  - Size of backup files.
  - Checksums of backup files.
  - Validity of WAL archives that belong to the backup.

If the difference is fixed for any backups, they acquire the `Invalid` status as they cannot be used to restore databases. Such backups will be removed during the next Cleanup action launch.

Before comparing data and metadata, ADBM additionally checks existence of configuration files (*pgbackrest.conf*) for all timelines on all segment hosts. If any of these files are not present, the last available version of the timeline configuration is taken, and a new configuration file is created on the basis of this configuration. Then this file is written to the segment host for the selected timeline. This is necessary to ensure that all configurations are present before backup data and metadata are checked.

> **NOTE**
>
> - If the repository is inaccessible during the ***Verify*** action execution, backups are transferred to the `Unknown` status. After the repository becomes available, these backups can be marked as `Done` as a result of the ***Verify*** action.
> - Backup data is checked starting with ADBM 2.5.0 via the `pgbackrest verify` command. Before that, the ***Verify*** action checked only backup metadata using the `pgbackrest info` command.
> - It is recommended to set the immutable attribute on the local NFS mount directory to ensure correct detection of repository availability.

To run the ***Verify*** action, do the following:

1. Select the ***Verify*** action on the cluster page or in the Clusters section.

Switch to the Verify action

2. In the window that opens, select a restore point (on the moment of which creation you need to check backup data) and click ***Run***.

The Verify action form

3. As a result, the ***Verify backup*** action runs. This action, in turn, generates several subactions. You can see all of them on the ***Actions*** tab (for more details, see View actions in ADBM).
4. As a successful result of all actions, some of the backups can acquire the `Invalid` or `Unknown` status. You can search such backups using the ***Status*** filter on the Backups tab.

Terminate

The ***Terminate*** action allows you to terminate some of the currently running actions. This is the only action type for which a separate distributed lock is used (in etcd), which allows you to run ***Terminate*** in parallel with other types of actions.

> **IMPORTANT**
>
> - You can terminate actions of the following types:
>   - Backup
>   - Cleanup
> - After applying the ***Terminate*** action, you can resume the latest terminated backup via the Resume action.

Below is an example how to terminate the ***Backup*** action:

1. Run the ***Backup*** action as shown above.
2. Select the ***Terminate*** action on the cluster page or in the Clusters section.

Switch to the Terminate action

3. In the window that opens, click ***Run***.

The Terminate action form

4. As a result, the ***Terminate*** action runs. This action, in turn, generates several subactions. You can see all of them on the ***Actions*** tab (for more details, see View actions in ADBM). The current backup acquires the `Terminating` status. You can check it in the list of backups on the ***Backups*** tab.

   As a successful result of all actions, the backup acquires the `Stopped` status.

The backup status is switched to Stopped

Resume

> **IMPORTANT**
>
> - Starting with ADBM 2.1.2 the ***Resume*** action is available only for `Full` backups.
> - Bundled files cannot be reused when a backup is resumed. The number of files that can be reused depends on the Bundle limit value configured in the ***Storages*** section of the backup configuration: the higher the value, the fewer files are available for reuse when resuming a backup.

The ***Resume*** action allows you to resume the latest terminated backup launch. This action is available in the list of cluster actions only when backups with the `Stopped` status exist. Below is an example how to resume the backup that was terminated in the previous section:

1. Select the ***Resume*** action on the cluster page or in the Clusters section.

Switch to the Resume action

The window that opens contains the following fields:

- ***Restore point*** — an automatically generated name for a new restore point.
- ***Configuration*** — the configuration version that will be applied. You can click it to check the configuration parameters in the list of configurations.
- ***Date*** — a current timestamp.

The Resume action form

2. Change the restore point name in the ***Restore point*** field if necessary.
3. Click ***Run***.
4. As a result, the ***Resume backup*** action runs. This action, in turn, generates several subactions. You can see all of them on the ***Actions*** tab (for more details, see View actions in ADBM).
   The backup acquires the `Resume` status. You can check it in the list of backups on the ***Backups*** tab.

The backup has the Resume status

5. As a successful result of all actions, the backup acquires the `Done` status.

The backup status is switched to Done

Create configuration

> **NOTE**
>
> The ***Create configuration*** action is available starting with ADBM 2.3.2.

The ***Create configuration*** action redirects you to the ***Backup manager*** → ***Backup*** → ***Cluster*** → ***<Current ADB cluster name>*** → ***Configuration*** tab, where you can add a new backup configuration based on one of the previously created.

Switch to the Create configuration action

View backups in ADBM

Daria Barysheva

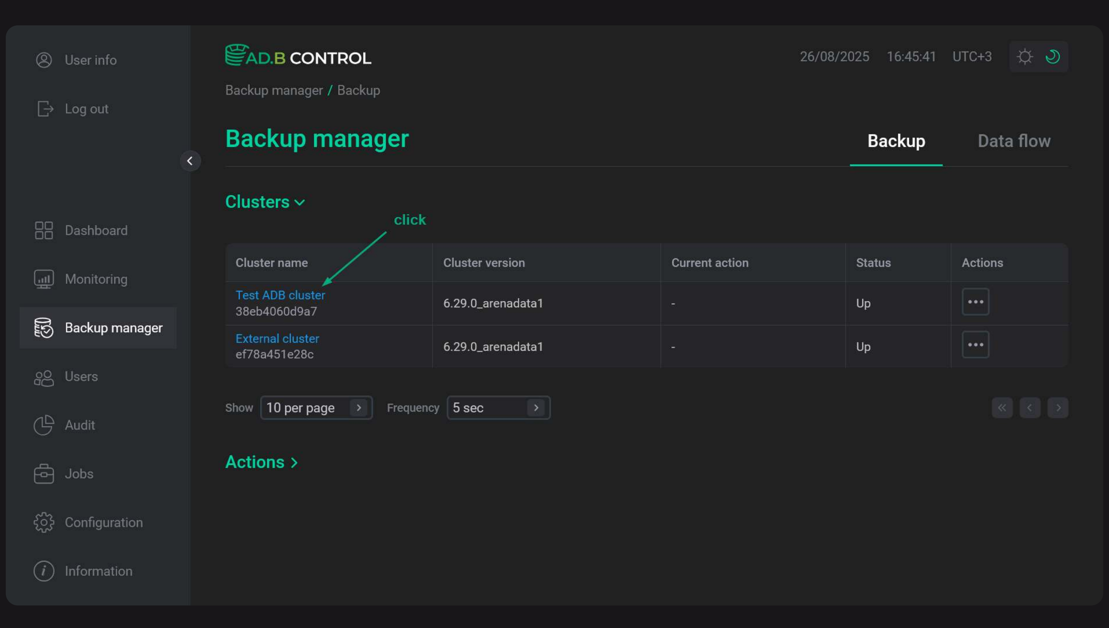
Contents

To Table of Contents

List of backups

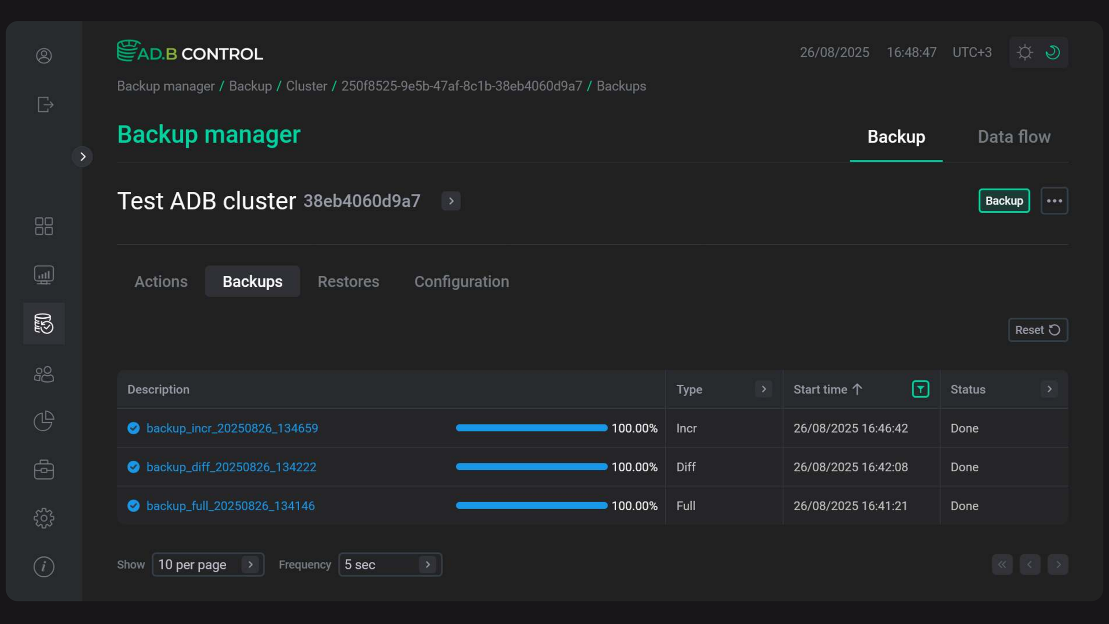
Regardless of how backups are created (manually or on a schedule), all of them are displayed on the ***Backups*** tab. To open this tab, follow the steps:

1. Open the Backup manager page via the ADB Control web interface.
2. In the table that is located in the ***Clusters*** section of the opened page, click the ADB cluster name.



Select the ADB cluster

3. Select the ***Backups*** tab.

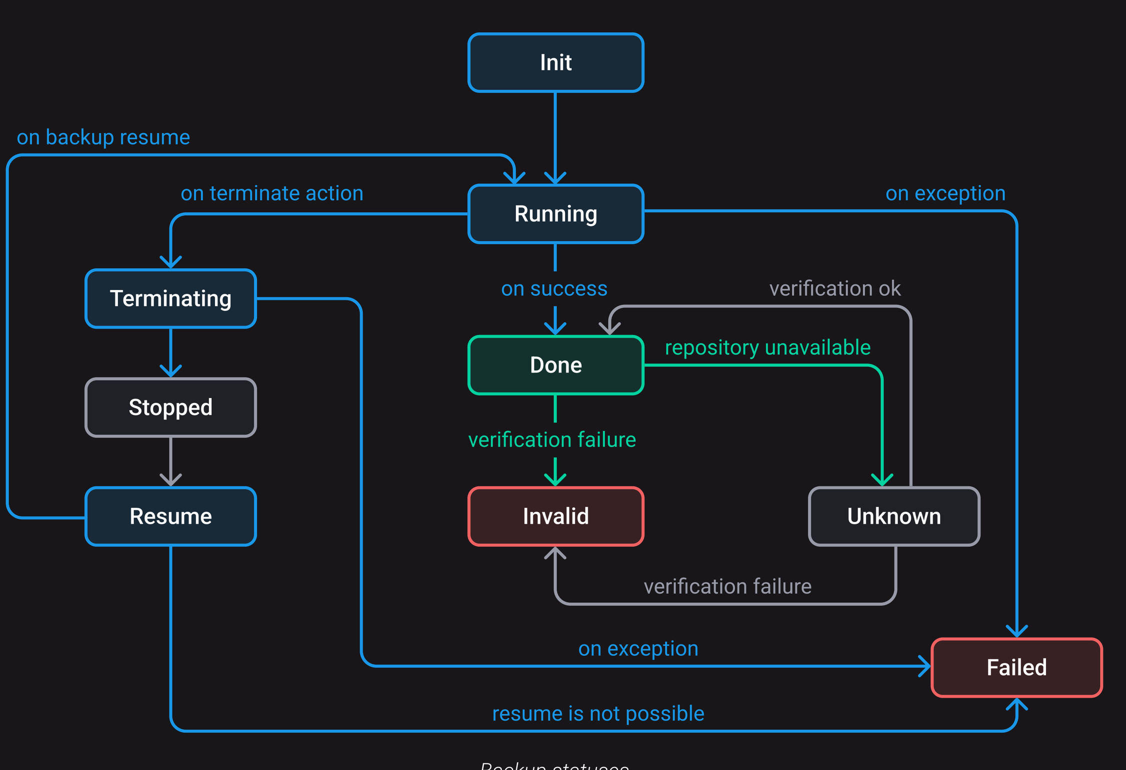


The Backups tab

On the page that opens, there is a table with the following information on backups.

Backup list fields

Field	Description
Description	A backup name. Filled in automatically when performing a backup according to the `backup_<type>_<start_date>_<start_time>` template, where: ▪ `<type>` — backup type (see ***Type*** below); ▪ `<start_date>` — backup start date in the `YYYYMMDD` format; ▪ `<start_time>` — backup start time in the `HHmmss` format. Starting with ADBM 2.4.0, there is a progress bar next to a backup name, which displays the backup execution progress (as a percentage of the estimated duration for all cluster segments). Note that backup progress is calculated only for the physical cluster reservation and does not include other actions
Type	The initial backup type (which was selected when creating a backup). Possible values: ▪ `Full` — full backup; ▪ `Diff` — differential backup; ▪ `Incr` — incremental backup.
Start time	A backup start timestamp in the `DD/MM/YYYY HH:mm:ss` format
Status	A backup status. Possible values: ▪ `Init` — the backup is being initialized. ▪ `Running` — the backup is running. ▪ `Terminating` — the backup is being terminated. ▪ `Invalid` — after verification, the backup is considered incorrect. Such backups will be automatically dropped when the next cleanup procedure starts. ▪ `Stopped` — the backup launch is stopped (after successful termination). ▪ `Resume` — the backup is being resumed. ▪ `Failed` — the backup launch is failed. ▪ `Done` — the backup launch is successful. ▪ `Unknown` — a repository that is specified in the backup configuration is inaccessible. For more information on possible transitions between statuses, see the diagram below



Backup statuses

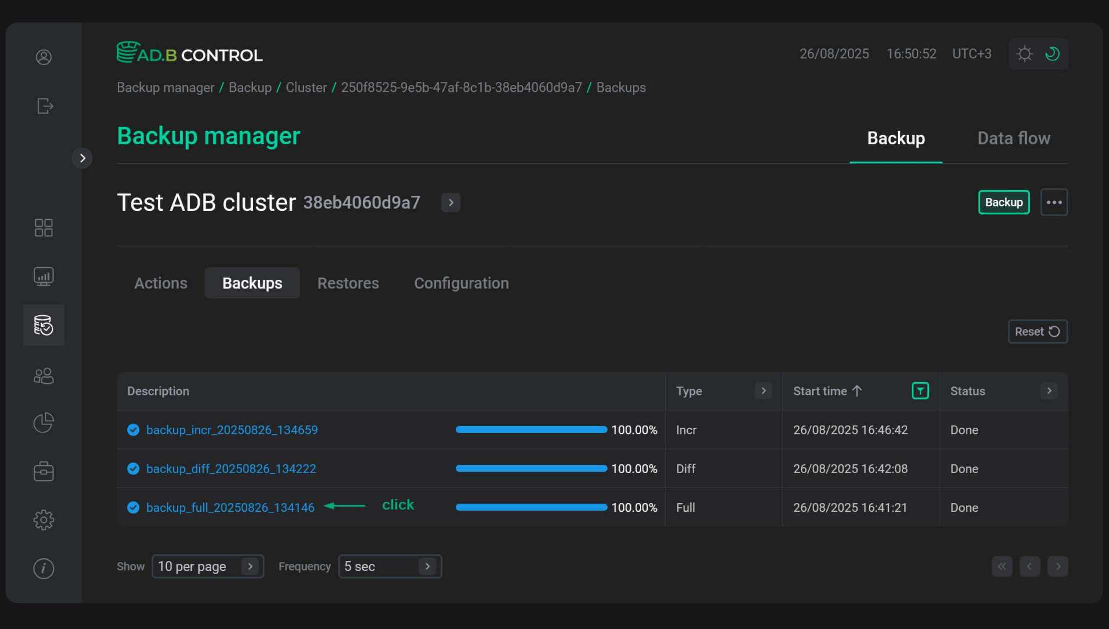
In the column headers of the table with a list of backups, there are filters that you can use to select specific data. To open a filter, click the > icon. For those columns where the set of possible values is limited (e.g. ***Type***), you can select a value from the drop-down list. For columns that show date and time (e.g. ***Start time***), the time range can be selected from the calendar.

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

> **IMPORTANT**
>
> By default, the table on the ***Backups*** tabs shows the current-day backups. To view all backups, use the time filters.

Backup details

To view the backup details, click the backup description (***Description***) in the table on the ***Backups*** tab.

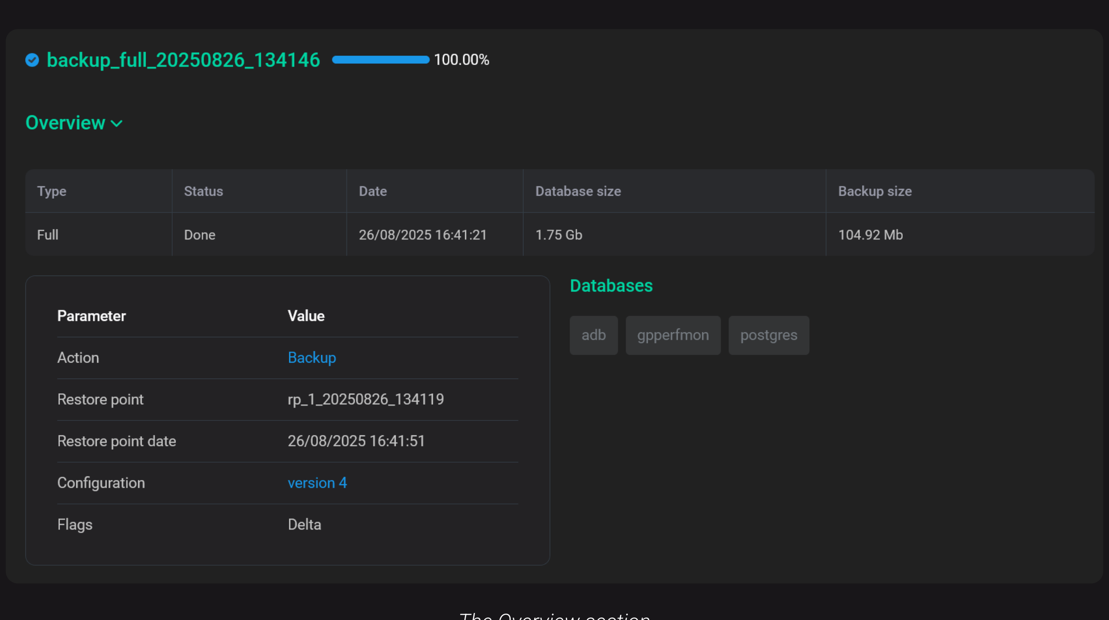


Select a backup

The next page contains the following sections:

- ***Overview***
- ***Topology***
- ***Statuses***

The ***Overview*** section shows the general backup information. At the top of the section, you can see the backup name. Starting with ADBM 2.4.0, there is a progress bar next to the backup name, which displays the backup execution progress (as a percentage of the estimated duration for all cluster segments).

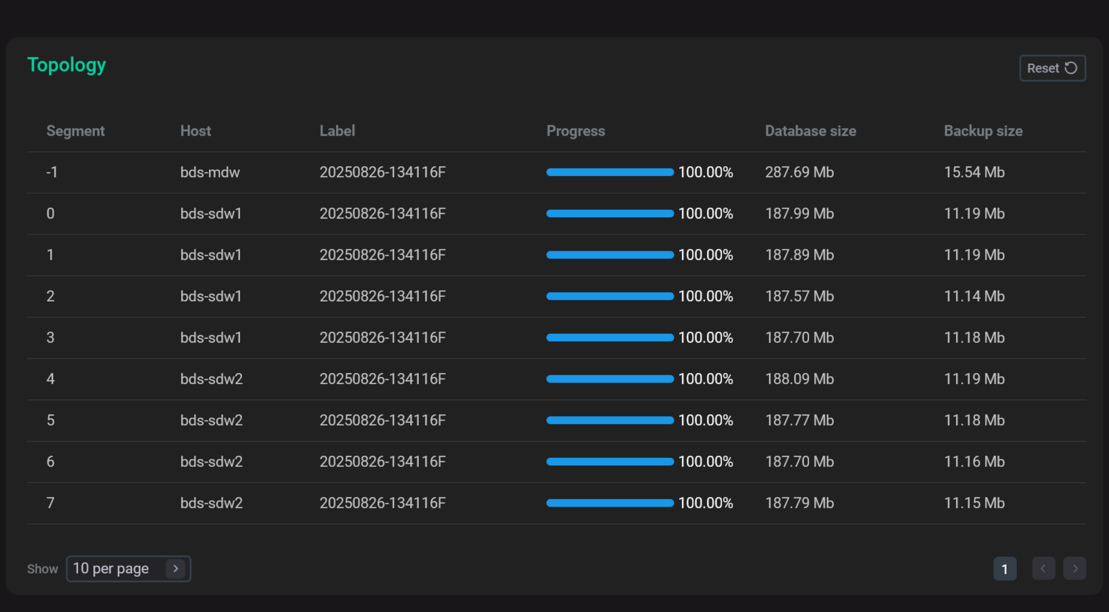


The Overview section

Fields of the Overview section

Field	Description
Type	A backup type. See ***Type*** above
Status	A backup status. See ***Status*** above
Date	A backup start timestamp in the `DD/MM/YYYY HH:mm:ss` format
Database size	The total volume of data files that have been backed up
Backup size	The total backup volume
Action	A name of the action within which the current backup was launched. You can click the name to view the detailed action information
Restore point	A name of the restore point created after the backup was completed
Restore point date	The restore point creation timestamp in the `DD/MM/YYYY HH:mm:ss` format
Configuration	A backup configuration version. To see the configuration details, click the version number
Flags	The flags that are selected for the backup operation (see ***Delta*** in the Backup description)
Databases	A list of databases that were in the DBMS at the time of the backup generation

The ***Topology*** section contains the backup topology information.

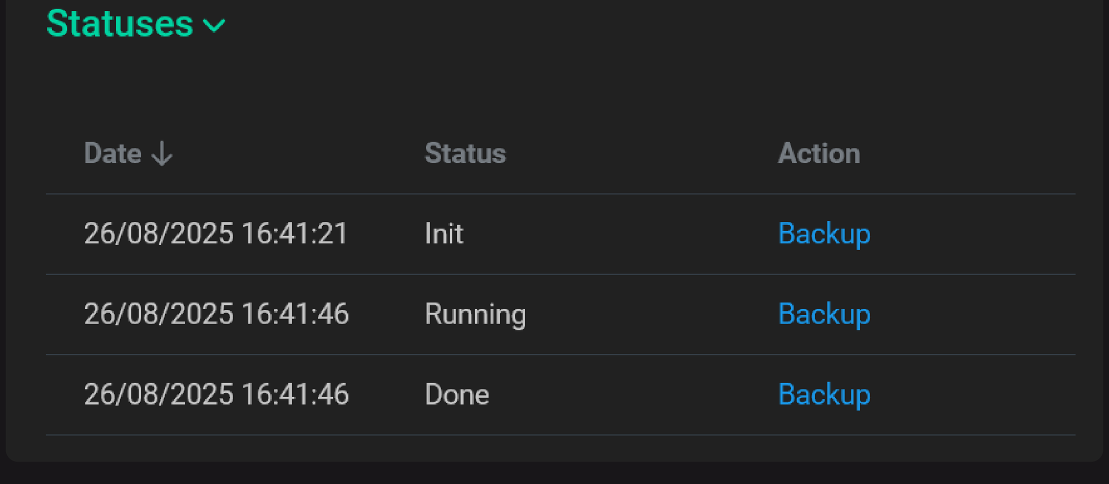


The Topology section

Fields of the Topology section

Field	Description
Segment	The content identifier of a segment (or master) instance. Corresponds to `gp_segment_configuration.content`
Host	A host name
Label	A backup name according to the internal logic of `pgbackrest`
Progress	Displays the backup execution progress for the current segment (as a percentage of the estimated duration). Available starting with ADBM 2.4.0
Database size	The full volume of uncompressed database data on a segment
Backup size	The full volume of compressed database data on a segment

The ***Statuses*** section displays a history of changing the backup status.



The Statuses section

Fields of the Statuses section

Field	Description
Date	A timestamp of the backup status change in the `DD/MM/YYYY HH:mm:ss` format
Status	A backup status. See ***Status*** above
Action	A name of the action during which the backup status was changed. To view the action details, click the name

View restores in ADBM

Daria Barysheva

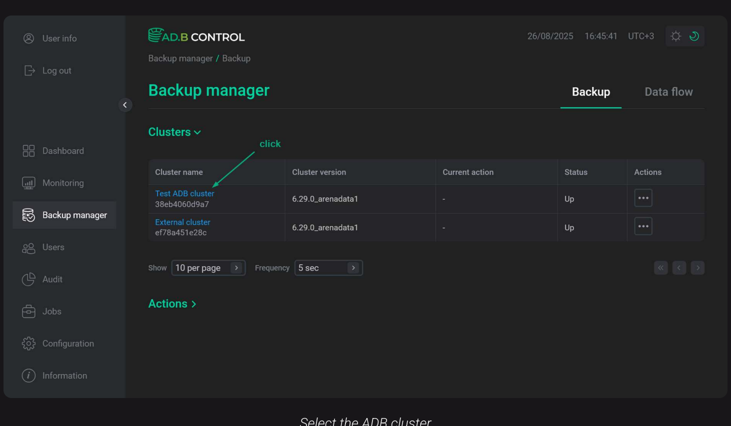
Contents

To Table of Contents

List of restores

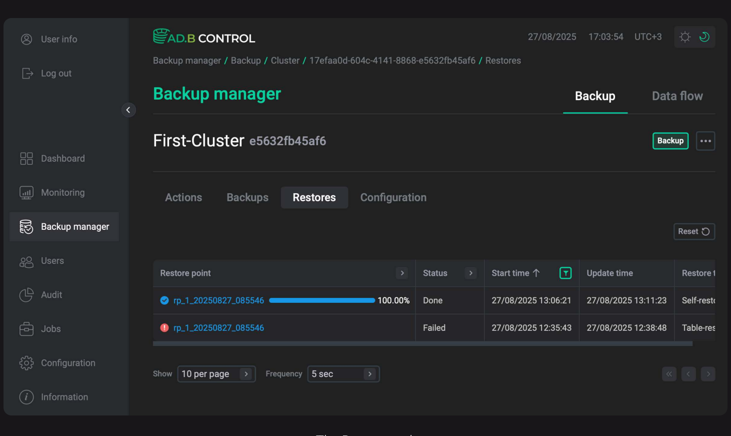
All restore operations that are launched by users in ADBM (via the Common restore, Partial restore actions or Disaster recovery) are displayed on the ***Restores*** tab. To open this tab, follow the steps:

1. Open the Backup manager page via the ADB Control web interface.
2. In the table that is located in the ***Clusters*** section of the opened page, click the ADB cluster name.



Select the ADB cluster

3. Select the ***Restores*** tab.



The Restores tab

On the page that opens, there is a table with the following information on the data restores.

Restore list fields

Field	Description
Restore point	A restore point that was selected to restore data. Starting with ADBM 2.5.0, there is a progress bar next to a restore point name, which displays the restore execution progress (as a percentage of the estimated duration for all cluster segments). This indicator is shown only for data restores with the `Self-restore` type
Status	A restore status. Possible values: ▪ `Init` — the restore is being initialized; ▪ `Running` — the restore is running; ▪ `Stopped` — the restore is stopped; ▪ `Failed` — the restore launch is failed; ▪ `Done` — the restore launch is successful.
Start time	A start timestamp of the restore operation in the `DD/MM/YYYY HH:mm:ss` format
Update time	A timestamp of the last restore status update in the `DD/MM/YYYY HH:mm:ss` format
Restore type	A data restore type. Possible values: ▪ `Self-restore` — a standard data restore operation that is launched via the Common restore action; ▪ `Table-restore` — a table-level data restore operation that is launched via the Partial restore action; ▪ `Copy` — a data restore operation that is launched during copying data between two clusters as a part of Disaster recovery with the `Copy` type (cold standby). ▪ `Streaming` — a data restore operation that is launched during copying data between two clusters as a part of Disaster recovery with the `Streaming` type (warm standby).

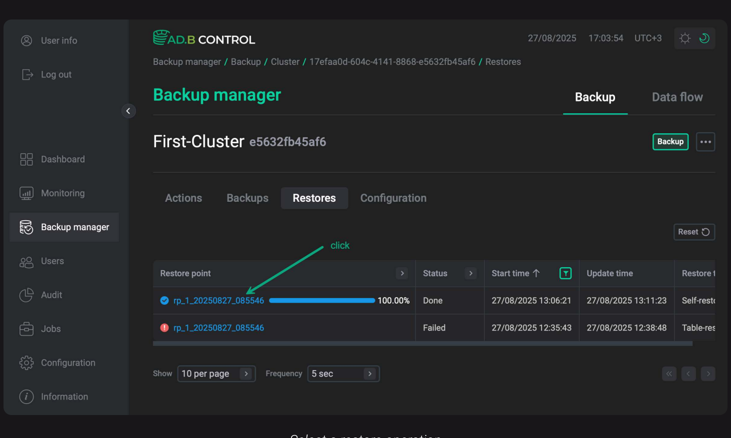
In the column headers of the table with a list of restore operations, there are filters that you can use to select specific data. To open a filter, click the > icon. For those columns where the set of possible values is limited (e.g. ***Status***), you can select a value from the drop-down list. For some columns (e.g. ***Restore point***), the search value should be entered. For columns that show date and time (e.g. ***Start time***), the time range can be selected from the calendar.

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

> **IMPORTANT**
>
> By default, the table on the ***Restores*** tab shows the current-day restore operations. To view all restores, use the time filters.

Restore details

To view the restore details, click the restore point name (***Restore point***) in the table on the ***Restores*** tab.



Select a restore operation

The next page displays the following information:

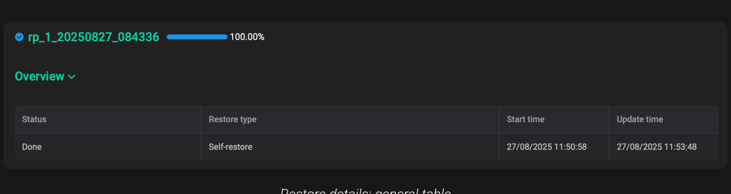
- The table with general restore information.
- The ***Parameters*** tab with additional information on parameters.
- The ***Tables*** tab with information on the restored tables.
- The ***Topology*** section with information on the cluster topology information.

> **NOTE**
>
> - The ***Tables*** tab is displayed only for the Partial restore actions starting with ADBM 2.2.2.
> - The ***Topology*** section is displayed starting with ADBM 2.5.0.

In the table, there are the following fields.

Fields of the table

Field	Description
Status	A restore status. See ***Status*** above
Restore type	A data restore type. See ***Restore type*** above
Start time	A start timestamp of the restore operation in the `DD/MM/YYYY HH:mm:ss` format
Update time	A timestamp of the last restore status update in the `DD/MM/YYYY HH:mm:ss` format

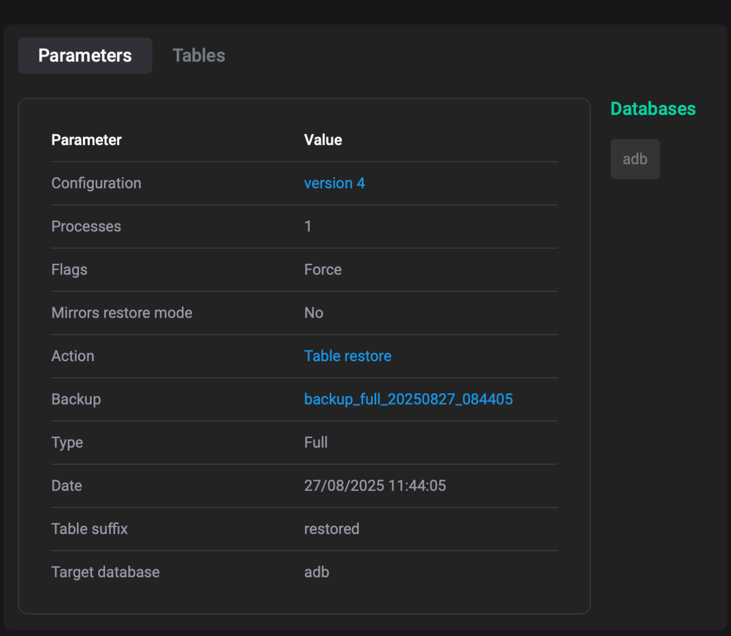


Restore details: general table

On the ***Parameters*** tab, there are the following fields.

Fields of the Parameters tab

Field	Description
Configuration	A configuration version. To view the configuration details, click the version number
Processes	A maximum number of processes that is selected to restore one segment
Flags	The flags that are selected for the restore operation (see ***Delta***, ***Force***, ***Verify*** in the Common restore action and data flow launch descriptions)
Mirrors restore mode	The restore mode selected for mirrors during the data restore configuration (see ***Restore mirrors*** in the Common restore action and data flow launch descriptions)
Action	A name of the action within which the restore operation was launched. To view the action details, click the name
Backup	A name of the latest backup that was created before the selected restore point. To view the backup details, click the name
Type	A type of the latest backup that was created before the selected restore point
Date	A creation timestamp of the latest backup that was completed before the selected restore point
Table suffix	The suffix that is added to all table names in the target database during the restore process. Displayed only for the Partial restore actions starting with ADBM 2.2.2
Target database	The target database where tables are restored during the restore process. Displayed only for the Partial restore actions starting with ADBM 2.2.2
Databases	A list of source databases selected for the data restore (from their backups)



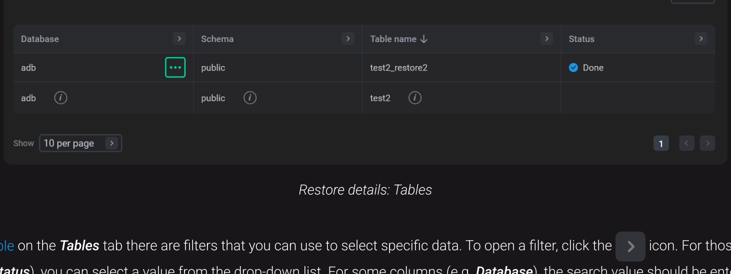
Restore details: Parameters

On the ***Tables*** tab, there are the following fields.

Fields of the Tables tab

Field	Description
Database	A name of the target database where the selected table is restored
Schema	A name of the target schema where the selected table is restored
Table name	A name of the restored table (after the restore operation)
Status	A table restore status. See ***Status*** above

> **TIP**
>
> If you click the ••• icon in the ***Database*** column, you can see the original names of the database, schema, and table from which the table data was restored — in the new row that appears.



Restore details: Tables

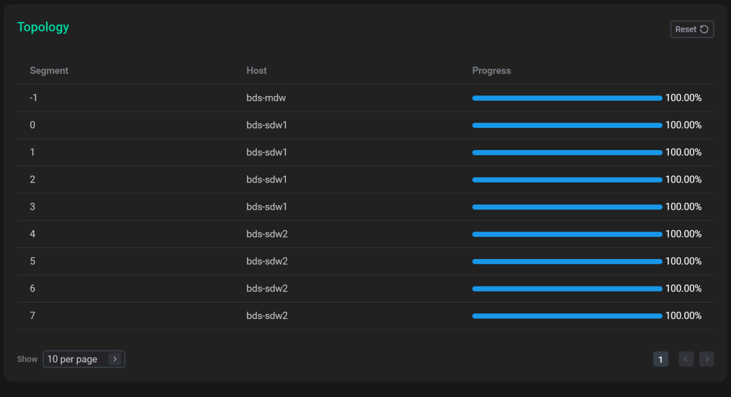
In the column headers of the table on the ***Tables*** tab there are filters that you can use to select specific data. To open a filter, click the > icon. For those columns where the set of possible values is limited (e.g. ***Status***), you can select a value from the drop-down list. For some columns (e.g. ***Database***), the search value should be entered.

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

In the ***Topology*** section, there are the following fields.

Fields of the Topology section

Field	Description
Segment	The content identifier of a segment (or master) instance. Corresponds to `gp_segment_configuration.content`
Host	A host name
Progress	Displays the restore execution progress for the current segment (as a percentage of the estimated duration). Available starting with ADBM 2.5.0 for data restores with the `Self-restore` type



Restore details: Topology

Disaster recovery

Daria Barysheva

Contents

To Table of Contents

Configure and run

To configure disaster recovery (DR) in ADBM, follow the steps:

1. Using the ADB Control web interface, add a new cluster to which data from the source ADB cluster will be copied. Next, the added cluster will be called **target** or **DR cluster**. Note that both ADB Control and ADBM services should be added to the target cluster via ADCM.
2. Open the ***Backup manager*** page in the ADB Control web interface. Ensure that the source and target clusters are available on the page in the `UP` status.

Both clusters are available on the Backup manager page

> **CAUTION**
> - To properly configure data copying, use two clusters with the same major version (the first two digits in the bundle number). For example, clusters with the `6.23.4` and `6.23.5` bundle versions can interact. But you cannot configure data copying between clusters with the `6.23.4` and `6.24.1` versions.
> - To restore data on a target ADB cluster from a backup created on another cluster, you need the same number of primary segments on both clusters. The rest of the cluster topology settings (such as mirroring, standby, number of host segments) do not affect the restore option.
> - The operating system versions and families should be compatible on the source and target cluster hosts.

3. Add backup configurations for both clusters. Note that the source ***Storage*** section (which is defined for a source cluster) should describe the storage that is available for both clusters (for example, S3).

Storage section for a source cluster

Example of filling the Storage section for a source cluster

Storage section for a target cluster

Example of filling the Storage section for a target cluster

4. To verify the successful result of data copy in the future, check which data is currently available in both clusters.

Check a source cluster

- List tables of the `adb` database via the `\dt` `psql` command:

```
\dt
```

Result:

```
                    List of relations
 Schema |      Name       | Type  |  Owner  |       Storage
--------+-----------------+-------+---------+----------------------
 public | best_books      | table | gpadmin | heap
 public | book_type       | table | gpadmin | heap
 public | spatial_ref_sys | table | gpadmin | heap
 public | test            | table | gpadmin | append only columnar
 public | test2           | table | gpadmin | append only columnar
(5 rows)
```

- View the `best_books` table data:

```
SELECT * FROM best_books;
```

Result:

```
 id |   author   |         title         | public_year
----+------------+-----------------------+------------
  1 | Harper Lee | To Kill a Mockingbird |        1960
(1 row)
```

Check a target cluster

View tables of the `adb` database via the `\dt` `psql` command:

```
\dt
```

Result:

```
                  List of relations
 Schema |      Name       | Type  |  Owner  | Storage
--------+-----------------+-------+---------+---------
 public | spatial_ref_sys | table | gpadmin | heap
(1 row)
```

5. On the ***Data flow → Configurations*** tab of the ***Backup manager*** page, click ***Create*** to add a new DR configuration.

Switch to adding a DR configuration

6. In the window that opens, fill in the following fields:
   - ***Source cluster*** — a source cluster name.
   - ***Target cluster*** — a target cluster name.
   - ***Autostart*** — a flag that indicates whether to create a data flow with the `Copy` restore type for a selected pair of clusters automatically each time a backup with the ***Backup type*** type is created in the source cluster.
   - ***Backup type*** — types of backups in the source cluster that will trigger copying data (data flow) to the target cluster. The field becomes available if the ***Autostart*** flag is set. Possible values:
     - `Full` — full backups;
     - `Diff` — differential backups;
     - `Incr` — incremental backups.

Add a DR configuration

7. Click ***Create***. As a result, the cluster pair is displayed on the ***Backup manager → Data flow → Configurations*** tab.

Configuration is added

8. Ensure that a source cluster has at least one restore point in ADBM. For this, a cluster should have a backup. Backups can be created either automatically according to the configured schedule or manually using the Backup action.

A Full backup exists for a source cluster

9. Click ***Create data flow*** on the ***Backup manager → Data flow → Data flows*** tab to switch to configuring a data flow for the selected cluster pair.

Switch to adding a data flow

10. In the window that opens, fill in the following fields:
    - ***Configuration*** — a configuration that you created earlier (which defines a pair of clusters).
    - ***Restore type*** — a restore type. Possible values:
      - `Copy` — cold standby. The target cluster data is recovered only once based on the selected restore point.
      - `Streaming` — warm standby. The target cluster data is constantly recovered based on the restore points created in the source cluster. After the replication process is terminated, the target cluster data will be actual at the moment of the latest restore point applied from the source cluster.

    > **NOTE**
    > Starting with ADBM 2.3.0, you can run a data flow with the `Copy` type, even if a target cluster is stopped (has the `Down` status). This allows you to restore data in a cluster if it has become inoperative and there are no backups for Common restore.

    - ***Restore point*** — the restore point at the moment of which you want to restore the source cluster databases in the target cluster. If the `Streaming` restore type is selected, the latest restore point is always used, and the ***Restore point*** field is disabled.
    - ***Databases*** — a list of the source cluster databases to restore in the target cluster.
    - ***Repository path*** — a path to the repository that stores the source cluster backups. The default value is extracted from the current backup configuration.
    - ***Processes*** — a maximum number of processes to restore one segment.
    - ***On restore points streaming failure*** — when the `Streaming` restore type is selected, this field determines the behavior of the system in case of errors during the application of restore points to the target cluster (for example, damage of WAL archives, unplanned data flow interruption, storage unavailability, etc.). The field is available starting with ADBM 2.4.0. Possible values:
      - `Pause` — transfer the data flow to the `Paused` status for its temporary suspension. In this status, three actions become available for the data flow:
        - Resume — resumes the application of restore points to the target cluster from the state when processing failed last time; and returns the data flow status to `Streaming`. This action checks whether the application of changes can be resumed without ***Restore*** (which significantly reduces the time needed to return to `Streaming` status). If it is impossible (for example, if an archiving error has been detected), then ***Resume*** will follow the logic of the ***Restart*** action.
        - ***Restart*** — resumes the application of restore points to the target cluster from the state when processing failed last time; and returns the data flow status to `Streaming`. Unlike ***Resume***, which requires manual corrections to be made beforehand, ***Restart*** re-initiates the streaming process from the failure point without prior intervention.
        - Cancel — reverts the target cluster state to the previous restore point (preceding the one whose processing failed) and finishes the data flow (the data flow acquires the `Cancelled` status).

        This mode is suitable when you need to keep the target cluster state as close to the current state of the source cluster as possible and do not require unlocking the target cluster (for running DB requests) while making corrections.
      - `Bring online` — revert the target cluster state to the previous restore point (preceding the one whose processing failed), restore data, and finish the data flow (by unlocking the target cluster for running database queries and changing the data flow status to `Restored`).
    - ***Restore mirrors*** — the drop-down list, which value indicates whether to restore data for mirrors and standby if they exist in the cluster.
      - `No` — do not restore mirrors.
      - `Parallel` — restore mirrors concurrently with primary segments.
      - `Sequential` — restore mirrors after all primary segments are restored.
    - ***Flags***:
      - ***Delta*** — a flag that indicates the need to use checksums during the restore operation.
      - ***Force*** — a flag that indicates the need to completely overwrite data and tablespace paths in PostgreSQL.
      - ***Verify*** — a flag that indicates the need to verify backup data before data restore.

Data flow configuration with the Copy type

[illegible]

- Mirror segments
  - pg_hba.conf
  - pg_ident.conf
  - postgresql.conf
  - postgresql.auto.conf
  - internal.auto.conf
  - recovery.conf

DR configuration management

View a list of configurations

All cluster pairs between which DR is configured are displayed on the ***Data flow → Configurations*** tab of the ***Backup manager*** page.

DR configurations on the Backup manager → Data flow → Configurations tab

The tab displays the following information.

The "Backup manager → Data flow → Configurations" tab fields

Field	Description
Source cluster	The source cluster from which the data is to be copied
Target cluster	The target cluster to which the data is to be copied
Username	A login of the user who created the current configuration via the ADB Control web interface
Data flow ID	A unique identifier of the latest data flow that was launched for a pair of clusters defined in the current configuration
Status	A status of the ***Data flow ID*** data flow. See Status below
Restore type	The restore type that was selected during the data flow configuration

Above the table with DR configurations, there are filters that you can use to select specific data. Available filters are listed below.

- ***Source cluster*** — filter by the source cluster name (see ***Source cluster*** above). Select a value from the drop-down list.
- ***Target cluster*** — filter by the target cluster name (see ***Target cluster*** above). Select a value from the drop-down list.
- ***Username*** — filter by the user name (see ***Username*** above). Enter a full value.

Filters on the Backup manager → Data flow → Configurations tab

Edit a configuration

To edit a DR configuration, follow the steps:

1. Click the icon in the ***Actions*** column on the ***Data flow → Configurations*** tab of the ***Backup manager*** page.

Switch to editing a configuration

2. In the window that opens, edit necessary fields. The ***Source cluster*** and ***Target cluster*** values cannot be changed.

Edit a configuration

3. Click ***Save***.

Delete a configuration

> **CAUTION**
> Do not delete a configuration of the currently running data flow.

To delete an existing configuration, follow the steps:

1. Click the icon in the ***Actions*** column on the ***Data flow → Configurations*** tab of the ***Backup manager*** page.

Switch to deleting a configuration

2. In the window that opens, confirm the operation by clicking ***Delete***.

Confirm the operation

As a result, the configuration data is removed from the system and no longer displayed on the ***Backup manager → Data flow → Configurations*** tab.

Configuration is removed

Data flow management

View active data flows

All active data flows that are created to copy data between two clusters are displayed on the ***Data flow → Data flows*** tab of the ***Backup manager*** page. Active data flows have the statuses different from `Impossible`, `Down`, and `Restored` (the ***Data flow → History*** tab is designed to show data flows in these three statuses).

The Backup manager → Data flow → Data flows tab

The tab displays the following information.

The "Backup manager → Data flow → Data flows" tab fields

Field	Description
Data flow ID	A unique identifier of the data flow
Source cluster	The source cluster from which the data is to be copied within the current data flow
Target cluster	The target cluster to which the data is to be copied within the current data flow
Username	A login of the user who created the current data flow via the ADB Control web interface
Status	A data flow status. Possible values: • `Init` — the data flow is being initialized; • `Ready` — the data flow is ready for data restore; • `Paused` — the data flow with the `Streaming` type is temporarily stopped (see ***On restore points streaming failure*** above); • `Copying` — the backup is being copied; • `Restoring` — the restore operation is running; • `Streaming` — the data flow with the `Streaming` type is running; • `Terminating` — the data flow launch is being terminated manually or automatically (see the note above). For more information on possible transitions between statuses, see the diagram below.
Start time	A data flow creation timestamp in the `DD/MM/YYYY HH:mm:ss` format
Restore type	The restore type that was selected during the data flow configuration
Replication lag	How much the target cluster lags from the source cluster. Calculated as the difference between the current time and the time when the latest restore point was applied to the target cluster during the streaming process. Available only for data flows with the `Streaming` type in the `Streaming` status

Data flow statuses

In the column headers of the table with a list of data flows, there are filters that you can use to select specific data. To open a filter, click the icon. For those columns where the set of possible values is limited (e.g. ***Status***), you can select a value from the drop-down list. For some columns (e.g. ***Username***), the search value should be entered. For columns that show date and time (e.g. ***Start time***), the time range can be selected from the calendar.

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

View finished data flows

The ***Backup manager → Data flow → History*** tab is used to display completed data flows, which have the following statuses:

- `Impossible` — the data flow launch is impossible.
- `Down` — the data flow launch is failed.
- `Restored` — the data flow launch is successful.
- `Cancelled` — the previously stopped data flow with the `Streaming` type is cancelled (see ***On restore points streaming failure*** above).

The content of the ***History*** tab is completely identical to the ***Data flows*** tab described above, except that the former does not contain the ***Replication lag*** column. Also, there are no available actions for data flows on the ***History*** tab.

> **IMPORTANT**
> By default, the table on the ***Backup manager → Data flow → History*** tab shows the current-day data flows. To view all data flows, use the time filters.

The Backup manager → Data flow → History tab

View the data flow details

To view the data flow details, click the data flow identifier (***Data flow ID***) in the table on the ***Backup manager → Data flow → Data flows*** or ***Backup manager → Data flow → History*** tab.

Select a data flow

The next page contains the following sections:

- ***Overview***
- ***Statuses***
- ***Restore points*** (shown only for the `Streaming` restore type)

The ***Overview*** section displays the general data flow information.

The Overview section

The Overview section fields

Field	Description
Source cluster	A source cluster name
Target cluster	A target cluster name
Username	A login of the user who created the current data flow via the ADB Control web interface
Date	A data flow creation timestamp in the `DD/MM/YYYY HH:mm:ss` format
Restore ID	A unique identifier of the restore operation in the target cluster. To view the restore details, click the identifier
Database	A list of databases that were selected during the data flow configuration
Restore type	The restore type that was selected during the data flow configuration
Processes	A number of processes that was selected during the data flow configuration
Flags	The flags that were selected during the data flow configuration (see the ***Delta***, ***Force***, ***Verify*** flags above)
Replication lag	How much the target cluster lags from the source cluster. Calculated as the difference between the current time and the time when the latest restore point was applied to the target cluster during the streaming process. Available only for data flows with the `Streaming` type in the `Streaming` status
Action name	A name of the action within which the data flow was launched. To view the action details, click the name
Restore point	An identifier of the restore point that was selected during the data flow configuration
Repository path	A repository path that was selected during the data flow configuration
Mirrors restore mode	The restore mode selected for mirrors during the data flow configuration (see ***Restore mirrors*** above)
Error handling mode	The error handling mode specified during the data flow configuration (see ***On restore points streaming failure*** above)

The ***Statuses*** section displays the history of changing the selected data flow status.

Statuses

Data flow history ID	Status	Date
19	Init	25/08/2025 20:48:45
20	Ready	25/08/2025 20:48:51
21	Copying	25/08/2025 20:48:55
22	Restored	25/08/2025 20:51:17

The Statuses section

The Statuses section fields

Field	Description
Data flow history ID	A unique identifier of the data flow status change event
Status	A data flow status at the time specified in the ***Date*** field
Date	A timestamp of the data flow status change event in the `DD/MM/YYYY HH:mm:ss` format

The ***Restore points*** section displays a list of restore points that have been applied to the target cluster during the current streaming process. The section is shown only for data flows with the `Streaming` type.

Restore points

Restore point name	Status	Time applied
rp_1_20250825_174755	Done	25/08/2025 21:03:37
rp_1_20250825_180610	Done	25/08/2025 21:11:31

Show 10 per page

The Restore points section

The Restore points section fields

Field	Description
Restore point name	A name that was assigned to the restore point in the source cluster
Status	A status of applying the restore point to the target cluster: • `Init` — initialized; • `Done` — finished successfully; • `Failed` — finished with an error.
Time applied	A timestamp when the restore point was applied to the target cluster (in the `DD/MM/YYYY HH:mm:ss` format)

Terminate a data flow

> **IMPORTANT**
> The ***Terminate*** action is available only for data flows with the `Streaming` type in the `Streaming` status.

To terminate a running data flow with the `Streaming` type, follow the steps:

1. Apply the ***Terminate*** action to the specified data flow on the ***Backup manager → Data flow → Data flows*** tab. To open a drop-down list with possible actions, click `...` in the ***Action*** column.

Go to terminating the data flow

2. In the window that opens, confirm the action by clicking ***Run***.

Confirm the action

3. As a result, the ***Terminate stream*** action starts for the target cluster. This action, in turn, generates several subactions. You can see all of them on the ***Actions*** tab (for more details, see View actions in ADBM).
   During the termination process:
   - A status of the previously created action ***Streaming cluster*** is changed to `Done`.
   - A status of the restore operation for the target cluster is changed to `Done`.
   - A status of the selected data flow is changed as follows: `Streaming` → `Terminating` → `Restoring` → `Restored`.
   - The selected data flow is moved from the ***Backup manager → Data flow → Data flows*** tab to the ***Backup manager → Data flow → History*** tab.

Resume a data flow

> **IMPORTANT**
> The ***Resume*** action is available only for data flows with the `Streaming` type in the `Paused` status starting with ADBM 2.4.0.

To resume a stopped data flow with the `Streaming` type (see ***On restore points streaming failure*** above), follow the steps:

1. Apply the ***Resume*** action to the specified data flow on the ***Backup manager → Data flow → Data flows*** tab. To open a drop-down list with possible actions, click `...` in the ***Action*** column.

Go to resuming a data flow

2. In the window that opens, you can change the data flow error handling mode by selecting one of the values in the ***Handler*** field:
   - `No operations` — leave the `Pause` data flow handler without any changes (see ***On restore points streaming failure*** above).
   - `Set stream handler to Bring Online` — change the data flow error handling mode from `Pause` to `Bring online` (see ***On restore points streaming failure*** above).
   - `Immediate recovery` — terminate the current data flow immediately and restore the target cluster data at the moment of the last valid restore point.

   After you select the ***Handler*** value, confirm the action by clicking ***Run***.

Confirm the action

3. As a result, the ***Resume Stream*** action starts for the target cluster. This action, in turn, generates several subactions. You can see all of them on the ***Actions*** tab (for more details, see View actions in ADBM).
   If the data flow resumes successfully, it acquires the `Streaming` status. If you selected the target cluster restore in the action settings and it does not fail, the data flow status changes to `Restored`.

Cancel a data flow

> **IMPORTANT**
> The ***Cancel*** action is available only for data flows with the `Streaming` type in the `Paused` status starting with ADBM 2.4.0.

To cancel a stopped data flow with the `Streaming` type (see ***On restore points streaming failure*** above), follow the steps:

1. Apply the ***Cancel*** action to the specified data flow on the ***Backup manager → Data flow → Data flows*** tab. To open a drop-down list with possible actions, click `...` in the ***Action*** column.

Go to cancelling a data flow

2. In the window that opens, confirm the action by clicking ***Run***.

Confirm the action

3. As a result, the ***Cancel Stream*** action starts for the target cluster. This action, in turn, generates several subactions. You can see all of them on the ***Actions*** tab (for more details, see View actions in ADBM).
   After all actions are successfully finished, the data flow status changes to `Cancelled`.

View actions in ADBM

Daria Barysheva

Contents

To Table of Contents

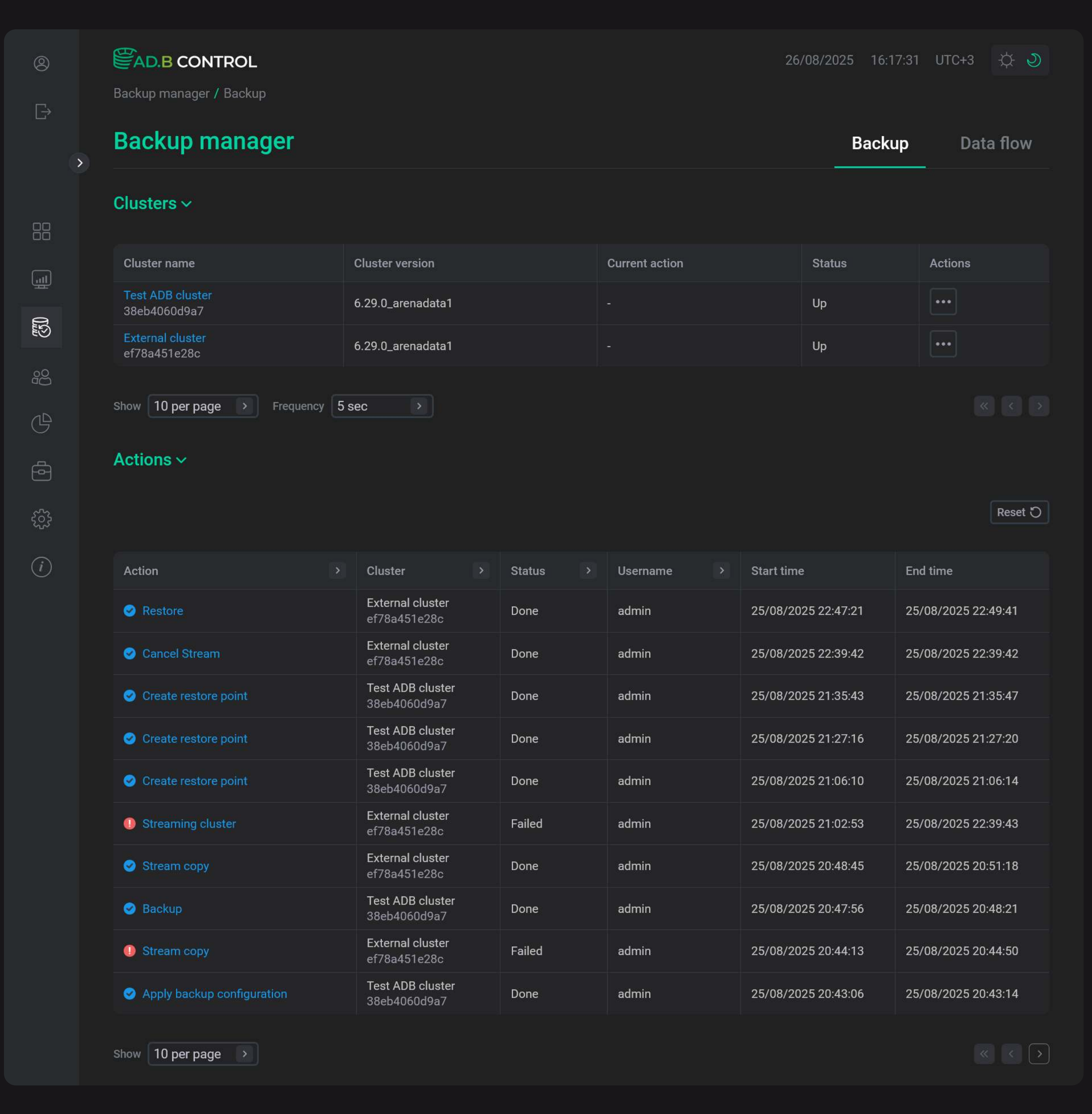
In ADBM, an **action** means any API call, SQL request, or internal procedure that starts when backup configurations or operations related to backups and DR are being applied.

Each action can trigger several other related actions — **subactions**.

List of actions

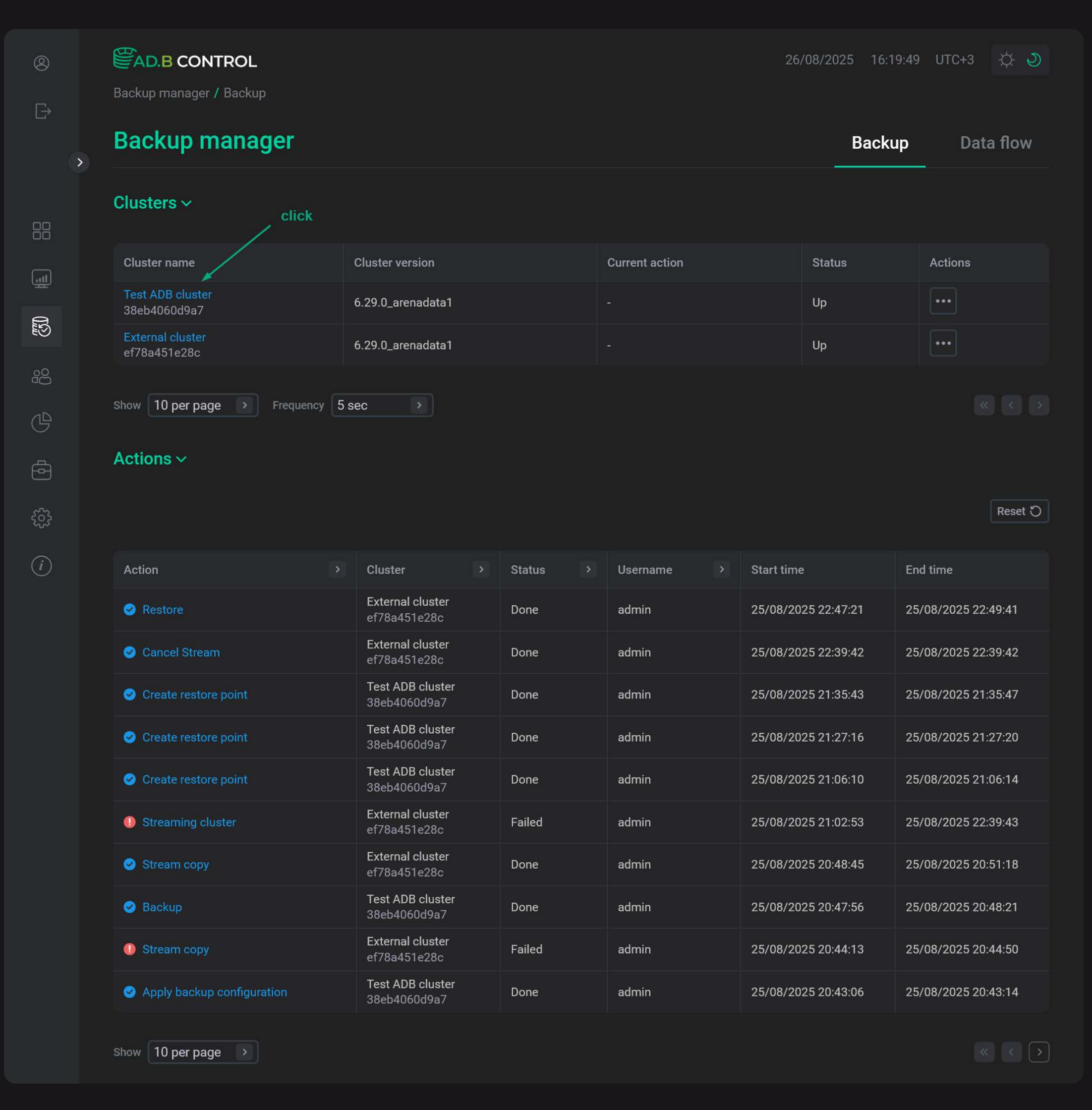
To view a list of actions, follow the steps:

1. Open the Backup manager page via the ADB Control web interface. The ***Actions*** section of the next page contains a table with general information on actions for all available ADB clusters.



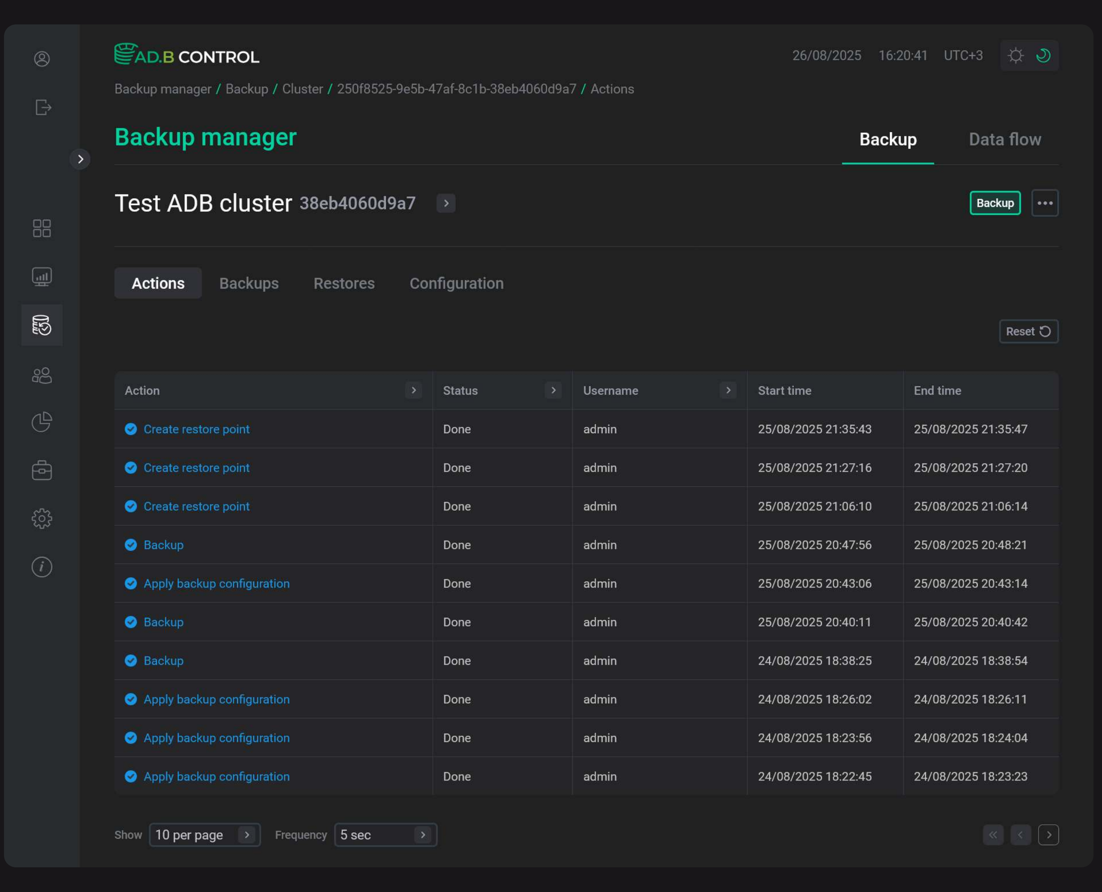
The Actions section

2. If you want to display actions for a specified cluster, click the cluster name in the table that is located in the ***Clusters*** section of the opened page.



Select the ADB cluster

The ***Actions*** tab of the opened page contains a table with general information on actions for the selected cluster.



The Actions tab

Both tables mentioned above contain the following information on actions.

Action list fields

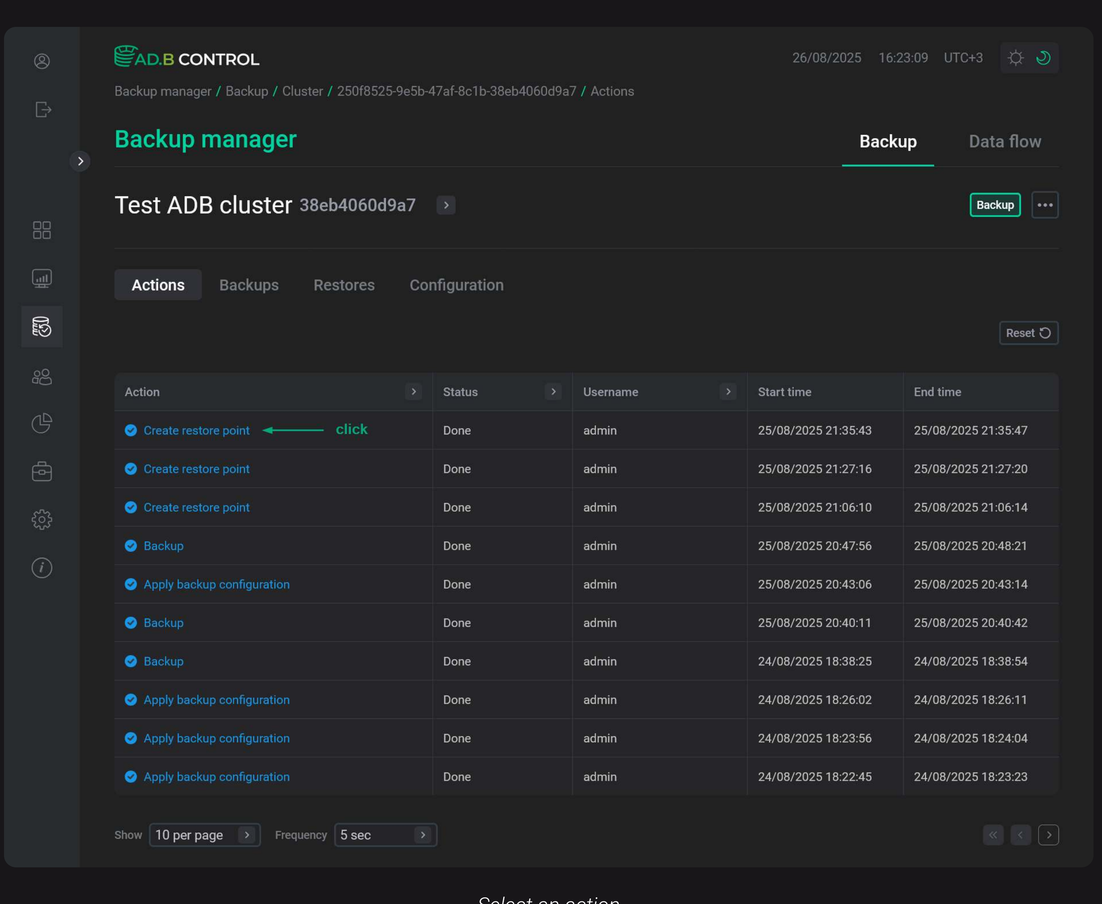
Field	Description
Action	An action full name
Cluster	A name of the cluster on which the selected action was executed. Displayed only if no specific cluster is selected on the ***Backup manager*** page
Status	An action status. Possible values: ▪ `Started` — the action is being started. ▪ `Stopped` — the action is stopped. ▪ `Done` — the action is completed successfully. ▪ `Failed` — the action is failed. ▪ `Terminating` — the action is being terminated. ▪ `Awaiting` — the action is waiting for something. ▪ `Unknown` — the action status is unknown. ▪ `Warning` — something went wrong during the action, but it is not critical for the parent action flow.
Username	The user who performed the selected action
Start time	An action start timestamp in the `DD/MM/YYYY HH:mm:ss` format
End time	An action end timestamp in the `DD/MM/YYYY HH:mm:ss` format

In the column headers of the tables that are located in the Actions section and on the Actions tab, there are filters that you can use to select specific data. To open a filter, click the icon. For those columns where the set of possible values is limited (e.g. ***Action***), you can select a value from the drop-down list. For some columns (e.g. ***Username***), the search value should be entered.

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

Action details

To view the action details, click the action name (***Action***) in the table located in the Actions section or on the Actions tab.

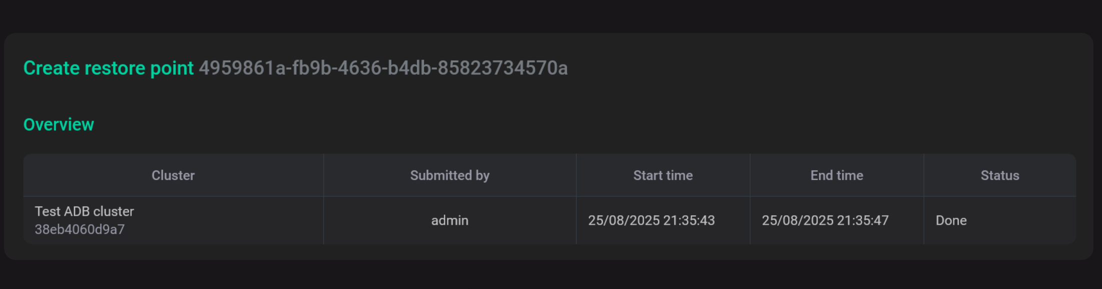


Select an action

The next page contains the following sections:

- ***Overview***
- ***Subactions***

The ***Overview*** section displays the action name, identifier, and general information.

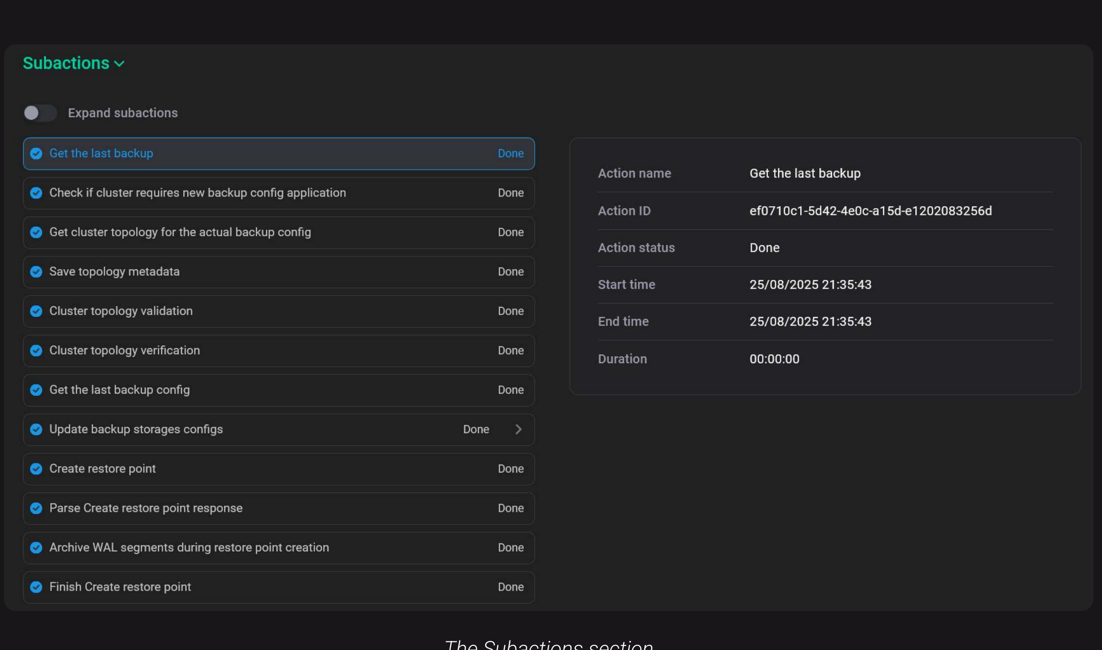


The Overview section

Fields of the Overview section

Field	Description
Cluster	A cluster name (see ***Cluster*** above)
Submitted by	The user who submitted the selected action
Start time	An action start timestamp in the `DD/MM/YYYY HH:mm:ss` format
End time	An action end timestamp in the `DD/MM/YYYY HH:mm:ss` format
Status	An action status (see ***Status*** above)

The ***Subactions*** section displays a list of subactions that are triggered by the top-level action. All subactions are ordered by the start timestamp — from the first action to the last one. For each action in the list, the action name and status are displayed.



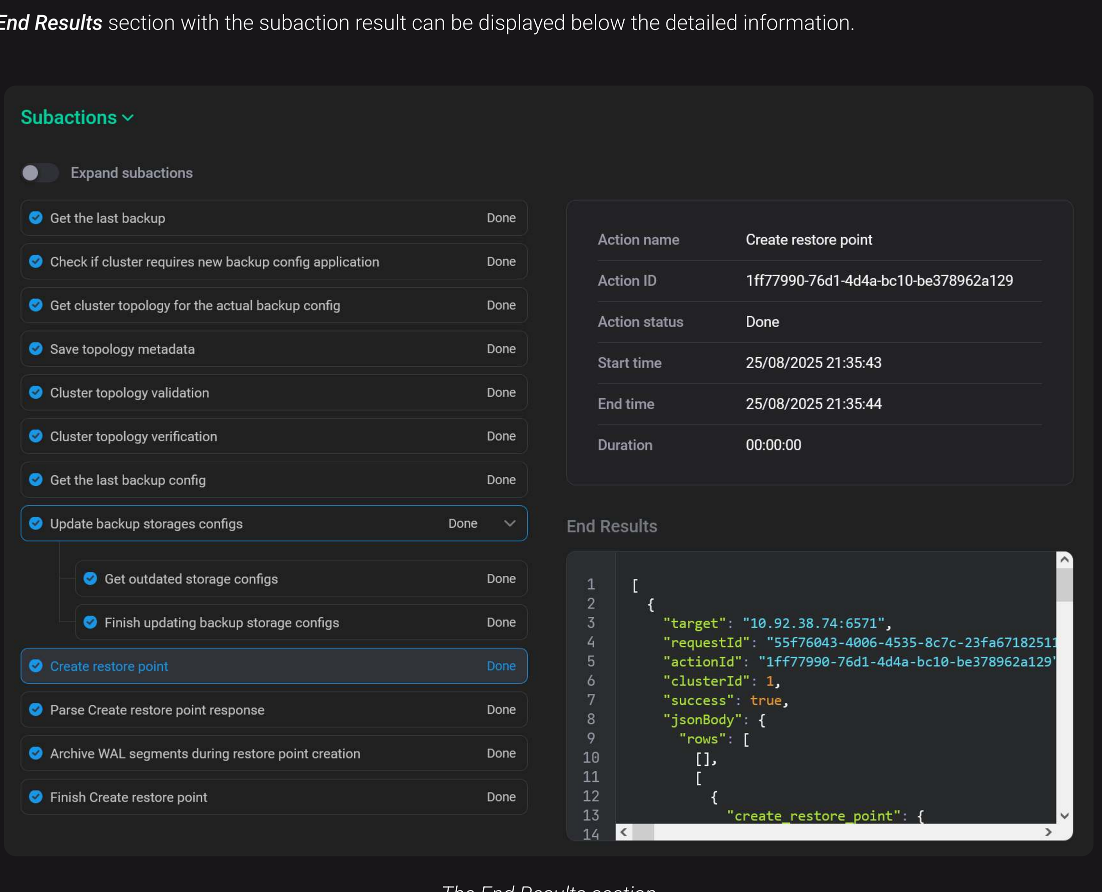
The Subactions section

When you click any subaction in the list, the detailed information on it is displayed in the right part of the screen.

Subaction details

Field	Description
Action name	A subaction name
Action ID	A subaction identifier
Action status	A subaction status (see ***Status*** above)
Start time	A subaction start timestamp in the `DD/MM/YYYY HH:mm:ss` format
End time	A subaction end timestamp in the `DD/MM/YYYY HH:mm:ss` format
Duration	A subaction duration in hours, minutes, and seconds

For some low-level actions, the ***End Results*** section with the subaction result can be displayed below the detailed information.



The End Results section

> **TIP**
>
> You can copy the result by clicking `Copy` in the top right corner of the code sample. This functionality can be useful for debugging errors.

Configuration parameters

Anton Vasilev

AD.B ES

Contents

To Table of Contents

This article describes parameters that can be configured for ADB ES services via ADCM.

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

AD Eureka

Parameter	Description	Default value
Xmx ram parameter	A maximum Java heap size for the AD Eureka component (in GB)	2
Log level	Logging level for the AD Eureka component. Possible values: `TRACE`, `DEBUG`, `INFO`, `WARN`, and `ERROR`	INFO
Registry app parameters	Read-only configuration of the AD Eureka component in YAML format. The content can be used as a reference if you need to modify AD Eureka properties (see below)	—
AD eureka server app environment key-value	This section lets you customize the behavior of the AD Eureka component using environment variables. Click ***Add property*** and enter the name of a variable and its value. Use the ***Registry app parameters*** read-only field to get the names of available variables	—

ADB Control

UI component parameters

Parameter	Description	Default value
ADB user name	ADB Control system user. A role with the same name is also created in ADB when you install the ***ADBC agents*** service in an ADB cluster. The name can only be changed before the ***ADB Control*** service is installed	adcc
Service password	The password for the ADB Control system user	—
Listening port	A port on which ADB Control UI is available via HTTP	8890
Xmx ram parameter	A maximum Java heap size for the ADB Control UI component (in GB)	8
Log level	Logging level for the ADB Control UI component. Possible values: `TRACE`, `DEBUG`, `INFO`, `WARN`, and `ERROR`	INFO
MIDDLE_PLACEHOLDER
	web.console.libraries	Path to console management libraries	/usr/share/admprom/prometheus/console_libraries
	web.console.templates	Path to Prometheus server console templates	/usr/share/admprom/prometheus/consoles
	storage.tsdb.retention.time	Specifies how long to retain samples in the storage. Supported units: `y`, `w`, `d`, `h`, `m`, `s`, `ms`	15d

Grafana settings

Parameter	Description	Default value
Grafana administrator’s password	Password for the `admin` user to access the Grafana web interface	—
Grafana listen port	Port to access the Grafana web interface	11210

Node Exporter settings

Parameter	Description	Default value
Listen port	Port to access ADB ES host system metrics in the Prometheus format	11203
Metrics endpoint	Endpoint to access ADB ES host system metrics in the Prometheus format	/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 the 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

Process exporter settings

Parameter	Description	Default value
Listen port	Port to access Process Exporter metrics in the Prometheus format	9256

ADB ES releases

Daria Barysheva

Contents

To Table of Contents

1.0.0

Date: 31.03.2026

New features

- Implemented the first version of Arenadata DB Enterprise Services (ADB ES) — a set of Enterprise services that extend the Arenadata DB (ADB) functionality
- Monitoring is now based on Prometheus and Grafana
- Added support for Astra Linux 1.8.4 "Orel" and "Voronezh"
- Implemented ADB Control 4.16.0
- Implemented ADBM 2.10.0

Improvements

- SSL management is now handled through the ***Manage SSL*** action

Misc/Internal

- ADBM and ADB Control are now delivered as packages instead of containers
- Introduced Process Exporter and Node Exporter in ADB ES
- CentOS support has been discontinued
- The legacy monitoring service (based on Graphite) is no longer supported
- Removed Docker dependencies

Improvements

- Improved a process of filtering in agents for gathering system metrics
- Added an automated cluster archivation when a new cluster with the same JDBC URL is added
- Removed the **_Create connection_** button from the **_Configuration → Clusters_** page due to the existing feature of automated adding external clusters via ADCM
- Support for SSL/TLS in gRPC services was added

Bug fixes

- A user IP address was not displayed in the **_Host_** column on the **_Audit → Operations_** and **_Audit → Autorizations_** pages
- Fixed an absence of a record for a denied action attempt on the **_Audit → Operations_** page

4.4.0

Date: 31.08.2023

New features

- The custom timezone setting is now available on the **_Configuration_** tab
- New parameters for the metrics offloading job (`Export job`) were added

Improvements

- Changed the order of side menu items
- Deleting the last user with the `Owner` role is now prohibited

Bug fixes

- The **_Show_** and **_Frequency_** settings for jobs and actions lists were not saved after switching to another page
- Fixed the error of filtering by **_Job group_** on the **_Jobs → Audit_** page

4.3.3

Date: 10.08.2023

New features

- ADB is now available as a target database for exporting metrics from ADB Control

Improvements

- Filters for **_Object name_**, **_Username_**, **_Host_** columns on the Audit → Operations and Audit → Authorizations pages are now case insensitive and partial (allow you to search by the first characters entered)

Bug fixes

- Password for an offload database was shown on the **_Audit → Operations_** page
- Sorting by columns was not available on the **_Jobs_** page
- User role was displayed as a digit on the **_Audit → Operations_** page
- After selecting a cluster, agents for other clusters were displayed on the **_Information_** page
- It was impossible to switch to the next list page in the **_Resource groups → Overview_** section if there were more than 5 segments in the list

4.3.2

Date: 20.07.2023

Improvements

- New tables and filters were implemented

Bug fixes

- Error when changing the user role
- Error when launching the metrics offload job
- Execution time was not showed in plan nodes after the command finished
- Details for relation audit were not showing
- Audit operation for updating the resource group configuration was not registered

4.3.1

Date: 30.06.2023

New features

- Audit for unsuccessful ADB logon attempts was added
- Audit for ADB Control users management actions is available now
- Checking that a new password differs from the current one was implemented
- Added an archiving action for a cluster to stop monitoring of the selected cluster with the ability to activate it again
- Added filtering of queries shorter than a threshold on the adcc-agent side to improve agents stability
- SQL signature (`sql_id`) was added for every command

Improvements

- **_Start time_**/**_End time_** filters are now visible by default on **_Commands History_**, **_Transactions History_**, and **_Backups_** pages
- Status names formatting in filters and table columns is unified

Bug fixes

- No message was displayed when a connection was successfully created
- Toggle button for a table was enabled when there was no further information
- **_Reset_** button didn't reset sorting
- Displayed `NaN` for **_Avg load rate / sec_** on the ADCC agent information page
- Cluster information data didn't update immediately after switching the cluster on **_Actions_**, **_Backups_**, **_Restores_** tabs
- Black screen was displayed when switching to another cluster from restore details
- **_Backup details_** action row was not displayed
- Page path not changed when switching tabs
- Performance metrics list for commands/transactions had duplicates

4.2.1

Date: 27.04.2023

New features

- Arenadata DB Command Center (ADBCC) is renamed to Arenadata DB Control (ADB Control)
- Added the ability to export metrics to an external database for a long-term storage
- Added the CSRF (Cross-Site Request Forgery) protection support
- Added a forced password change at the first user authentication in the system
- Added the temporary blocking of the user account after several failed authentication attempts
- Added calculation of total monitoring metrics. For active commands and transactions, `cpu-total`, `read-total`, and `write-total` are calculated at the current time from the start of execution
- Added pagination for the **_Job monitoring_** page

Bug fixes

- Fixed checking the name of the user that is being created. Only latin characters, numbers, and special characters in the user name are available
- Fixed: no pop-up window when receiving an error
- Fixed: a query could become "frozen" in the unknown status after 14-16 seconds

Misc/Internal

- Decreased the log level in `gpadcc`

4.1.0

Date: 15.02.2023

New features

- Using ADBCC along with `gpperfmon` within the same ADB cluster is now supported
- LDAP search in several OUs during ADBCC authentication is now available
- Added the ability to sort by almost all columns on the **_Command online/Command history_** and **_Transaction online/Transaction history_** pages
- Added the ability to show all available filters on the **_Command online/Command history_** and **_Transaction online/Transaction history_** pages

Improvements

- Added the ability to edit the `JDBC URL` and the `Username` cluster connection settings after the cluster creation
- Added a pop-up window with information about the application of the resource group settings
- On the detailed transaction page in the table of commands belonging to the transaction, renamed the **_Workload_** column to **_Resource group_**
- Added the ability to specify the sorting direction on the **_Resource groups_** page
- Fixed breadcrumbs that are shown when viewing resource groups. Instead of cluster and group IDs, resource group names are now displayed

Bug fixes

- Sometimes the **_cpu skew_** metric obtained negative values
- The statistics calculation for an interval of more than an hour
- The calculation for the **_Total queue duration_** column on the **_Resource groups_** page
- Removed the ability to enter negative values for the **_Batch size_** and **_Expire duration_** settings on the **_Configuration_** page for a job policy
- The pagination error on the **_User_** page
- The calculation for the number of commands in the `running` status on the **_Command online_** page
- The search for a command in the text when there is a newline character in the query text

4.0.0

Date: 06.12.2022

New features

- Added a new user interface for ADBCC
- Added a new module for ADB binary backups management — Arenadata DB Backup Manager (ADBM)
- Added a new module responsible for performing scheduled tasks — `adcc-scheduler`
- Added a functionality for monitoring resource groups:
  - The page with existing resource groups.
  - The page with detailed information for the resource group including active commands of the group.
  - The ability to configure the resource group.

Improvements

- All datetime metrics are now stored in UTC
- For clusters being removed from the ADBCC configuration, uncompleted commands and transactions are not updated
- The `status`, `finishAt`, and system metrics (`readTotal`, `writeTotal`, and `cpuUsageTotal`) of the current active command are now updated immediately for a terminated transaction. The status of the command is set to `CANCELLED`

Bug fixes

- The calculation of `Spill Skew` on the **_Command monitoring_** and **_Command history_** pages
- The calculation of the node execution time for a command in **_PLAN & PROGRESS_**
- The calculation of system metrics with negative values

3.7.0

Date: 20.10.2022

New features

- Added new performance metrics for commands and transactions: `Cpu usage total`, `Read bytes total`, `Write bytes total`
- Added the ability to repeatedly change a resource group for a transaction

Improvements

- Removed the ability to stop execution of a separate command by canceling or terminating it
- Renamed the **_Workload_** column to **_Resource Group_** on the **_Monitoring_** and **_History data_** pages for commands and transactions, as well as for detailed pages
- Removed the ability to change a resource group for a command on the **_Monitor Command_** page
- For commands and transactions, rows are now sorted by the **_Status_** column value
- Excluded the large bold font style for system metrics on detailed pages of commands and transactions

Bug fixes

- Corrected the data sorting on the **_Audit_** page based on numbers, symbols, letters
- It was impossible to enter data using the keyboard into the **_Start Time_** filter on the **_Monitor/Transaction_** pages
- The error with appearance of multiple filters by cluster and by database, if a filter by database was being added from the filter list
- Fixed the mapping of memory dimension from MiB to Mb
- Fixed calculating the number of tuples affected by the request without updating the page

Misc/Internal

- Included the Planchecker database objects into migration

3.6.0

Date: 30.08.2022

New features

- Added a new column **_Planner_**
- Added the ability to filter commands by a planner
- Added the ability to cancel a transaction on the **_Detailed transaction_** page
- Added a button to reset and delete all previously applied filters on the **_Monitor_** and **_History command_** pages
- Added the ability to reassign queries to another resource group
- Added a configuration for a Planchecker address

Improvements

- Changed the display of the `Spill Files (skew)`, `Row Skew` (in the **_Node_** panel), `Cpu Skew`, `Memory Skew`, `Read Skew`, `Write Skew` metrics to the uniform format: `<metric_name> Skew <metric_value>%`

Bug fixes

- The ADB unavailability error in case of the `OutOfMemoryError` error in ADBCC
- Fixed an actual tuples count for commands
- Fixed the incorrect display of the command status on the **_Detailed transaction_** page if the transaction was cancelled

Misc/Internal

- Added a Planchecker image to the ADBCC service (docker-compose) in the ADB bundle

3.5.1

Date: 21.07.2022

New features

- Processing of NodeStatusMetric messages is supported
- Added transaction pages: **_Transactions monitoring_**, **_Transactions history_**, page with detailed information about the transaction and a list of requests made within this transaction
- Added the ability to filter data by schema and relation on the **_Query history_** page
- Added a link to **_Transactions history_** in order to navigate to queries from the **_Audit_** page
- Added a link to the parent transaction in the **_Command details_** tab
- The SQL statement groups DDL, DML, DCL, and TCL are now displayed on the **_Monitor_** and **_History_** pages
- Added **_Spill_** and **_Spill Skew_** columns to the customization table on the **_Monitor/Command_** and **_History/Command_** pages

Improvements

- A non-blocking socket is now used to communicate with an agent

Bug fixes

- Long values of the **_Query ID_** column overlapped the next column with data on the **_History/Command_** page
- The incorrect sorting according to the schema with a heavy load on the **_Relation audit_** page
- When a user moved from the statistics to the list of commands on the **_Relation audit_** page, the command was being duplicated
- The missing tooltip for the **_Reset_** button
- The `NullPointerException` exception occurred during attempts to get system metrics
- Empty values were at the end of the list when the data was being sorted
- The **_queuedTime_** field displayed different values on the **_Monitor_** page and the **_Command details_** page
- The cache did not work properly, and the data was obtained directly from the database
- After the **_Cluster name_** column was added, the cluster value was not displayed in the table on the **_Monitor/Command_** and **_History/Command_** pages
- The `Duplicate key value violates unique constraint` error
- The bug with sorting on the **_Monitor_** and **_History_** pages when all the records became hidden

3.4.0

Date: 11.04.2022

New features

- Integrated the Avg cluster query metrics
- Implemented compression for huge queries
- Added a configuration for Registry Service to configure the addresses of backend servers

Improvements

- The actual statistics from `EXPLAIN ANALYZE` for finished queries is processed now
- The **_Plan&Progress_** and **_Text_** panels are hidden now if the content is empty

Bug fixes

- ADBCC exceeded the limit of connections to ADB
- Fixed the relation name display for a Dynamic Seq Scan node
- The color-coding for plan node statuses was broken
- The login error with cyrillic passwords

Misc/Internal

- Migrated the `adcc-extension` to the adbcc repository

3.3.1

Date: 09.03.2022

Bug fixes

- Filtering invalid metrics from `gpperfmon`

3.3.0

Date: 17.02.2022

New features

- Data audit support
- Background jobs history
- Support for virtual process memory in system metrics of commands
- Service load ratio support

Improvements

- Changed the sort order for metrics with `NULLS LAST` on the **_Query Monitor_**/**_History_** pages
- Reworked top menu

Misc/Internal

- Bumped log4j2 dependencies
- New system endpoints

3.2.5

Date: 20.12.2021

New features

- HTTPS support in ADBCC SSL
- The new `maxMessageKbSize` parameter in the `adcc-agent` configuration file

Bug fixes

- The agent stopped processing query metrics due to `RejectedExecutionException`

3.2.4

Date: 18.11.2021

New features

- Added the `maxMessageKbSize` setting to *agent.properties*

Improvements

- Added a message size check

Bug fixes

- Fixed re-sending of incorrect messages
- Fixed getting system metrics from ClickHouse

3.2.2

Date: 12.11.2021

New features

- Multi-clusters
- New system metrics: `CPU`, `RAM`, `IO`
- Adding columns dynamically to the **_History_** and **_Monitoring_** pages
- Information about background processes on the system status information page

Improvements

- Actualization for hanging queries
- adcc-extension: database and username are now shown in a plan message
- adcc-extension: lock polling is now disabled to avoid sending huge locks snapshots to agents

Misc/Internal

- Using a ClickHouse database for saving draft system metrics

3.1.3

Date: 07.09.2021

New features

- Added an agent build for Power8 LE platform (ppc64le arch)

3.1.0

Date: 17.08.2021

New features

- Time-based Retention Policy
- Fine-grained access to multiple databases for Advanced users
- Extended information on errors
- Column search filters and sorting on the **_Query Monitor_** and **_History_** pages

Improvements

- Information on ADBCC/ADB versions and documentation
- Scaling of the plan modal window
- UI updates

3.0.3

Date: 03.08.2021

New features

- New UI for Backend Server
- Ability to cancel and terminate queries on the **_Query Details_** and **_Monitoring_** pages
- Ability to view locks and blocks on the **_Query Details_** page
- Support of displaying information for the ADBCC agent
- LDAP authentication

Improvements

- Truncating of query plan details

ADB Control known issues

Daria Barysheva

To Table of Contents

Issue	From	Fixed	Workaround
Clusters that are connected to an external ADB Control instance should be upgraded along with the main cluster to ADB 6.29.0.3 in order for ADBM and ADB Control to work properly	4.14.2	—	Upgrade clusters to ADB 6.29.0.3

In AltLinux 8, the Docker engine update may break the *docker-compose* package, which causes the ADB Control start to fail	—	AltLinux fix	To fix */usr/lib/python3/site-packages/compose/service.py*, replace the following:
			`container.image_config['ContainerConfig'].get('Volumes') or {}`
			by:
			`container.image_config['Config'].get('Volumes') or {}`

The ***Reconfigure*** service action may fail for ADB Control hosted on CentOS in case of using external Query DB, since the CentOS repository does not have the required PostgreSQL version	4.13.0	—	Follow the steps: 1. Uninstall completely PostgreSQL 9 from the host with ADB Control. 2. Install PostgreSQL client of version 14 or higher. Do not install ADB Control on the ADB master host
After the ADB Control service is configured to use LDAP authentication with `FreeIPA` type, a list of groups is not loaded when you create a new mapping on the ***Users → LDAP group mapping*** page	4.0.0	—	Use basic authentication or LDAP with `MSAD` type (Microsoft Active Directory)
When a cluster is upgraded from 6.27.1.56 to 6.27.1.58, an error occurs if external ADB Control is used	4.10.3	4.10.3	Upgrade a cluster with ADB Control before upgrading clusters that use this ADB Control as external (in ***External ADB Control server parameters***)
After the transaction cancellation, plan nodes in the Plan & progress section will have the `executing` status and display the increasing execution time	4.8.8	4.9.1	Keep in mind that cancelled queries do not have executing processes
ADB Control 4.8.8 implements changes in API that affect communication between the main and external ADB Control services of different versions. When an ADB cluster with the main ADB Control is upgraded to 6.27.1.56.b1, clusters of older versions that use this ADB Control as external will not be able to perform ***Reconfigure***/***Start*** actions with the ADB Control service	4.8.8	4.8.8	In order to make all ADB Control service actions to work, upgrade clusters with the external ADB Control and versions older than 6.27.1.56.b1 to 6.27.1.56.b1
After ADB Control upgrade to 4.3.2, the `Export job` settings (that are defined on the ***Configuration → Job policy → Metrics offload*** page) are reset to defaults	4.3.2	4.3.2	Fill in and save the required settings. The problem does not affect future upgrades
Standby activation after switching from master fails on clusters with ADB Control/ADBM installed	4.3.1	4.3.2	Use external ADB Control/ADBM
In ADB Control 4.2.1, you cannot register new external ADB clusters with versions earlier than 6.23.3	4.2.1	4.2.1	Update ADB cluster to 6.23.3

Switching off sending metrics of inner queries to ADB Control may cause hangs of ADB queries	4.1.0	4.10.3	Turn on the GUC that enables sending metrics of inner queries in ADB Control or in ADB (as follows):
			`$ gpconfig -c adcc.monitor_inner_queries -v on`
			`$ gpstop -u`

Commands appear in the `Running` status on the ***Monitoring → Commands → Online*** page but after 14-16 seconds they freeze in the `Unknown` status (with no metrics updated). After the command completion, all metrics become available on the ***Monitoring → Commands → History*** page, but the `Unknown` status persists	4.1.0	4.2.0	1. Using `psql` or other SQL IDE, set the `adcc` user password via the following command: `ALTER USER adcc WITH ENCRYPTED PASSWORD '<password>';` 2. Open the ***Configuration → Clusters*** page in the ADCC UI and go to editing the cluster connection. 3. In the ***Password*** field, set the `<password>` value from step 1. Click ***Save***.

There is a potential memory leak with specific queries affected by the `gpperfmon` engine. If you use queries against huge tables with UDF that generate inner per-tuple queries, you may experience the significant RAM consumption, and even get OOM errors. The example is the `information_schema.columns` view, which can cause memory leak when being selected against a large system catalog	4.0.0	4.1.0	The possible solution is to switch off monitoring metrics (this operation requires cluster restart):
			1. Remove `gpadcc` from *shared_preload_libraries*, e.g.:
			`$ gpconfig -c shared_preload_libraries -v ""`
			2. Switch off the following parameters:
			`$ gpconfig -c gp_enable_gpperfmon -v off`
			`$ gpconfig -c adcc.explain_log_verbose -v off`
			`$ gpconfig -c adcc.explain_log_analyze -v off`
			3. Restart your cluster:
			`$ gpstop -a -M fast`
			`$ gpstart a`
			Remember that this workaround does not disable the ADCC UI and Arenadata DB Backup Manager

Backend ADCC server cannot connect to ADCC UI server. Data update may not always work on the details page	4.0.0	4.1.0	1. Connect to the ADCC server via SSH.
			2. Go to the */opt/adcc* directory.
			3. Edit the *docker-compose.yaml* file. For the *adcc_ui_server* and *adcc_backend_server* services, change the value of the `ADCC_EUREKA_CLIENT_PREFER_IP_ADDRESS` environment variable to `true`.
			4. Run the command and wait until all Docker containers stop:
			`$ docker-compose down`
			5. Run the following command:
			`$ docker-compose up -d`

It is impossible to configure the `Data cleanup` job on the ***Configuration*** page. In general, ADCC can work without data cleanup, but you need to monitor the metrics database size	4.0.0	4.1.0	1. Connect to the ADCC server via SSH.
			2. Go to the */opt/adcc* directory.
			3. Edit the *docker-compose.yaml* file. For the *adcc_scheduler* service, add the following environment variable: `ADCC_SCHEDULER_GRPC_PORT: 6572`.
			4. Run the command and wait until all Docker containers stop:
			`$ docker-compose down`
			5. Run the following command:
			`$ docker-compose up -d`

ADBM releases

Contents

2.10.0

Date: 31.03.2026

New features

- Introduced cluster groups
- Added an option to restart streaming (see the ***Restart*** action in the ***On restore points streaming failure*** description in Disaster recovery)

Improvements

- Improved stream resume behavior
- Allowed restore point creation during backup operations
- Added detection of segment timeline changes during backup to prevent invalid restore

Bug fixes

- Fixed restore of mirrors during ***Stream Copy*** after a previous stream failure

2.8.1

Date: 23.12.2025

Bug fixes

- Fixed an issue where table restore would fail on tables with a high number of partitions

2.8.0

Date: 28.11.2025

New features

- ADBM now supports ADB 7
- `pgbackrest` updated to version 2.54

Bug fixes

- Fixed an issue where the status was changed to `failed` when an error occurred
- A backup that failed after the ***Resume*** action now shows 100% progress
- Added timeline ID comparison before and after the backup

2.6.6

Date: 08.10.2025

Bug fixes

- Fixed a bug where the ***Cleanup*** action would delete valid backups

2.6.5

Date: 17.09.2025

New features

- Added audit of operations with clusters and storage configurations
- Added the Storages tab and modified backup configurations accordingly
- Added representation of storage space utilization and availability

Improvements

- Implemented compatibility verification for database and agent versions
- Added processing of a new `XLogRecord` type during the ***Partial restore*** action

Bug fixes

- Fixed: WAL archives for backup were not verified during the ***Verify*** action
- Fixed: agents were not displayed on the ***Backup manager*** page when ADB was stopped
- Fixed: backup progress was sometimes incorrectly updated

Misc/Internal

- Changed the name of the stream entity to data flow and modified naming in UI accordingly

2.5.7

Date: 12.08.2025

Improvements

- Changed errors in jobs to more informative ones

Bug fixes

- Storage access was checked only from the master, which could cause incorrect backup status acquisition
- After launching a stream of the `Copy` type with the parallel mirror restore option, the *pg_hba.conf* file was missing on target segments if the source segments had no mirrors

2.5.5

Date: 05.08.2025

Improvements

- Added a check of the corresponding WAL chain during the ***Verify*** action
- Added a check of the PostgreSQL backend availability during the table restore

Bug fixes

- Fixed filtering WAL for headers that were divided across two pages
- Fixed handling backup blocks during filtering WAL in `pgbackrest` to avoid `gpdb` assertion triggering during the ***Partial restore*** action
- In `pgbackrest`, fixed handling a long record at the end of a previous WAL file during the ***Partial restore*** action
- Fixed errors that occurred during the ***Partial restore*** task execution in temporary PostgreSQL instances when running the `pgackrest archive-get` command
- Fixed the search for neighboring WAL files in other directories during filtering WAL
- In filtering WAL, replaced the error of missing the next WAL file with a warning
- Fixed the overwrite `contrecord` (continuation record) handling at the beginning of the next file when filtering the WAL
- Fixed the `record is too big` error when filtering WAL
- Fixed filtering WAL when a checksum is in the next segment file
- Fixed filtering WAL when there is no unfinished record at the end of the segment file
- Fixed handling WAL records larger than a WAL segment
- Fixed: a segment did not select timeline of a restore point during ***Partial restore*** after switching to a mirror
- Fixed reading an unfinished record at the end of the file when the next file is partial

2.5.0

Date: 23.06.2025

New features

- Added display of the Common restore action progress
- Removed the ***Skip validation*** option from the data restore form if a cluster is down. Now it is detected automatically and works for data restore, copy and streaming, giving the ability to recover a stopped cluster. ADBM agents should be alive on the stopped cluster hosts for this option to work
- Reworked the Verify action: backup files are now verified along with metadata. Added the corresponding option ***Verify*** to the Common restore, Partial restore, stream launch actions

Improvements

- Changed the `archive_mode` setting value to `on` to avoid the `Duplicates found in archive for WAL segment` error that occurred when primary and mirror segments tried to archive the same file. Note: ADB cluster will be restarted after ADBM upgrade because of the `archive_mode` change
- Implemented storing the `ic_proxy_port_delta` parameter on a cluster level for proper handling of the `ic_proxy` configuration during data restore
- Created the new page Backup manager → Configuration → Schedule to manage scheduling for a cluster. Also, reworked a schema of scheduling data
- Reworked a schema of cluster-related data and changed cluster addressing from a name to GUID
- In the ADBM agent, improved processing of the mirrors absence

Bug fixes

- Fixed handling of auto terminate actions for timed out backups
- Fixed the `A conflicting conditional operation is currently in progress against this resource` error that occurred when checking S3 writability

2.4.0

Date: 07.03.2025

New features

- Estimated progress is now shown for backups
- Added server-side encryption options for S3: `kms-key-id` and `sse-customer-key` (see ***Server-side encryption options*** in Manage configurations in ADBM → Add a new configuration → Repository)
- Added a choice of the restore point streaming error handling mode on stream creation (see ***On restore points streaming failure*** in Disaster recovery). Added the `Paused` status for streams and Resume/Cancel options for this status

Improvements

- Added storing of the `gp_interconnect_type` value in the cluster topology to support force restore on clusters with configured `ic_proxy`
- Changed the ***Apply backup configuration*** action logic to restart a cluster only when the `archive_mode` value is different from `always` and has to be set
- Added caching of the ***Recover segments*** action on the agent side for high availability (HA) purposes
- Adjusted the `await-segment-ready` parameters in the ADBM agent properties to optimize recovery time and avoid streaming failures by timeout

Bug fixes

- Data restore with the ***Force*** and ***Skip validation*** flags set could fail if there was switch to standby
- In ADBM CLI, some parameters were not returned in the command output and parameter compatibility was not checked

2.3.2

Date: 27.12.2024

New features

- Removed the etcd component from the ADBM architecture and assigned its functions to PostgreSQL
- To avoid illogical assignment of the `Invalid` status to backups, modified the ***Verify storage*** action to check repository accessibility from all agents instead of the master host only

Improvements

- A list of cluster actions now contains only currently allowed actions to eliminate errors that occurred when the ***Verify*** action was not applied after cluster upgrade
- For the backup report command of ADBM CLI, added the ability to select the output format in the `--output` option
- Added actualization of agent roles before data restore with the ***Force*** option — to support force restore on a target cluster with a lost dispatcher role
- Added filters, sorting, and pagination to the table on the restore details page

Bug fixes

- Fixed handling of `gp_interconnect_proxy_addresses` during data restore in an ADB cluster
- Fixed: the Partial restore action launched on a source cluster could terminate streaming
- Fixed: a filtering configuration for Partial restore was not used
- Fixed: a target cluster was shown in the ADB Control UI with the `Up` status while streaming

2.2.3

Date: 14.11.2024

New features

- Introduced an option of partial recovery from binary backups

Improvements

- Added parallel Cleanup execution on agents to avoid termination of this action by timeout on large installations
- Updated the backup configuration version display
- In the ***Subactions*** section on the page with action details, added line breaks to the display of long action names
- In the process of cluster reinstallation, added restart after the ADBM client installation to set the `PGBACKREST_CONFIG` environment variable

Bug fixes

- The `isQuiet`, `isVerbose`, `backupDir` options did not work for the `backup` and `backup config` commands in ADBM CLI
- Some elements were not aligned in the ADBM action details
- ADBM restarted ADB upon backup configuration changes even when `archive_command` was not changed
- The shortened version of the `database` option for commands did not work in ADBM CLI
- Backups in the `Invalid` status were not deleted during the ***Cleanup*** action
- The `archiver` process failed after the host recovery

2.1.2

Date: 29.08.2024

New features

- Implemented new parameters for S3 in backup configurations: ***Upload chunk size*** and ***Certificate verification*** (see the ***Repository*** section in Manage configurations in ADBM)
- Added integration with Vault

Improvements

- The Resume action functionality is now available only for full backups
- Restricted the number of backups in the `Stopped` status to `1` and added a history of backup status changes
- Added a limit for a total number of backups
- Added validation for a compression level in a backup configuration

Bug fixes

- Fixed the ***Cleanup*** action for backups in the `Unknown` status
- Fixed: backups were not transferred to the `Invalid` status
- Fixed: `gprecoverseg` failed on an agent during the ***Restore*** action due to running data backup or restore in ADB
- Fixed: no information on a backup after changing its status from `Unknown` to `Done`

2.0.4

Date: 27.06.2024

New features

- Implemented CLI for logical backups
- Enabled the feature of block-level incremental backups: the corresponding flag is added to the backup configuration page

Improvements

- Added support for expanding an ADB cluster for ADBM
- Added the ***Verify backup*** action to a stream flow

Bug fixes

- Removed the ability to delete a configuration for an active stream
- Fixed: the restore process failed for a restore point created after one of the segments went down
- Question marks were displayed in the ***Exclude path*** field if its value was added in Russian
- Fixed the incorrect `Warning` status of the ***Terminate*** action applied to a stream after several clicks on the ***Run*** button in the ***Terminate stream*** modal window
- Fixed: the only cluster backup configuration was deleted after several streams and cleanups were launched
- A backup with the `Applied` configuration failed if the `Draft` configuration existed
- Backups failed after upgrading an ADB cluster from 6.25.1 to 6.26.2
- ADB master stopped during the restart of *adbm-agent*

1.7.3

Date: 11.04.2024

New features

- Added a new option for Disaster Recovery: standby cluster can now be restored from WAL changes of a source cluster. Restore from a source backup is required only on the initial step, not on every iteration

Improvements

- Improved handling of an S3 repository in ADBM actions considering essential configuration parameters
- Cleanup logic was updated to delete all incremental and differential backups that might be left by policy
- Added the `liquibase.releaseLocks` parameter for ADBM migration service

Bug fixes

- Fixed: copy restore failed after stopping one of the source cluster segments
- Fixed a stream status type on applying impossible streaming configuration
- Fixed: the `Unknown` status for backups was absent

1.6.3

Date: 28.12.2023

New features

- Added new statuses for management of backup configurations
- Implemented changes in the ***Cleanup*** action to support modifications of a repository type or path, and to protect backups during Disaster Recovery (DR) copy stream

Improvements

- Added handling of repository unavailability to the process of backup status determination

Bug fixes

- The ***Backup*** action flow was being stopped when subactions were in the `Warning` status
- Some details of the ***Restore*** action were not visible in the light theme

1.5.2

Date: 02.11.2023

New features

- Implemented the ability to restore mirrors from binary backups in parallel with primary segments — via the Restore mirrors field. Note that it is an experimental feature

Improvements

- Verifying the repository accessibility was added for a copy stream command flow
- Added the Warning status for ADBM actions
- Date filter defaults were changed to the last day on the ***Backups*** and ***Restores*** pages
- Added the forced ***Skip validation*** flag display to the Restore modal window for restoring data in stopped ADB clusters

Bug fixes

- Fixed the error of running incremental backups connected to zero-sized files
- Fixed the verification error that occurred when an S3 repository with self-signed certificates was used
- Fixed the error of creating ADBM configuration for S3 storages with the `path` URI type
- Fixed downgrade in the ADBM UI performance when one of clusters was unavailable
- Fixed the error of restoring from a backup after a successful copy stream on a DR cluster

1.4.0

Date: 31.08.2023

New features

- CIFS repository support was added
- Added the option of automated backup copying to a target cluster when a new backup on a source cluster is completed

Improvements

- Action id was added into action details
- Auto detection of an active master host was added to support a backup process after switching to standby

Bug fixes

- It was impossible to restore backups in clusters with non-unique names within one ADBM
- Self-restore (via the ***Restore*** action) failed after changing roles on mirrors
- S3 certificate was deleted from segments after the host reboot

1.3.3

Date: 10.08.2023

New features

- Added the restore with mirrors option for Disaster Recovery (DR) copy process
- The repository validation step was added before saving a backup configuration

Improvements

- Backup topology data is now sorted by ***Segment***

Bug fixes

- ***Restores*** list was empty for clusters with spaces in names
- DR copy action failed when source and target clusters had different standby configurations

1.3.2

Date: 20.07.2023

New features

- Restore for standby master was implemented
- Automatic checking of ADB versions compatibility on source and target clusters was added to the Disaster Recovery (DR) copy process
- Availability check for a backup configuration on a target cluster was added to the DR copy process

Bug fixes

- Copy stream failed on a cluster with standby
- Could not restore a backup on DR cluster with a number of datanodes different from a source cluster
- Choosing particular databases from the database list caused the Disaster Recovery (DR) copy process to fail
- DR cluster could not be started after restore

1.3.0

Date: 30.06.2023

New features

- Implemented Disaster Recovery with Cold Standby — ability to recover a cluster from a backup located on another cluster. Beta version! Please pay attention to the ADBM known issues section and backup a target cluster before starting a stream
- Added `pgbackrest` options (`-delta`, `--force`, `--max-process`) for the restore action
- Added an option to save a backup configuration without schedule for running backups on request

Improvements

- ADBM action tree view was enhanced
- Enabled resuming failed backups considering segments that have already been backed up

Bug fixes

- Fixed: the ***Apply backup config*** action failed when S3 was selected

1.2.1

Date: 27.04.2023

New features

- Added support for PowerPC-based computers
- Added the ability to restore data from backups without mirror segments
- Added the ability to filter restore points for the ***Restore retry*** mechanism
- Added the Restores tab to view ***Restore*** actions

Bug fixes

- Creating an additional configuration when reinstalling without creating a backup
- The error with using an S3 repository to store backups

Misc/Internal

- Added the `openapi-yaml` generator to generate OpenAPI files
- Added a timeout to read an OpenAPI file

1.1.0

Date: 15.02.2023

New features

- The list of available cluster actions is supported now
- Added the ability to delete the last backup from the stanza. It expands the cleaning functionality logic
- Added the ability to restore specific databases from backups

Improvements

- Added the ability to use the ***Force*** flag when a cluster is being restored from a backup. It means bypassing all checks
- Added the validation for the `cron` expressions that are used when ADBM configuration is being created or edited
- Added a check for `archive_command` when saving the ADBM configuration. It allows avoiding unnecessary restart of the ADB cluster

1.0.0

Date: 06.12.2022

New features

- Implemented the first ADBM version. For more details, see Arenadata DB Backup Manager overview

ADBM known issues

Daria Barysheva

To Table of Contents

Issue	From	Fixed	Workaround
ADBM agents with a version lower than 2.8.0 (compatible with ADB 6) cannot connect to the ADBM server 2.8.0	2.8.0	—	Use separate ADBM deployments for ADB 6 and ADB 7 clusters

If an NFS or CIFS storage used for backup becomes unavailable, the `pgbackrest` process can hang, making ADBM operations unresponsive. For example, a backup operation can remain in a hung state even after the ***Protocol timeout*** period (defined in the backup configuration) expires	—	—	1. Identify the `pgbackrest` process and check its state:
			`$ pgrep -a pgbackrest`
			`$ ps -o pid,state,command -p <PID>`
			If the `state` parameter has the value of `D`, the process has entered an `Uninterruptible Sleep` state and cannot be terminated via a SIGKILL signal or `pgbackrest` timeouts until connectivity with the storage is restored.
			2. Mount your NFS or CIFS storage with the `soft` option, which allows system calls to return errors instead of waiting indefinitely. For example, add the following to the */etc/fstab* file:
			• For NFS:
			`server:/share /mnt/share nfs soft,timeo=300,retrans=3 0 0`
			• For CIFS:
			`//server/share /mnt/share cifs soft,timeo=300,retrans=3,credentials=/path/to/credentials 0 0`

The ***Audit*** page does not show operations	2.6.5	—	In ADB Control → UI component parameters, set ***Listening port*** to `8890` for the main and external clusters. Run the ***Reconfigure & Restart*** action to apply changes
Clusters that are connected to external ADB Control should be upgraded along with the main cluster to ADB 6.29.0.3 in order for ADBM and ADB Control to work properly	2.6.5	—	Upgrade clusters to ADB 6.29.0.3
The standby's WAL directory is not recycled properly if `wal_sender_archiving_status_interval` is greater than `0` and `archive_mode` is `always`, which causes data restore in ADBM to crash in some cases	2.0.4	—	Set `wal_sender_archiving_status_interval` to `0`
The backup repository configuration creation fails in ADBM CLI when the S3 plugin is used with default `uploadChunkSize` and `uploadConcurrency` settings	2.0.4	—	Configure `uploadChunkSize` and `uploadConcurrency` based on your system resources before creating a backup configuration. The S3 plugin will try to allocate the memory size equal to `segment_per_host` * `uploadChunkSize` * `uploadConcurrency`
The ***Information*** → ***Backup manager*** page does not display information for segment hosts in the `DOWN` status	1.0.0	—	This is the expected behavior that may be improved in future releases
Termination fails for streams with the ***Restore mirrors*** option set to `In parallel` (during the stream creation)	2.2.0	2.4.0	When creating streams with the `Streaming` restore type, set the `After primary` value for the ***Restore mirrors*** option
After upgrade to ADB 6.26.2, backups with the configuration created before upgrade fail	1.7.3	2.3.2	After upgrade, run the ***Verify*** action for clusters that have backup configurations

Restore fails if one of the cluster segment hosts is down for restore points created after the host was stopped	1.7.0	2.0.4	1. Edit the *ssh_config* file under `root` on the master host:
			`$ vi /etc/ssh/ssh_config`
			2. Add the following line to the end of the file:
			`ConnectTimeout 20`
			3. Restart the SSH daemon (`sshd`):
			`$ systemctl restart sshd`

The ***Create Stream*** modal window has the following restriction: the `Streaming` restore type is only a prototype, do not use it	1.3.1	1.7.3	Use only the ***Copy*** option for a stream
Mirrors are not restored automatically on a target cluster. If a target cluster already has mirrors, the restore job will fail, and if it doesn't then the restore job will finish without errors but the restored cluster will not have mirrors even if the source cluster had	1.3.1	1.3.3	DR cluster shouldn't have mirror segments
The ***Create Stream*** and ***Restore*** modal windows have the following restriction: the ***Restore mirrors*** option should not be used	1.3.1	1.3.3	Reset the ***Restore mirrors*** checkbox when setting up a stream or restore
The Disaster recovery copy action fails when: • A source cluster doesn't have a standby and a target cluster has it. • A source cluster has a standby and a target cluster has not.	1.3.1	1.3.3	Standby should be present or absent on both source and target clusters
Restores list is empty for clusters with spaces in names	1.1.0	1.3.3	Cluster names shouldn't have spaces
Standby restore for a target cluster is not available	1.3.1	1.3.2	DR cluster shouldn't have standby master
It is impossible to restore data from a backup on the DR cluster when the source cluster number of data nodes differs from the DR cluster	1.3.1	1.3.2	DR cluster and source cluster should have the same topology
Cluster is marked `Down` in ADBM when master is on the same host as segment	1.3.1	1.3.2	Edit */opt/adbm-agent/service-configuration* and leave only one record for this host that describes ADBM agent roles: `ADBM_AGENT_ROLES=EXECUTOR,DISPATCHER`, then restart agent with the `systemctl restart adbm-agent` command
DR cluster cannot be used for writing	1.3.1	1.3.2	Do not use DDL or DML commands on DR cluster

DR cluster cannot be started after restoring	1.3.1	1.3.2	1. Backup *pg_hba.conf* of `-1` and `0` segments on DR cluster before running a restore stream. Save it to persistent storage (do not use */tmp*). Use the following commands (under the `gpadmin` user) on master host:
			`$ cp /data1/master/gpseg-1/pg_hba.conf /home/gpadmin`
			`$ gpssh -v -f /home/gpadmin/arenadata_configs/arenadata_segment_hosts.hosts -e 'cp /data1/primary/gpseg*/pg_hba.conf /home/gpadmin/'`
			2. Restore *pg_hba.conf* after completing ***Disaster recovery copy stream*** action via the following commands (under the `gpadmin` user) on master host:
			`$ cp /home/gpadmin/pg_hba.conf /data1/master/gpseg-1/`
			`$ gpssh -v -f /home/gpadmin/arenadata_configs/arenadata_segment_hosts.hosts -e 'find /data1/primary/gpseg* -maxdepth 0 -exec cp /home/gpadmin/pg_hba.conf {} \;'`
			3. Change the `synchronous_standby_names` parameter on all primary segments from `'*'` to `''` in */<data_directory>/primary/gpseg*/postgresql.auto.conf*.

The ***Create Stream*** and ***Restore*** modal windows have the following restriction: choosing particular databases from the ***Database*** list will cause the job to fail, do not use it	1.3.1	1.3.2	Select all databases when setting up a stream or restore

It is impossible to add external ADB clusters to the general-purpose ADBM from the ADBM UI	1.0.0	1.3.1	1. Ensure the external ADB cluster with the installed ADBM service exists in ADCM.
			2. Connect to the Linux console. Generate the Basic Authorization hash with base64 for a user with access to Arenadata DB Backup Manager via the following command: `echo -n <adbm_user>:<adbm_user_password> | base64`. For example:
			`$ echo -n adbm_user:Password | base64`
			Result:
			`YWRibV91c2VyOlBhc3N3b3Jk`
			3. Send the following POST request to the general-purpose ADBM host:
			• Header parameters:
			  ◦ `Accept`.
			  ◦ `Accept-Charset`.
			  ◦ `Authorization`. It should have the value started with `Basic ` and then the base64 value from step 2.
			  ◦ `Content-Type`.
			  ◦ `X-ADBM_USER`. It should have the `adbm_user` value from step 2.
			• Body:
			`{`
			`"clusterName": "<name of external ADB cluster in ADCM>",`
			`"clusterType": "<purpose of ADB cluster>",`
			`"clusterVersion": "<version of ADB cluster>"`
			`}`
			The following is an example of the POST request with the Linux `curl` command:
			`$ curl --request POST \`
			`--url http://adbm-main:8890/backend/adbm/api/v1/clusters \`
			`--header 'Accept: application/json' \`
			`--header 'Accept-Charset: ISO-8859-1' \`
			`--header 'Authorization: Basic YWRibV91c2VyOlBhc3N3b3Jk' \`
			`--header 'Content-Type: application/json' \`
			`--header 'X-ADBM-USER: adbm_user' \`
			`--data '{`
			`"clusterName": "adb-demo",`
			`"clusterType": "ADB demo stand",`
			`"clusterVersion": "6.22"}'`
			A successful response contains a JSON with the generated identifier of an external cluster in ADBM. The cluster becomes visible in ***Arenadata DB Backup Manager*** → ***Clusters*** with the `Down` status.
			4. In ADCM, go to the external ADB cluster and select ***Services*** → ***ADBM*** → ***Configuration***.
			5. Set the ***Show advanced*** flag and edit the ADBM agent service configuration parameter `ADBM_EUREKA_CLIENT_SERV_URL_DEF_ZONE` located in the bottom area. Replace its value with URL of the service registry for a general-purpose ADBM host, e.g. `http://adbm-main:8761/eureka`.
			6. Apply the ***Reconfigure & Restart*** action to ADBM. Upon success, the cluster status in ***Arenadata DB Backup Manager*** → ***Clusters*** is changed to `Up`.

FQDN names in the `hostname` and `address` columns in the `gp_segment_configuration` table are not supported. Essentially, it is a blocker for ADBM utilizing in clusters where FQDNs are used	1.0.0	1.1.0	Modify names in the `gp_segment_configuration` table to short ones. Take into account that it is an extremely dangerous operation, which can harm a cluster
Segment servers have an additional role that they don't need to have. ADBM will work, but with extra queries on segment servers. Users can discover it only from a log	1.0.0	1.1.0	Change a configuration file on the ***ADCM*** → ***ADBM*** → ***Configuration*** page: 1. Set the ***Show advanced*** flag. 2. Find the ***ADBM agent service config parameters*** field. 3. Replace `master` by `adb.master` and `segment` by `adb.segment`. 4. Click ***Save*** and then run the ***Reconfigure & Restart*** action.

Create a cluster

Anton Vasilev

Contents

To Table of Contents

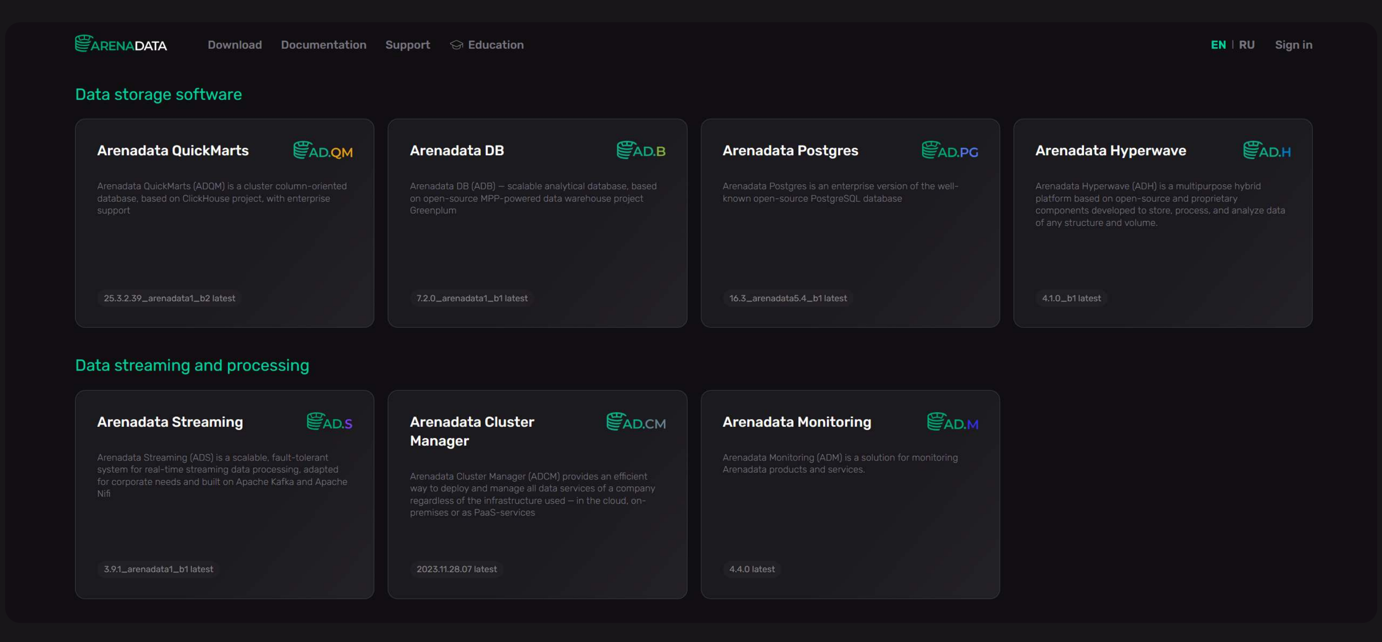
In ADCM, a **cluster** means a set of services running on a set of hosts.

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

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

Step 1. Download a cluster bundle

1. Go to https://network.arenadata.io/ and choose ***Arenadata DB Enterprise Services***.



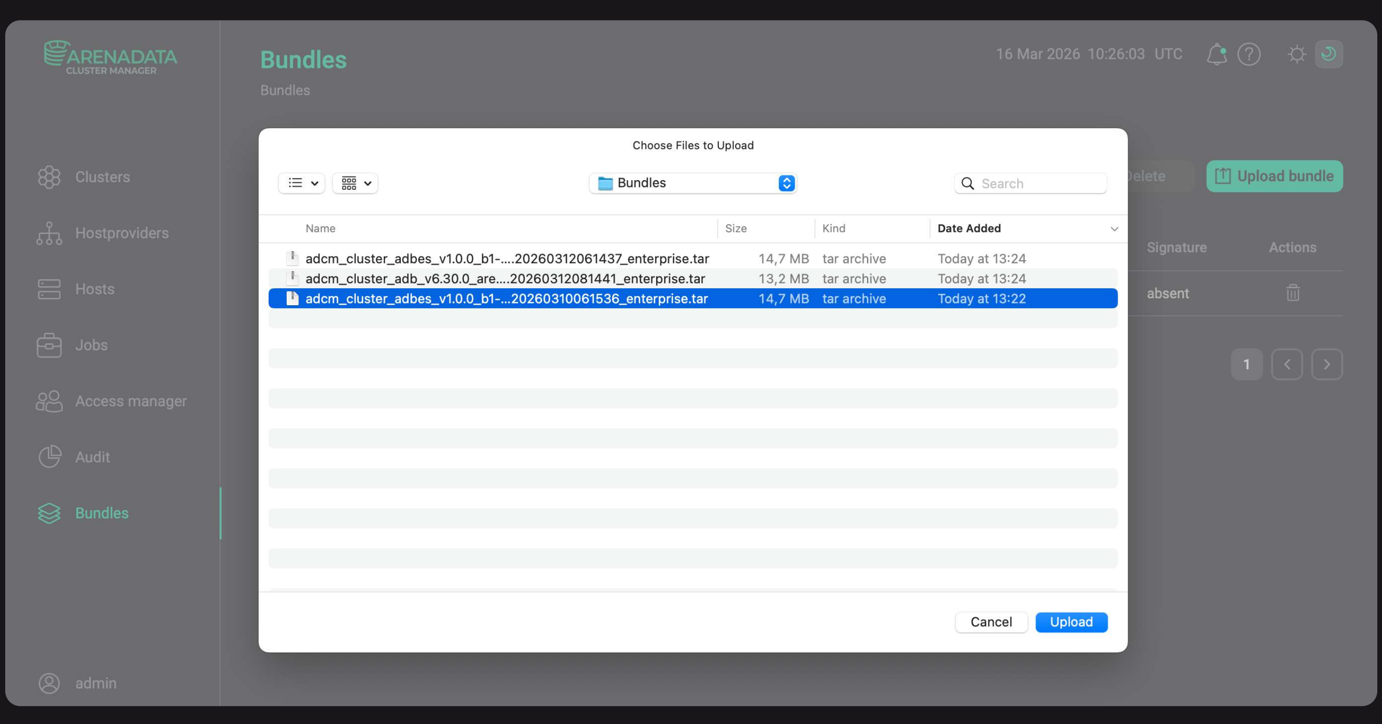
Switch to the bundle download page

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

Step 2. Upload a cluster bundle to ADCM

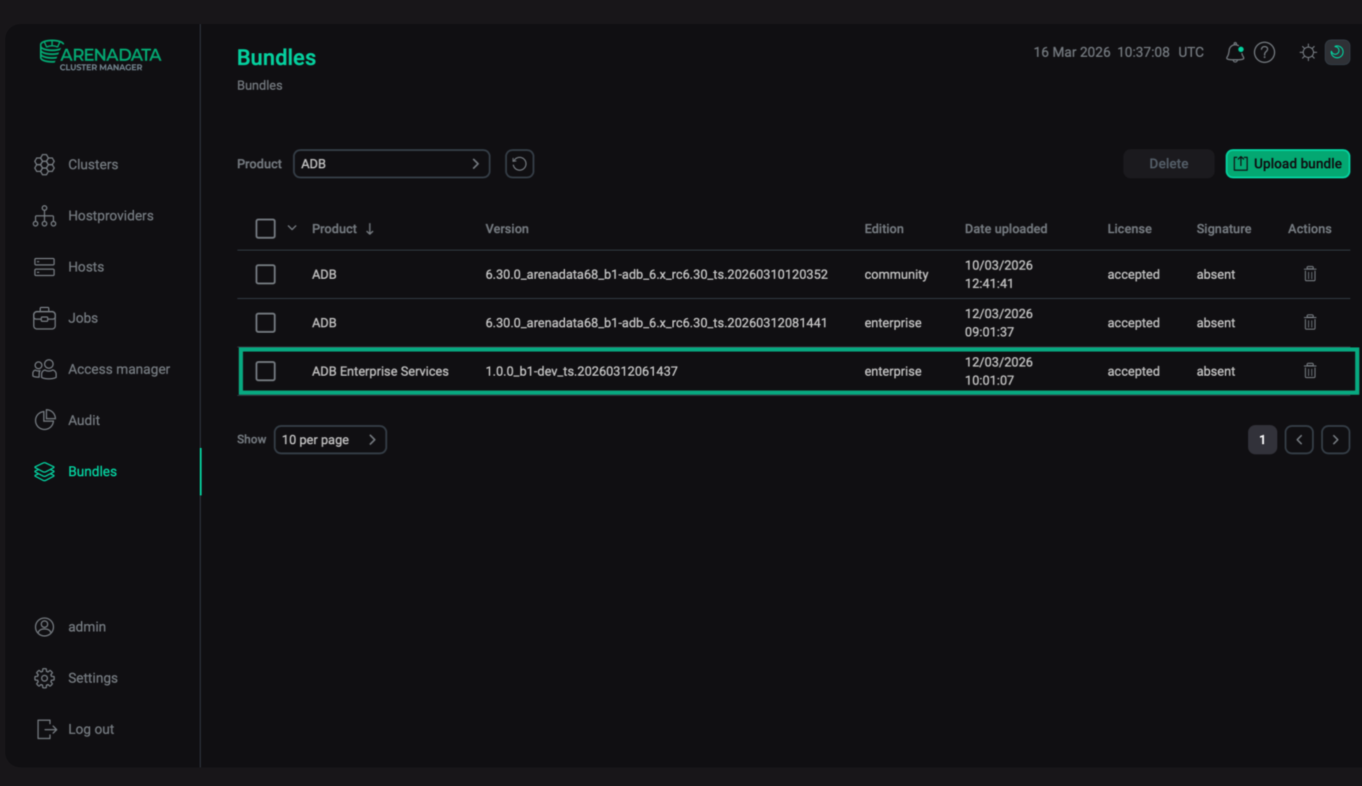
To upload a bundle to ADCM, follow the steps:

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

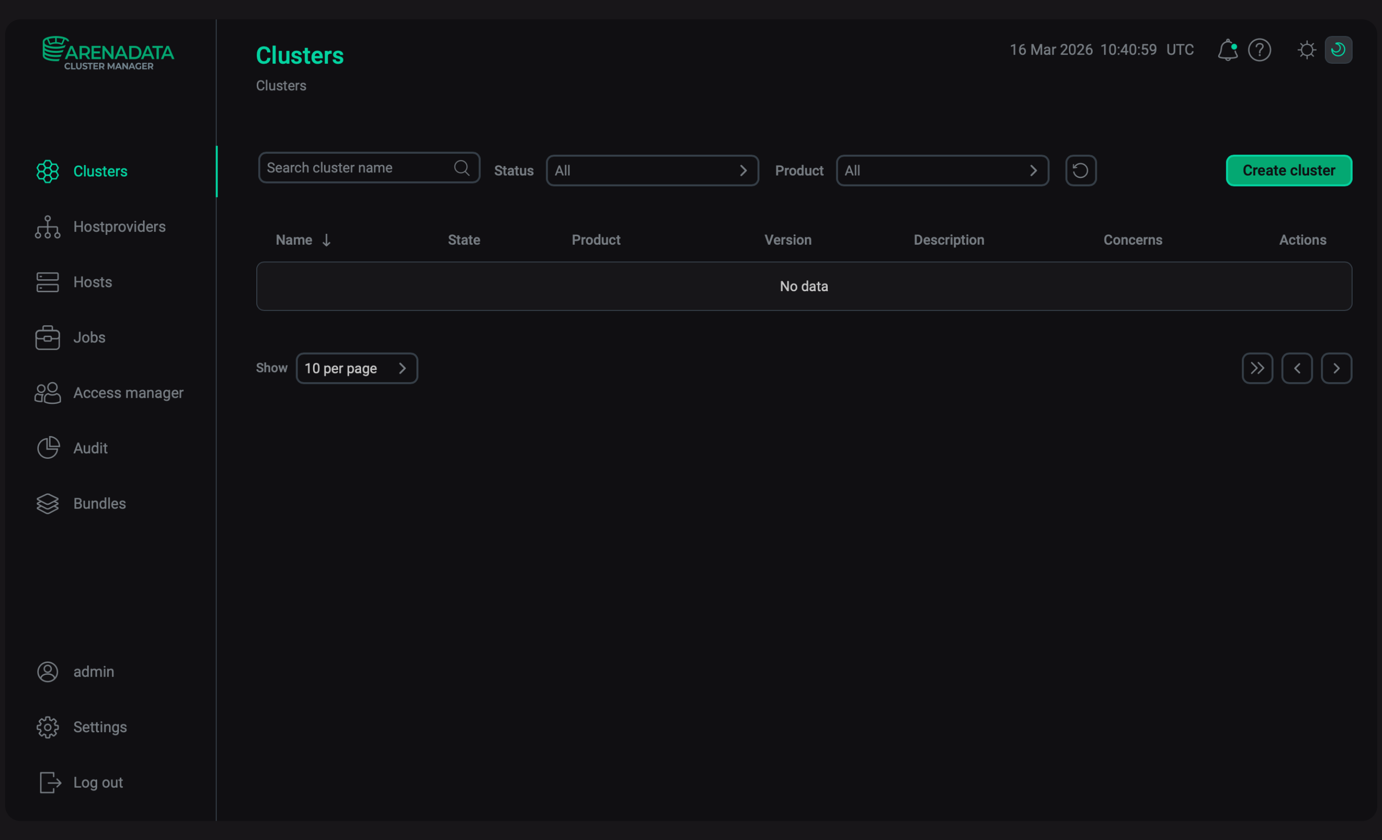
Step 3. Create a cluster based on the uploaded bundle

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:

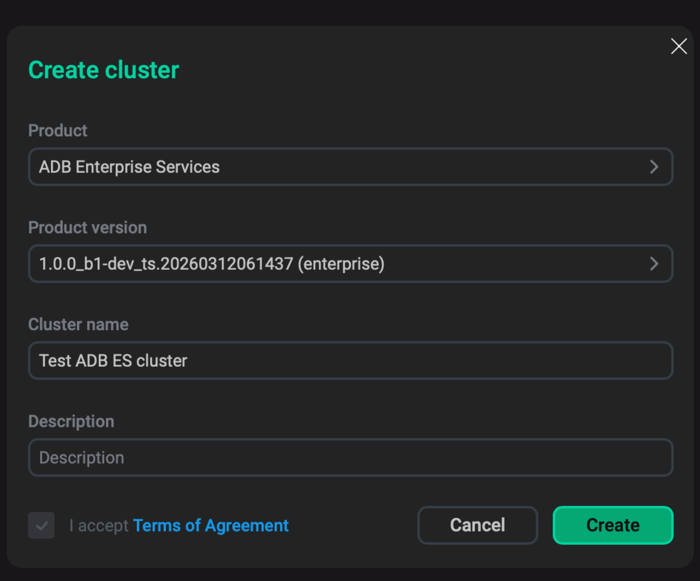
1. Select the ***Clusters*** item in the left navigation menu and click ***Create cluster***.



Switch to cluster creation

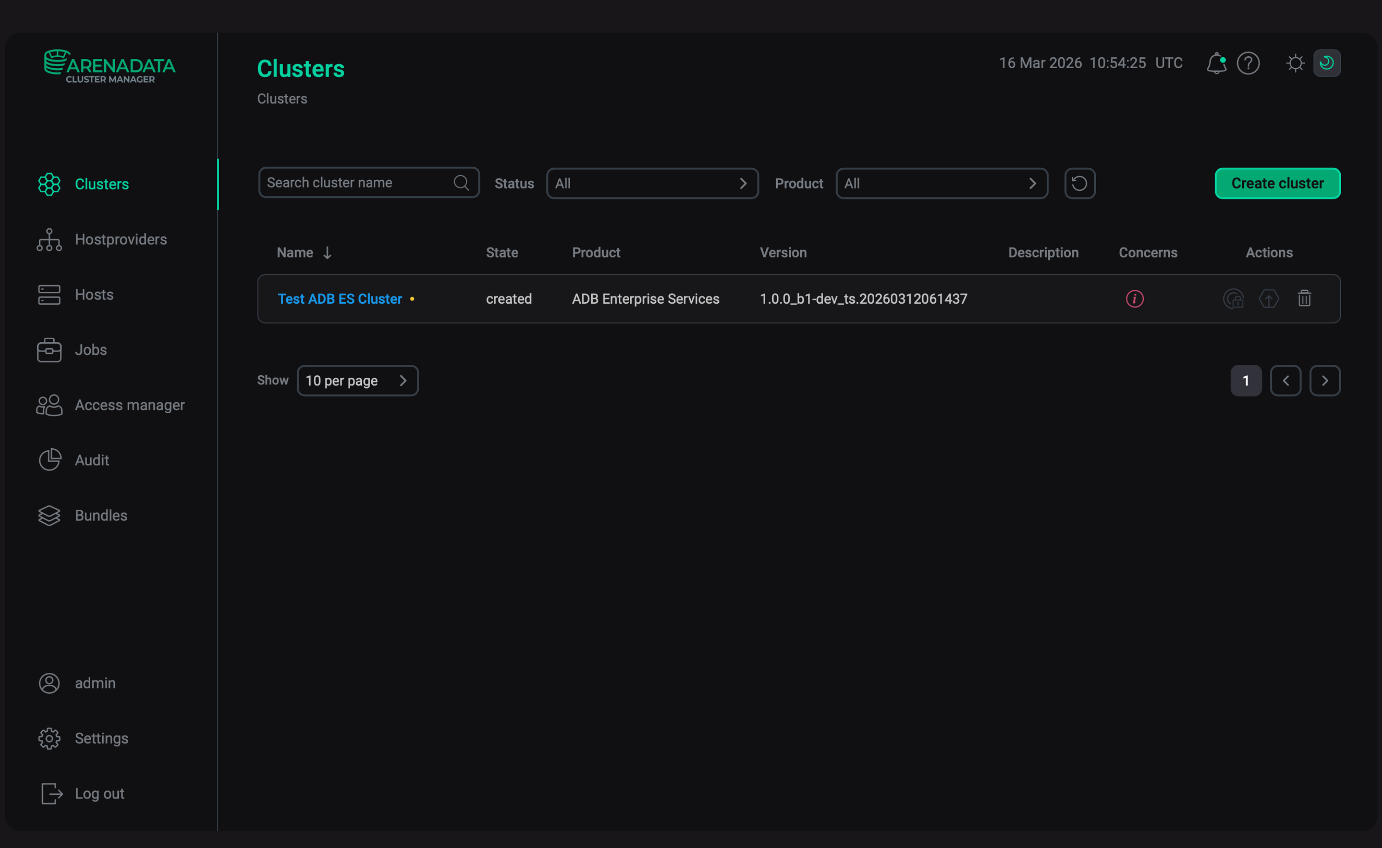
2. In the opened dialog, describe a new cluster instance:

   - 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

Arenadata DB Enterprise Services

Contents

To Table of Contents

Arenadata DB Enterprise Services (ADB ES) is a set of Enterprise services that extend the Arenadata DB (ADB) functionality. The core ADB ES services are:

- Arenadata DB Control (ADB Control) — real-time ADB query monitoring system.
- Arenadata DB Backup Manager (ADBM) — ADB binary backup management system.

TOP-5 popular articles

Migration of ADB Control and ADBM metadata during ADB ES installation

This article describes how to migrate ADB Control and ADBM metadata from ADB during initial ADB ES installation.

Arenadata DB Backup Manager overview

Arenadata DB Backup Manager (ADBM) is a fault-tolerant system for ADB binary backups management, which is built on the base of pgbackrest. Learn about ADBM features, architecture, and some internal details.

Arenadata DB Control overview

Arenadata DB Control (ADB Control) is a real-time Arenadata DB query monitoring system. Learn about ADB Control features, architecture, and some internal details.

Manage configurations in ADBM

This article describes how to configure and use backup configurations in ADBM.

Connect to ADB Control and ADBM

This article describes how to start working with the ADB Control and ADBM web interface.