ADET

Arenadata Enterprise Tools



Arenadata Enterprise Tools (ADET) provides the infrastructure needed to deliver, manage, and maintain dependencies for Arenadata products in secure, isolated, or restricted network environments.

Version CURRENT Language: EN

Contents

1. Planning guide
   - Hardware requirements
   - Network requirements
   - Software requirements
2. Get started
   - Installation
     - Install ADCM
     - Prepare hosts
     - Install ADET cluster
     - Create a cluster
     - Add services
     - Add hosts to a cluster
     - Add components
     - Configure services
     - Configure a cluster
     - Install a cluster
3. Services
   - Diamond
     - Service management via ADCM
     - Configuration parameters
   - Docker Registry
     - Service management via ADCM
     - Configuration parameters
   - Grafana
     - Service management via ADCM
     - Configuration parameters
   - Graphite
     - Service management via ADCM
     - Configuration parameters
   - HTTP Mirror
     - Service management via ADCM
     - Configuration parameters
4. Cluster-level operations
   - Cluster management via ADCM
     - Cluster actions
       - Manage SSL
       - Reinstall
       - Reinstall Statuschecker
       - Restart
       - Uninstall
       - Upload pack
     - Service actions
5. References
   - Configuration parameters
6. Release notes

Hardware requirements

Lisa Vaysman

To Table of Contents

For an Arenadata Enterprise Tools cluster, the following minimum requirements apply:

- Virtual server.
- CPU: 4 cores or more.
- RAM: 8 GB or more.
- Disk: 50 GB or more.
- Disk space requirements:
  - Docker Root directory. Ensure there is enough free space in the directory where Docker stores images and temporary layers (*/var/lib/docker* by default). It is recommended to reserve at least twice the size of the uncompressed product packages, as Docker creates temporary layers during the loading process before the final image formation.
  - Installation directory. Ensure that the */opt/arenadata/etools/* directory (or another selected directory) has enough available disk space to store uploaded archives and service data.

Network requirements

Lisa Vaysman

To Table of Contents

The network requirements for ADET installation are listed below:

- All cluster hosts should have known IP addresses, fully qualified domain names (FQDN), and the ability to communicate with each other at the required speeds.
- `ping` access to any cluster host (ICMP) should be provided.
- There should be no firewall or other software that blocks or slows down the traffic (`firewalld`, `iptables`, `ufw`, etc.). If deactivation of such software is not possible, then add the ports used by ADET services to the firewall exception list using the table below.

When identifying port numbers, keep in mind the following rules:

- SSL mode. HTTP Mirror and Grafana use different default ports depending on whether SSL is enabled or disabled. Only add to your firewall exception list the ports that correspond to your SSL mode.
- Custom ports. If you modified the default ports (for example, during ADET installation in ADCM), use your custom values instead of the values specified in the table below.

NOTE

Each port refers to the host where a specific service is installed.

ADET ports

Service	Port	Purpose
Common	22	TCP port for SSH connections to the ADET host
Docker Registry	443	TCP port for Docker Registry to download container images. This value can be changed either via the HTTPS port setting of Docker Registry or via the Manage SSL ADET cluster action
HTTP Mirror	81	TCP port (HTTP) to access ADET repositories on a dedicated ADET host during offline installation. This value can be changed via the HTTP port setting of HTTP Mirror
	8443	TCP port (HTTPS) to access ADET repositories on a dedicated ADET host during offline installation. This value can be changed via the Manage SSL ADET cluster action
Grafana	3000	TCP port to access the Grafana web interface via HTTP. This value can be changed via the Port setting of Grafana
	3003	TCP port to access the Grafana web interface via HTTPS. This value can be changed via the Manage SSL ADET cluster action
Graphite	80	TCP port to access the Graphite web interface. This value can be changed via the Web-interface TCP port setting of Graphite
	2015	TCP port for receiving metrics from Diamond agents via Carbon Plaintext Protocol. This value can be changed via the Metrics receiver TCP port setting of Graphite
	2016	UDP port for receiving metrics from Diamond agents via Carbon Plaintext Protocol. This value can be changed via the Metrics receiver UDP port setting of Graphite

Software requirements

Lisa Vaysman

Contents

To Table of Contents

Operating system

ADCM

Operating system

ADET supports only x86_64 architecture and the following operating systems:

- ALT Linux 8.4 SP
- ALT Linux 10.2 SP
- Astra Linux SE 1.7.6 Orel and Voronezh
- Astra Linux SE 1.8.4 Orel and Voronezh
- RED OS 7.3 Certified edition
- RED OS 8.0 Certified edition
- RHEL 9
- Ubuntu 22.04 LTS
- Ubuntu 24.04 LTS

ADCM

It is required to use ADCM 2.11.0 or higher.

Installation

To Table of Contents

This tutorial guides you through the process of installing an Arenadata Enterprise Tools (ADET) cluster.

The easiest way to install it is to use Arenadata Cluster Manager (ADCM) — a separate Arenadata product designed for simple, convenient, and fast software deployment and operation.

This guide focuses specifically on **offline** installation, making it suitable for target hosts that lack internet access.

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

Step 1.

Install ADCM

Step 2.

Prepare hosts

Step 3.

Install ADET cluster

Install AXCM

Contents

Step 1. Check prerequisites
- Software and hardware requirements
- Check the KVM port availability
- Customer startup process

Step 2. Install AXCM

Step 3. Start AXCM
- Check the installation status
- Check the Docker container status
- Check the AXCM web interface

Step 4. Check AXHA

Step 5. Update AXCM

Step 1. Check prerequisites

Software and hardware requirements

AXCM is delivered as a Docker container image and runs on Ubuntu Linux. To check your host:

1. Run the following operating system:

```
lsb_release -a
```

Example output:

```
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.5 LTS
Release:        22.04
Codename:       jammy
```

2. Run the following CPU:

```
lscpu
```

Example output:

```
Architecture:            x86_64
  CPU op-mode(s):        32-bit, 64-bit
  Address sizes:         46 bits physical, 48 bits virtual
  Byte Order:            Little Endian
CPU(s):                  8
  On-line CPU(s) list:   0-7
Vendor ID:               GenuineIntel
  Model name:            Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz
    CPU family:          6
    Model:               85
    Thread(s) per core:  1
    Core(s) per socket:  8
    Socket(s):           1
    Stepping:            7
    BogoMIPS:            6000.00
    Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc rep_good nopl xtopology nonstop_tsc cpuid aperfmperf tsc_known_freq pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch invpcid_single pti ssbd ibrs ibpb stibp fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid avx512f avx512dq rdseed adx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves arat pku ospke avx512_vnni md_clear flush_l1d arch_capabilities
Virtualization features:
  Hypervisor vendor:     VMware
  Virtualization type:   full
Caches (sum of all):
  L1d:                   256 KiB (8 instances)
  L1i:                   256 KiB (8 instances)
  L2:                    8 MiB (8 instances)
  L3:                    176 MiB (8 instances)
NUMA:
  NUMA node(s):          1
  NUMA node0 CPU(s):     0-7
Vulnerabilities:
  Itlb multihit:         KVM: Mitigation: VMX unsupported
  L1tf:                  Mitigation; PTE Inversion; VMX flush not necessary, SMT disabled
  Mds:                   Mitigation; Clear CPU buffers; SMT Host state unknown
  Meltdown:              Mitigation; PTI
  Mmio stale data:       Mitigation; Clear CPU buffers; SMT Host state unknown
  Retbleed:              Mitigation; IBRS
  Spec store bypass:     Mitigation; Speculative Store Bypass disabled via prctl and seccomp
  Spectre v1:            Mitigation; usercopy/swapgs barriers and __user pointer sanitization
  Spectre v2:            Mitigation; IBRS, IBPB conditional, STIBP conditional, RSB filling, PBRSB-eIBRS Not affected
  Srbds:                 Not affected
  Tsx async abort:       Not affected
```

3. Run the following memory:

```
free -h
```

Example output:

```
               total        used        free      shared  buff/cache   available
Mem:            31Gi       1.0Gi        29Gi       1.0Mi       1.1Gi        30Gi
Swap:           2.0Gi          0B       2.0Gi
```

4. Run the following disk:

```
df -h
```

Example output:

```
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           3.2G  1.1M  3.2G   1% /run
/dev/sda2       100G   11G   90G  11% /
tmpfs            16G     0   16G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/sda1       511M  6.1M  505M   2% /boot/efi
tmpfs           3.2G   12K  3.2G   1% /run/user/1000
```

5. Run the following GPU:

```
lspci | grep -i vga
```

Example output:

```
00:0f.0 VGA compatible controller: VMware SVGA II Adapter
```

6. Run the following network:

```
ip -br addr
```

Example output:

```
lo               UNKNOWN        127.0.0.1/8 ::1/128
ens160           UP             10.10.10.10/24 fe80::250:56ff:fe8e:1234/64
```

Check KVM, CPU, RAM, Disk, and GPU

1. Check KVM:

```
kvm-ok
```

Example output:

```
INFO: /dev/kvm exists
KVM acceleration can be used
```

2. Check CPU:

```
lscpu | grep -E 'Model name|CPU\(s\)|Thread|Core|Socket'
```

Example output:

```
CPU(s):                  8
  On-line CPU(s) list:   0-7
  Model name:            Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz
    Thread(s) per core:  1
    Core(s) per socket:  8
    Socket(s):           1
```

3. Check RAM:

```
free -h
```

Example output:

```
               total        used        free      shared  buff/cache   available
Mem:            31Gi       1.0Gi        29Gi       1.0Mi       1.1Gi        30Gi
Swap:           2.0Gi          0B       2.0Gi
```

4. Check Disk:

```
df -h /
```

Example output:

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       100G   11G   90G  11% /
```

5. Check GPU:

```
lspci | grep -i vga
```

Example output:

```
00:0f.0 VGA compatible controller: VMware SVGA II Adapter
```

Check KVM Port

1. Check if KVM is running:

```
systemctl status libvirtd
```

Example output:

```
● libvirtd.service - Virtualization daemon
     Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2026-04-20 09:12:34 KST; 2 days ago
       Docs: man:libvirtd(8)
             https://libvirt.org
   Main PID: 1234 (libvirtd)
      Tasks: 19 (limit: 32768)
     Memory: 18.4M
        CPU: 1.234s
     CGroup: /system.slice/libvirtd.service
             └─1234 /usr/sbin/libvirtd
```

2. Check if KVM is running:

```
virsh list --all
```

Example output:

```
 Id   Name   State
--------------------
```

Check AXCM Port

1. Check if AXCM is running:

```
docker ps | grep axcm
```

Example output:

```
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                                                
```

Prepare hosts

To Table of Contents

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

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

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

For offline installation, the SSH hostprovider is recommended.

CAUTION

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



Create a cluster

Lisa Vaysman

Contents

To Table of Contents

Step 1. Get the Enterprise version of a cluster bundle

Step 2. Upload a cluster bundle to ADCM

Step 3. Create a cluster based on the uploaded bundle

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

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

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

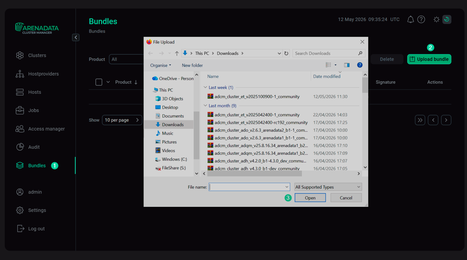
Step 1. Get the Enterprise version of a cluster bundle

Please get a cluster bundle from the Arenadata support team.

Step 2. Upload a cluster bundle to ADCM

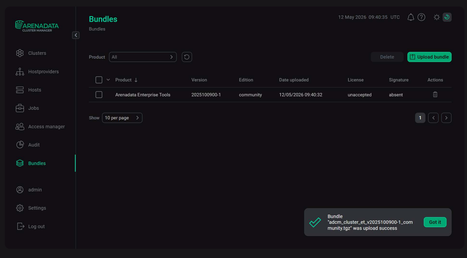
To upload a bundle to ADCM, follow the steps:

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



Upload a bundle

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



The result of successful uploading a bundle

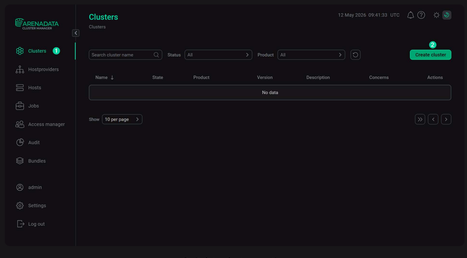
Step 3. Create a cluster based on the uploaded bundle

IMPORTANT

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

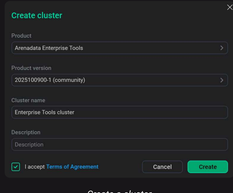
The steps for cluster creation are given below:

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



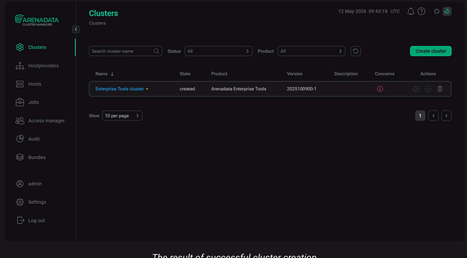
Switch to cluster creation

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



Create a cluster

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



The result of successful cluster creation

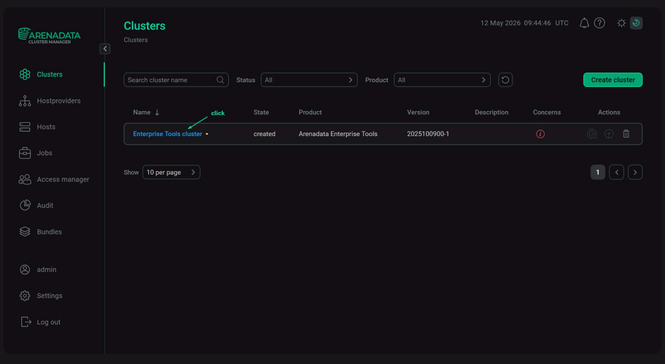
Add services

Lisa Vaysman, Daria Barysheva

To Table of Contents

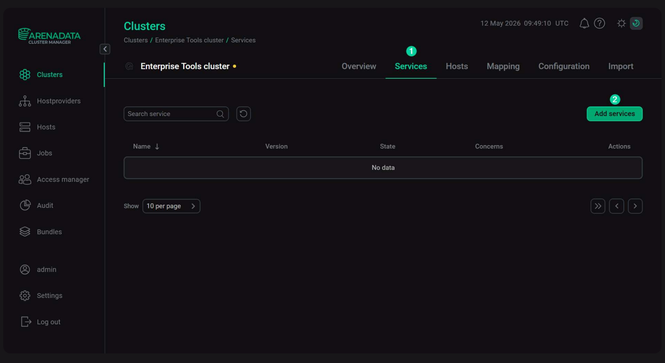
In ADCM, a **service** means a software that performs some function. The steps for adding services to a cluster are listed below:

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



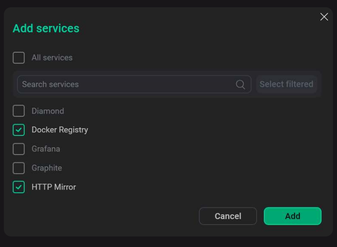
Select a cluster

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



Switch to adding services

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



Select services

The brief description of available services is listed below.

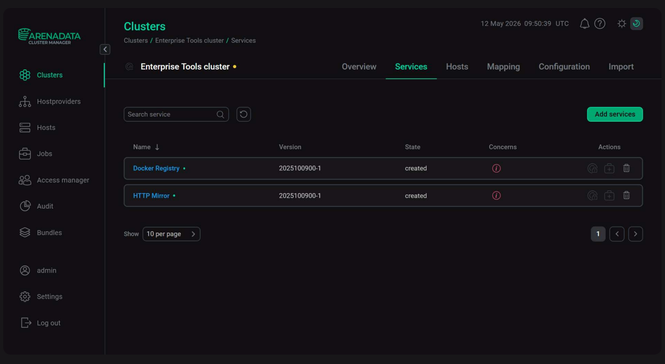
Services that should be added to the Arenadata Enterprise Tools cluster

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

NOTE

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

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



The result of successful adding services to a cluster

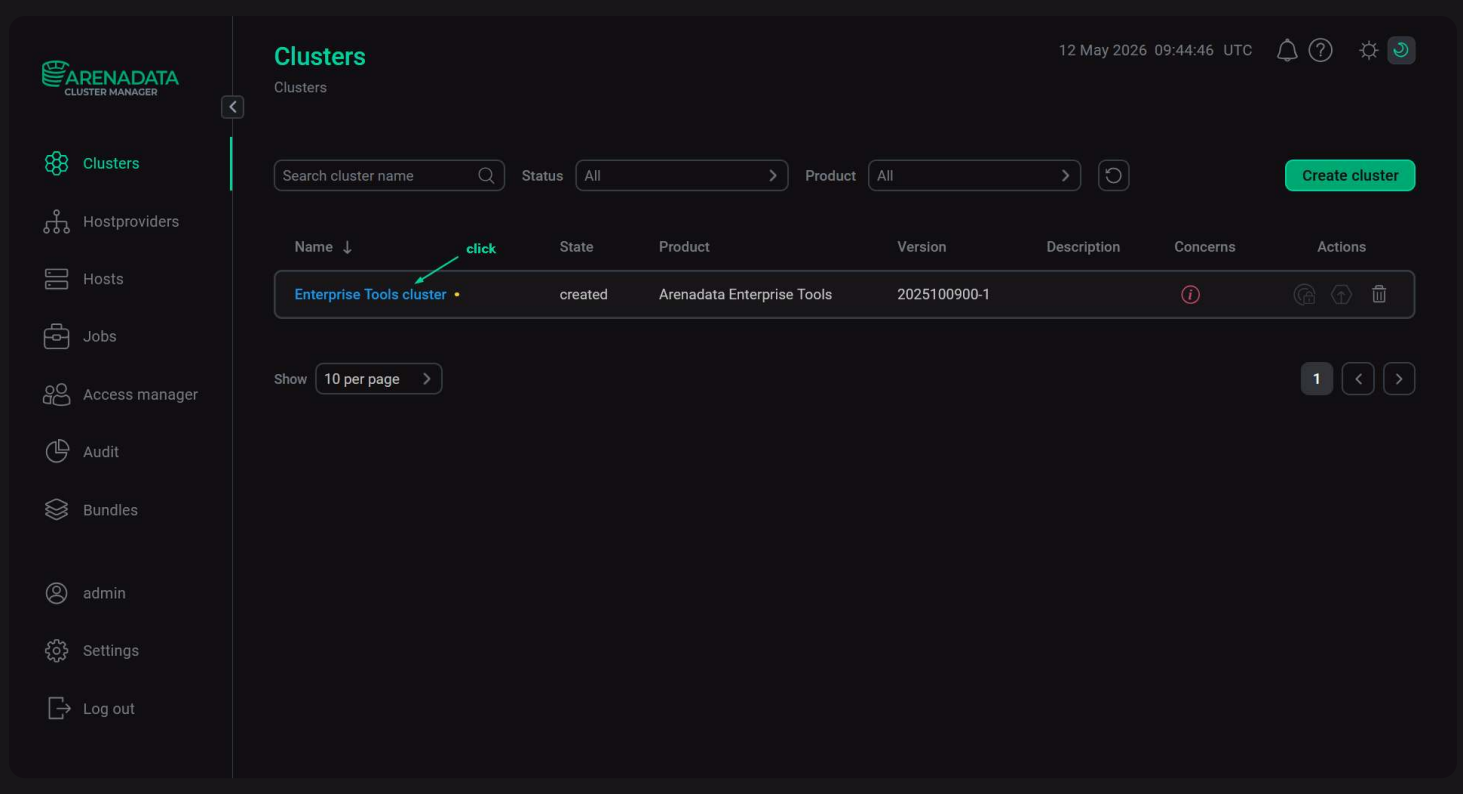
Add hosts to a cluster

Lisa Vaysman, Daria Barysheva

To Table of Contents

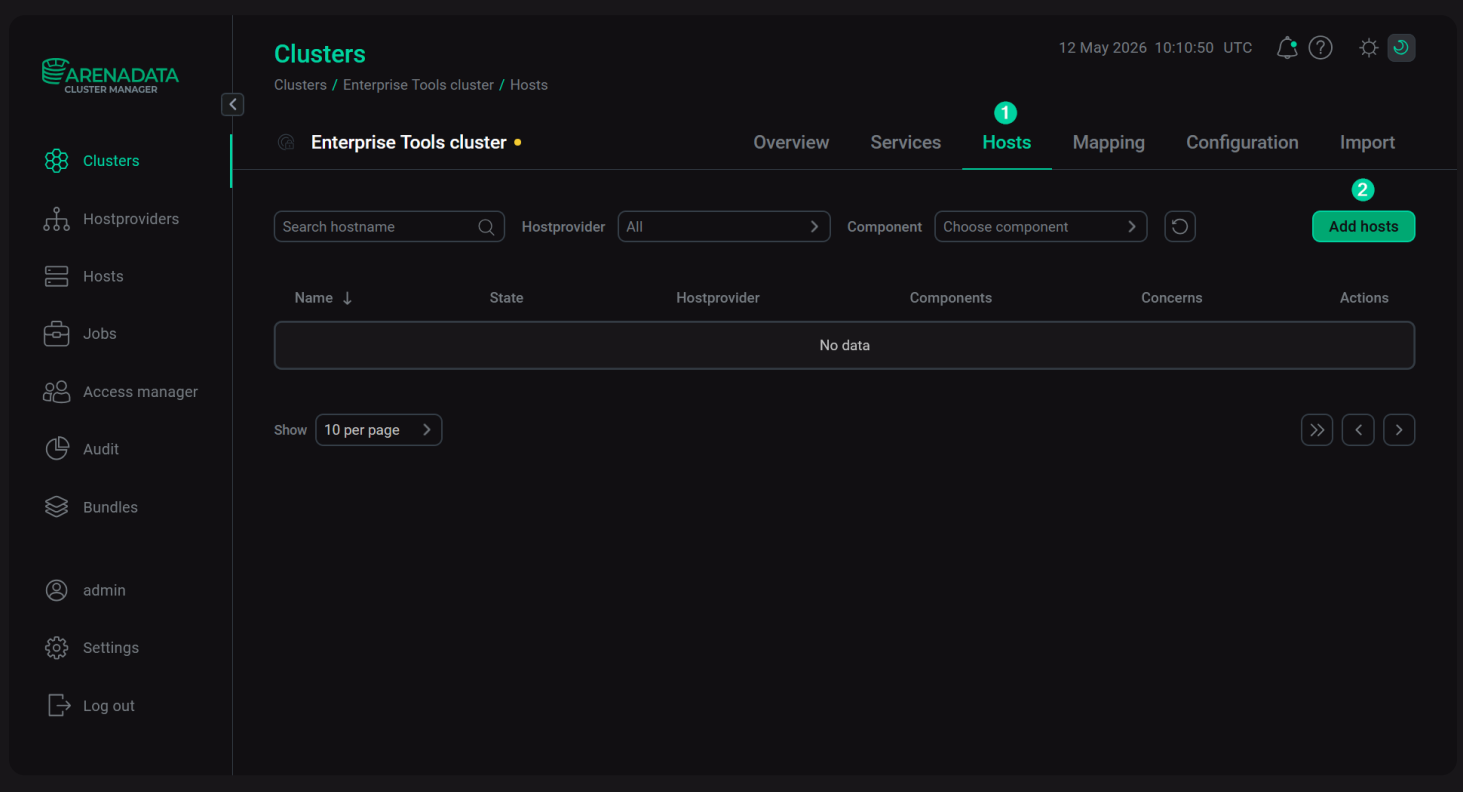
The steps for adding hosts to a cluster are listed below:

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



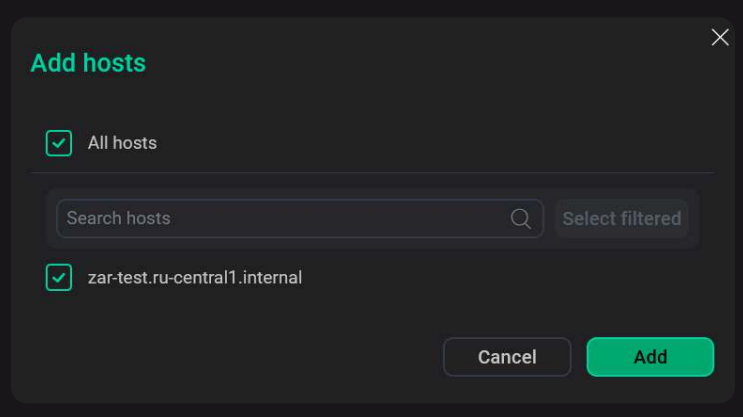
Select a cluster

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



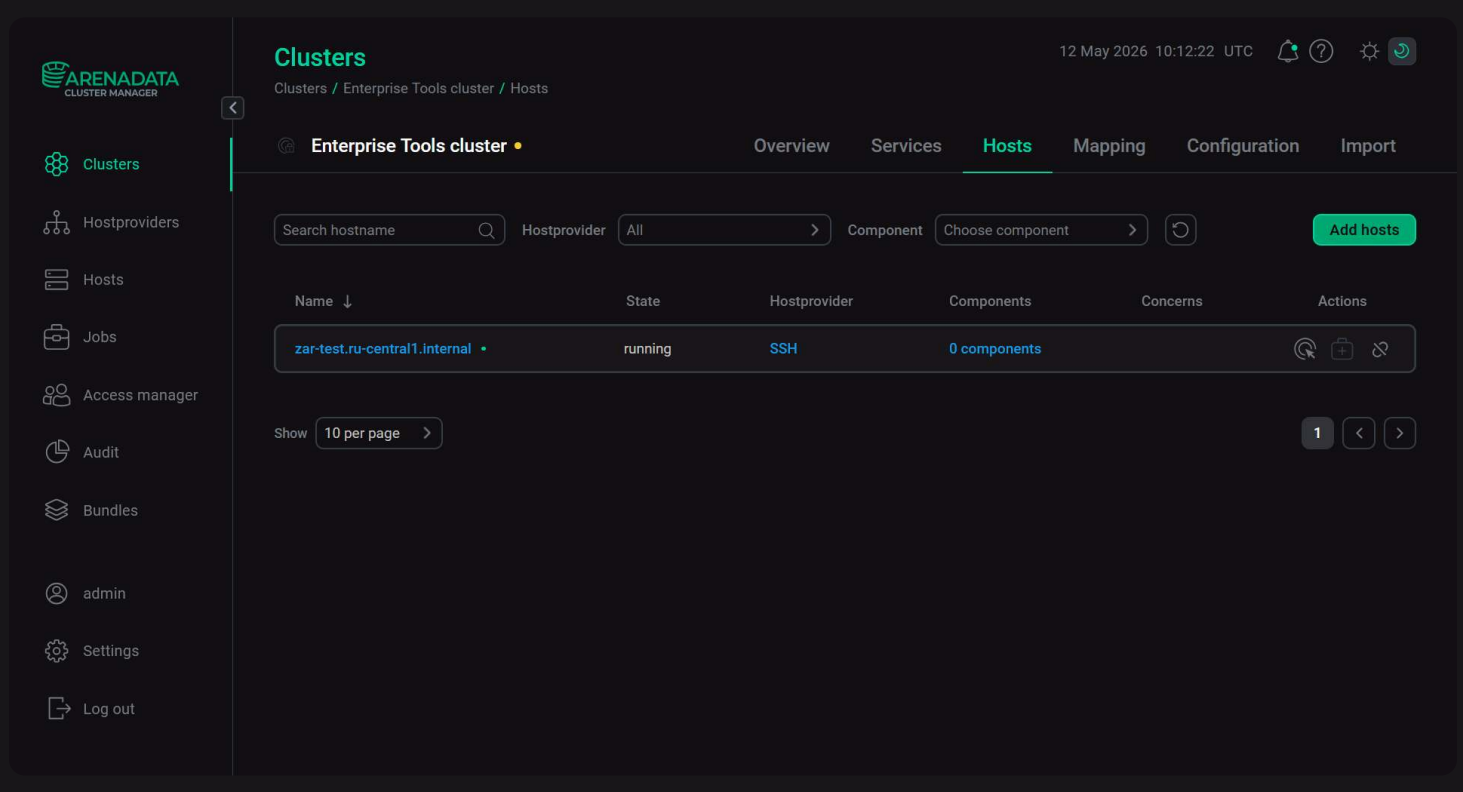
Switch to adding hosts

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



Select hosts

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



The result of successful adding hosts to a cluster

Add components

Lisa Vaysman, Daria Barysheva

To Table of Contents

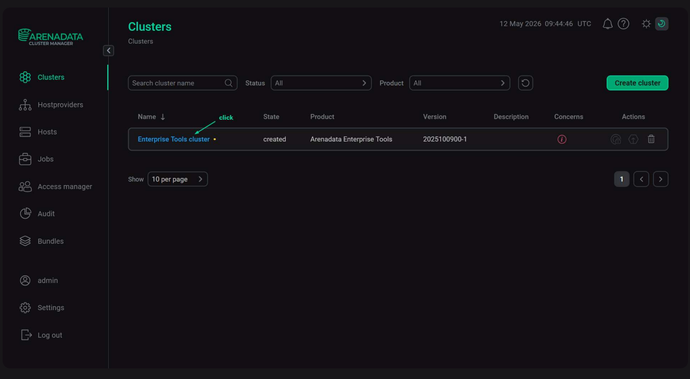
In ADCM, a component means a part of a service that should be deployed on one or several cluster hosts.

IMPORTANT

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

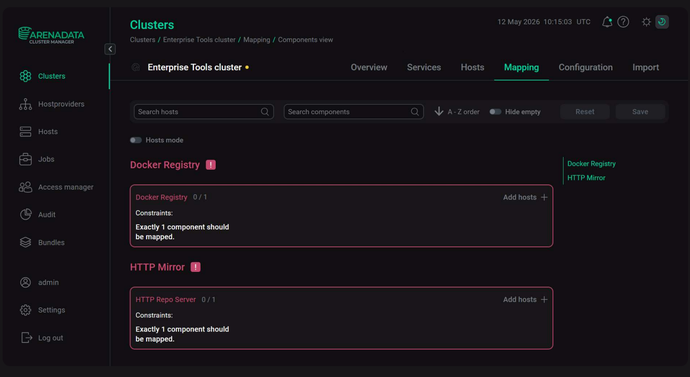
The steps for adding components to hosts are listed below:

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



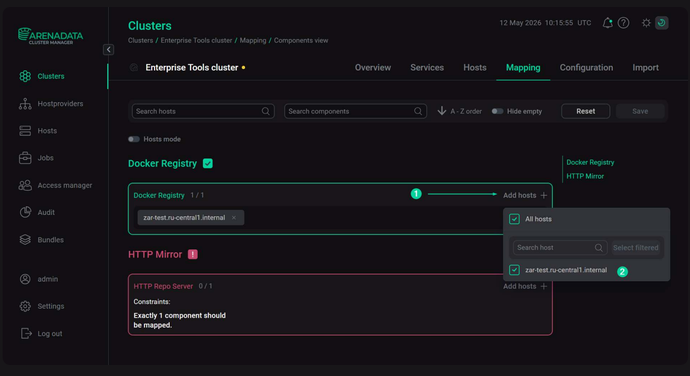
Select a cluster

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



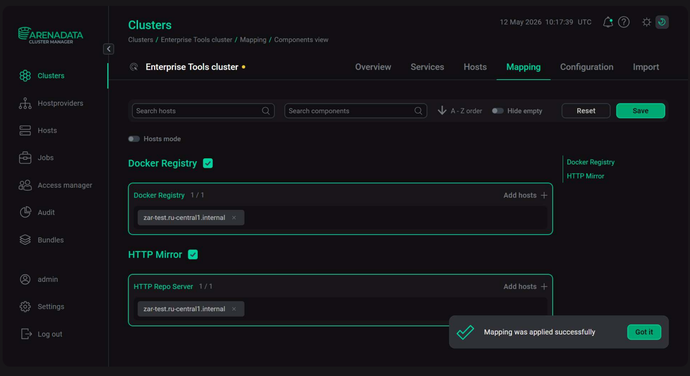
Switch to mapping service components

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



Select hosts for a component

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



Save mapping of components

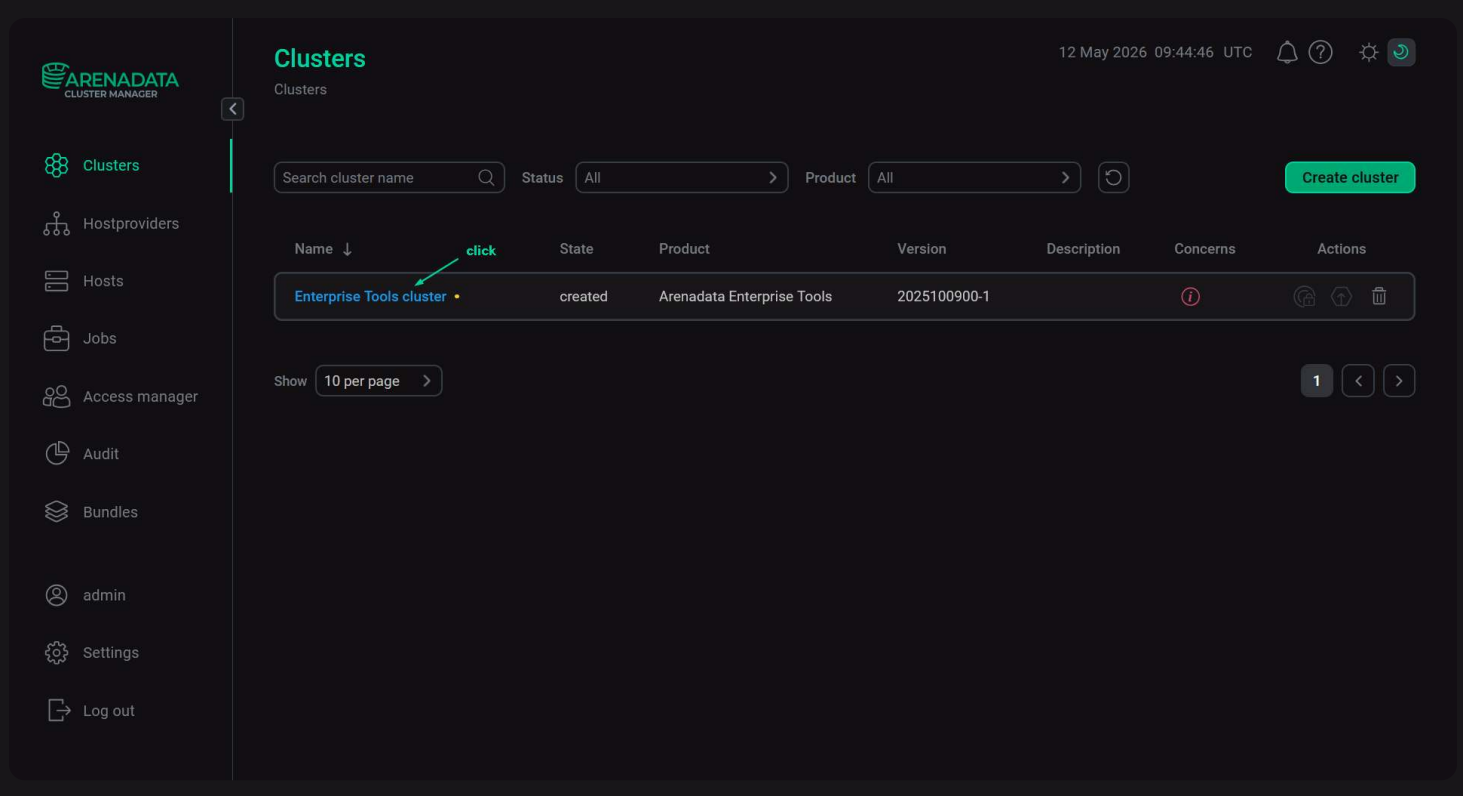
Configure services

Lisa Vaysman, Daria Barysheva

To Table of Contents

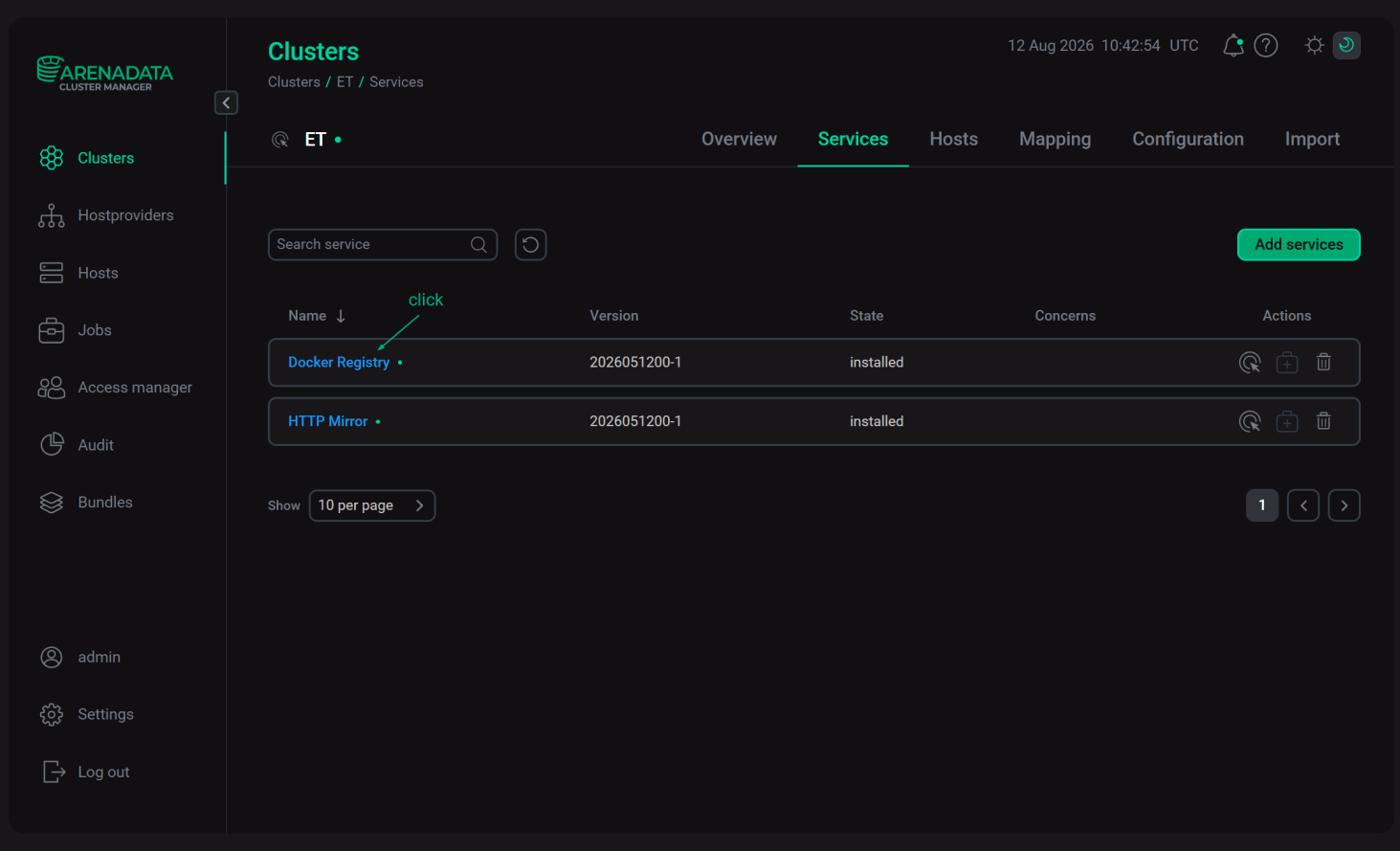
After adding all necessary services, you can configure its parameters. The steps for configuring cluster services are listed below:

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



Select a cluster

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



Switch to configuring a service

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

Configure a service

NOTE

For more information about service configuration parameters, see Configuration parameters.

Contents

Table Of Contents

Primary configuration
- Parameters
- Prerequisites for "Use existing docker"
  - Docker
  - Podman with Docker compatibility

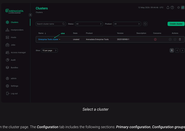
Configuration groups
Ansible settings

IMPORTANT

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

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

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



Select a cluster

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

Primary configuration

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

Parameters

Available configuration parameters are listed below:

- The **Timeout to activate package** parameter. It is displayed if the **Show advanced** switch is activated. The section contains the following parameters:
  - **retries** – the number of retries for cluster operations (`3` by default).
  - **delay** – the delay between retries in seconds (`30` by default).
- **Use custom registry** section. Using this parameter group is generally not recommended. Enable this parameter group only if you need to install an ADCM cluster from a custom Docker registry section. Using this parameter overrides the required ADCM images, such as `dl.frieslab.com` or `dl.arenadata.io`, that are subsequent in the custom Docker registry. For a seamless offline ADCM installation, using a custom Docker Registry is not required, because after installation ADCM provides Docker Registry and HTTP mirror services on its own. The section contains the following parameters:
  - **registry** – network address of the custom Docker Registry in the `<host>:<port>` format. For example, `registry.company.local:5043`. During ADCM cluster installation, ADCM will request ADCT service images from this registry. Custom registries that require authentication with a login and password are not supported. ADCM checks the registry address during cluster installation, and the installation fails if it is unavailable.
  - **path** – repository path within the custom Docker Registry where the necessary images are stored. Nested paths are supported, for example *dataplane/subfolder*.
- **Use self-signed certificate** – flag that defines that you trust the Docker registry's TLS certificate:
  - **false** – allows only a certificate issued by a trusted Certificate Authority (CA).
  - **true** – allows using a self-signed certificate. Set this flag to **true** if your Docker Registry uses an internal or manually generated certificate; otherwise, Docker will block the connection.

NOTE

Currently, the self-signed certificate specified in the **Self-signed certificate** field is not automatically added to the trusted certificate storage on cluster nodes. So before deploying the cluster, manually place the certificate on each node of the cluster to be used by Docker for certificates of this registry.

For example, on Ubuntu, save the certificate as *ca.docker.crt* in */usr/local/share/ca-certificates/* (a the registry address (for example, `registry.company.local:5043`).

If the certificate is not added manually, putting images from the custom registry may fail with a TLS verification error, for example: `tls: failed to verify certificate: x509: certificate signed by unknown authority`.

- **Self-signed certificate** – flag that defines that you trust the Docker registry's TLS certificate:
  - **false** – ADCM only uses a certificate issued by a trusted Certificate Authority (CA).
  - **true** – ADCM checks the Docker registry's self-signed certificate. Set this flag to **true** if your Docker Registry uses an internal or manually generated certificate; otherwise, Docker will block the connection.
- The **Use existing docker** section. This parameter defines whether ADCM installs and manages Docker on cluster hosts or uses an existing Docker-compatible environment. Possible values:
  - **false** – recommended — ADCM fully controls the container environment. During cluster deployment, ADCM automatically installs, configures, and starts a validated version of Docker.
  - **true** – ADCM skips the Docker installation step and connects to an existing Docker-compatible container engine on ADET hosts. If you use this option, verify the existing Docker engine as described in [Prerequisites for "Use existing docker"](#).
- The read-only **OS** section. The parameters in this section can be changed via the Manage OS. ADET cluster action.

Prerequisites for "Use existing docker"

If **Use existing docker** is set to **true**, verify one of the following Docker-compatible environments on each ADET host.

Docker

Use this scenario if Docker is installed on ADET hosts.

1. Check that the Docker CLI is installed and available as `docker`:

```
$ docker --version
```

Sample result:

```
Docker version 29.1.1, build f1268e4
```

2. Check that the Docker socket and service are active:

```
$ systemctl is-active docker.socket
$ systemctl is-active docker.service
```

Sample result:

```
active
active
```

3. Check that the Docker CLI can connect to the Docker engine:

```
$ sudo docker info
```

Sample result


```
Client:
 Version:    29.1.3
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Docker Buildx (Docker Inc.)
  Version:  v0.30.1
  Path:     /usr/libexec/docker/cli-plugins/docker-buildx

Server:
 Containers: 4
  Running: 4
  Paused: 0
  Stopped: 0
 Images: 4
 Server Version: 29.1.3
 Storage Driver: overlay2
  Driver-type: io.containerd.snapshotter.v1
 Logging Driver: json-file
 Cgroup Driver: systemd
 Cgroup Version: 2
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog
 CDI spec directories:
  /etc/cdi
  /var/run/cdi
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version:
 runc version:
 init version:
 Security Options:
  apparmor
  seccomp
   Profile: builtin
  cgroupns
 Kernel Version: 5.15.0-105-generic
 Operating System: Ubuntu 22.04.5 LTS
 OSType: linux
 Architecture: x86_64
 CPUs: 4
 Total Memory: 7.753GiB
 Name: vm-at-s
 ID: 5ee2f823-ea2c-45c5-b53b-f075ef1b4ca9
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 Experimental: false
 Insecure Registries:
  ::1/128
  127.0.0.0/8
 Live Restore Enabled: false
 Firewall Backend: iptables
```


Podman with Docker compatibility

Use this scenario if you want to use Podman as the container engine on ADET hosts.

NOTE

ADET does not currently support calling `podman` directly or using Podman without Docker compatibility.

For distributions with Podman, install the `podman-docker` package on each ADET host. This package provides a Docker-compatible CLI.

1. Check that the Docker-compatible CLI is installed and available as `docker`:

```
$ docker --version
```

Sample result

```
Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.
podman version 4.9.3
```

2. Check that the Podman socket is active:

```
$ systemctl is-active podman.socket
```

Sample result

```
active
```

3. Check that the Docker-compatible CLI can connect to Podman:

```
$ sudo docker info
```

Sample result


```
Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.
host:
  arch: amd64
  buildahVersion: 1.33.7
  cgroupControllers:
  - cpuset
  - cpu
  - io
  - memory
  - hugetlb
  - pids
  - rdma
  - misc
  cgroupManager: systemd
  cgroupVersion: v2
  conmon:
    package: conmon-2.1.10-1+b1+ld12_amd64
    path: /usr/bin/conmon
    version: 'conmon version 2.1.10, commit: unknown'
  ociRuntime:
    defaultPath: /bin/runc
    systemPath: /bin/runc
    version: 'runc version 1.1.12'
  os: linux
  pasta:
    executable: /usr/bin/pasta
    package: passt_0.0~git20240220.1e6f92b-1_amd64
    version: |
      pasta unknown version
      Copyright Red Hat
      GNU General Public License, version 2 or later
        <https://www.gnu.org/licenses/gpl-2.0.html>
      This is free software: you are free to change and redistribute it.
      There is NO WARRANTY, to the extent permitted by law.
  remoteSocket:
    exists: true
    path: /run/podman/podman.sock
  security:
    apparmorEnabled: true
    capabilities: CAP_CHOWN,CAP_DAC_OVERRIDE,CAP_FOWNER,CAP_FSETID,CAP_KILL,CAP_
NET_BIND_SERVICE,CAP_SETFCAP,CAP_SETGID,CAP_SETPCAP,CAP_SETUID,CAP_SYS_CHROOT
    rootless: false
    seccompEnabled: true
    seccompProfilePath: /usr/share/containers/seccomp.json
    selinuxEnabled: false
  serviceIsRemote: false
  slirp4netns:
    executable: /usr/bin/slirp4netns
    package: slirp4netns_1.2.1-1build2_amd64
    version: |-
      slirp4netns version 1.2.1
      commit: 09e31e92fa3d68c52ced0eb42ce2cbe7b7df3af4
      libslirp: 4.7.0
      SLIRP_CONFIG_VERSION_MAX: 4
      libseccomp: 2.5.5
  swapFree: 0
  swapTotal: 0
  uptime: 0h 3m 59.00s
  variant: ""
plugins:
  authorization: null
  log:
  - k8s-file
  - none
  - passthrough
  - journald
  network:
  - bridge
  - macvlan
  - ipvlan
  volume:
  - local
registries: {}
store:
  configFile: /usr/share/containers/storage.conf
  containerStore:
    number: 0
    paused: 0
    running: 0
    stopped: 0
  graphDriverName: overlay
  graphOptions: {}
  graphRoot: /var/lib/containers/storage
  graphRootAllocated: 15625357952
  graphRootUsed: 1478624000
  imageStore:
    number: 0
  runRoot: /run/containers/storage
  transientStore: false
  volumePath: /var/lib/containers/storage/volumes
version:
  APIVersion: 4.9.3
  Built: 0
  BuiltTime: Thu Jan  1 00:00:00 1970
  GitCommit: ""
  GoVersion: go1.22.2
  Os: linux
  OsArch: linux/amd64
  Version: 4.9.3
```


Configuration groups

The **Configuration** → **Configuration groups** tab is designed to add [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 [2.1.0](#).

Ansible settings

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

Install a cluster

Lisa Vaysman, Daria Barysheva

AD.ET

Contents

To Table of Contents

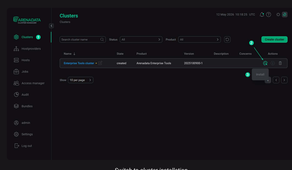
Step 1. Run installation

Step 2. Upload packages for offline installation

Step 1. Run installation

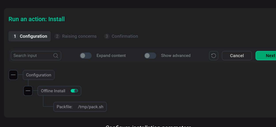
The last step of adding a new Arenadata Enterprise Tools cluster is to install its services. To do this:

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



Switch to cluster installation

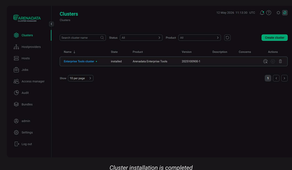
2. In the opened dialog box, on the ***Configuration*** tab:
   - Enable the ***Offline install*** toggle.
   - In the ***Packfile*** field, enter a full path to the unpacked shell script received from the Arenadata support team. The file must be located on the ADET host in a directory where script execution is allowed by the operating system security policy.
   - Click ***Next***



Configure installation parameters

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

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



Cluster installation is completed

Step 2. Upload packages for offline installation

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

Disk space consumption specifics

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

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

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

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

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

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

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

1. Determine the Docker data storage directory:

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

The result should look like this:

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

2. Check available disk space:

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

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

3. Check actual disk space usage:

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

4. If necessary, remove unused Docker objects:

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

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

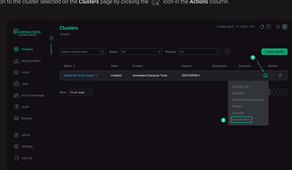
To upload the packages, perform the following steps:

1. Copy the pack file received from the Arenadata support team to the host where the ADET cluster is deployed.
   The pack file is a self-extracting shell script that contains product packages and images required for offline installation. It is usually distributed as a compressed archive with a name similar to *adqm_v26.3.3.20_arenadata1_ee_centos_7_x86_64.sh.xz*.
   Before running the ***Upload Pack*** action, unpack the archive on the ADET host, for example:

```
$ xz -d adqm_v26.3.3.20_arenadata1_ee_centos_7_x86_64.sh.xz
```

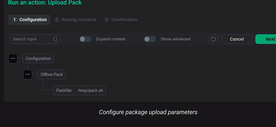
   After unpacking, the ***Packfile*** field should point to the resulting *.sh* file, not to the original *.xz* archive.

   You can place the pack file in any directory on the ADET host, for example, */tmp*. However, the selected directory must allow script execution according to the operating system security policy. In some distributions, for example, ALT Linux, execution of scripts from */tmp* can be prohibited. In this case, create and use another directory where script execution is allowed.
2. 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

3. In the opened dialog box, on the ***Configuration*** tab:
   - In the ***Packfile*** field, enter the full path to the unpacked pack file on the ADET host. The path must point to the *.sh* file located in a directory where script execution is allowed, for example: */tmp/adqm_v26.3.3.20_arenadata1_ee_centos_7_x86_64.sh*.
   - Click ***Next***



Configure package upload parameters

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

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

Diamond service management via ADCM

Dmitry Bulgakov

To Table of Contents

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

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

Action	Description
Restart	Restarts the service. When you run this action, the option ***Apply configs from ADCM*** is available. If it is set to `true`, all service configurations defined in ADCM settings will be applied during the service restart. Otherwise, the service restarts without applying configurations from ADCM

Diamond configuration parameters

Lisa Vaysman

To Table of Contents

To configure the service, use the following configuration parameters in ADCM.

NOTE

To apply a changed parameter in ADCM, perform one of the following actions:

- Restart the Diamond service using the Restart service action.
- Reinstall the ADET cluster using the Reinstall cluster action.

Diamond demon user and group

Parameter	Description	Default value
Diamond user	The username used to run the Diamond process	diamond
Diamond group	The group name used to run the Diamond process	diamond

Docker Registry service management via ADCM

Dmitry Bulgakov

To Table of Contents

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

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

Action	Description
Restart	Restarts the service. When you run this action, the option ***Apply configs from ADCM*** is available. If it is set to `true`, all service configurations defined in ADCM settings will be applied during the service restart. Otherwise, the service restarts without applying configurations from ADCM

Docker Registry configuration parameters

Lisa Vaysman

To Table of Contents

To configure the service, use the following configuration parameters in ADCM.

NOTE

To apply a changed parameter in ADCM, perform one of the following actions:

- Restart the Docker Registry service using the Restart service action.
- Reinstall the ADET cluster using the Reinstall cluster action.

Repositories

Parameter	Description	Default value
Working directory	The absolute path on the host where Docker Registry stores container images and installation artifacts. This directory is required for both the initial deployment and subsequent cluster operations. Do not delete the unpacked bundle installation files from this directory; otherwise, system package managers (`apt` / `dnf` / `yum`) will fail, causing ADCM UI actions to fail with errors	/opt/arenadata/etools/registry

HTTPS listener

Parameter	Description	Default value
Host	The IP address through which the Docker Registry repository is accessible to the hosts where Arenadata products are being deployed	—
HTTPS port	The port through which the Docker Registry repository is accessible	443
Server certificate	The TLS/HTTPS certificate for the Docker Registry repository in PEM (Privacy-Enhanced Mail) format. By default, this field contains the self-signed certificate generated during the installation of Arenadata products	—
Server private key	The cryptographic private key matching the server certificate. It is used to secure TLS/HTTPS connections to the Docker Registry repository. RSA algorithm is recommended. Note that ADCM does not automatically validate whether the private key matches the provided server certificate. Ensure that the key pair is correct before saving the configuration	—
TLS Min Version	The minimum allowed version of the TLS protocol. Possible values: - `tls1.0` - `tls1.1` - `tls1.2` - `tls1.3`	tls1.2

Enable Auth

Parameter	Description	Default value
Auth Type	Authentication method used to control access to the Docker Registry repository. If ***Enable Auth*** is enabled, this field is mandatory. Possible values: `token`, `htpasswd`. Use `htpasswd` to authenticate users with a username and password managed by Arenadata Enterprise Tools. Use `token` to delegate authentication to an external token service compatible with Docker Registry token authentication. For details, see Docker Registry authentication documentation	—
List of users	List of users allowed to access the Docker Registry repository when `htpasswd` authentication is used. You can add properties to this list in the `<username>:<password>` format. Arenadata Enterprise Tools uses this list to generate and manage the *.htpasswd* file that Docker Registry uses to verify user credentials	—
htpasswd Authentication realm	Authentication realm displayed to Docker clients when `htpasswd` authentication is requested. The realm identifies the protected access area for which the client must provide credentials. Example: `Registry Realm`	Registry Realm
htpasswd file path (managed by ET)	Read-only path to the ADET-managed *.htpasswd* file, which is used by Docker Registry for `htpasswd` authentication	/opt/arenadata/etools/registry/config/.htpasswd
Token auth realm URL	URL of the external endpoint used for token-based authentication. When a Docker client requests access to the registry, Docker Registry redirects the client to this URL to obtain an access token. Example: `http://10.1.1.1:8080/realms/registry/protocol/docker-v2/auth`. For details, see Docker Registry authentication documentation	—
Token issuer	Issuer of tokens used for token-based authentication. Docker Registry checks this value to verify that an access token was issued by a trusted token service. The value is case-sensitive and must exactly match the issuer value included in tokens by the token service. For example, if Keycloak is used and the realm configured for Docker Registry authentication is `registry`, the token issuer can be `http://<keycloak-address>:8080/realms/registry`	—
Token service name	Service name for which tokens must be issued. Docker Registry uses this value to verify that an access token is intended for this registry service. The value must match the service name used by the token service. For example, when Keycloak is used, this value can match the client ID configured for Docker Registry authentication. Only one service name is supported. The value must not contain spaces and special characters such as `&`, `?`, `#`, and `/`. Example: `registry-client`	—
CA bundle for token validation	CA bundle used by Docker Registry to validate tokens received from the external token service. The value must contain one certificate or a certificate bundle in PEM format. If you specify multiple certificates, place them one after another in the same value, with each certificate starting with `-----BEGIN CERTIFICATE-----` and ending with `-----END CERTIFICATE-----`	—

Other

Parameter	Description	Default value
Enable image deletion	Allows deleting images from the Docker Registry repository. When enabled, clients can delete image manifests and tags through the Docker Registry API. Deleted data may not be removed from disk immediately	false
Storage directory	Directory where Docker Registry stores its internal data	/var/lib/registry
Registry listen address	Address and port on which Docker Registry listens for incoming connections. This address can differ from the external HTTPS port used by clients to access the registry. The default value `0.0.0.0:8443` means that Docker Registry listens on port `8443` on all available network interfaces	0.0.0.0:8443
Log verbosity	Logging level used by Docker Registry. Possible values ordered from most detailed to least detailed: `debug`, `info`, `warn`, `error`	info
Log formatter	Format used for Docker Registry logs. Possible values: `text`, `json`, `logstash`	text
Container restart policy	Restart policy applied to the Docker Registry container. Possible values: `no`, `always`, `on-failure`, `unless-stopped`. The default value `always` helps keep Docker Registry available by restarting the container after failures or host restarts. See details in Docker Engine documentation	always
Custom Configuration Overrides	Custom configuration parameters that override the default Docker Registry configuration. Custom configuration parameters have priority over individual configuration fields. Use this parameter for advanced settings that are not available as individual configuration fields. Nested keys are specified via dot notation, for example, `storage.delete.enabled`	—

Grafana service management via ADCM

Dmitry Bulgakov

To Table of Contents

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

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

Action	Description
Restart	Restarts the service. When you run this action, the option ***Apply configs from ADCM*** is available. If it is set to `true`, all service configurations defined in ADCM settings will be applied during the service restart. Otherwise, the service restarts without applying configurations from ADCM

Grafana configuration parameters

Lisa Vaysman

To Table of Contents

To configure the service, use the following configuration parameters in ADCM.

NOTE

To apply a changed parameter in ADCM, perform one of the following actions:

- Restart the Grafana service using the Restart service action.
- Reinstall the ADET cluster using the Reinstall cluster action.

security

Parameter	Description	Default value
Username	Grafana administrator username used to log in to the Grafana web interface	admin
Password	Password for the Grafana administrator	—
Allow Sign Up	Option for allowing users to self-register an account	false
Anonymous enabled	Option that makes Grafana accessible without requiring a login	true

HTTP Listener

Parameter	Description	Default value
Host	The IP address of the Grafana web interface. By default, this field is filled in automatically during the installation	—
Port	The network port used by Grafana to send and receive data	3000

Other

Parameter	Description	Default value
HTTPS listener	Read-only group of settings. To change HTTPS listener settings, use the Manage SSL cluster action	—
Plugins	A comma-separated list of Grafana plugins that will be installed automatically inside the Docker container when Grafana starts (e.g. `grafana-clock-panel,grafana-simple-json-datasource`). For details, see Grafana documentation	—
Default UI theme	The visual theme, which is used by default for the Grafana dashboards	Dark

Paths

Parameter	Description	Default value
Configuration directory	The path to the directory that contains Grafana configuration files	/etc/grafana
Logs directory	The path to the directory for storing Grafana service logs	/var/log/grafana

SMTP

Parameter	Description	Default value
SMTP host and port	The IP address and the port of the SMTP mail server. This parameter is mandatory when the ***SMTP*** toggle is enabled	—
SMTP username	The username used to authenticate against the SMTP server. Leave this field empty if the mail server does not require authentication	—
SMTP password	The password corresponding to the specified SMTP username. Leave this field empty if the mail server does not require authentication	—
SMTP FROM ADDRESS	The sender email address that will appear in the "From" header of all automated notifications (for example, `noreply@example.com`)	—
SMTP FROM NAME	Sender's name displayed in email notifications (for example, `Grafana Alerts`)	—

Proxy settings

Parameter	Description	Default value
HTTP proxy	The address (URL or IP address) and the port of the proxy server, which is used for HTTP requests. If the ***Proxy settings*** toggle is enabled, then this field is mandatory. Example: `192.168.1.254:8080` or `http://example.com:8080`	—
HTTPS proxy	The address (URL or IP address) and the port of the proxy server, which is used for HTTPS requests. If the ***Proxy settings*** toggle is enabled, then this field is mandatory. Example: `192.168.1.254:8080` or `https://example.com:8080`	—
No proxy	A comma-separated list of domain names, IP addresses, or subnets that must be accessed directly, bypassing the proxy server. Crucial for internal cluster component communication. Example: `example.com`	localhost,127.0.0.1,::1

Graphite service management via ADCM

Dmitry Bulgakov

To Table of Contents

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

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

Action	Description
Restart	Restarts the service. When you run this action, the option ***Apply configs from ADCM*** is available. If it is set to `true`, all service configurations defined in ADCM settings will be applied during the service restart. Otherwise, the service restarts without applying configurations from ADCM

Graphite configuration parameters

Lisa Vaysman

To Table of Contents

To configure the service, use the following configuration parameters in ADCM.

NOTE

To apply a changed parameter in ADCM, perform one of the following actions:

- Restart the Graphite service using the Restart service action. Additionally, restart the Grafana service (this is necessary since Grafana uses Graphite).
- Reinstall the ADET cluster using the Reinstall cluster action.

Paths

Parameter	Description	Default value
Data storage	A path to the directory for storing metrics	/opt/monitoring
Configuration directory	A path to the directory that contains Graphite configuration files	/etc/graphite
Logs directory	A path to the directory for storing Graphite service logs	/var/log/graphite

ip_and_ports

Parameter	Description	Default value
Host IP address	IP address where the metrics receiver is available. Filled automatically after the installation is completed or can be set manually	—
Web-interface TCP port	TCP port on which the Graphite web interface is available	80
Metrics receiver TCP port	TCP port used by the metrics receiver to accept data in Carbon plaintext protocol format. For details, see Graphite documentation	2015
Metrics receiver UDP port	UDP port used by the metrics receiver to accept data in Carbon plaintext protocol format. For details, see Graphite documentation	2016

Other

Parameter	Description	Default value
Carbon services command-line options	Additional command-line arguments for all Carbon services (e.g. `--debug` for debug logging)	—

Storage schema

Parameter	Description	Default value
Carbon retentions	Retention schema configuration for storing metrics in the Carbon cache, defined as `<precision>:<retention_period>`, where - `<precision>` is the time interval between individual datapoints; - `<retention_period>` is the total duration for keeping data. For example, `60:90d` value saves 1 datapoint every 60 seconds and keeps data for 90 days	60:90d
Default graphite retentions	The rules that control how long Graphite keeps data and how data granularity decreases over time. The value is a comma-separated list of retention stages, ordered from the newest to the oldest. When data moves to an older stage, Graphite automatically merges multiple high-granularity datapoints into an individual lower-granularity datapoint. Each stage uses the `<precision>:<retention_period>` format, where - `<precision>` is the time interval between individual datapoints; - `<retention_period>` is the total duration for keeping data. Possible time units: `s` (seconds), `m` (minutes), `h` (hours), `d` (days), `w` (weeks), or `y` (years). For example, `60s:90d,10m:2y` value keeps high-granularity data (one datapoint every 60 seconds) for 90 days. After 90 days, the service aggregates these datapoints into one historical datapoint every 10 minutes and stores them for 2 years	60s:90d,10m:2y

HTTP Mirror service management via ADCM

Dmitry Bulgakov

To Table of Contents

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

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

Action	Description
Restart	Restarts the service. When you run this action, the option ***Apply configs from ADCM*** is available. If it is set to `true`, all service configurations defined in ADCM settings will be applied during the service restart. Otherwise, the service restarts without applying configurations from ADCM

HTTP Mirror configuration parameters

Lisa Vaysman

To Table of Contents

To configure the service, use the following configuration parameters in ADCM.

NOTE

To apply a changed parameter in ADCM, perform one of the following actions:

- Restart the HTTP Mirror service using the Restart service action.
- Reinstall the ADET cluster using the Reinstall cluster action.

Repositories

Parameter	Description	Default value
Working directory	The absolute path to the local directory where the HTTP Mirror service stores its data	/opt/arenadata/etools/httprepo

HTTP listener

Parameter	Description	Default value
Host	The IP address on which the HTTP Mirror service listens for incoming connections. By default, this value is automatically created during the installation based on the actual IP address of the deployment host	—
HTTP port	The network port used by the HTTP Mirror service	81

Other

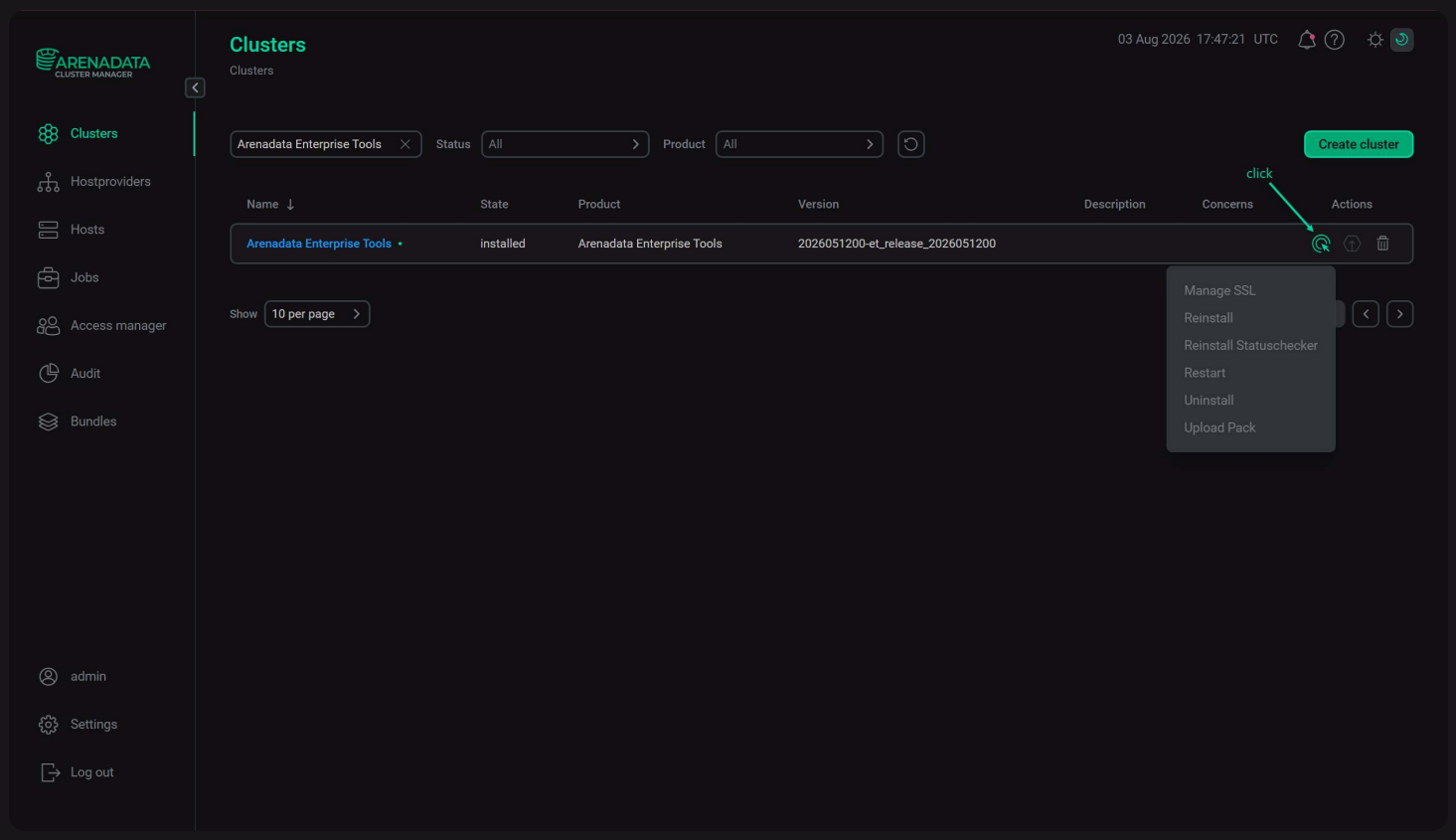
Parameter	Description	Default value
HTTPS listener	Read-only group of settings. To change HTTPS listener settings, use the Manage SSL cluster action	—

Cluster actions

Dmitry Bulgakov

To Table of Contents

You can manage an Arenadata Enterprise Tools cluster on the ***Clusters*** page in the ADCM UI.



ADET cluster in ADCM UI

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

The following actions are available for an ADET cluster:

- Manage SSL
- Reinstall
- Reinstall Statuschecker
- Restart
- Uninstall
- Upload Pack

Manage SSL

Dmitry Bulgakov

Contents

To Table of Contents

Enable SSL
- Manage SSL action parameters

Disable SSL in ADET

The ***Manage SSL*** action allows you to control the SSL encryption of services in an Arenadata Enterprise Tools cluster (Docker Registry, HTTP Mirror, Grafana).

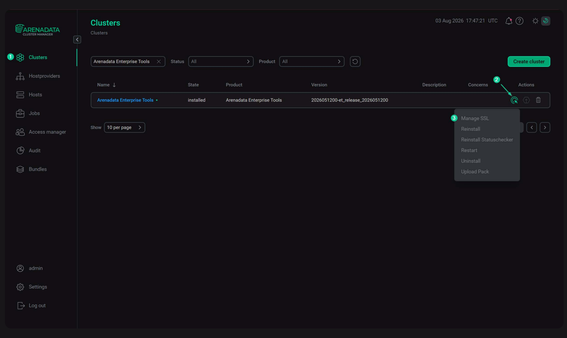
Key points when enabling SSL for an ADET cluster:

- Although some ADET services have their own HTTPS settings (***HTTPS listener*** block), it is strongly recommended to use the ***Manage SSL*** cluster action rather than configuring services individually.
- SSL certificates should be trusted by the ADET cluster host(s).
- Do not upgrade an ADET cluster with SSL enabled. Disable SSL first.
- During the execution of the ***Manage SSL*** action, ADCM reconfigures and restarts required services, so no explicit cluster restart is required after the action.

Enable SSL

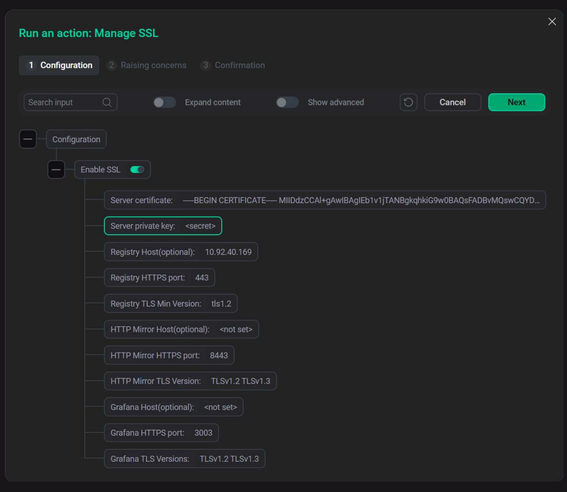
Follow the steps below to enable SSL in ADET:

1. Go to ***Clusters*** page, click in the ***Actions*** column, and select the ***Manage SSL*** action in the drop-down list.



Execute the "Manage SSL" action

2. Activate the ***Enable SSL*** configuration group and specify the SSL parameters.



SSL parameters

3. Click ***Next*** and then click ***Run*** to start the action.
4. You can find the execution process, logs, and result of the action on the ***Jobs*** page. On this page, you can view action execution details, including verbose Ansible STDOUT/STDERR outputs.

Manage SSL action parameters

When running the ***Manage SSL*** action, ADCM displays a dialog window where you can specify the following SSL parameters.

Parameter	Description	Default value
Server certificate	SSL certificate in the PEM format. SSL certificates must be trusted by ADET cluster host(s)	—
Server private key	Private key in the PEM format	—
Registry Host(optional)	Filled automatically with the IP address, where the Docker Registry web UI is available. Can be set manually for custom configuration	<et-host-ip>
Registry HTTPS port	HTTPS port for the Docker Registry service	443
Registry TLS Min Version	Minimum TLS version allowed for the Docker Registry service	tls1.2
HTTP Mirror Host(optional)	Filled automatically with the IP address, where the HTTP Mirror web UI is available. Can be set manually for custom configuration	—
HTTP Mirror HTTPS port	HTTPS port for the HTTP Mirror service	8443
HTTP Mirror TLS Version	TLS version allowed for the HTTP Mirror service	TLSv1.2 TLSv1.3
Grafana Host(optional)	Filled automatically with the IP address, where the Grafana web UI is available. Can be set manually for custom configuration	—
Grafana HTTPS port	HTTPS port for the Grafana service	3003
Grafana TLS Versions	TLS version allowed for the Grafana service	TLSv1.2 TLSv1.3

Disable SSL in ADET

To disable SSL for an ADET cluster, run the ***Manage SSL*** action with ***Enable SSL*** configuration group disabled.

Reinstall

Dmitry Bulgakov

Contents

To Table of Contents

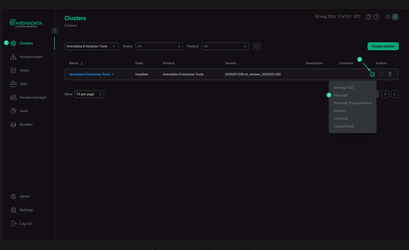
Online reinstall

Offline reinstall

Follow the steps below to reinstall an Arenadata Enterprise Tools cluster via ADCM.

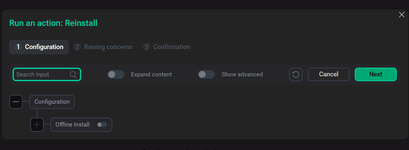
Online reinstall

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



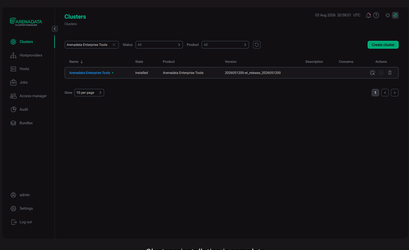
Execute the "Reinstall" action

2. In the opened dialog box, on the ***Configuration*** tab click ***Next***.



Configure reinstallation parameters

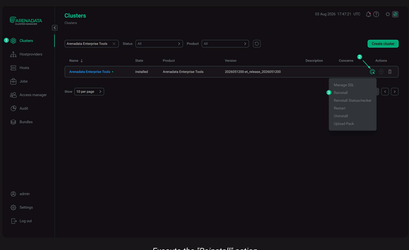
3. On the ***Raising concerns*** tab:
   - If necessary, enable the ***Raise non-blocking concern*** toggle to run the action without blocking the cluster during action execution.
   - Click ***Next***.
4. On the ***Confirmation*** tab:
   - Add a comment with a brief description of the changes.
   - Select the ***Verbose*** checkbox to display additional information about the action execution on the ***Jobs*** page.
   - Click ***Run***.
5. Wait until the reinstallation process is complete.
6. You can find the execution process, logs, and result of the action on the ***Jobs*** page. On this page, you can view action execution details, including verbose Ansible STDOUT/STDERR outputs.



Cluster reinstallation is complete

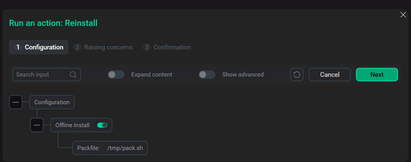
Offline reinstall

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



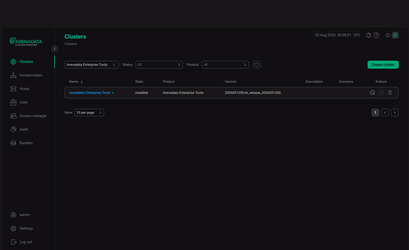
Execute the "Reinstall" action

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



Configure reinstallation parameters

3. On the ***Raising concerns*** tab:
   - If necessary, enable the ***Raise non-blocking concern*** toggle to run the action without blocking the cluster during action execution.
   - Click ***Next***.
4. On the ***Confirmation*** tab:
   - Add a comment with a brief description of the changes.
   - Select the ***Verbose*** checkbox to display additional information about the action execution on the ***Jobs*** page.
   - Click ***Run***.
5. Wait until the reinstallation process is complete.
6. You can find the execution process, logs, and result of the action on the ***Jobs*** page. On this page, you can view action execution details, including verbose Ansible STDOUT/STDERR outputs.



Cluster reinstallation is complete

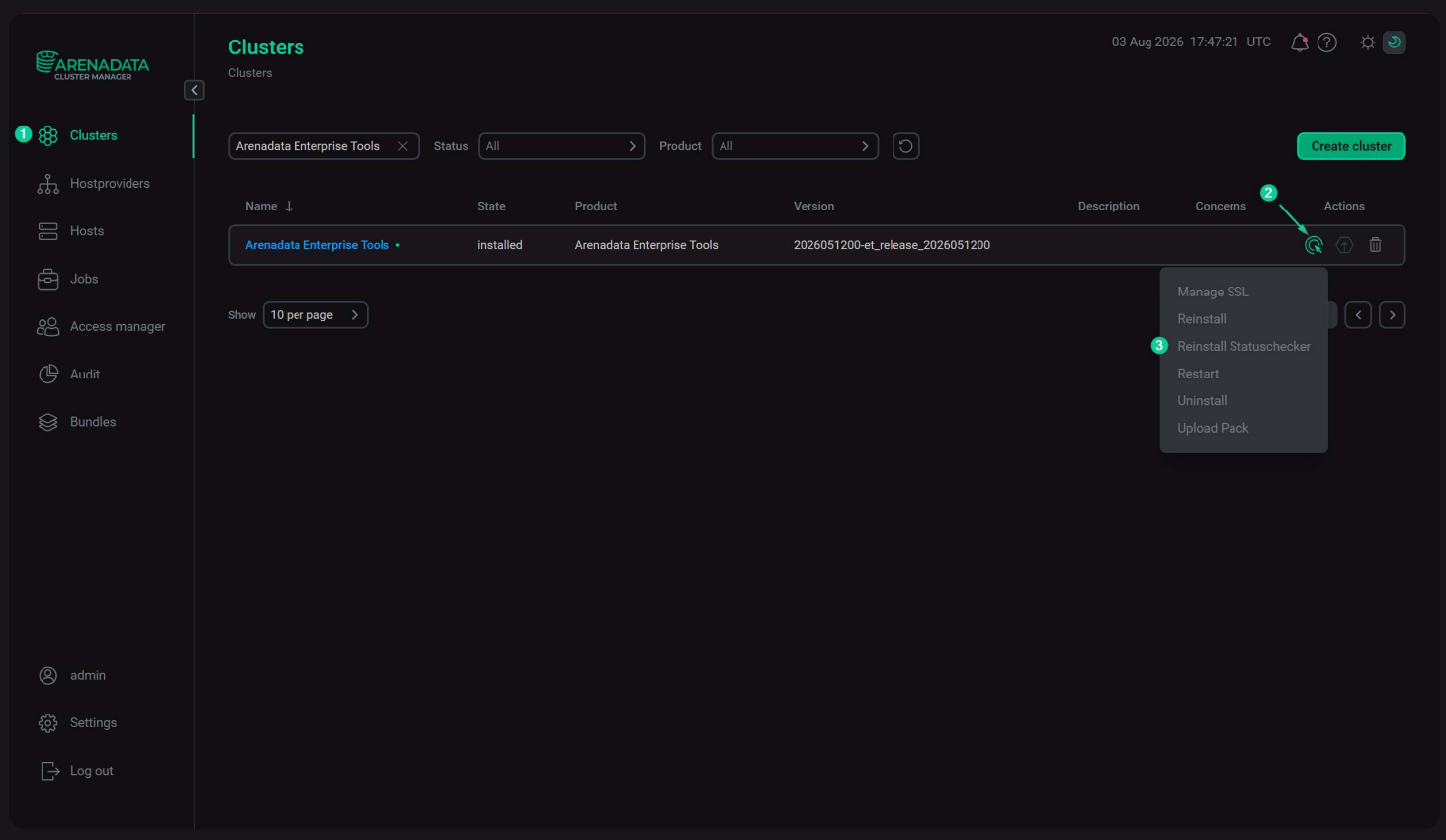
Reinstall Statuschecker

Dmitry Bulgakov

To Table of Contents

The ***Reinstall Statuschecker*** action reinstalls statuschecker for all services one by one.

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



Execute the "Reinstall Statuschecker" action

2. On the ***Raising concerns*** tab:
   - If necessary, enable the ***Raise non-blocking concern*** toggle to run the action without blocking the cluster during action execution.
   - Click ***Next***.
3. On the ***Confirmation*** tab:
   - Add a comment with a brief description of the changes.
   - Select the ***Verbose*** checkbox to display additional information about the action execution on the ***Jobs*** page.
   - Click ***Run***.
4. Wait until the statuschecker installation process is complete.
5. You can find the execution process, logs, and result of the action on the ***Jobs*** page. On this page, you can view action execution details, including verbose Ansible STDOUT/STDERR outputs.

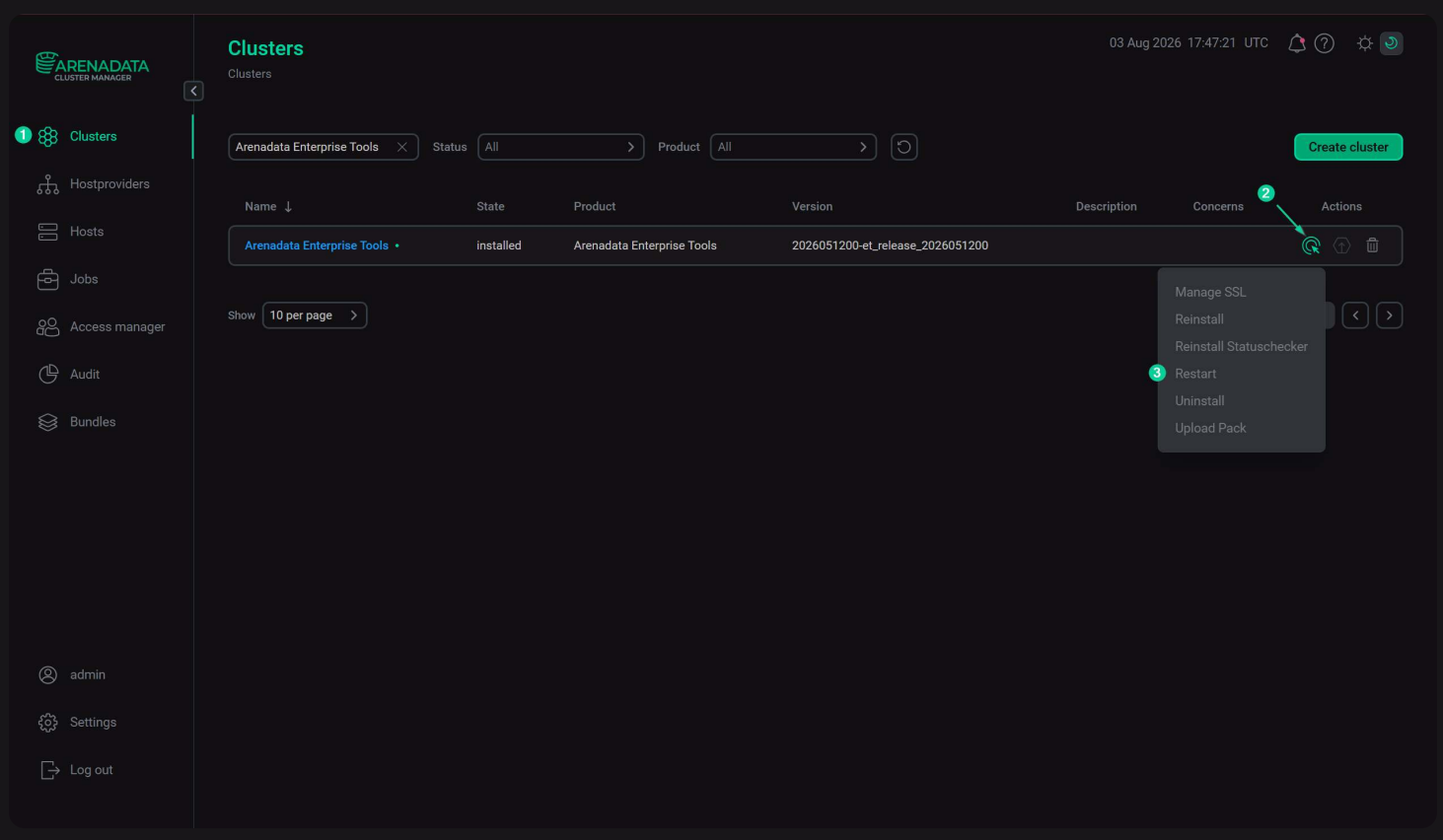
Restart

Dmitry Bulgakov

To Table of Contents

The ***Restart*** action stops and then starts over all cluster services and their components. During a restart cluster services are unavailable.

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



Execute the "Restart" action

2. On the ***Raising concerns*** tab:
   - If necessary, enable the ***Raise non-blocking concern*** toggle to run the action without blocking the cluster during action execution.
   - Click ***Next***.
3. On the ***Confirmation*** tab:
   - Add a comment with a brief description of the changes.
   - Select the ***Verbose*** checkbox to display additional information about the action execution on the ***Jobs*** page.
   - Click ***Run***.
4. You can find the execution process, logs, and result of the action on the ***Jobs*** page. On this page, you can view action execution details, including verbose Ansible STDOUT/STDERR outputs.

NOTE

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

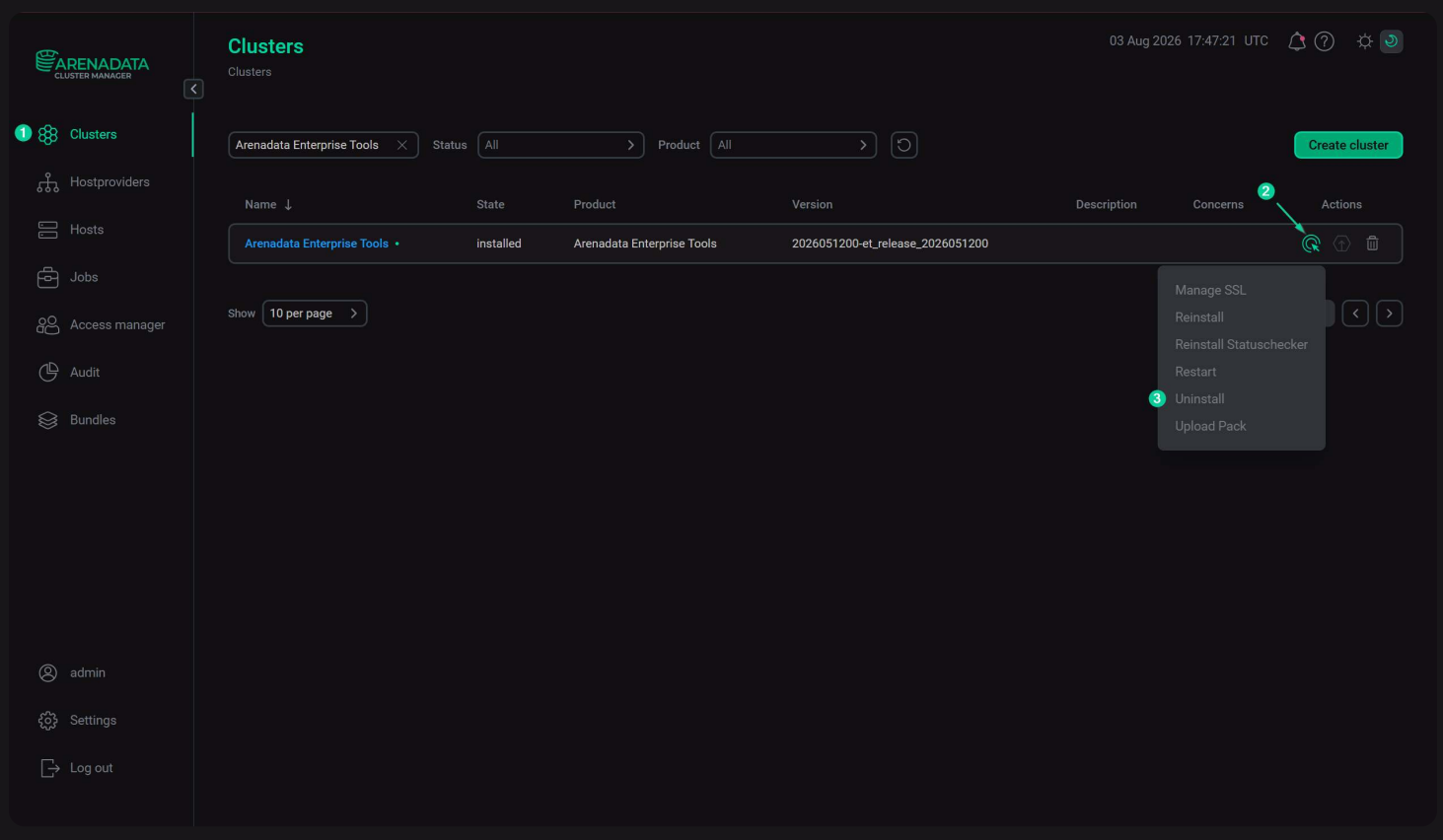
Uninstall

Dmitry Bulgakov

To Table of Contents

The ***Uninstall*** action stops and then uninstalls all cluster services, their components, and mapping.

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



Execute the "Uninstall" action

2. On the ***Raising concerns*** tab:
   - If necessary, enable the ***Raise non-blocking concern*** toggle to run the action without blocking the cluster during action execution.
   - Click ***Next***.
3. On the ***Confirmation*** tab:
   - Add a comment with a brief description of the changes.
   - Select the ***Verbose*** checkbox to display additional information about the action execution on the ***Jobs*** page.
   - Click ***Run***.
4. Wait until the uninstallation process is complete.
5. You can find the execution process, logs, and result of the action on the ***Jobs*** page. On this page, you can view action execution details, including verbose Ansible STDOUT/STDERR outputs.

IMPORTANT

The ***Uninstall*** action removes Enterprise Tools from the hosts where it was installed. This action should not be considered a procedure for preparing the hosts for secure reuse. If completely clean hosts are required, reinstall their operating systems to remove possible leftover artifacts.

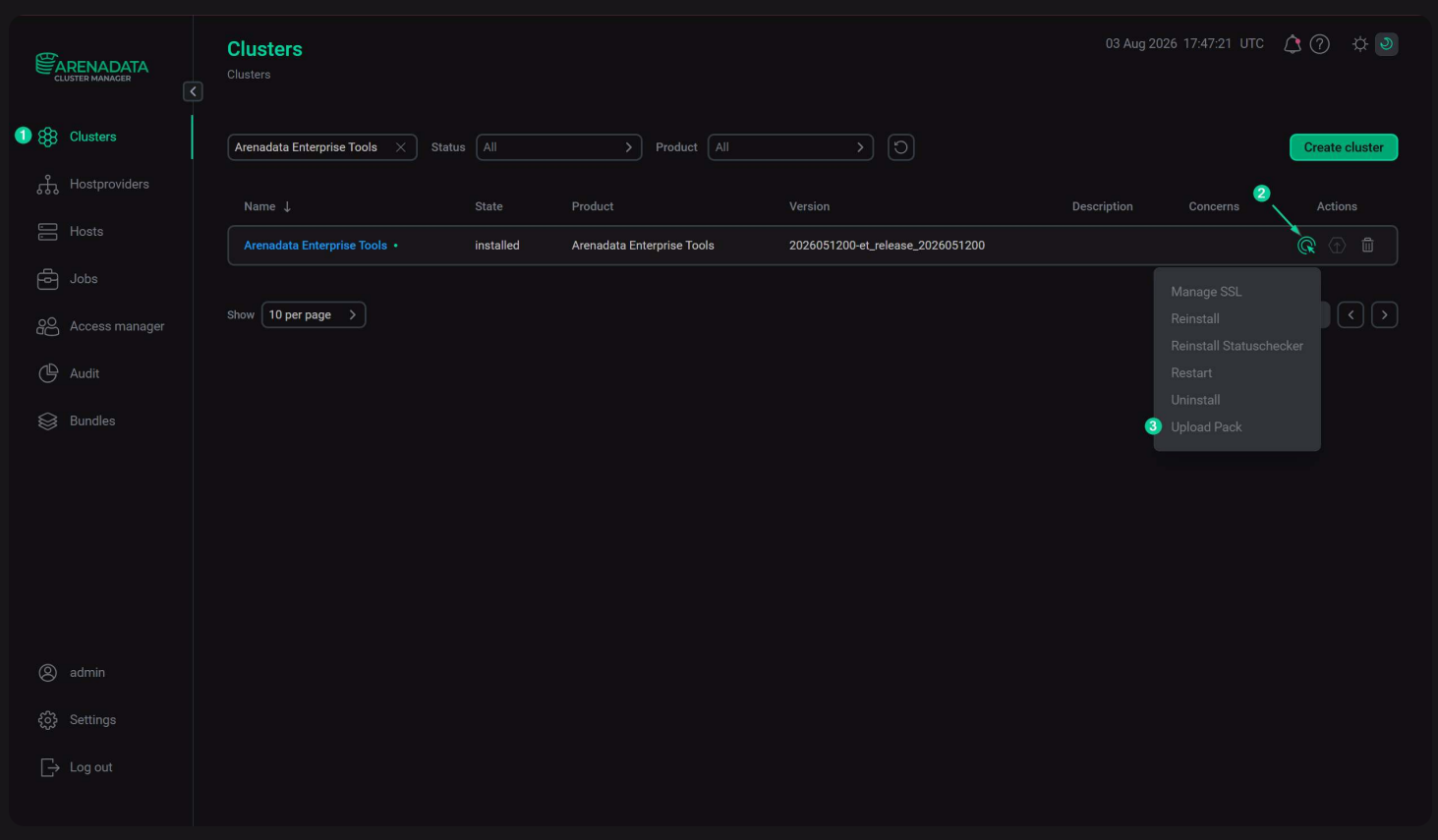
Upload Pack

Dmitry Bulgakov

To Table of Contents

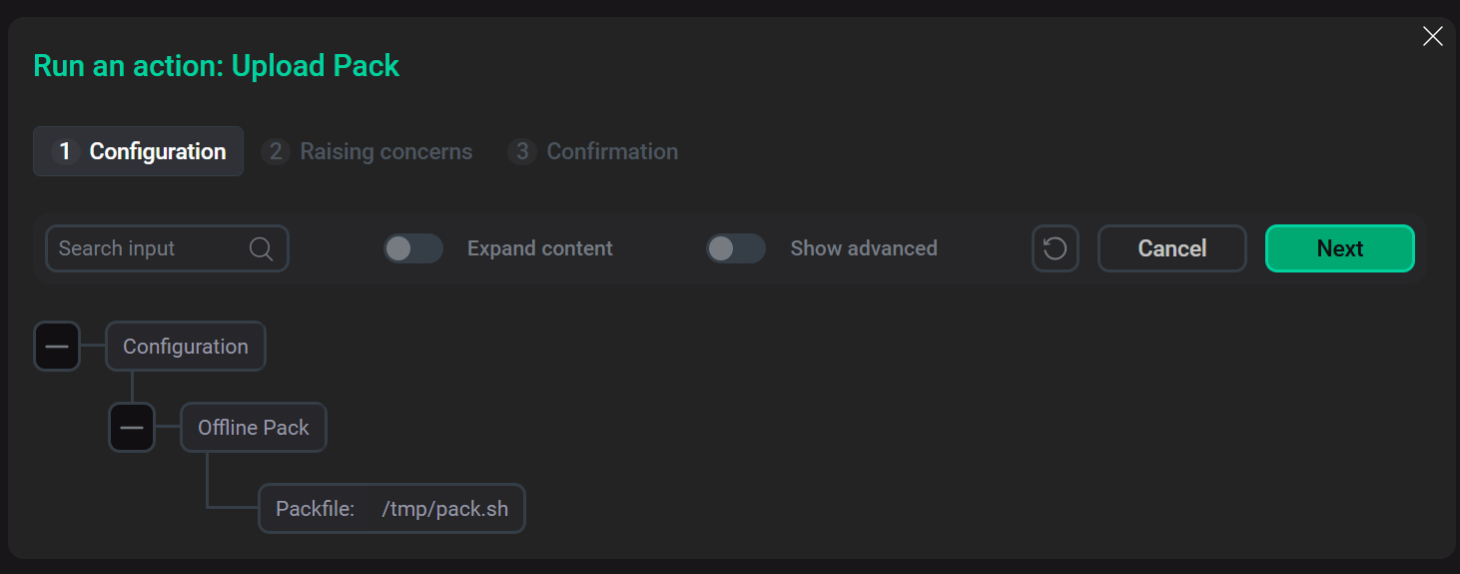
In ADCM, follow the steps below to upload a package for Arenadata Enterprise Tools cluster, which is used for offline Arenadata product installation. To understand the most common scenario of installation with offline packages, see Upload packages for offline installation.

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



Execute the "Upload Pack" action

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



Configure upload parameters

3. On the ***Raising concerns*** tab:
   - If necessary, enable the ***Raise non-blocking concern*** toggle to run the action without blocking the cluster during action execution.
   - Click ***Next***.

4. On the ***Confirmation*** tab:
   - Add a comment with a brief description of the changes.
   - Select the ***Verbose*** checkbox to display additional information about the action execution on the ***Jobs*** page.
   - Click ***Run***.

5. Wait until the upload process is complete.

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

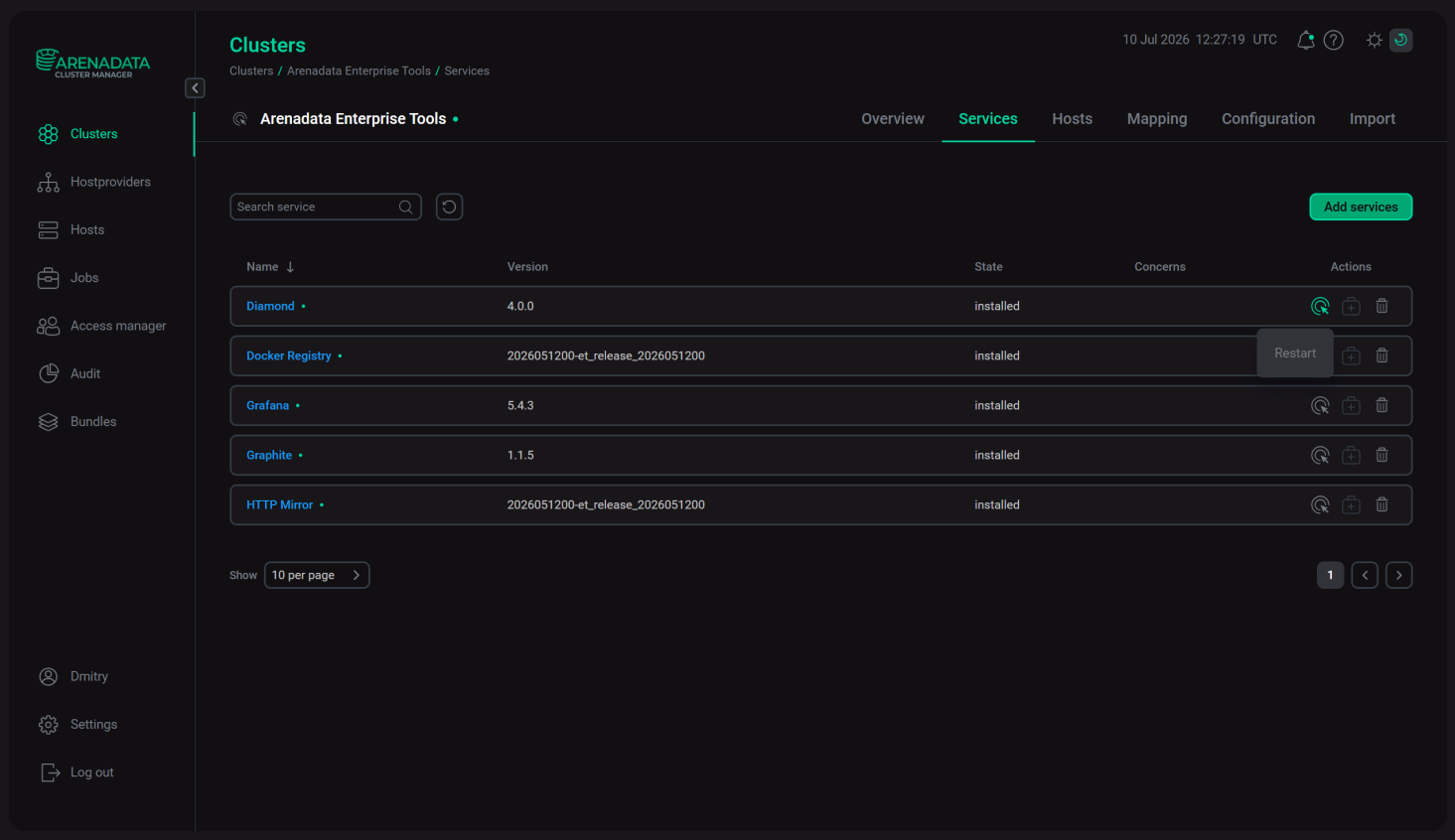
Service actions

Dmitry Bulgakov

To Table of Contents

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

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



Running service actions

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

- Diamond
- Docker Registry
- Grafana
- Graphite
- HTTP Mirror

Configuration parameters

Lisa Vaysman

To Table of Contents

The following articles provide information about the configuration parameters of ADET services:

- Diamond
- Docker Registry
- Grafana
- Graphite
- HTTP Mirror

To read about the configuring process, refer to the Configure services article.

ADET release notes

Lisa Vaysman

Contents

To Table of Contents

ET-2026083100
ET-2026062900
ET-2026051200
ET-2025100900
ET-2025081100
ET-2025042400

ET-2026083100

Date: 14.09.2026

New features

Added support for Ubuntu 24.04 LTS

Bug fixes

Fixed an issue where offline ADET installation could fail because the Docker Registry container could not access */certs/registry.crt* due to an incorrect `umask` value

Misc/Internal

Discontinued support for CentOS 7.9 and RHEL 7.9

ET-2026062900

Date: 29.06.2026

New features

Added Docker Registry configuration via ADCM. ADET now allows administrators to manage key Docker Registry behavior and security settings without manually editing Docker Registry configuration files

Added Docker Registry authentication configuration via ADCM. ADET supports Docker Registry authentication configuration, including `htpasswd`-based authentication and integration parameters for token-based authentication with an external token service

Added Docker Registry lifecycle management via ADCM. ADET now provides actions for managing Docker Registry availability: ***Start Registry*** and ***Stop Registry***

Improvements

Updated Docker Registry base image from 2.8.3 to 3.0.0

Updated bundled NGINX image to 1.30.1

Misc/Internal

Migrated ADET bundle to ADCM bundle contract 2.1

Existing product bundles may not support authenticated Docker Registry access. Additional manual Docker client configuration may be required on target hosts when authentication is enabled

Docker Registry artifact inventory and artifact lifecycle management through ADCM are not included in this release. ADET can configure Docker Registry behavior related to image deletion, but it does not provide full artifact inventory or artifact deletion management

ET-2026051200

Date: 12.05.2026

New features

Added support for Astra Linux SE 1.8.4 Orel/Voronezh. Legacy monitoring components are not supported on this OS

Added support for RED OS 8.0 Certified edition

Added support for RHEL 9

Added support for Ansible 2.16

Improvements

Added validation for upgrade packs used during offline reinstallation. ADET now checks that the offline pack contains the required repositories and provides a clearer error when an incorrect pack is used

Bug fixes

Fixed an issue where Docker attempted to pull images during offline installation or upgrade instead of using images from the offline package

Fixed `remove_coredns` behavior that could incorrectly require Docker on all cluster hosts

Misc/Internal

Updated legacy monitoring component support. Legacy monitoring components are supported only on operating systems that provide Python 2

Migrated ADET bundle to ADCM bundle contract 2.0

ET-2025100900

Date: 09.10.2025

New features

Added support for ALT Linux 10.2 SP

Bug fixes

Fixed ADET upgrade failure when the Diamond monitoring service was not installed

Fixed an issue where ADET cluster services and host could remain with the yellow state indicator after successful installation

Fixed ***Upload Pack*** behavior for broken *pack.sh* files. ADET no longer marks upload-related tasks as successfully changed when the uploaded pack is invalid or incomplete

Misc/Internal

Updated the minimum supported ADCM version to 2.6.0

ET-2025081100

Date: 11.08.2025

New features

Added the ***Uninstall*** cluster action for Arenadata Enterprise Tools. The action allows removing Arenadata Enterprise Tools from a host using ADCM

Added the ***Restart*** cluster action. The action allows restarting ADET services from ADCM

Added support for Astra Linux SE 1.7.6 Voronezh

Improvements

Added a service check before upgrade. ADET checks key services before upgrade to reduce the risk of failed upgrade scenarios

Updated the upgrade workflow for ADET services

Added validation for mandatory configuration fields. ADET now prevents saving required configuration fields with empty values

Added support for FQDN values in `httprepo_host` and `registry_host`

Improved handling of string and text configuration values without Jinja2 templating

Bug fixes

Fixed failures in the ***Manage SSL*** action

Fixed offline installation failure

Fixed incorrect statuschecker restart behavior

ET-2025042400

Date: 24.04.2025

New features

Added the ***Manage SSL*** cluster action. The action allows administrators to manage SSL configuration for Arenadata Enterprise Tools services via ADCM, including providing a server certificate and private key

Added the ***Reinstall statuschecker*** cluster action. The action allows reinstalling ADET statuschecker components, for example, after cluster import or when service or cluster identifiers have changed

Improvements

Added ADET documentation. Documentation coverage was extended for ADET installation, upgrade, requirements, accounts, and offline upgrade scenarios

Added cluster information for Arenadata Enterprise Tools in ADCM

Bug fixes

Fixed ***Install Monitoring*** failure when HTTP Mirror worked with HTTPS listener enabled

Fixed the incorrect display of ADET cluster and service indicators in ADCM on ALT Linux 8.4 SP after host restart

Fixed the incorrect cluster state after ADET post-upgrade actions

Fixed issues in service restart actions, including Docker Registry and Graphite restart behavior

Fixed Grafana HTTPS proxy behavior when HTTPS listener was disabled

Fixed the incorrect availability of the ***Restart*** action for Grafana and Graphite services before monitoring was installed

Misc/Internal

Updated a bundled Ansible version to 2.9

Arenadata Enterprise Tools

To Table of Contents

Arenadata Enterprise Tools (ADET) provides the infrastructure needed to deliver, manage, and maintain dependencies for Arenadata products in secure, isolated, or restricted network environments.