

Trove Openstack Service

- [Introduction](#)
- [Creating the Database instance](#)
 - [1\) Using the OpenStack dashboard](#)
 - [2\) Using the OpenStack command line](#)
 - [Prerequisite](#)
 - [Software required](#)
 - [Setting up the environment variables](#)
 - [Create the database instance](#)
 - [Check the status of the database instance](#)
- [Accessing the Database instance](#)
- [How to list the available datastore and their version using OpenStack CLI](#)

Introduction

Trove is a database as a service (DBaaS) component of OpenStack. It provides a RESTful API for creating and managing databases and database users. It is a pluggable service that supports multiple database engines.

Trove is a part of the OpenStack project and is available as a service in ADA Cloud CINECA infrastructure.

The Database service provides scalable and reliable cloud provisioning functionality for both relational and non-relational database engines. Users can quickly and easily use database features without the burden of handling complex administrative tasks. Cloud users and database administrators can provision and manage multiple database instances as needed.

The Database service provides resource isolation at high-performance levels, and automates complex administrative tasks such as deployment, configuration, patching, backups, restorations, and monitoring.

More details on Trove service can be found in the [OpenStack Database Service User Guide](#).

What follows describes how to create a Database instance: **two alternative approaches** are described, one by using the **OpenStack dashboard** and the other by using the **OpenStack command line**. At the end, it is shown how to access the Database instance.

Creating the Database instance

1) Using the OpenStack dashboard

To create a database instance using the OpenStack dashboard, you can follow the steps below:

- Click on "[Project](#) [Database](#) [Instances](#) [Launch Instance](#)" and fill in all the fields described below :
 - **Details** tab:
 - Availability Zone: nova
 - Instance Name: <the name you prefer>
 - Volume Size: <size in GB for the volume that will contain the databases>. By default the maximum allowed is 10 GB.
 - Volume Type: choose between "___DEFAULT___" or "LUKS". The second one is for encrypted volumes.
 - Datastore: choose among the available datastores. They are listed in the drop-down menu showing also the available versions.
 - Flavor: it is the dimension of the VM that will have the database volume attached. **Insert fl.ada.xxs, since you will not be allowed by design to login into this VM.**
 - Locality: None
 - **Networking** tab:
 - Selected networks: <network_name>, Choose one among the available network in the project. Make sure to create the network before creating the database instance.
 - **Database access** tab:
 - Is public: Check this box if you want to allow access to the database instance from the public network; otherwise leave blank.
 - Allowed CIDRs: <xx.xx.xx.xx./y> . Specify the allowed IP or IP-ranges from which to access the database service.
 - **Initialize Databases** tab:
 - Initial Databases: <the name you want, for the first databases that will be created>. Note that additional databases can be created later.
 - Initial Admin Users: <the username for the admin database user>
 - Password: <the password for the database admin user>
 - Allowed Hosts: optional value, to further restrict for this specific database the allowed Host or IP addresses able to connect to the database.
 - **Advanced** tab:
 - Configuration Group and Source from Initial State: Fill these two fields only if you want to create the database using a previous backup, or as a replica of another database instance.
 - Replica Count: fill in this field only if you want to have multiple replicas of this database instance.

At the end, click on "Launch" on the right bottom to launch the Database instance.

To access the created database you can refer to [dedicated section](#).

2) Using the OpenStack command line

Prerequisite

Software required

To use OpenStack command line interface, the following packages are needed. It is recommended to install these packages in a virtual environment .

```
pip install python-openstackclient==5.8.0
pip install python-troveclient
```

For more information, see [python-troveclient](#), a command-line client for the Trove API.

Setting up the environment variables

Make sure to have a valid Application Credential for the project in ADA Cloud. Please refer to the [dedicated page](#) where it is described how to get OpenStack Application Credentials.

Create the database instance

To create a database instance, you need to execute a command as "**openstack database instance create <DB-INST-NAME>**" specifying at least the following parameters:

- **name**: The name of the database instance.
- **flavor**: The flavor of the database instance. **Insert fl.ada.xxs, since you will not be allowed by design to login into this VM.**
- **datastore**: The datastore of the database instance. Go to the [dedicated section](#) to see how to get the list of the available datastore in ADA cloud.
- **datastore-version**: The version of the datastore to use. Go to the [dedicated section](#) to see how to get the version of each datastore present in ADA cloud.
- **size**: The size of the instance disk volume in GB. By default the maximum allowed is 10 GB.
- **nic**: The network interface card of the database instance.
- **net-id**: The network id of the database instance.
- **allowed-cidr**: The allowed cidr of the database instance. It is an IP or a range of IPs from which the database instance can be accessed.
- **database**: The name of the initial database.
- **users**: The username and password of the admin user in the database.
- **is-public**: Add this flag to be able to access the database from internet. Otherwise, the database will be accessible only from an other VM present in the same network of the project.

For the full list of options please type the command: "**openstack database instance create --help**".

Make sure to create a network stack and copy the id of the network before creating a database instance. In the [AdaCloud documentation](#) you can find the instruction to create the network stack.

The example below shows how to create a database instance:


```
$ openstack database instance show <id-of-the-database-instance>
```

```
+-----+
+-----+
| Field |
+-----+
Value |
+-----+
+-----+
| addresses | [{ 'address': 'xx.xx.xx.xx', 'type': 'private'}, { 'address': 'xx.xx.xx.xx', 'type': 'public' }] |
| allowed_cidrs | ['xx.xx.xx.xx/y', 'xx.xx.xx.xx/y'] |
| created | 2023-06-09T09:19:11 |
| datastore | mysql |
| datastore_version | 5.7.29 |
| datastore_version_number | 5.7.29 |
| flavor | 7595d735-6de4-415f-a958-838089a09080 |
| id | --- the ID of the DB instance |
| ip | xx.xx.xx.xx, xx.xx.xx.xx |
| name | MyTroveDB |
| operating_status | HEALTHY |
| public | True |
| region | RegionOne |
| service_status_updated | 2023-06-09T09:22:21 |
| status | ACTIVE |
| updated | 2023-06-09T09:20:48 |
| volume | 10 |
| volume_used | 0.22 |
+-----+
+-----+
```

Also, check the `status` and the `operating_status`. If the status is `ACTIVE`, then the database instance is ready to use. It may takes some time to get the database instance in `ACTIVE` state after being created.

To access the created database you can refer to [dedicated section](#).

Accessing the Database instance

To access the database instance, the typical `SQL` command (depending on the datastore you chose for your database instance) can be used. In the below example of a MySQL database instance, it is shown how to access the database. After you successfully install the MySQL client, use the following commands to access the database:

```
$ mysql -h <ip-address-of-db-instance> -u<user-name> -p<password>
```

```
-----

mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.7.29 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Once you accessed the database, a `mysql` prompt will be available. You can check the list of databases using the following command:

```
mysql> show databases;

-----
show databases
-----

+-----+
| Database |
+-----+
| information_schema |
| test |
+-----+
2 rows in set (0,02 sec)
```

How to list the available datastore and their version using OpenStack CLI

A datastore is a database engine that is supported by Trove and it is used to create the database instance. On ADA Cloud the following datastores namely MySQL, MariaDB, PostgreSQL are available.

- You can check the available datastores with the command "**openstack datastore list**". The **ID** and **Name** of the datastores supported will be shown.

```
$ openstack datastore list # shows all datastores available in Cloud infrastructure

+-----+-----+
| ID | Name |
+-----+-----+
| b2103dff-9331-4be2-8193-170f2a509e16 | mariadb |
| ed541d5a-d260-4b6b-ac80-74ac38167d70 | mysql |
| e8d12fef-3c54-4e83-818c-4a89a104780d | postgresql |
+-----+-----+
```

- With the command "**openstack datastore version list <name of the datastore>**" you can check the list of datastores with their version. Below we show an example for mysql datastore.

```
$ openstack datastore version list mysql
```

ID	Name	Version
a6ee8255-2d71-4e22-b68d-1d0e5919e74a	5.7.29	5.7.29
434292f3-2074-4033-8f19-07874cbe5b7d	8.0.29	8.0.29

You will find Version, Name, and ID of the mysql datastores available in AdaCloud.

Note: The Version is important, as we have to specify the version of the datastore while creating the database instance. For more information about the version, use the following command:

```
$ openstack datastore version show a6ee8255-2d71-4e22-b68d-1d0e5919e74a # shows details of a version
```

Field	Value
datastore	ed541d5a-d260-4b6b-ac80-74ac38167d70
id	a6ee8255-2d71-4e22-b68d-1d0e5919e74a
name	5.7.29
version	5.7.29