



CloudBeaver User Guide v.24.1.ea

User Guide

Table of contents

[Application overview](#)

[Demo Server](#)

Administration

[Server configuration](#)

[Connection Templates Management](#)

Access Management

[Users](#)

[Teams](#)

[Users Provisioning](#)

[Password policy](#)

Authentication methods

[Local Access Authentication](#)

[Anonymous Access Configuration](#)

[Reverse proxy header authentication](#)

[Single Sign On](#)

[SAML](#)

[OpenID](#)

[AWS OpenID](#)

[AWS IAM](#)

[AWS OpenId via Okta](#)

[Snowflake SSO](#)

[Okta OpenId](#)

[Cognito OpenId](#)

[JWT authentication](#)

[NTLM](#)

[Azure AD authentication](#)

[Google authentication](#)

[User credentials storage](#)

[Cloud Explorer](#)

[Cloud storage](#)

[Query Manager](#)

[Drivers Management](#)

[Supported databases](#)

[Localization](#)

Features

[Create Connection](#)

[Database navigator](#)

[DB Navigator toolbar](#)

[DB Navigator Settings menu](#)

[DB Navigator folders](#)

[Simple and Advanced View](#)

[Data editor](#)

[Data Filters](#)

[Data Ordering](#)

[Value Panel](#)

[Grouping Panel](#)

[Managing Charts](#)

[JSON and Document View](#)

[Data export](#)

[Data import](#)

[Entity Diagrams](#)

[SQL Editor](#)

[Query Execution Plan](#)

[Visual Query Builder](#)

[AI Smart Assistance](#)

[Log Viewer](#)

[Query History](#)

[Resource Manager](#)

Installation

[Installation](#)

[Version upgrade](#)

[Workspace backup](#)

Configuration

[Server configuration](#)

[Query manager database configuration](#)

[Configuring server datasources](#)

[Connection configuration](#)

[CloudBeaver and Nginx](#)

[Domain manager](#)

[Configuring HTTPS for Jetty server](#)

[Product configuration parameters](#)

[Authentication](#)

[Theming](#)

[Localization](#)

[Database Navigator](#)

[Data Editor](#)

[SQL Editor](#)

[Log Viewer](#)

[Data Export](#)

[ERD](#)

[Connections](#)

[Command line parameters](#)

[Local Preferences](#)

CloudBeaver EE for AWS

[Overview](#)

[AWS Settings](#)

[Cloud Explorer](#)

CloudBeaver Enterprise Edition

[Overview](#)

[License Management](#)

Team Edition

[Team Edition Overview](#)

[Getting started with Team Edition](#)

[Team Edition Server Configuration](#)

[Projects in Team Edition](#)

[Teams in Team Edition](#)

[Team Edition Deployment](#)

[Roles in Team Edition](#)

[Git integration in Team Edition](#)

[Datasets in Team Edition](#)

Deployment

CloudBeaver Community

[Docker image](#)

[AWS Marketplace](#)

[Google Cloud](#)

[Microsoft Azure](#)

CloudBeaver AWS

[AWS Marketplace](#)

[Docker image](#)

CloudBeaver Enterprise

[Docker image](#)

[AWS AMI](#)

[Google Cloud](#)

[Microsoft Azure](#)

Deployment options

[SSL certificate configuration](#)

[How to connect CloudBeaver to a database on a separate machine in Azure](#)

Application overview

Table of contents

[Toolbar](#)

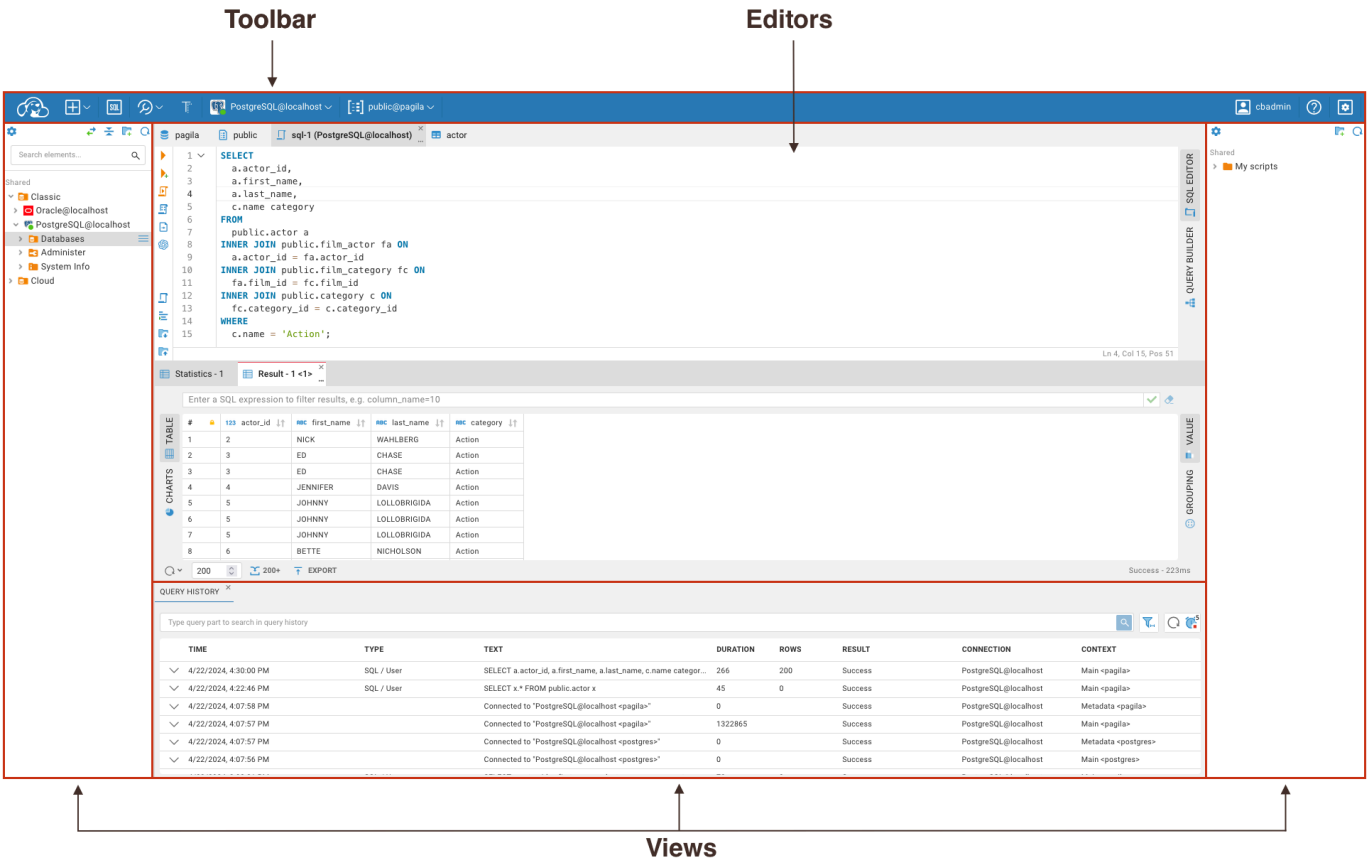
[Views](#)

[Editors](#)

[Managing Views and Editors](#)

Overview

The CloudBeaver window contains a **Toolbar** and a workspace with one or more **Views** and **Editors**:



Toolbar

The **Toolbar** may contain the following menus and buttons:

- The **Project Selector** contains a list of user's projects. [Learn more about the Projects](#)

- The **Connection** menu contains buttons for [creating new connections](#), finding local databases, and opening [Cloud Explorer](#).
- The **SQL Editor** button is for opening the [SQL Editor](#).
- The **Tools** menu provides options to open the Cloud Storage view, Scripts viewer, Query history, and Log viewer and Datasets viewer.
- The menu for switching between [Auto and Manual commit mode](#).
- The **Database** menu allows for the management of database drivers and connections.
- The **Profile** menu contains user info, [preferences](#), and capability for password management.
- The **Shortcuts** button contains shortcuts for CloudBeaver usage.
- The **Settings** menu provides options to open an [Administration menu](#), view Product information, open [Local preferences](#), and Logout.

Note: The availability of these buttons depends on the version of CloudBeaver you are using.

As an administrator, you can customize the menu bar. To do this, navigate to **Settings -> Administration -> Server Configuration** tab.

Views

The **Views** in the workspace are designed to display data and provide navigation and may contain the following buttons:

- **Database Navigator:** Provides a tree structure view of all the databases you have access to, allowing you to navigate through schemas, tables, and other database objects.
- **[Query History](#):** Records the queries you have executed and allows you to revisit and analyze them.
- **Scripts viewer:** Manages and organizes your SQL scripts.
- **Cloud Storage:** Enables you to interact with files stored in cloud services.
- **Log viewer:** Displays the CloudBeaver logs.

- [Datasets viewer](#): Presents the list of saved Datasets.

Except for the Database Navigator, to access these views, click on the **Tools** button and choose the necessary **View** from the list.

Note: The availability of these buttons depends on the version of CloudBeaver you are using.

Editors

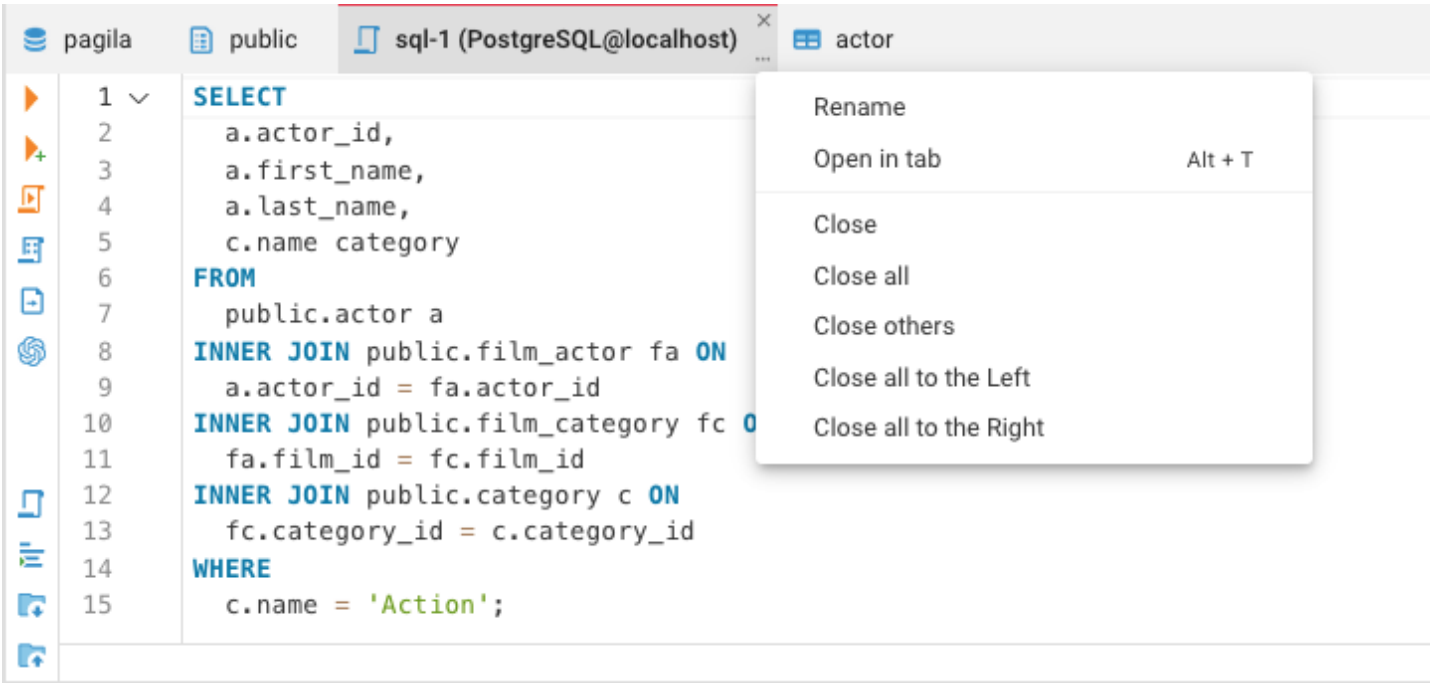
The **Editors** within the workspace are interactive windows that allow you to directly work with data and SQL queries.

- [Data Editor](#): An interface for modifying and browsing table data.
- [SQL Editor](#): An interface for writing, editing, and running SQL queries.

To open these **Editors**, navigate through the Database Navigator to the desired database object or use **SQL Editor** button in the **Toolbar**.

Managing Views and Editors

All **Editors** have the following menu buttons. To access them, click on the three dots under the corresponding tab.



Button	Description
Rename	Rename the SQL Editor's tab.

Open in Tab	Opens the current editor in a separate tab.
Close	Close the current editor window or tab.
Close all	Close all open editor windows or tabs.
Close others	Close all editor windows or tabs except the currently active one.
Close all to the Left	Close all editor windows or tabs to the left of the current tab.
Close all to the Right	Close all editor windows or tabs to the right of the current tab.

For the **Views**, you have the ability to adjust their size or change how they are displayed.

CLOUD STORAGE

QUERY HISTORY ×

Type query part to search in query history

TIME	TYPE	TEXT	DURATION	ROWS	RESULT	CONNECTION	CONTEXT
4/15/2024, 1:54:16 PM	SQL / User	SELECT * FROM SiteActivity	112	0	Success	Firestore	Main Firestore Connecti...
4/15/2024, 1:54:13 PM	SQL / User	SELECT * FROM SiteActivity	349	0	Success	Firestore	Main Firestore Connecti...
4/15/2024, 1:54:06 PM		Connected to "Firestore"	9959		Success	Firestore	Main Firestore Connecti...
4/15/2024, 1:52:26 PM		Connected to "MongoDB@localhost"	0		Success	MongoDB@localhost	Main MongoDB Connect...
4/15/2024, 1:52:16 PM		Disconnected from "MongoDB@localhost"	24720		Success	MongoDB@localhost	Main MongoDB Connect...
4/15/2024, 1:51:51 PM		Connected to "MongoDB@localhost"	24720		Success	MongoDB@localhost	Main MongoDB Connect...

To do this:

- **Resize:** Click and drag the borderline separating the **View** from the **Editor**. This allows you to customize the size of the **View** pane to your preference.
- **Maximize/Restore:** Use the maximize and restore buttons to toggle the View pane between the full workspace and its original size.
- **Minimize:** Click the minimize button to collapse the **View** pane, providing more space for the **Editor** area.

Demo Server

Table of contents

[Pre-configured databases access:](#)

[Custom connections](#)

[Security](#)

We host a demo server where you can see what CloudBeaver looks in real life.

It is a simple server with a few sample databases.

[Demo Server](https://demo.cloudbeaver.io) - <https://demo.cloudbeaver.io>

Pre-configured databases access:

Database	User	Password
SQLite	n/a	n/a
MySQL	demo	demo
PostgreSQL	demo	demo

Custom connections

CloudBeaver does not store/cache your credentials or any user data. You could try to connect to some of your databases using the Custom connection wizard.

Warning: it is not secure to open direct access to your database so do not use this on databases with sensitive data.

The Demo server is for testing only.

If you want to use CloudBeaver with your real databases then [deploy it in your infrastructure](#).

Security

We did not pay too much attention on the Demo server security.

It is a completely isolated server with no sensitive data.

But you could try to hack it - please do it gently and please let us know if you will find any security holes.

Thank you ;-)

Have fun!

Learn more

Table of contents

[Server information](#)

[Configuration](#)

[Custom connections](#)

[Navigator simple view](#)

[Services](#)

[AWS](#)

[Authentication settings](#)

[Security](#)

[Save credentials](#)

[Save users credentials](#)

CloudBeaver offers different settings that allow configuring the server. The administrator can set the Server configuration settings when configuring the app for the first time, or it can be done later in the Administration Menu.

The screenshot displays the CloudBeaver Administration interface. The top navigation bar is blue with the CloudBeaver logo on the left and a user profile icon labeled 'cbadmin' on the right. A left sidebar contains icons and labels for 'Connection Management', 'Access Management', 'Server configuration' (which is highlighted), 'Identity Providers', 'License', and 'Version update'. The main content area has a light gray background and is divided into several sections. At the top of the main area are 'SAVE' and 'CANCEL' buttons. The 'SERVER INFORMATION' section includes fields for 'Server Name *' (filled with 'Cloudbeaver EE Web Server'), 'Server URL *' (filled with 'http://localhost:3100'), and 'Session lifetime *' (filled with '30'). The 'CONFIGURATION' section features three toggle switches: 'Enable custom connections' (checked), 'Navigator simple view' (unchecked), and 'AWS' (unchecked). The 'AUTHENTICATION SETTINGS' section contains three toggle switches: 'Allow anonymous access' (unchecked), 'Local' (checked), and 'AWS IAM' (unchecked). Below these are sections for 'OpenId' and 'SAML', each with a toggle switch and a link to 'Edit configurations'. The 'SECURITY' section at the bottom has two toggle switches: 'Save credentials' (checked) and 'Save users credentials' (checked).

Server information

Basic settings such as Server name and Session lifetime.

Configuration

Custom connections

Whether users can create connections by themselves or it can be done only from the Administration Menu.

Navigator simple view

Defines how the [Database navigator](#) structure will look like.

You can read more about Simple and Advanced mode [here](#).

Services

AWS

Enables AWS Services.

Authentication settings

Define different authentication methods.

You can read more about authentication methods [here](#).

Security

Save credentials

Allow saving credentials for the pre-configured database.

Save users credentials

Allow saving credentials for non-admin users.

Connection Templates Management

Table of contents

- [Overview](#)
- [The Purpose](#)
- [Description](#)
- [Creation](#)
- [Template Form](#)
- [Usage](#)

Overview

The Purpose

The templates enable administrators to define various reusable connection parameters, subsequently allowing users to create multiple connections based on these templates. This process ensures consistency across the connections created from these templates, as they adhere to the defined parameters within the templates.

Description

You can add, edit, or remove database templates on the **Connection Templates** page in administration.

Query Manager

Driver Management

Connection Templates

Access Management

Server configuration

AWS Settings

Identity Providers

License

Version update

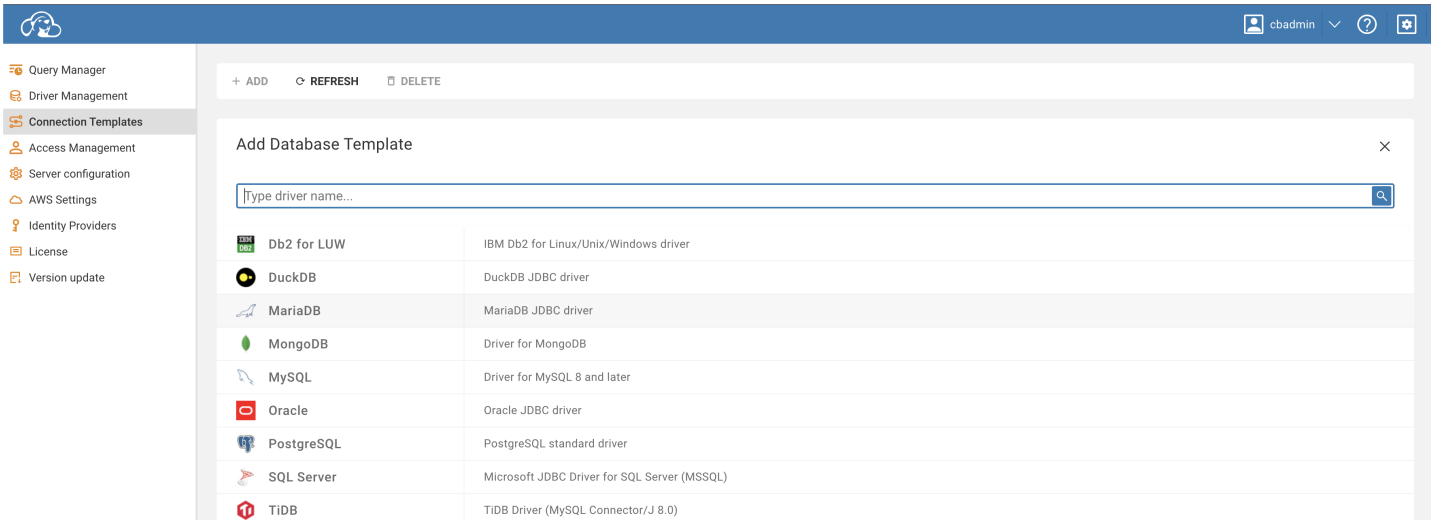
+ ADD

REFRESH

DELETE

	CONNECTION NAME	ADDRESS	
<input type="checkbox"/>			
<input type="checkbox"/>	42.5.4@localhost	localhost:5432	
<input type="checkbox"/>	Exasol_template	localhost:8563	
<input type="checkbox"/>	Greenplum_template	localhost:5432	
<input type="checkbox"/>	MongoDB_template	localhost:27017	
<input type="checkbox"/>	Oracle	localhost:1521	
<input type="checkbox"/>	Oracle@localhost	localhost:1521	
<input type="checkbox"/>	PostgreSQL_GIS_template	localhost:5432	
<input type="checkbox"/>	SQL Server@localhost	localhost:1433	
<input type="checkbox"/>	SQLite - Chinook (Sample)	localhost:1234	

Creation



To create a new template, follow the steps below:

1. Click the **Add** button located on the top toolbar. This will open the template creation form.
2. Fill out the fields in the form with the appropriate information.
3. Once all fields are completed, click **Create**.

Your new template should be successfully created and it will appear at the top of the templates table.

Template Form

Below are the detailed steps to set the template parameters:

1. Open the template form. This can be done by clicking the **Add** button in the toolbar.
2. Fill in the appropriate information in the base template parameters, driver settings, SSH tunnel, and access fields.
3. To verify the template connection, click the **Test** button. If SSH is configured, the test will use it to establish the connection.
4. When users attempt to establish a connection using the template, they will be prompted to enter credentials.

You can manage access to the database at the **Access** tab. You can select users or roles to provide access to.

Usage

Once a template is prepared, connections can be created using the template from the public section of the CloudBeaver.



Users

Table of contents

[CloudBeaver CE](#)

[Adding a New User in the Administration Menu](#)

[CloudBeaver EE](#)

[AWS and Federated users](#)

[CloudBeaver AWS](#)

The Administrator can create users for local name/password based authentication in the Administration Menu.

CloudBeaver CE

Adding a New User in the Administration Menu

1. **Navigating to the Access Management Tab:** Within the Administration Menu, find and select the **Access Management** tab.
2. **Initiating the Creation of a New User:** To begin the process of creating a new user, click on the **Add** button located within the **Access Management** tab.
3. **Entering User Details:** Here, you will be required to input a username and password. These credentials will be used by the new user for logging into the system.
4. **Assigning a Team to the New User:** A team will define the permissions the user has within the system. For additional information regarding teams and their definitions, please refer to the [Team management](#) article.
5. **Setting Connection Access:** If necessary, you can provide the user with connection access. This setting can be found and adjusted within the **Connection Access** tab.
6. **Finalizing User Creation:** To complete the process and create the new user, click on the **Create** button.

Connection Templates

Access Management

Server configuration

Version update

USERS TEAMS

+ CREATE REFRESH

Search for the user name...

User Creation

INFO CONNECTIONS ACCESS

CANCEL CREATE

CREDENTIALS

Username *

User password *

Repeat password *

USER STATUS

☒ Enabled

USER TEAM

☐ admin

☒ user

PARAMETERS

First Name

Last Name

The newly created user can now be authenticated to CloudBeaver using local authentication. The user's permissions are defined according to their assigned profile.

CloudBeaver EE

CloudBeaver Enterprise Edition also allows you to configure AWS and SSO users.

AWS and Federated users

When a user is authorized to the CloudBeaver EE instance via AWS IAM or Federated authentication for the first time, a corresponding user is automatically created within the application, with the 'User' team assigned by default. Post-creation, the administrator can alter the user's team as needed.

It is important to note that administrators cannot create new AWS or Federated users directly within the application. The system is designed to work with existing, legitimate AWS and Federated users.

CloudBeaver AWS

CloudBeaver AWS exclusively supports configuration for AWS and Federated users, as it does not provide local access. As a consequence, local users cannot be created within the Product environment.

Remember, user management is an important aspect of maintaining system security. Always ensure that users are only granted access and permissions necessary for their tasks.

Teams

Table of contents

[Overview](#)

[Team creation](#)

[Predefined Team types](#)

[Integration with Identity Providers](#)

[Configuration steps](#)

[Automatic membership management](#)

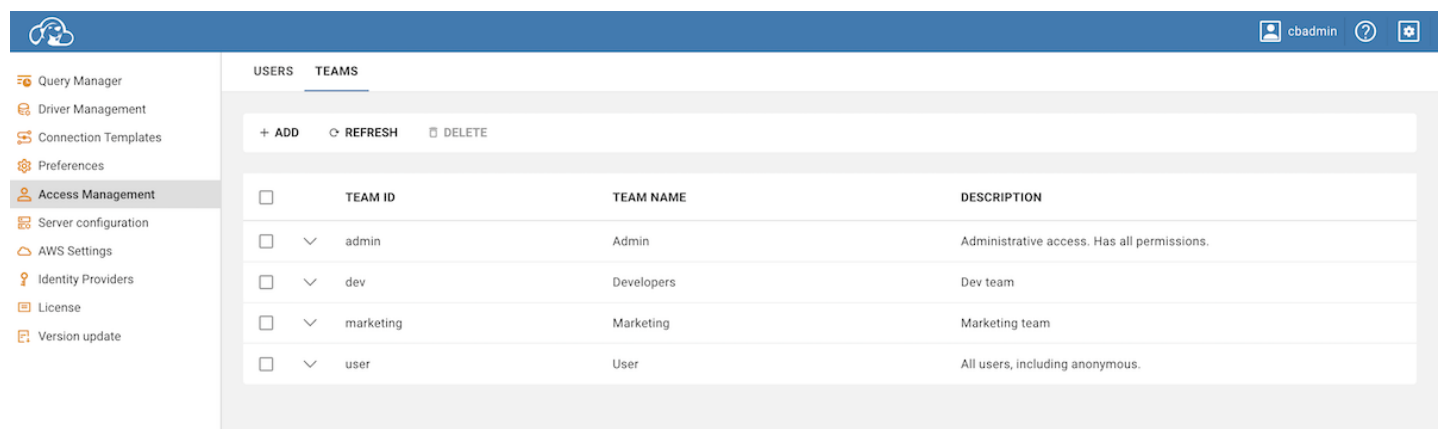
[Updating Team memberships](#)

[User management](#)

[Connection management](#)

Overview

The CloudBeaver provides a team management feature, allowing administrators to create and manage teams. This feature is integral for organizing users into groups and controlling their access to various databases.



The screenshot displays the CloudBeaver web interface. On the left is a sidebar menu with options like Query Manager, Driver Management, Connection Templates, Preferences, Access Management (highlighted), Server configuration, AWS Settings, Identity Providers, License, and Version update. The main content area is titled 'TEAMS' and includes a header bar with '+ ADD', 'REFRESH', and 'DELETE' buttons. Below this is a table listing existing teams.

	TEAM ID	TEAM NAME	DESCRIPTION
<input type="checkbox"/>	admin	Admin	Administrative access. Has all permissions.
<input type="checkbox"/>	dev	Developers	Dev team
<input type="checkbox"/>	marketing	Marketing	Marketing team
<input type="checkbox"/>	user	User	All users, including anonymous.

Team creation

To create a new team, follow these steps:

1. Navigate to the **Settings -> Administration -> Access Management -> Teams**.

- 2. Click on the **+ Add** button.
- 3. Fill in the necessary details in the provided fields.

cbadmin ?

Query Manager

Driver Management

Connection Templates

Preferences

Access Management

Server configuration

AWS Settings

Identity Providers

License

Version update

USERS **TEAMS**

+ ADD ◁ REFRESH ▢ DELETE

Team Creation ×

OPTIONS **USERS** CONNECTIONS

Team ID *

dev

Team name

Developers

Description

PERMISSIONS

☒ Admin Full Access
Provides access to the CloudBeaver configuration

PARAMETERS

AWS Role ARN

DEVTEAM

SAML Group ID

Microsoft Entra ID Group ID

OKTA Group ID

CANCEL

CREATE

Field Name	Description	Additional Info
Team ID	A unique identifier for the team.	
Team Name	The name of the team.	
Description	A brief description of the team and its purpose.	
Permissions	Specifies the level of access the team has.	Admin Full Access checkbox provides complete access to CloudBeaver configuration.
Parameters	Additional parameters based on the authentication provider.	Read more about Integration with Identity Providers .

Note:

- The **Parameters** section is available in [Enterprise](#), [AWS](#) and [Team](#) editions only and becomes accessible when configuring certain [authentication providers](#).
- In CloudBeaver [Team Edition](#), **Permissions** are set through assigned [Roles](#).

Predefined Team types

CloudBeaver includes two predefined Team types:

Types	Description
<code>admin</code>	Members of this Team have full administrative privileges within CloudBeaver.
<code>user</code>	This Team is for regular users. Administrators assign access to databases (in Team Edition , access to projects) to this team.

Integration with Identity Providers

You have the ability to integrate Teams with various identity providers. This integration allows for the utilization of roles and groups defined by your identity provider to manage Team memberships automatically.

Configuration steps

1. When creating or editing a Team, navigate to the **Parameters** section.
2. Here, depending on your identity provider, you can associate the Team with a specific identity attribute:

Provider	Attribute	Related articles
AWS	<code>AWS Role ARN</code>	AWS OpenID , AWS OpenID via Okta
SAML	<code>SAML Group ID</code>	SAML configuration
Microsoft Entra ID	<code>Microsoft Entra ID Group ID</code>	Microsoft Entra ID
Okta OpenID	<code>OKTA Group ID</code>	Okta OpenID

Automatic membership management

Once the integration is set up, whenever a user authenticated by the configured identity provider logs into CloudBeaver, the application will check for matching identity attributes. If there is a match with any of the defined parameters within CloudBeaver's Teams, the user will be automatically assigned to the appropriate Team.

Updating Team memberships

For the changes to take effect, especially in cases where group memberships are updated:

- Users may need to log off and log back in through the Single Sign-On (SSO).
- Alternatively, users can wait for the session to timeout.

These actions ensure that the updated claims from the identity provider are received by CloudBeaver, thereby refreshing the Team memberships.

User management

In the **Users** tab, you can manage Team memberships:

- To add a user to the Team, click **Edit**, select the desired users, and then click **Add**.
- To remove a user from the Team, select the user and click **Delete**.

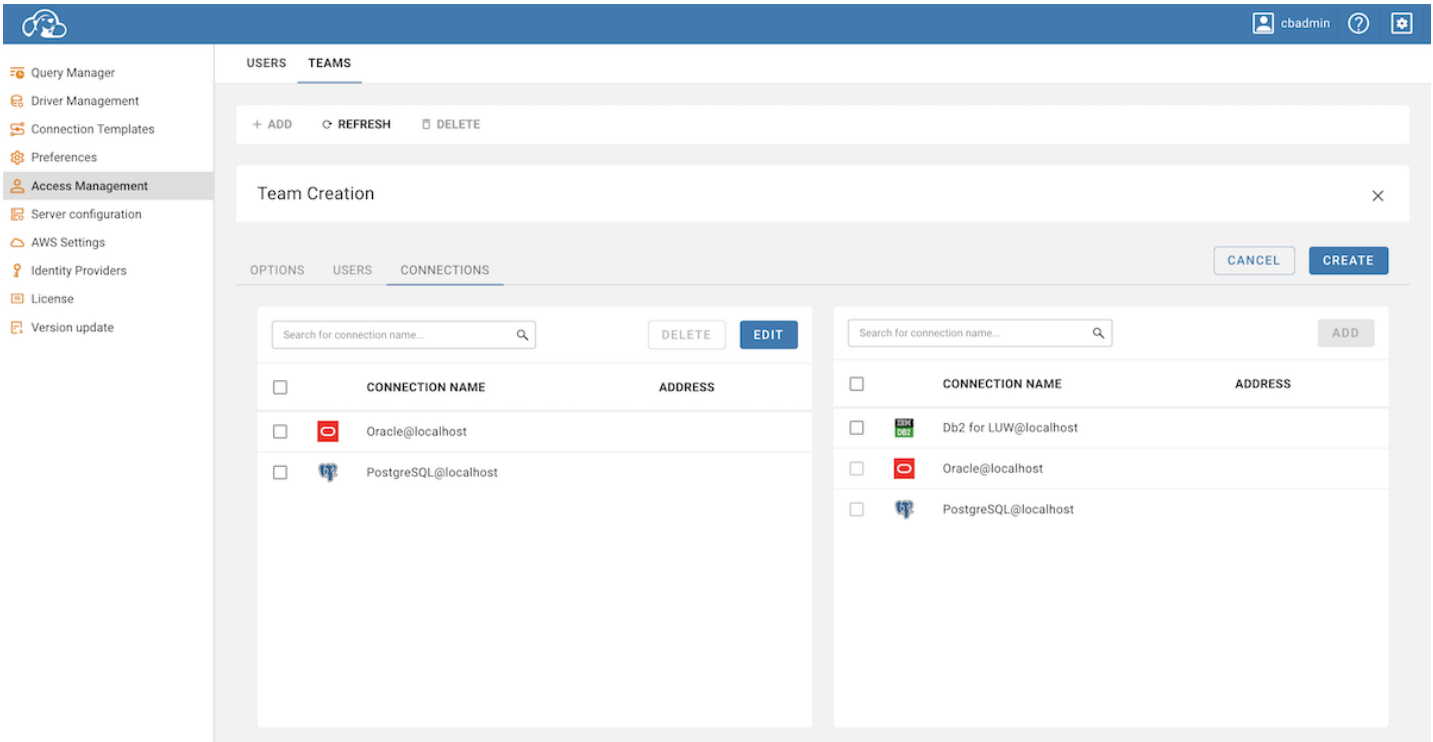
The screenshot shows the CloudBeaver interface with a sidebar on the left containing navigation links: Query Manager, Driver Management, Connection Templates, Preferences, Access Management (highlighted), Server configuration, AWS Settings, Identity Providers, License, and Version update. The main area displays the 'Team Creation' dialog box. At the top, there are buttons for '+ ADD', 'REFRESH', and 'DELETE'. Below this, the 'Team Creation' title is followed by a close button 'X'. The dialog has three tabs: 'OPTIONS', 'USERS', and 'CONNECTIONS'. The 'USERS' tab is selected. It contains two side-by-side user selection panels. The left panel has a search bar 'Search for user ID...', a 'DELETE' button, and an 'EDIT' button. It lists one user: 'Max (developer)'. The right panel has a search bar 'Search for user ID...', an 'ADD' button, and lists four users: 'Alex (Testing)', 'cbadmin (You)', 'Felix (Marketing)', and 'Max (developer)'. At the bottom right of the dialog are 'CANCEL' and 'CREATE' buttons.

Tip: One user can be a member of a multiple Teams.

Connection management

In the Connections tab, you can manage which connections are available to the Team:

- To add connections to the Team, click **Edit**, choose the desired connections, and then click **Add**.
- To remove connections from the Team, select the connection and click **Delete**.



Note: In [Team Edition](#), teams are granted access to projects, not directly to connections.

Learn more

Table of contents

[Provisioning](#)

Note: This feature is available in [Enterprise](#) and [Team Edition](#) editions only.

Users can be provisioned from the third-party system. This is useful when you want to pre-configure access for users.

Supported providers:

- 1. Microsoft AzureAD
- 2. Okta OpenID

Provisioning

You can find the **+ Import** button on the Users management page in administration. (only in Enterprise products)

cbadmin

?

+

Query Manager

Driver Management

Connection Templates

Access Management

Server configuration

AWS Settings

Identity Providers

License

Version update

USERS

TEAMS

+ CREATE

+ IMPORT

REFRESH

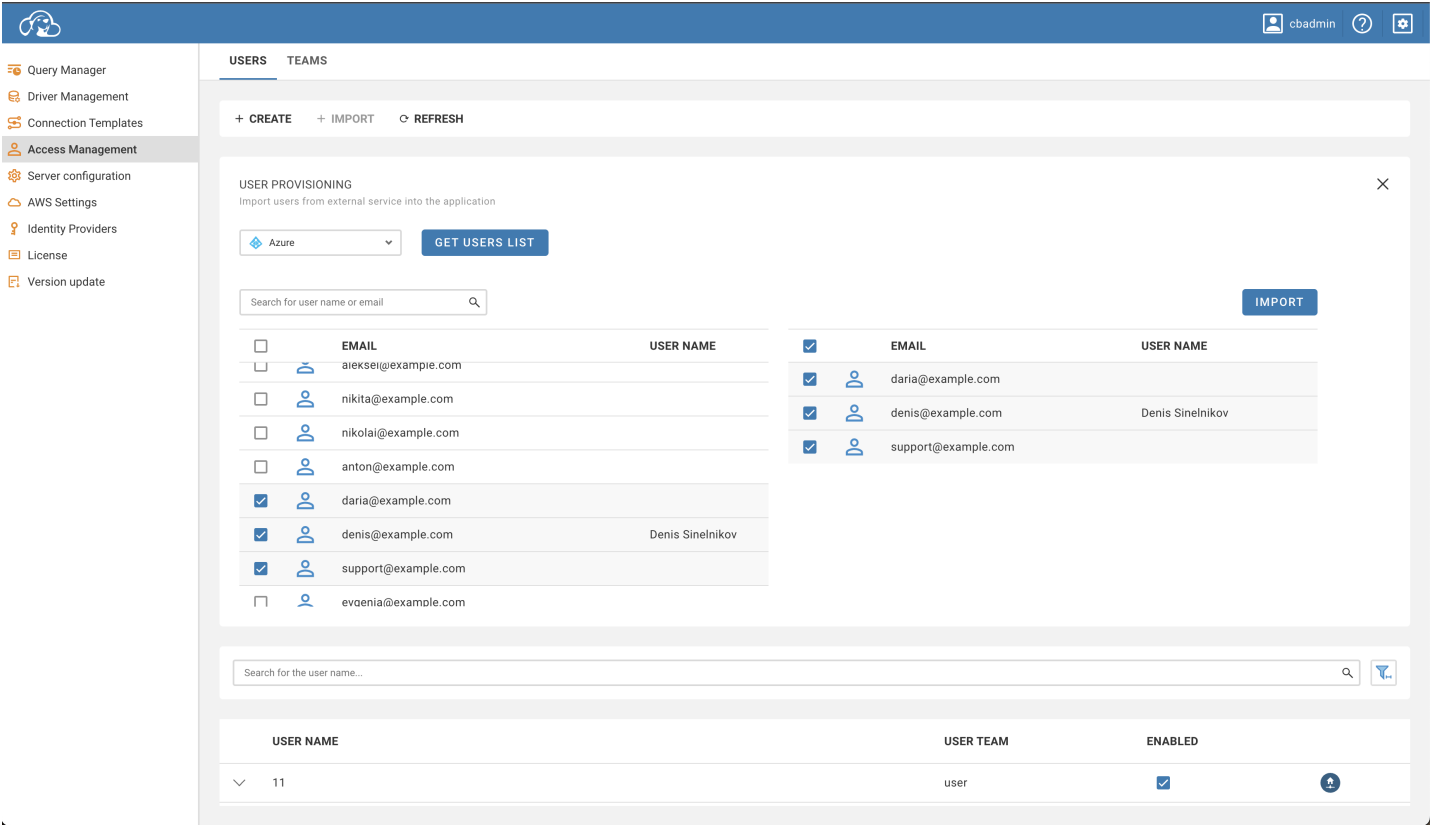
Search for the user name...

🔍

🔼

USER NAME	USER TEAM	ENABLED	
admin1	admin, user	☑	👤
admin.yamamoto@beaver.com	user	☑	👤🔑🔒🔥
cbadmin	admin, dum, team_analytics, user	☐	👤🔑🔒🔥
empty	simple	☑	👤
new	user	☑	👤
wroud	simple, user	☑	👤

Click on this button will open users importing form.



To import users, you need to select users provisioning provider and click **Get Users List** . So that you know, you need to configure at least one supported provider before.

You will be asked to log in with the selected provider (in case you haven't logged in before) to load users.

After loading users, You can select users to import and verify them in the table on the right side. In the **Team Edition**, you must also select the role assigned to users.

To confirm importing, click on the **Import** button. Users will appear in the table.

Password policy

Table of contents

[Password Policy Configuration](#)

[Overview](#)

[Configuration Properties](#)

[Applying Password Policy](#)

[Example snippet:](#)

Password Policy Configuration

Overview

This document provides guidelines for configuring the password policy in CloudBeaver. The password policy settings allow administrators to define rules for user passwords, ensuring security and compliance with organizational requirements.

Configuration Properties

The following properties can be adjusted in the [configuration file](#) to customize the password policy. These settings will be applied globally throughout the application.

1. minLength:

- Description: Specifies the minimum length requirement for user passwords.
- Default Value: 8
- Example:

```
minLength: "${CLODBEAVER_POLICY_MIN_LENGTH:10}",
```

2. **requireMixedCase:**

- Description: Enforces the use of both uppercase and lowercase letters in passwords.
- Default Value: true (mixed case required)
- Example:

```
requireMixedCase: "${CLOUDBEAVER_POLICY_REQUIRE_MIXED_CASE:false}",
```

3. **minNumberCount:**

- Description: Sets the minimum number of numeric characters required in passwords.
- Default Value: 1
- Example:

```
minNumberCount: "${CLOUDBEAVER_POLICY_MIN_NUMBER_COUNT:2}",
```

4. **minSymbolCount:**

- Description: Defines the minimum number of special symbols required in passwords.
- Default Value: 1
- Example:

```
minSymbolCount: "${CLOUDBEAVER_POLICY_MIN_SYMBOL_COUNT:3}"
```

Applying Password Policy

The configured password policy will be applied during password create and change processes. The CloudBeaver will check the entered passwords against the defined policy, and users will be prompted to update their passwords if they do not meet the specified requirements.

cbadmin
?
⚙️

- Query Manager
- Driver Management
- Connection Templates
- Access Management
- Server configuration
- Identity Providers
- AI Settings
- License
- Version update

USERS TEAMS

+ CREATE + IMPORT ↻ REFRESH

User Creation

INFO CONNECTIONS ACCESS

CREDENTIALS

Username *

User password *

USER STATUS

☒ Enabled

USER TEAM

☐ admin
☒ user

PARAMETERS

First Name

Last Name

CANCEL

CREATE

⚠️ Password must contain both upper and lower case letters

Example snippet:

```
passwordPolicy: {
  minLength: "${CLOUDBEAVER_POLICY_MIN_LENGTH:8}",
  requireMixedCase: "${CLOUDBEAVER_POLICY_REQUIRE_MIXED_CASE:true}",
  minNumberCount: "${CLOUDBEAVER_POLICY_MIN_NUMBER_COUNT:1}",
  minSymbolCount: "${CLOUDBEAVER_POLICY_MIN_SYMBOL_COUNT:0}"
}
```

Local Access Authentication

Table of contents

[Overview](#)

[Configuration steps](#)

[Step 1: Enabling local authentication](#)

[Step 2: Granting local access](#)

[Recovering from disabled authentication methods](#)

Note: This feature is available in [Enterprise](#) and [Team](#) editions only.

Overview

Local access authentication is a method for validating users based on usernames and passwords managed within the system.

Configuration steps

Step 1: Enabling local authentication

1. As an administrator, navigate to the **Settings** -> **Server configuration**.
2. Locate the **Local** option and activate this setting to allow local authentication.

SAVE CANCEL

SERVER INFORMATION

Server Name *
CloudBeaver CE Server

Server URL *
http://localhost:8090

Session lifetime, min *
30

CONFIGURATION

☐ Enable private connections
Allows users to create private connections

☐ Navigator simple view
By default, all user's new connections will contain only basic information in navigation tree

RESOURCE MANAGER

☒ Enable Resource Manager
Enable Resource Manager functionality

AUTHENTICATION SETTINGS

☒ Allow anonymous access
Allows to work with CloudBeaver without user authentication

☒ Local
Local name/password based authentication

☐ Reverse proxy
Reverse proxy header based authentication

SECURITY

☒ Save credentials
Allow to save credentials for pre-configured database

☒ Save users credentials
Allow to save credentials for non-admin users

3. Save changes.

Step 2: Granting local access

To grant local access, administrators set up individual user accounts. This involves creating each account with a unique username and a secure password in the Administration section. For detailed guidance on creating user accounts, refer to the [Users](#) article.

USERS TEAMS

+ CREATE REFRESH

Search for the user name...

User Creation

INFO CONNECTIONS ACCESS

CREDENTIALS

Username *
User password *
Repeat password *

USER STATUS

☒ Enabled

USER TEAM

☐ admin
☒ user

PARAMETERS

First Name
Last Name

CANCEL CREATE

Recovering from disabled authentication methods

If an administrator disables all types of authentication, including **Local**, the administrator cannot log in with their credentials after logging out. In such a situation, you need to:

1. Open the file `{cb_workspace}/.data/.cloudbeaver.runtime.conf`.
2. Find `enabledAuthProviders`.
3. Add `"local"` to the list.

```
"enabledAuthProviders": ["local"],
```

4. Restart the server to apply the changes.

Anonymous Access Configuration

Table of contents

[Overview](#)

[Configuration methods](#)

[Enabling private connections for anonymous users](#)

[Administrator defined anonymous access](#)

Overview

Anonymous Access Configuration in CloudBeaver enables users to interact with databases without authentication.

There are two methods for setting up anonymous access:

- **Administrator defined connections:** Administrators can configure connections that are accessible anonymously, allowing access to these predefined connections.
- **Private connections:** When enabled, this feature lets anonymous users configure their private connections from the main page. These connections are temporary and expire with the session.

By adhering to the settings below, CloudBeaver can be configured to allow anonymous access through both administrator-defined and user-customized connections.

Important: If the **Allow anonymous access** checkbox is not selected, the functionality for anonymous access is disabled, restricting the use of CloudBeaver to authenticated users only.

Configuration methods

Enabling private connections for anonymous users

1. As an administrator, navigate to the **Settings -> Server configuration**.
2. Locate the **Enable private connections** option and activate this setting to allow the creation of connections by anonymous users.

SAVE CANCEL

SERVER INFORMATION

Server Name *
CloudBeaver CE Server

Server URL *
http://localhost:8080

Session lifetime, min *
30

CONFIGURATION

☐ Enable private connections
Allows users to create private connections

☐ Navigator simple view
By default, all user's new connections will contain only basic information in navigation tree

RESOURCE MANAGER

☒ Enable Resource Manager
Enable Resource Manager functionality

AUTHENTICATION SETTINGS

☒ Allow anonymous access
Allows to work with CloudBeaver without user authentication

☒ Local
Local name/password based authentication

☐ Reverse proxy
Reverse proxy header based authentication

SECURITY

☒ Save credentials
Allow to save credentials for pre-configured database

☒ Save users credentials
Allow to save credentials for non-admin users

3. Save changes.

Note: Connections created by anonymous users are not permanent. These connections will be terminated following the session expiration.

Administrator defined anonymous access

1. Start the process of creating a connection as an administrator. For detailed instructions, refer to the [Create Connection](#) article.
2. Navigate to the **Access** tab within the connection settings and click on the **Edit** button to modify access settings.

Shared
> PostgreSQL@localhost

MAIN DRIVER PROPERTIES SSH TUNNEL SSL ACCESS

CANCEL TEST CREATE

Search for user or team name

DELETE EDIT

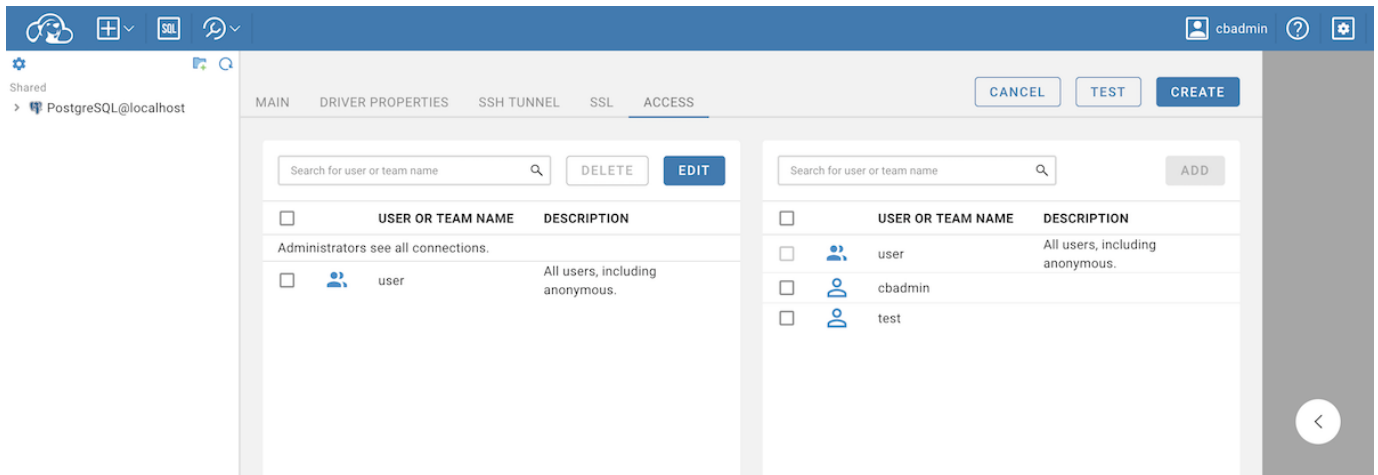
USER OR TEAM NAME	DESCRIPTION
There are no items yet.	

Search for user or team name

ADD

USER OR TEAM NAME	DESCRIPTION
<input type="checkbox"/> user	All users, including anonymous.
<input type="checkbox"/> cbadmin	
<input type="checkbox"/> test	

3. Choose **user** (representing all users, including anonymous) from the list and click **Add** to include this user group in the access rights.



4. Once all the connection settings are configured, click the **Create** button to establish the new connection.

Note: If the administrator has enabled the **Enable private connections** option, anonymous access cannot be configured for that specific database.

Reverse proxy header authentication

Table of contents

[Overview](#)

[Configuration Steps](#)

[Step 1: Enabling Reverse proxy authentication](#)

[Step 2.1: Reverse proxy identity provider configuration in Community Edition](#)

[Step 2.2: Reverse proxy identity provider configuration in Enterprise and Team Editions](#)

[Step 3: Configuring default HTTP header fields](#)

[Header example](#)

Overview

CloudBeaver offers a feature for authorization and authentication using reverse proxy headers. This method allows to authenticate users via specific HTTP header fields.

Configuration Steps

Step 1: Enabling Reverse proxy authentication

1. As an administrator, navigate to the **Settings -> Server configuration**.
2. Locate the **Reverse proxy** option and activate this setting to allow reverse proxy authentication.

3. Save changes.

Step 2.1: Reverse proxy identity provider configuration in Community Edition

To configure reverse proxy authentication, follow these steps:

1) Open your `.cloudbeaver.runtime.conf` configuration file. 2) Locate the `app` section within the file. 3) Add a new entry to the `authConfigurations` array with the following structure:

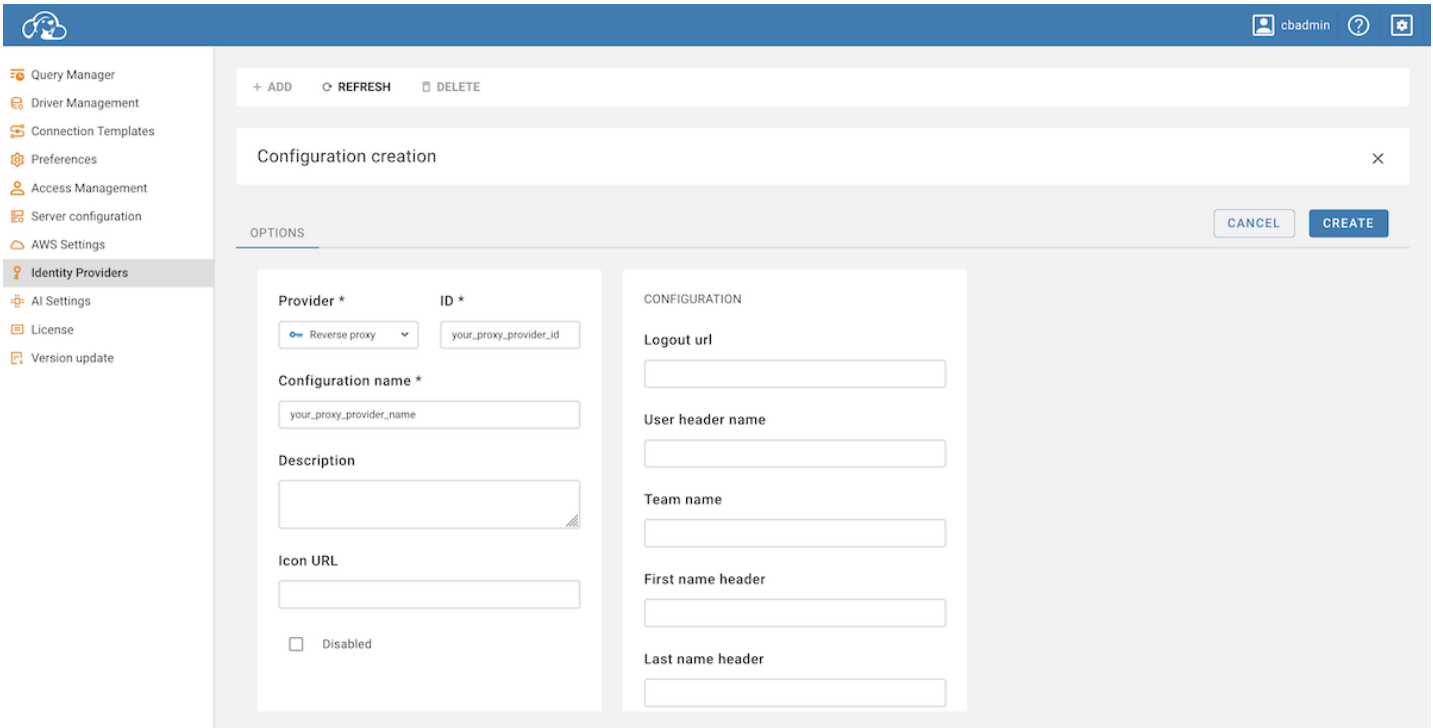
```
"app": {
  "authConfigurations": [
    {
      "id": "your_proxy_id",
      "provider": "reverseProxy",
      "displayName": "your_proxy_username",
      "disabled": true,
      "iconURL": "",
      "description": "",
      "parameters": {
        "logout-url": "https://link_if_needed",
        "user-header": "",
        "team-header": "",
        "team-delimiter": "",
        "first-name-header": "",
        "last-name-header": ""
      }
    }
  ]
}
```

Important: Ensure you include the mandatory fields `id`, `provider`, and `displayName`. The `provider` name must be set to `reverseProxy`.

Step 2.2: Reverse proxy identity provider configuration in Enterprise and Team Editions

To configure reverse proxy authentication in the Enterprise and Team Editions of CloudBeaver using the graphical user interface (GUI), follow these steps:

- 1) Log in as an administrator.
- 2) Navigate to **Settings -> Server configuration** in the CloudBeaver interface.
- 3) Click on the **+ Add** button to create a new authentication provider.
- 4) In the Provider dropdown menu, select **Reverse Proxy**.
- 5) Enter a unique identifier in the ID field and a name for the configuration in the **Configuration name** field.
- 6) Click on **Save** to apply the changes.



Step 3: Configuring default HTTP header fields

Configure the standard HTTP header fields as follows:

Header	Description
X-User	user login
X-Team	user teams
X-First-name	user profile firstname
X-Last-name	user profile lastname
X-Role	user roles, only for DBeaver Team edition

Header example

Consider a user named `newuser`, belonging to both `user` and `admin` teams. To access an application with reverse proxy header authentication enabled, the following HTTP headers should be set in the request to the CloudBeaver application:

```
X-User: newuser  
X-Role: user|admin  
X-First-name: John  
X-Last-name: Smith
```

Tip: CloudBeaver categorizes users into two default teams: `user` and `admin`. Default delimiter used to separate teams in the header is `|` (could be customized in `team-delimiter` parameter, all characters are allowed).

Single Sign On

Table of contents

[Single Sign-On](#)

[SSO for AWS](#)

[Proxy user permissions](#)

[Notes:](#)

Single Sign-On

CloudBeaver Enterprise supports federated authentication for Single Sign-On (SSO) access into the application.

SSO is an authentication service which permits a user to log in with single credentials to multiple applications.

SSO in Cloudbeaver allows to:

- log in to the application by users who have been given rights.
- get access to databases according to users' roles.

Cloudbeaver supports SAML and OpenID authentication methods for SSO.

SSO for AWS

You can configure SSO access for AWS. In order to provide users permission to your AWS cloud resources (RDS, DynamoDB, etc.) you need to configure AWS federated access proxy user. You can find more information here:

[configuring SAML assertions for the authentication response:](#)

1. Go to the AWS Settings tab and enable the Federated authentication.

2. Add the Proxy User on the same page. You can set the current user or add a new one.

3. Create SAML configuration. You can find more information here: !

When an AWS user is logged into CloudBeaver using SSO, it has [the Proxy User and the IAM user's identity-based permissions](#).

Actual permission set and user role are configured in attribute mappings of SAML integration.

Proxy user permissions

Proxy use must have permissions to access you databases. Besides that it must have permission to generate federated tokens for end-users based on requested roles. Make sure it have following AWS policies: Policy name | Description ---|--- arn:aws:iam::aws:policy/service-role/AWSQuickSightListIAM | Allows to list IAM policies and permissions

Also make sure it has following STS permissions: Permission | Description ---|--- sts:GetSessionToken | sts:TagSession | sts:GetFederationToken | sts:GetAccessKeyInfo | sts:GetCallerIdentity | sts:GetServiceBearerToken |

More info at [GetFederationToken policy](#)

Notes:

CloudBeaver does not keep your authentication information on the server-side and in configuration files.

Once your session expires, you will need to authenticate again. When a user logs out from the application, CloudBeaver also performs a session logout from Id Provider.

SAML

Table of contents

[SAML configuration](#)

[Enabling SAML authentication](#)

[Configuring an external identity provider](#)

[Configuring CloudBeaver integration in an external identity provider](#)

[AWS SSO configuration](#)

[Testing SAML authentication](#)

Note: This feature is available in [Enterprise](#), [AWS](#) and [Team](#) editions only.

SAML configuration

If your Identity Provider uses SAML (Security Assertion Markup Language), follow this guide.

Enabling SAML authentication

Go to the Administration menu and enable **SAML** in the Server configuration tab.

The screenshot shows the CloudBeaver 'Server configuration' page. The left sidebar contains navigation links: Connection Management, Access Management, Server configuration (selected), AWS Settings, Identity Providers, and License. The main content area has a top bar with 'SAVE' and 'CANCEL' buttons. Below this, the 'Server URL' field is empty. The 'Session lifetime' is set to 30. On the right, the 'Navigator simple view' toggle is off. Under 'SERVICES', the 'AWS' toggle is on. The 'AUTHENTICATION SETTINGS' section includes: 'Allow anonymous access' (off), 'Local' (on), 'AWS IAM' (off), 'OpenId' (on), and 'SAML' (on). The 'SECURITY' section includes: 'Save credentials' (on) and 'Save users credentials' (on).

Configuring an external identity provider

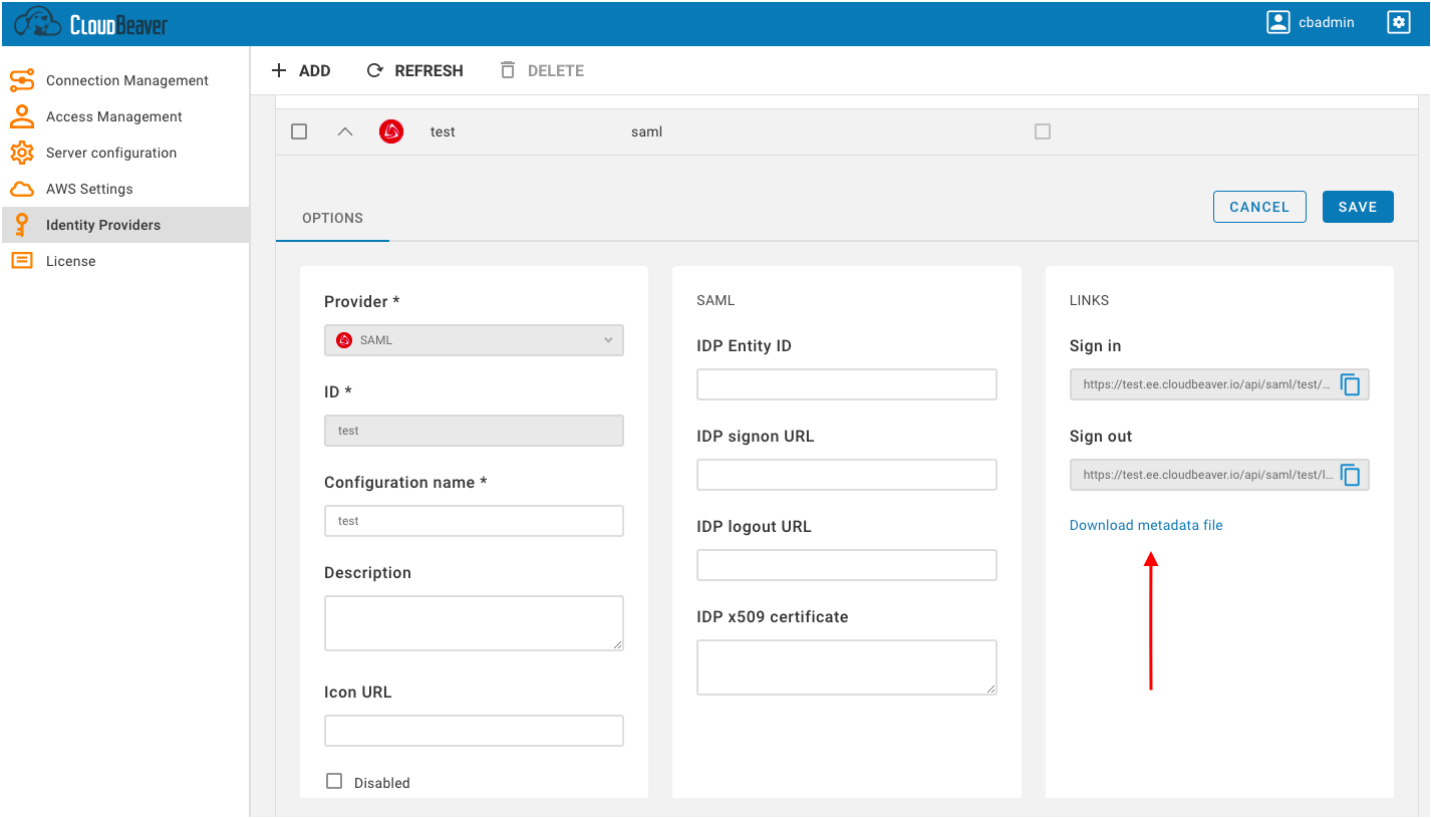
1. Go to the Identity Providers tab and create a new configuration using the SAML IdP details.

The screenshot shows the CloudBeaver 'Identity Providers' configuration page. The left sidebar has 'Identity Providers' selected. The main content area has a top bar with '+ ADD', 'REFRESH', and 'DELETE' buttons. Below this is a 'Configuration creation' modal window. The modal has a 'CANCEL' button and a 'CREATE' button. The 'OPTIONS' tab is selected. The 'Provider' dropdown is set to 'SAML'. The 'ID' field is empty. The 'Configuration name' field is empty. The 'Description' field is empty. The 'Icon URL' field is empty. The 'Disabled' checkbox is unchecked. The 'SAML' section includes: 'IDP Entity ID' field, 'IDP signon URL' field, 'IDP logout URL' field, and 'IDP x509 certificate' field.

2. Add details from your SAML IdP into the new configuration in CloudBeaver.

Configuring CloudBeaver integration in an external identity provider

1. Open the created configuration in CloudBeaver and download the metadata file.



2. Go to the SAML IdP website and add the metadata parameters from the file (entityID and Location) to the SSO access settings, assign users and add the attribute mappings according to the SAML IdP requirements.

Each identity provider has its own configuration procedure, we will show how to do it in AWS in the next chapter.

AWS SSO configuration

Configuration

1. Go to the Identity Providers tab and create a new configuration using the SAML IdP details as it is described above.
2. Add details from your SAML IdP into the new configuration in CloudBeaver.

Configuration in Amazon	Configuration in CloudBeaver
AWS SSO sign-in URL	IDP signon URL

AWS SSO sign-out URL	IDP logout URL
AWS SSO issuer URL	IDP Entity ID

3. You can upload the metadata file to fill parameters automatically.

4. Or you can specify parameters manually:

Parameter	Value
Application ACS URL	<code>https://HOST_NAME/api/saml/CONFIG_ID/acs</code>
Application SAML audience	<code>https://HOST_NAME/api/saml/CONFIG_ID/metadata</code>

Where HOST_NAME is the host name of your CloudBeaver installation, CONFIG_ID is the identifier of your SAML configuration.

Attributes

Attributes explanation:

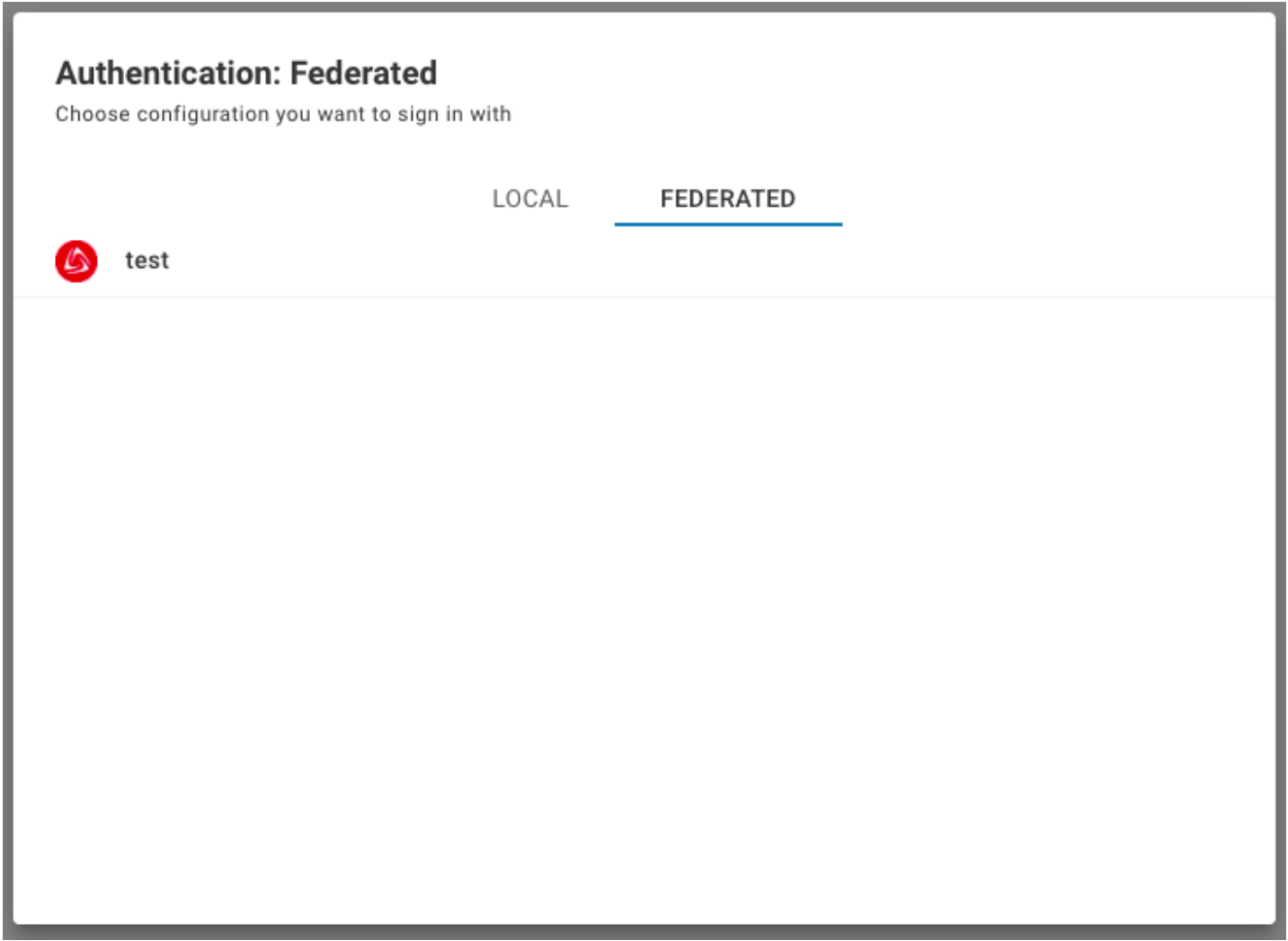
Attribute	Value	Meaning
Subject	<code>\${user:email}</code>	User unique identifier (nameId). It is usually an email address.
https://aws.amazon.com/SAML/Attributes/SessionDuration	1800	Session duration in seconds. 1800 (30 minutes) is the default value
https://aws.amazon.com/SAML/Attributes/Role	roleARN, idpARN	IAM role identifier

Role is the most important attribute, it defines which IAM role will be used for user federation session. Role format: roleARN, idpARN. You can get role ARN in AWS IAM section <https://console.aws.amazon.com/iamv2/home#/roles>. Role ARN looks like this: `arn:aws:iam::123678087624:role/RoleForSAMLAccess`.

You can get IDP ARN in AWS identity providers page https://console.aws.amazon.com/iamv2/home#/identity_providers. IDP ARN looks like this: `arn:aws:iam::123678087624:saml-provider/GSuiteSAML`.

Testing SAML authentication

The Federated tab becomes available in the CloudBeaver authentication dialog after creating the configuration. The user can select the configuration and thereafter login into the application using SSO.



OpenID

Table of contents

[Overview](#)

[Configuration steps](#)

[Step 1: Enabling OpenID Authentication](#)

[Step 2: Adding an Identity Provider](#)

[Step 3: Logging in](#)

Note: This feature is available in [Enterprise](#), [AWS](#) and [Team](#) editions only.

Overview

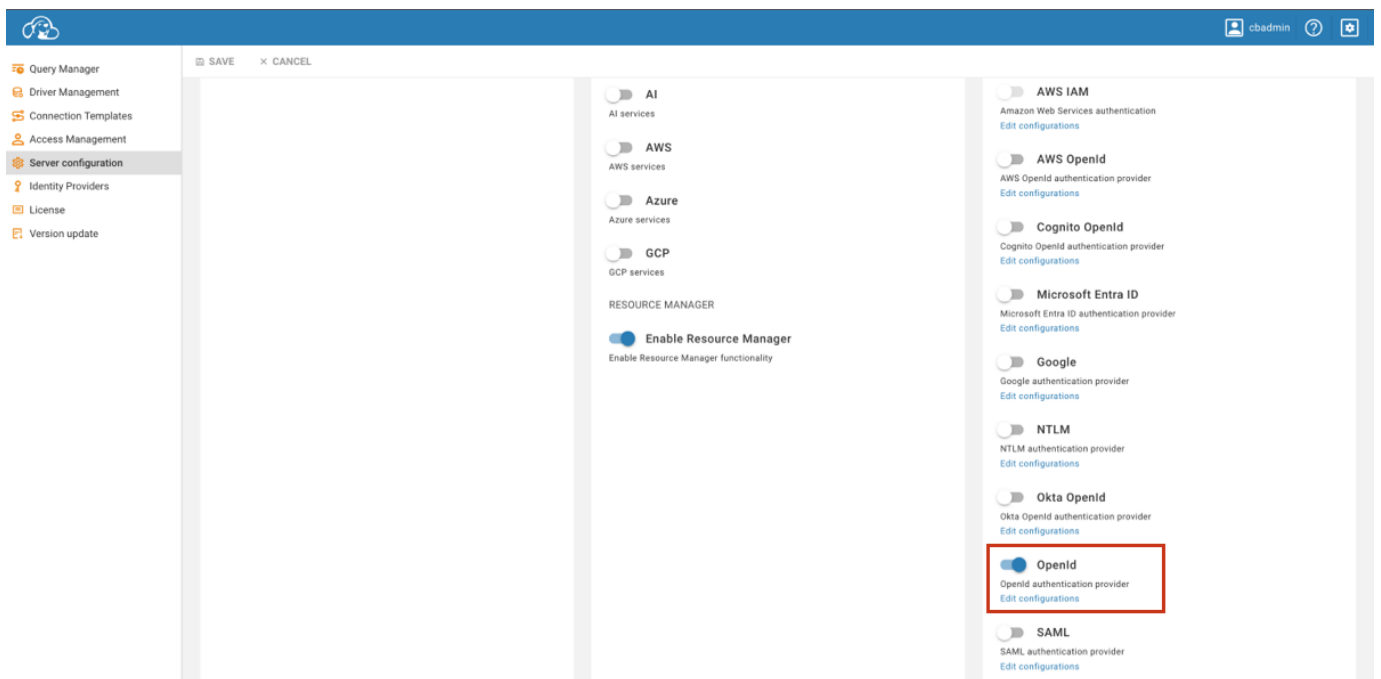
OpenID is an authentication protocol that allows users to authenticate by leveraging their existing identities from an OpenID provider. OpenID is designed for integration with third-party services, making all Identity Providers (IdPs) inherently built for integration with external services. While popular providers like Google can be utilized through OpenID, the protocol is particularly beneficial for custom, specific, or self-hosted identity providers.

For more comprehensive details on this authentication method, you can refer to the [official site](#).

Configuration steps

Step 1: Enabling OpenID Authentication

1. As an administrator, go to **Settings -> Server Configuration**.
2. Find the **OpenID** option in the Authentication Settings section and activate this setting to enable OpenID authentication.



3. Save the changes.

Step 2: Adding an Identity Provider

1. As an administrator, navigate to **Settings -> Identity Providers**.
2. Click on the **+ Add** button.
3. Fill in the following fields:

Field	Description

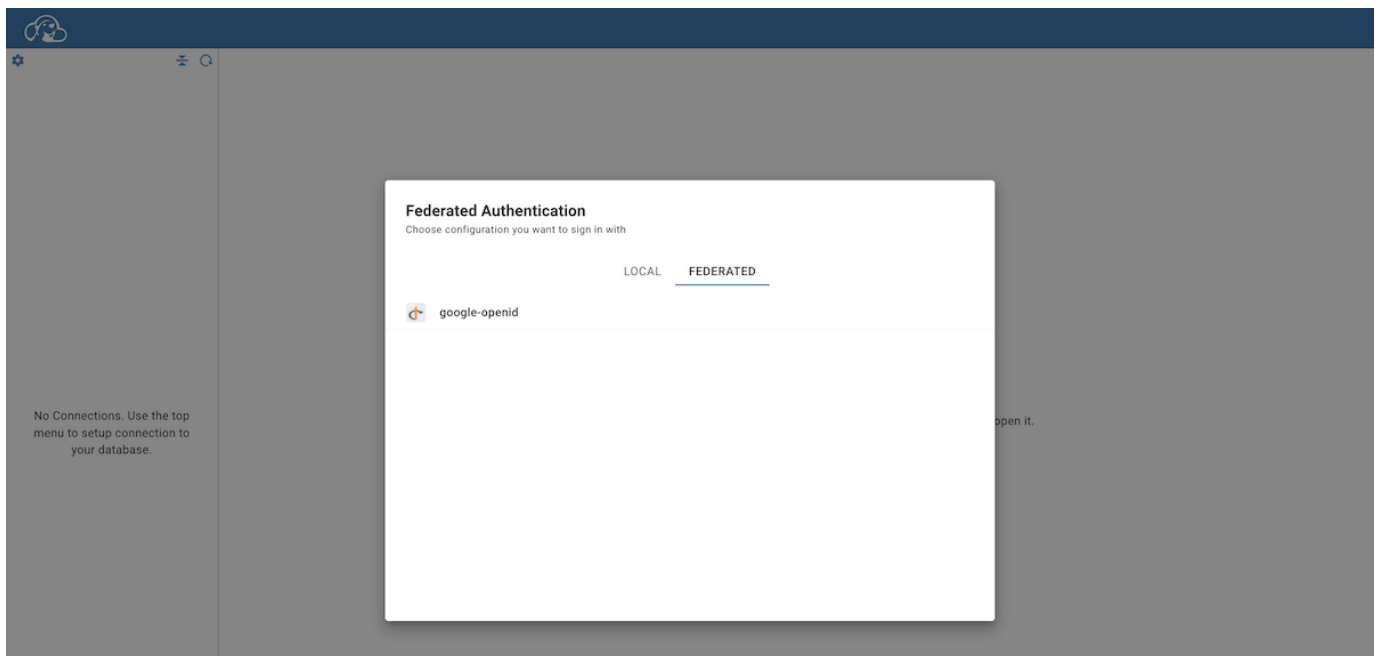
Provider	Select OpenID from the dropdown menu.
ID	Enter a unique identifier for the configuration.
Configuration name	Enter a descriptive name for this configuration.
Description	Provide a brief description of this identity provider configuration.
Icon URL	Enter the URL of an icon to represent this provider.
Disabled	Leave unchecked to enable this identity provider.
Client ID	The client identifier provided by the OpenID Connect provider.
Client Secret	A secret key associated with the client ID for authentication.
IDP auth endpoint URL	The endpoint for initiating the authentication process.
IDP token endpoint URL	The endpoint for obtaining access and refresh tokens.

Note: The values for the **Client ID**, **Client Secret**, **IDP auth endpoint URL**, and **IDP token endpoint URL** depend on the specific OpenID Connect provider being used.

4. Click on the **Create** button.
5. **Copy Redirect and Sign out Links:**
 1. Enter the newly created identity provider.
 2. Copy the **Redirect** link and the **Sign out** link.
6. **Update Redirect URIs in your service.**

Step 3: Logging in

1. With the OpenID configuration now established, proceed to the login screen.
2. Select the Federated authentication method, labeled with the **Configuration name** you specified.



3. Clicking on this authentication method will redirect you to your OpenID provider's sign-in page.
4. After successfully authenticating with your OpenID provider, you will be automatically redirected and logged into CloudBeaver.

AWS OpenID

Table of contents

[Overview](#)

[Configuration steps](#)

[Step 1: Enabling AWS OpenID Authentication](#)

[Step 2: Adding an Identity Provider](#)

[Step 3: Logging in](#)

Note: This feature is available in [Enterprise](#), [AWS](#) and [Team](#) editions only.

Overview

AWS-OpenID Authentication uses AWS credentials to authenticate users in applications, leveraging OpenID Connect with AWS IAM. It enables secure, efficient user access control, minimizing separate account management. For comprehensive setup information of AWS OpenID itself, refer to the [official AWS OpenID documentation](#).

Configuration steps

Step 1: Enabling AWS OpenID Authentication

1. As an administrator, go to **Settings -> Server Configuration**.
2. Find the **AWS** option (in the Configuration section) and **AWS OpenID** (in the Authentication Settings section).
Activate this setting to enable AWS authentication.

> **Note:** In CloudBeaver AWS Edition, the **AWS** option is enabled by default.

3. Save the changes.

Step 2: Adding an Identity Provider

1. As an administrator, navigate to **Settings -> Identity Providers**.

2. Click on the **+ Add** button.

3. Fill in the following fields:

Field	Description
Provider	Select AWS OpenID from the dropdown menu.
ID	Enter a unique identifier for the configuration.
Configuration name	Enter a descriptive name for this configuration.
Description	Provide a brief description of this identity provider configuration.
Icon URL	Enter the URL of an icon to represent this provider.
Disabled	Leave unchecked to enable this identity provider.
Client ID	The client identifier provided by the OpenID Connect provider.
Client Secret	A secret key associated with the client ID for authentication.
IDP auth endpoint URL	The endpoint for initiating the authentication process.
IDP token endpoint URL	The endpoint for obtaining access and refresh tokens.
Role ARN	ARN of the role that will be used by the users during authorization.

Important: The **Role ARN** added during this step acts as the default role. It's not advisable to use an administrator role at this step. It is recommended to use a role with minimum privileges during provider setup.

After the provider is configured, you will see an **AWS Role ARN** field for each user, where you can specify a role with higher privileges, if necessary.

Note: The values for the **Client ID**, **Client Secret**, **IDP auth endpoint URL**, and **IDP token endpoint URL** depend on the specific OpenID Connect provider being used.

4. Click on the **Create** button.

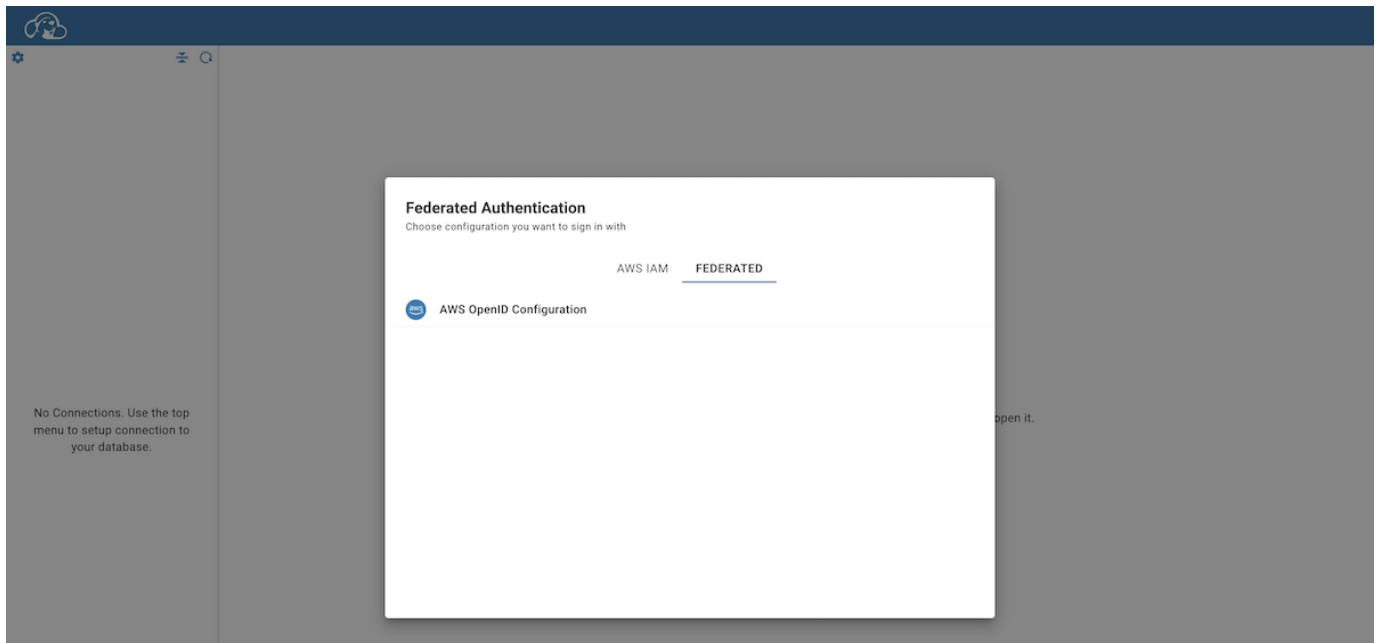
5. **Copy Redirect and Sign out Links:**

1. Enter the newly created identity provider.
2. Copy the **Redirect** link and the **Sign out** link.

6. **Update Redirect URIs in the authorization service.**

Step 3: Logging in

1. With the AWS OpenID configuration now established, proceed to the login screen.
2. Select the Federated authentication method, labeled with the **Configuration name** you specified.



3. Clicking on this authentication method will redirect you to the **Sign in with Google** page.
4. After selecting the necessary account, you will be automatically redirected and logged into the CloudBeaver.
5. **Verify the Integration of AWS and OpenID**
 1. Once logged in, click on your username in CloudBeaver and navigate to the **User Info** tab.
 2. Here, you should see two tokens. Their presence indicates that the integration of AWS and OpenID has been successfully completed, and CloudBeaver has access to the necessary credentials.

AWS IAM

Table of contents

[Overview](#)

[Configuration steps](#)

[For CloudBeaver Enterprise and Team editions](#)

[Step 1: Enabling AWS Authentication](#)

[Step 2: Adding an Identity Provider](#)

[Step 3: Logging in](#)

[For CloudBeaver AWS Edition](#)

[Special characteristics](#)

[IAM permissions](#)

Note: This feature is available in [Enterprise](#), [AWS](#) and [Team](#) editions only.

Overview

CloudBeaver supports AWS IAM authentication to enhance security for database operations within AWS environments.

This guide details the configuration process for AWS IAM authentication, applicable to all CloudBeaver versions, with an emphasis on the initial setup required for the AWS version.

For comprehensive setup information of AWS IAM itself, refer to the [official AWS IAM documentation](#).

Configuration steps

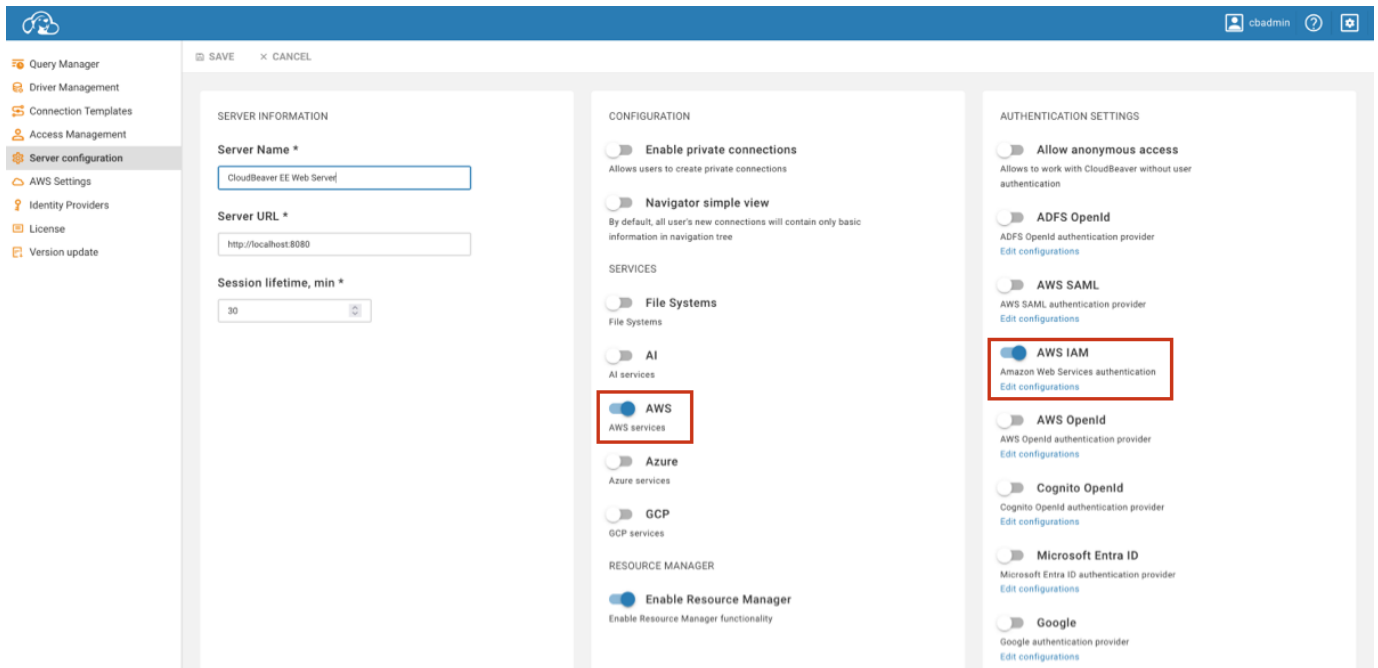
For CloudBeaver Enterprise and Team editions

Step 1: Enabling AWS Authentication

1. As an administrator, go to **Settings -> Server Configuration**.

2. Find the **AWS** option (in the Configuration section) and **AWS IAM** (in the Authentication Settings section).

Activate this setting to enable AWS authentication.



3. Save the changes.

Step 2: Adding an Identity Provider

1. As an administrator, navigate to **Settings -> Identity Providers**.
2. Click on the **+ Add** button.
3. Fill in the following fields:

Field	Description
Provider	Select AWS IAM from the dropdown menu.
ID	Enter a unique identifier for the configuration.
Configuration name	Enter a descriptive name for this configuration.
Description	Provide a brief description of this identity provider configuration.
Icon URL	Enter the URL of an icon to represent this provider.
Disabled	Leave unchecked to enable this identity provider.
Account IDs	Enter AWS Account IDs, separated by commas. Only users from these accounts are allowed.
AWS session lifetime	Specify the duration for the AWS session in seconds.

Tip for Account IDs: You can create entries for different Identity Providers for a more flexible configuration.

- Click on the **Create** button.

Step 3: Logging in

- With the AWS configuration now established, proceed to the login screen.
- You will be presented with two options for key types:

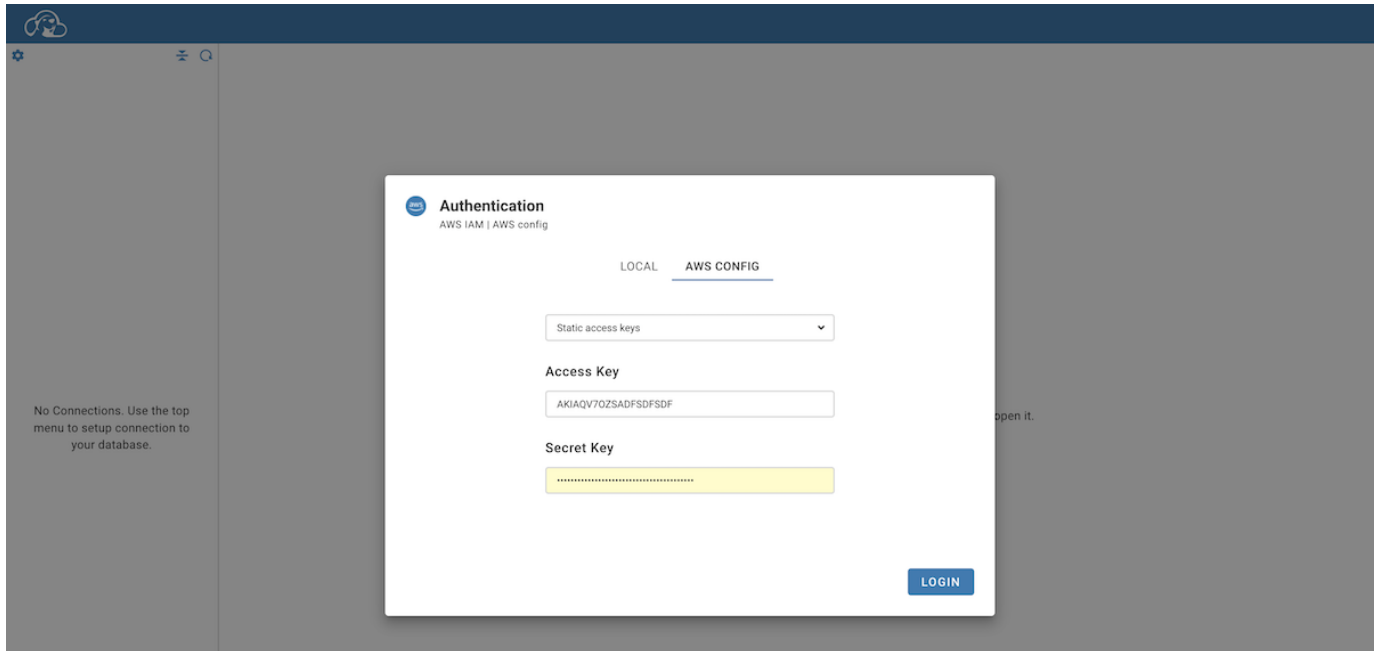
- For permanent credentials, select **Static access keys**.

- If using credentials that change regularly, select **Temporary access keys**.

3. Input your **Access Key** and **Secret Key** in the respective fields.

1. If you selected **Temporary access keys**, enter your **Session Token** in the additional field that appears.

4. Confirm your details and click the **LOGIN** button to authenticate.

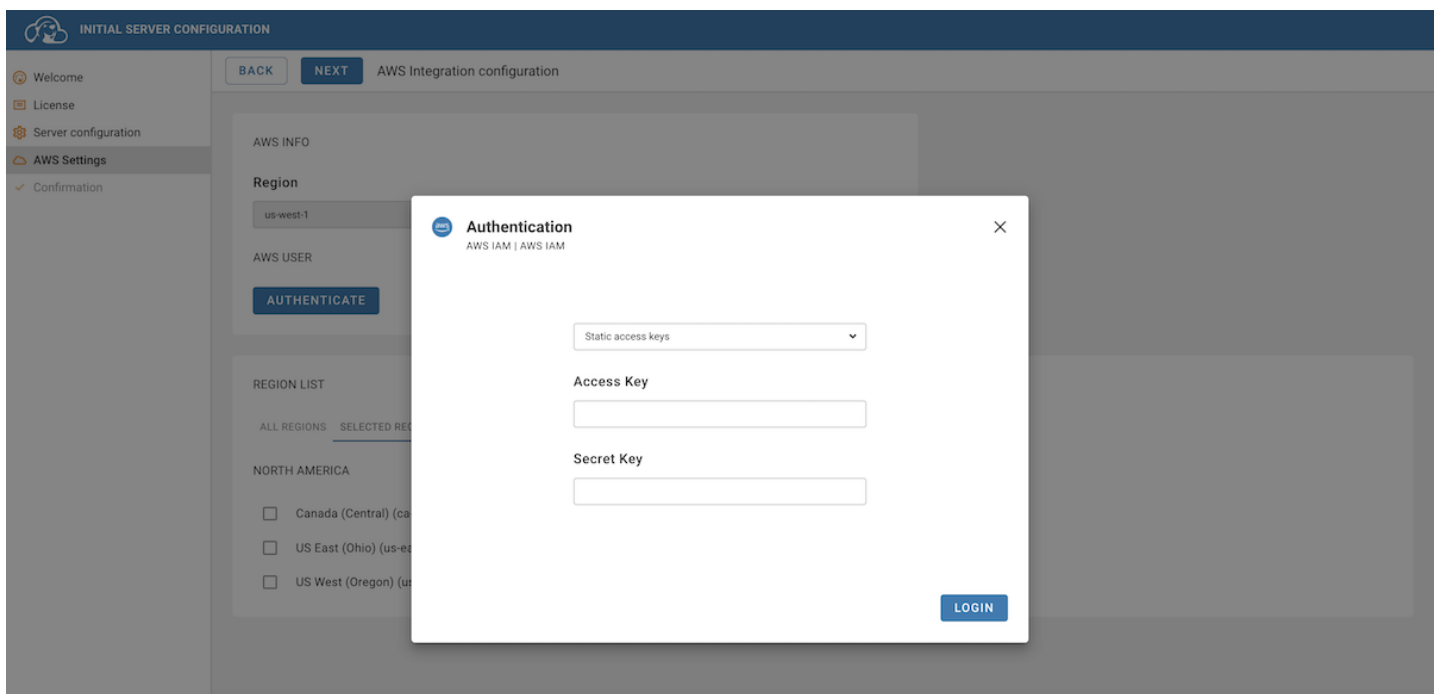


Note: To make databases available for users who log in via AWS IAM, the administrator must first add the desired databases. This is done by navigating to **Connections -> Cloud Connections** and including them in the Database Navigator.

For CloudBeaver AWS Edition

When configuring the CloudBeaver AWS Edition for the first time, AWS IAM credentials are mandatory. This version is optimized for the AWS Marketplace and specifically requires IAM authentication to integrate with AWS services.

During the initial launch, you are required to input your IAM user's Access Key ID and Secret Access Key to establish the necessary AWS integrations.



Special characteristics

- **No Server-Side Key Storage:** CloudBeaver Enterprise is designed not to store AWS access and secret keys on the server, ensuring they are not held in databases or configuration files. This approach is crucial for maintaining the security of your data.
- **Automatic administrator role assignment:** The AWS user responsible for configuring CloudBeaver Enterprise automatically receives administrator privileges in the CloudBeaver Enterprise instance. This user will have comprehensive control over the instance's settings and configurations.
- **AWS account association:** Upon completing the server configuration, the AWS account of the administrator is associated with the CloudBeaver Enterprise instance. This means that only AWS users belonging to this specific account can authenticate and access this instance of CloudBeaver Enterprise.
- **User management within AWS scope:** CloudBeaver Enterprise for AWS does not support the creation of new users within its platform. It solely operates with existing AWS user accounts. Consequently, every user who needs access must authenticate through their AWS account.
- **Database Authentication Requirements:** AWS databases typically have their own authorization mechanisms, requiring additional authentication parameters such as a username and password. For RDS/Aurora databases using IAM authentication, you may only need to provide the database username, leaving the password field empty.

IAM permissions

CloudBeaver Enterprise uses the following AWS services in order to operate with databases (most of them are optional):

- STS (required): used for user authentication
- RDS: list RDS/Aurora instances for cloud databases explorer (describeDBInstances)
- Redshift: list Redshift clusters for cloud databases explorer (describeClusters)
- DynamoDB: all DynamoDB services for DynamoDB operating. Can be read-only for read-only DynamoDB access.
- DocumentDB: list DocumentDB clusters for cloud databases explorer (describeDBClusters)
- IAM (optional): additional user/organization information read (like account organization name)

CloudBeaver Enterprise uses native database clients to connect and operate with most databases. It uses AWS services only to find database instances and configure database connection.

The only exception is the DynamoDB service which is a database driver by itself. You can limit DynamoDB access directly in the AWS console.

AWS OpenId via Okta

Table of contents

[Overview](#)

[Prerequisites](#)

[Configuration steps](#)

Note: This feature is available in [Enterprise](#), [AWS](#) and [Team](#) editions only.

Overview

CloudBeaver allows for database connections through AWS OpenID with Okta authentication. This guide details the process for establishing such connections. Please ensure you meet all the prerequisites outlined below before proceeding with the configuration steps.

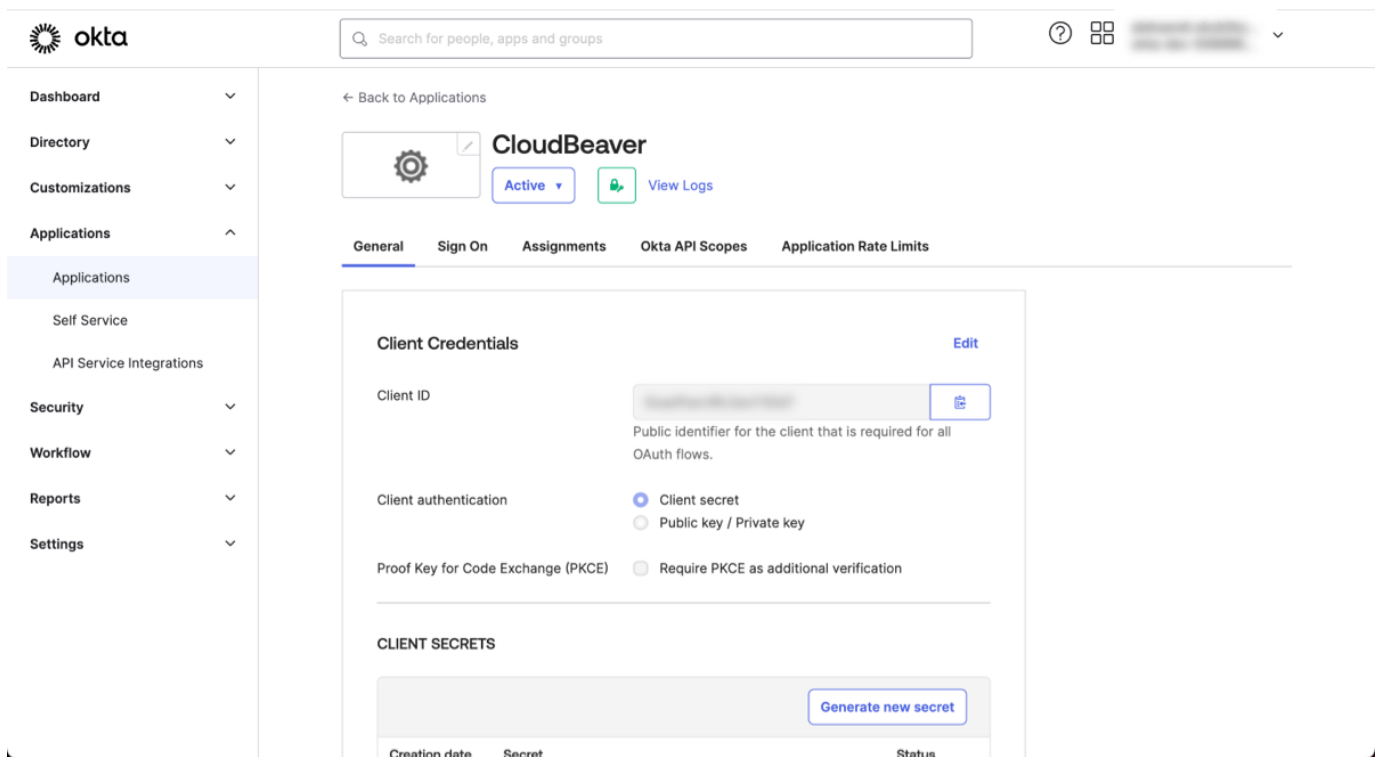
Prerequisites

- **AWS OpenID Configuration:** An active configuration of AWS OpenID is necessary. This includes a properly set up AWS account with OpenID Connect enabled. Additionally, ensure the account has the required permissions to create and manage identity providers and roles.
- **Okta setup:** Access an Okta account with the necessary permissions to configure applications.
- **CloudBeaver administrative access:** Ensure administrative privileges in CloudBeaver.

Configuration steps

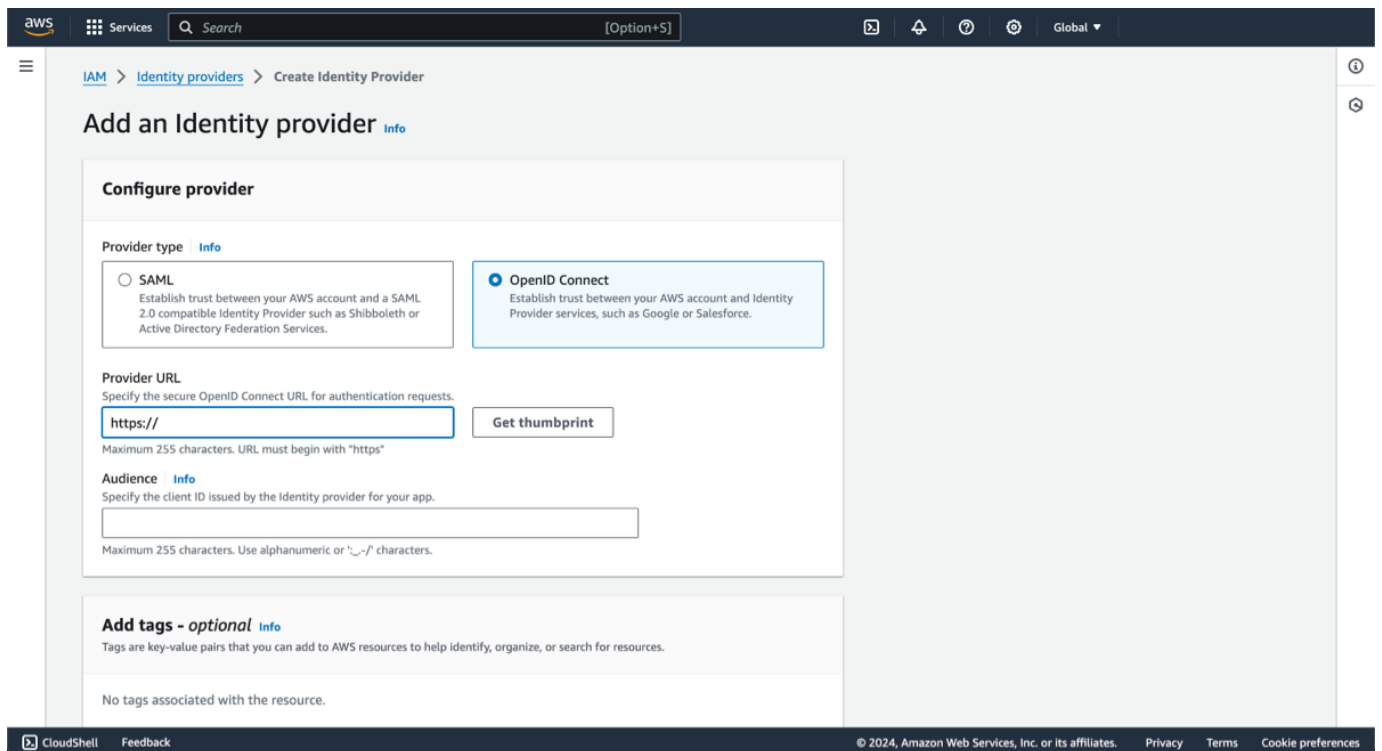
1. Create an Application in Okta:

- Initiate the process by creating an application in Okta. For detailed steps, consult the official [Okta documentation on application creation](#).



2. Add Identity Provider in AWS IAM:

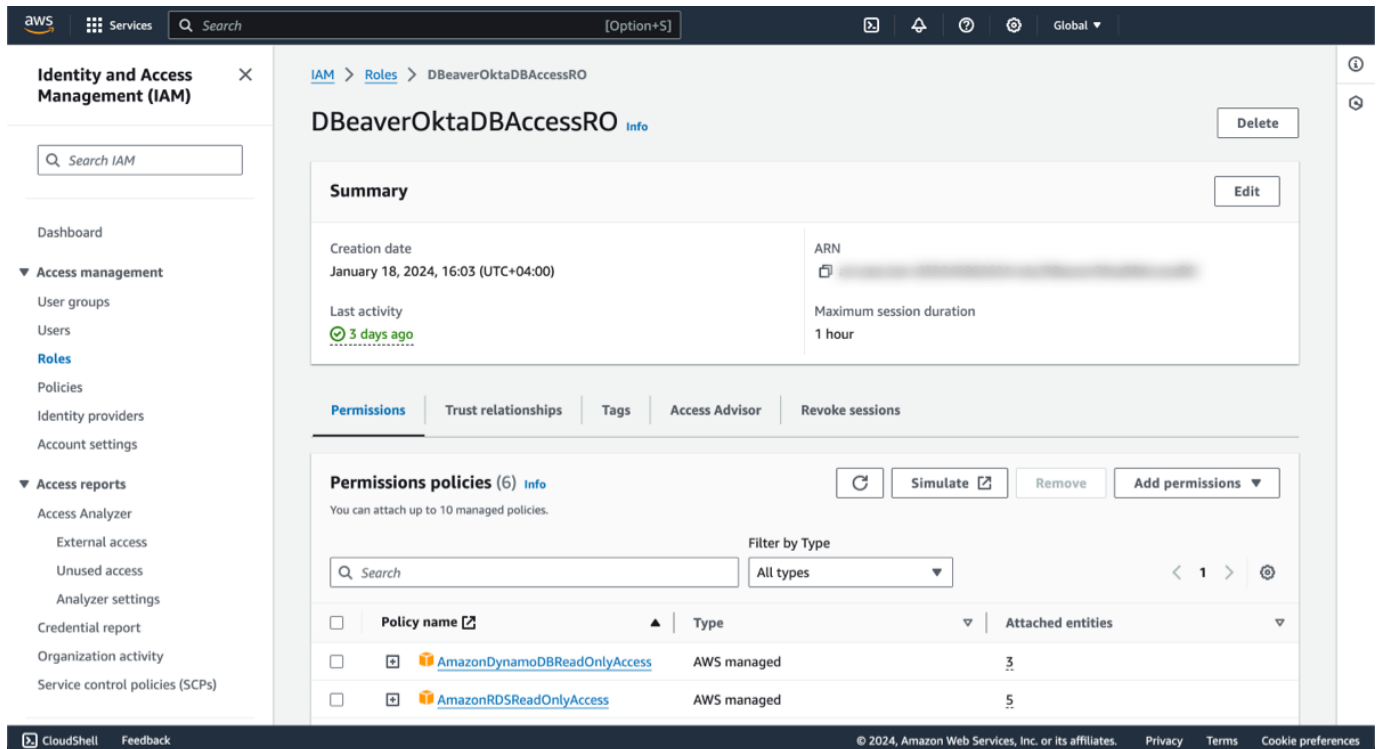
- Add an identity provider in AWS IAM. This allows AWS to authenticate users managed by Okta. For comprehensive instructions, refer to the official [AWS documentation on identity providers](#).



Hints for AWS IAM Identity Provider configuration: > - **Provider URL:** Use your Okta domain, for example, `your-domain.okta.com/`. > - **Audience:** Enter Okta's client ID, which can be copied from the application created in Okta.

3. Configure a Role for Web Identity in AWS:

- The next step is configuring an AWS role for web identity. This role will be used to grant permissions based on the authenticated identity from Okta. For a detailed walkthrough, visit the official [AWS documentation on creating roles for identity providers](#).



4. Log in to CloudBeaver as an Administrator.

5. Enable AWS Services and AWS OpenID Provider:

- Navigate to **Settings -> Administration -> Server Configuration** and select the checkboxes for both **AWS** and **AWS OpenID**.

6. Configure Identity Provider

- Continue to **Settings -> Administration -> Identity Providers**.
- Click on the **+ Add** button to begin configuring a new identity provider.

Below is the table with fields to be completed for configuring the identity provider:

	Field	Description
	Provider	Select AWS OpenID from the dropdown menu.
	ID	Enter a custom name for the identity provider.
	Configuration name	Specify the configuration name.
	Description (optional)	Provide a brief description of the identity provider.
	Icon URL (optional)	Enter the URL of an icon to represent this identity provider in CloudBeaver.
	Client ID	Use the Client ID from the Okta application.
	Client Secret	Use the Client Secret from the Okta application.
	IDP auth endpoint URL	Format as <code>https://{okta_domain}/oauth2/v1/authorize</code> .
	IDP token endpoint URL	Format as <code>https://{okta_domain}/oauth2/v1/token</code> .
	Role ARN	Enter the ARN for the WebIdentity role from AWS.

> **Important:** The **Role ARN** added during this step acts as the default role. It's not advisable to use an administrator role at this step. It is recommended to use a role with minimum privileges during provider setup. >

After the provider is configured, you will see an **AWS Role ARN** field for each user, where > you can specify a role with higher privileges, if necessary.

- After filling in the fields, click on the **Create** button to complete the identity provider configuration.

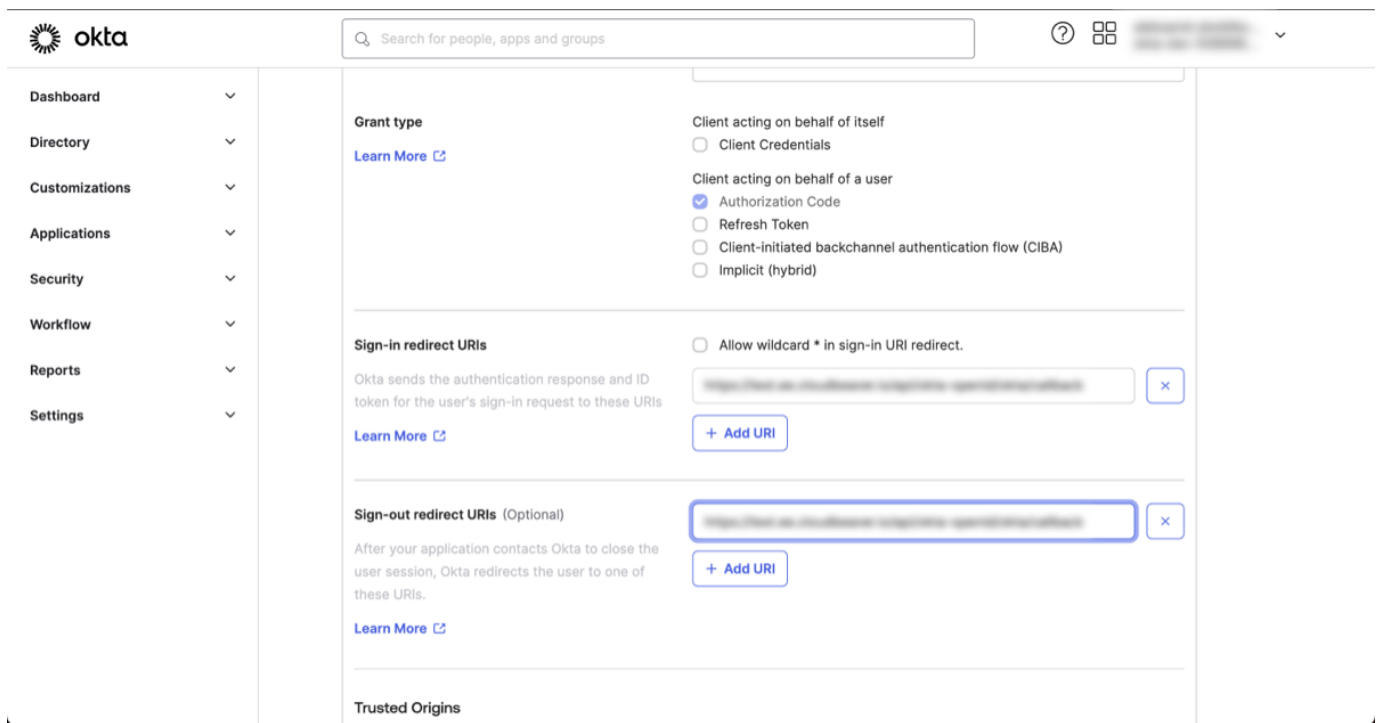
7. Copy Redirect and Sign out Links:

1. Enter the newly created identity provider.
2. Copy the **Redirect** link and the **Sign out** link.

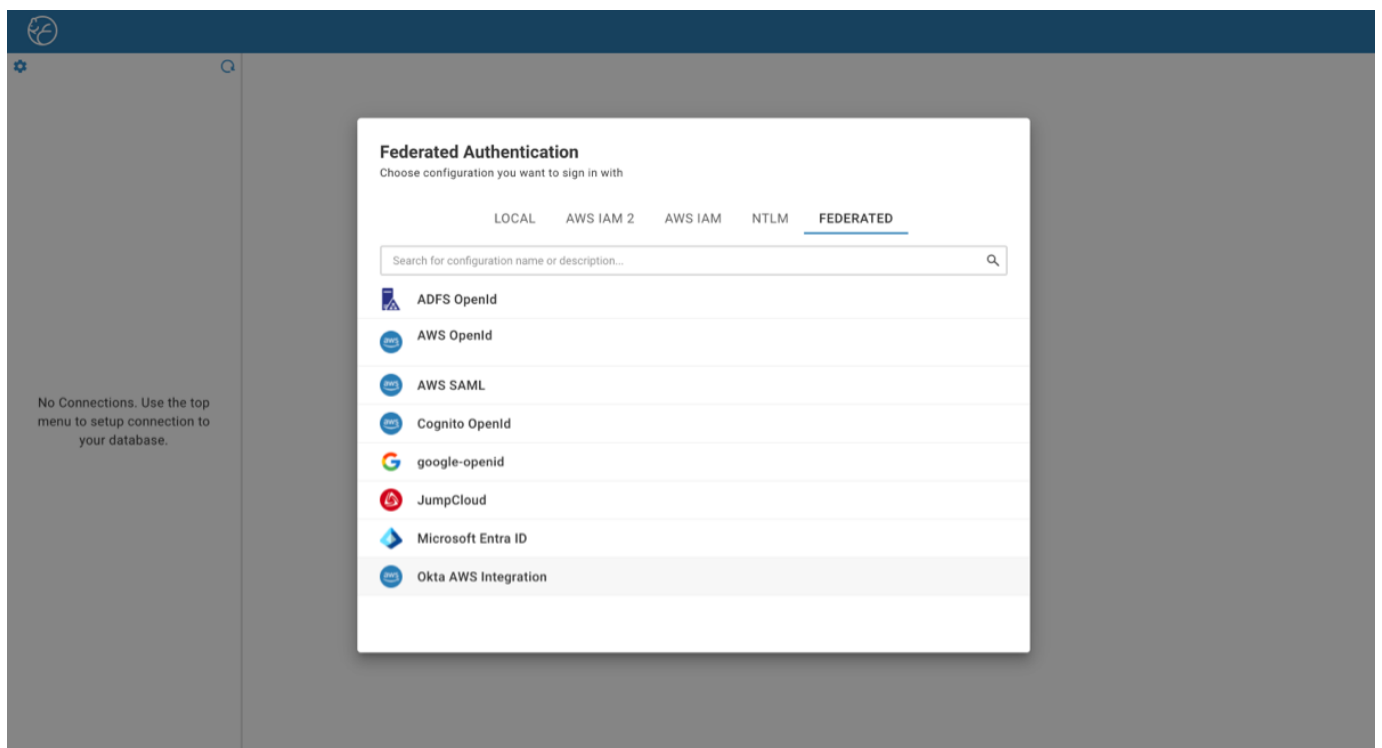
The screenshot shows the Okta Admin Console interface for configuring an identity provider. The left sidebar contains navigation links for Projects, Query Manager, Driver Management, Access Management, Server configuration, AWS Settings, Identity Providers, AI Settings, License, and Version update. The main content area displays a table of configurations with columns for Configuration Name, Provider, Description, and Disabled. A single configuration named 'Okta AWS Integration' is listed with the provider 'aws-openid'. Below the table, the 'OPTIONS' tab is selected, showing a form for configuring the provider. The form is divided into several sections: Provider (AWS Openid), ID (aws-okta), Configuration name (Okta AWS Integration), Description, Icon URL, OPENID (Client ID, Client Secret, IDP auth endpoint URL, IDP token endpoint URL), AWS (Role ARN), and LINKS. The LINKS section contains fields for Sign in, Sign out, and Redirect. The Sign out and Redirect fields are highlighted with a red box, indicating the links to be copied.

8. Update Redirect URIs in Okta:

1. In your Okta application, navigate to **General -> Login**.
2. Under **Sign-in redirect URIs**, paste the copied **Redirect** link.
3. In the same section, locate **Sign-out redirect URIs** and paste the **Sign out** link there.
4. Click **Save** in Okta to finalize these configurations.

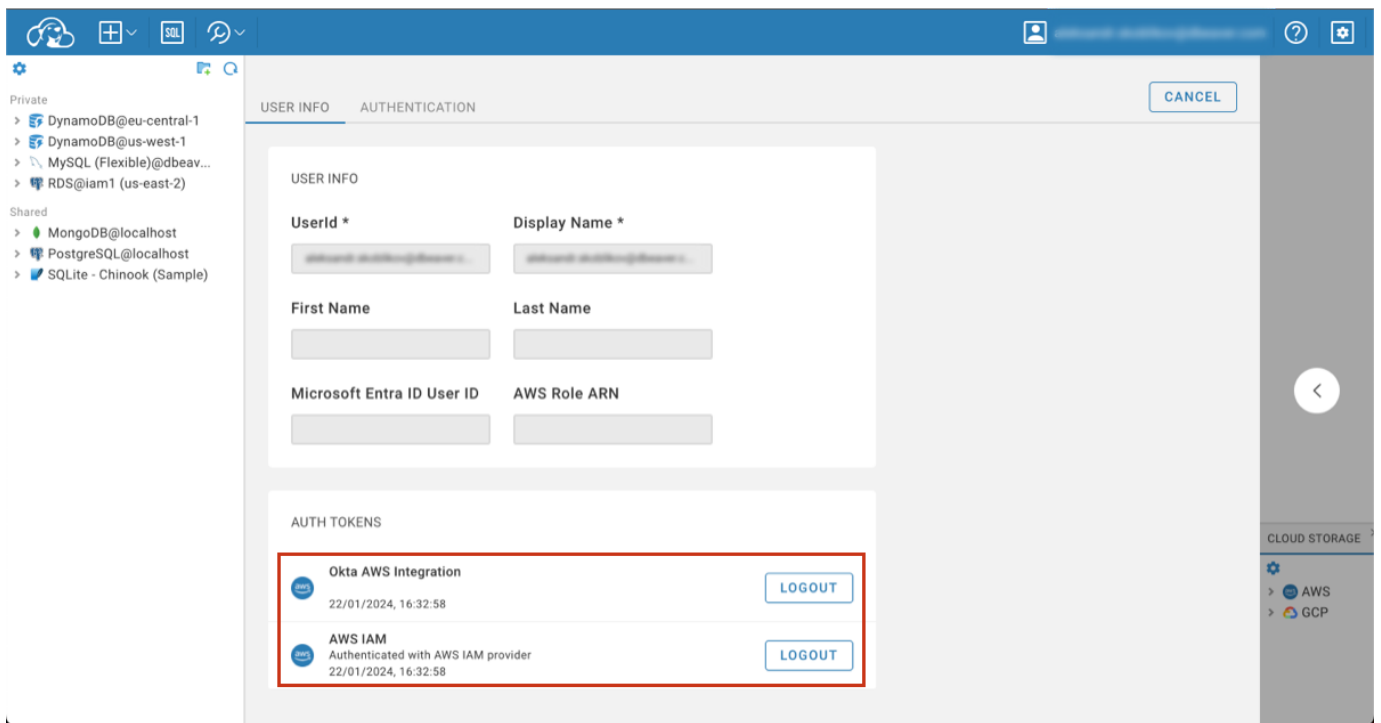


9. Now you can log in through the newly created Federated authentication method in CloudBeaver, using the **Configuration name** you assigned during the setup.



10. Verify the Integration of AWS OpenID and Okta

1. Once logged in, click on your username in CloudBeaver and navigate to the **User Info** tab.
2. Here, you should see two tokens. Their presence indicates that the integration of AWS OpenID and Okta has been successfully completed, and CloudBeaver has access to the necessary credentials.



11. Following successful login, you can access the databases listed in [Cloud Explorer](#) that are integrated with your AWS account.

Snowflake SSO

Table of contents

[Overview](#)

[Prerequisites](#)

[Configuration steps](#)

Note: This feature is available in [Enterprise](#), [AWS](#) and [Team](#) editions only.

Overview

CloudBeaver allows for Snowflake connections through OpenID with Okta authentication. This guide details the process for establishing such connections. Please ensure you meet all the prerequisites outlined below before proceeding with the configuration steps.

Note: While this article provides an example specifically for Okta, you may use any Snowflake-compatible identity provider.

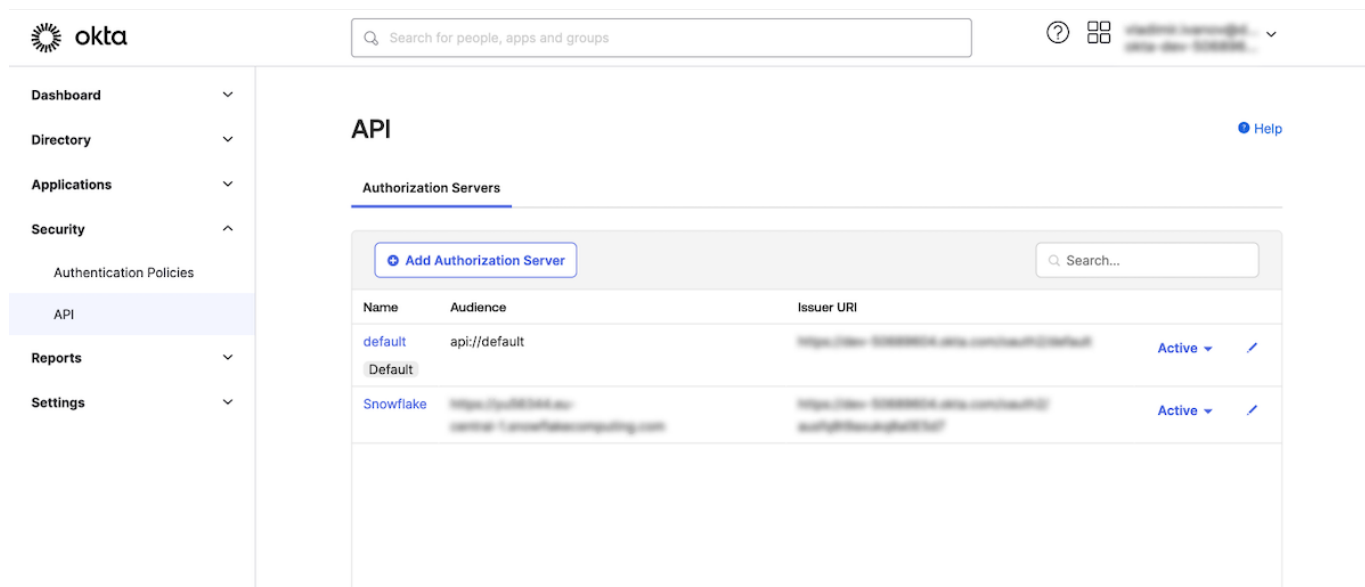
Prerequisites

- **Snowflake Configuration:** You must have a configured Snowflake account. This account should include the necessary roles and users for database access. For guidance on creating users and roles in Snowflake, refer to the [Snowflake documentation](#).
- **Okta setup:** Access an Okta account with the necessary permissions to configure applications.
- **CloudBeaver administrative access:** Ensure administrative privileges in CloudBeaver.

Configuration steps

1. **Create an Authorization Server in Okta:** Begin by setting up an authorization server in Okta. This server will manage the security tokens needed for user authentication and authorization. For a comprehensive guide on

how to create and configure an authorization server, refer to the [Okta documentation on custom authorization servers](#).



2. Create a security integration in Snowflake:

To facilitate secure communication between Snowflake and Okta, it is essential to create a security integration in Snowflake. This integration will authenticate tokens issued by Okta and assign Snowflake data access based on your roles linked with the OAuth token.



1. Proceed to your Snowflake account. Select **Projects -> Worksheets** to access the query editor.
2. In the **Worksheet**, execute an SQL command to create the security integration. The command should include the following parameters:

```
create security integration external_oauth_okta_2
type = external_oauth
enabled = true
external_oauth_type = okta
external_oauth_issuer = '<ISSUER_URL_FROM_OKTA>'
external_oauth_jws_keys_url = '<JWS_KEYS_URL_FROM_OKTA>'
external_oauth_audience_list = ('<AUDIENCE_ARRAY_FROM_OKTA>');
external_oauth_token_user_mapping_claim = 'sub'
external_oauth_snowflake_user_mapping_attribute = 'EMAIL_ADDRESS' or 'LOGIN_NAME'
```

SQL Parameter	Description	Source location
<code><EXTERNAL_OAUTH_ISSUER></code>	The unique identifier for the authorization server in Okta.	Okta: Security -> API -> Authorization Servers -> Settings (Issuer)
<code><AUDIENCE_ARRAY_FROM_OKTA></code>	Specifies the audience for which the token is intended to be used.	Okta: Security -> API -> Authorization Servers -> Settings (Audience)
<code><JWS_KEYS_URL_FROM_OKTA></code>	JSON Web Key Set (JWKS) for token validation.	Okta: Security -> API -> Authorization Servers -> Settings -> Metadata URI (jwks_uri)
<code>EMAIL_ADDRESS</code> or <code>LOGIN_NAME</code>	Attribute used for mapping the user in Snowflake to the Okta token.	Snowflake: Account -> Users -> Select User -> Edit (<code>User Name</code> or <code>Email</code>)

For more in-depth information on creating a security integration within Snowflake, consult the [Snowflake documentation](#).

3. Allow dynamic roles:

Once the security integration is established, you need to enable dynamic role assignment. This step allows users authenticated through Okta to be granted different roles based on their authentication context.



1. Execute the following SQL query to set up dynamic roles.
2. Replace `<SNOWFLAKE_ROLE_1>` , `<SNOWFLAKE_ROLE_2>` , etc., with the actual role names from Snowflake, enclosed in single quotes and separated by commas, forming an array.

```
alter security integration external_oauth_okta_2 SET
    external_oauth_any_role_mode = 'ENABLE'
    external_oauth_allowed_roles_list = ('<SNOWFLAKE_ROLE_1', '<SNOWFLAKE_ROLE_2')
grant USE_ANY_ROLE on integration external_oauth_okta_2 to <SNOWFLAKE_ROLE_1>
```

To determine the appropriate roles, navigate to **Admin -> Users and Roles** in the Snowflake web interface.

Tip: In addition to predefined roles, you can grant a universal role assignment capability within Snowflake.

Use the following command to allow all authenticated users to assume any role:

```
grant USE_ANY_ROLE on integration external_oauth_okta_2 to <SNOWFLAKE_ROLE_1>;
```

Use this setting with caution, as it gives broad permissions to all users.

4. Create API access Scopes in Okta:

After establishing the authorization server, the next step involves creating API access scopes within Okta.

These scopes define the access permissions that an OAuth token grants in Snowflake, ensuring that users can only perform actions aligned with their roles in Snowflake.

The screenshot shows the 'Add Scope' dialog in the Okta web interface. The dialog has the following fields and options:

- Name:** session:role:PUBLIC (with a hint: 'For example: email')
- Display phrase:** Snowflake Public access (with a hint: 'For example: Access your email', '52 characters remaining')
- Description:** (empty field, with a hint: 'For example: This allows you to use your email to login to the app')
- User consent:**
 - ☒ Implicit
 - ☐ Optional
 - ☐ Required
- Block services:**
 - ☒ Block services from requesting this scope
- Default scope:**
 - ☒ Set as a default scope
- Metadata:**
 - ☐ Include in public metadata

At the bottom right of the dialog are 'Create' and 'Cancel' buttons.

Follow these guidelines to create API access scopes:

1. Navigate to your Okta web interface.
2. Go to **Security -> API**.
3. Select the authorization server you created earlier.

4. Click on the **Scopes** tab.

5. Here, you can add new scopes that correspond to the roles and permissions you have defined in Snowflake.

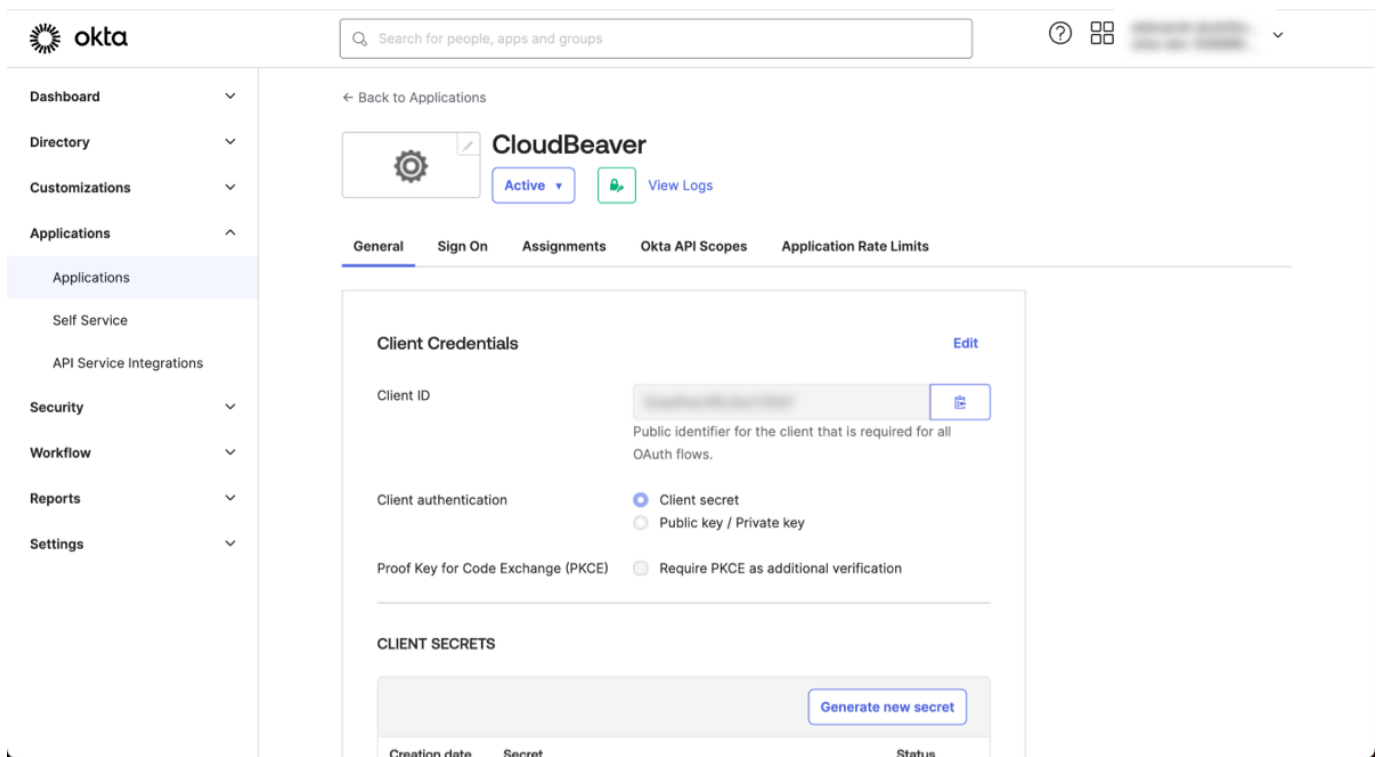
To make sure that users have the right access in Snowflake, match the Okta scopes with Snowflake roles. For instance:

Snowflake role	Corresponding Okta Scope
<code>PUBLIC</code>	<code>session:role:PUBLIC</code>
<code>SYSADMIN</code>	<code>session:role:SYSADMIN</code>
<code>ACCOUNTADMIN</code>	<code>session:role:ACCOUNTADMIN</code>
Granted by <code>grant USE_ANY_ROLE on integration external_oauth_okta_2 to <SNOWFLAKE ROLE_1></code>	<code>session:role-any</code>

For detailed instructions on setting up API access scopes in Okta, consult the [Okta documentation on API scopes](#).

5. Create an Application in Okta:

Continue the process by creating an application in Okta. For detailed steps, consult the official [Okta documentation on application creation](#).



6. Log in to CloudBeaver as an Administrator.

7. Enable OpenID:

Navigate to **Settings -> Administration -> Server Configuration** and select the checkboxes for **OpenID**.

8. Configure Identity Provider

1. Continue to **Settings -> Administration -> Identity Providers**.
2. Click on the **+ Add** button to begin configuring a new identity provider.
3. After filling in the fields, click on the **Create** button to complete the identity provider configuration.

Below is the table with fields to be completed for configuring the identity provider:

Field	Description
Provider	Select OpenID from the dropdown menu.
ID	Enter a custom name for the identity provider.
Configuration name	Specify the configuration name.
Description (optional)	Provide a brief description of the identity provider.
Icon URL (optional)	Enter the URL of an icon to represent this identity provider in CloudBeaver.
Client ID	Use the Client ID from the Okta application.
Client Secret	Use the Client Secret from the Okta application.
IDP auth endpoint URL	The URL for authorization from Okta, labeled as <code>authorization_endpoint</code> in the Okta Metadata URI.
IDP token endpoint URL	The URL for token exchange from Okta, labeled as <code>token_endpoint</code> in the Okta Metadata URI.
Custom scopes	The scopes you set up in Okta to control access, formatted like <code>session:role:PUBLIC</code> .

Tip: The Metadata URI is found in the Okta web interface. You can locate it by navigating to **Security -> API -> Authorization Servers -> Settings -> Metadata URI**.

9. Copy Redirect and Sign out Links:

1. Enter the newly created identity provider.

2. Copy the **Redirect** link and the **Sign out** link.

The screenshot shows the AWS IAM console configuration for an OpenID Connect provider. The 'LINKS' section on the right is highlighted with a red box, showing the 'Sign out' and 'Redirect' links. The 'Sign in' link is also visible above it. The 'Provider *' is set to 'Openid'. The 'Client ID *' is 'SnowflakeClientID'. The 'Client Secret' is 'SnowflakeClientSecret'. The 'IDP auth endpoint URL *' is 'https://okta.com/oidc/authorize'. The 'IDP token endpoint URL *' is 'https://okta.com/oidc/token'. The 'IDP userinfo endpoint URL' is 'https://okta.com/oidc/userinfo'. The 'Configuration name *' is 'Snowflake SSO'. The 'Description' is empty. The 'Icon URL' is 'https://icon-library.com/images/snowflake...'. The 'Disabled' checkbox is unchecked.

10. Update Redirect URIs in Okta:

1. In your Okta application, navigate to **General -> Login**.
2. Under **Sign-in redirect URIs**, paste the copied **Redirect** link.
3. In the same section, locate **Sign-out redirect URIs** and paste the **Sign out** link there.
4. Click **Save** in Okta to finalize these configurations.

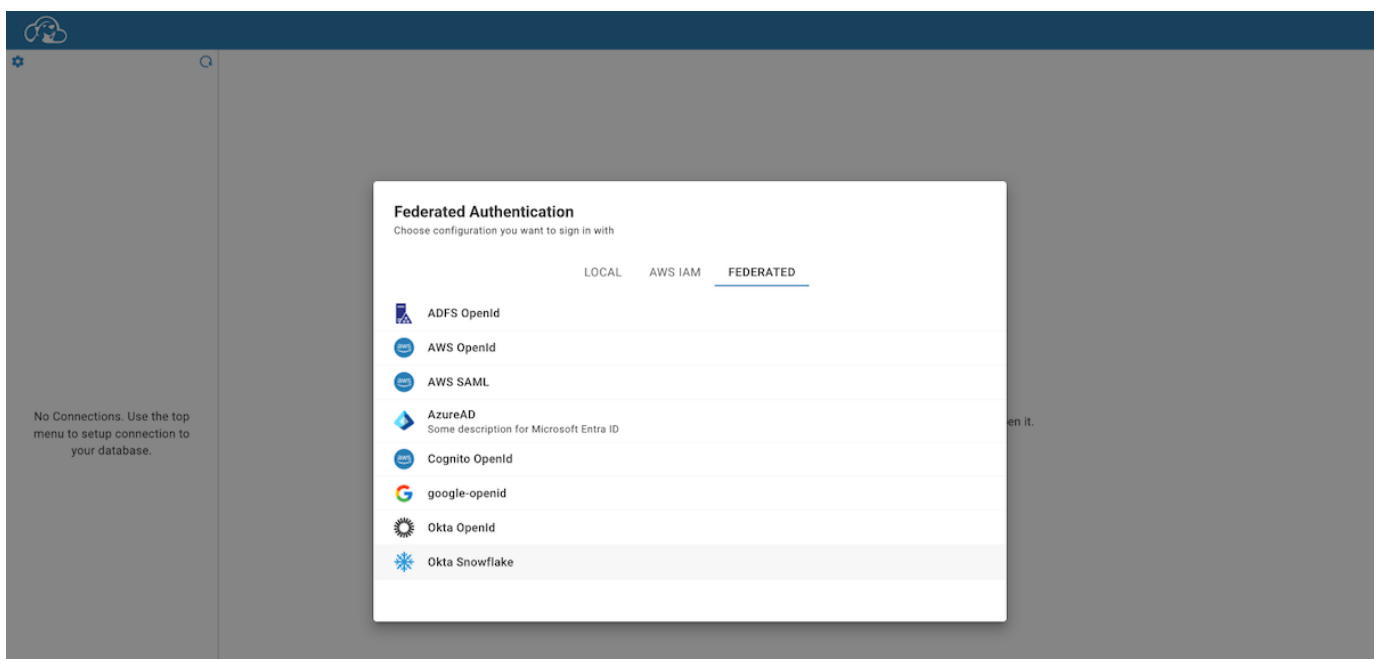
The screenshot shows the Okta application configuration page. The 'Sign-in redirect URIs' section is visible, with a list of URIs and a '+ Add URI' button. The 'Sign-out redirect URIs (Optional)' section is highlighted with a blue box, showing a list of URIs and a '+ Add URI' button. The 'Grant type' section shows 'Authorization Code' selected. The 'Client acting on behalf of a user' section shows 'Authorization Code' selected. The 'Client acting on behalf of itself' section shows 'Client Credentials' selected. The 'Client-initiated backchannel authentication flow (CIBA)' and 'Implicit (hybrid)' options are also visible.

11. Set SSO Authentication for a New Connection

To integrate Single Sign-On (SSO) authentication for database connections after configuring the identity provider:

- Create a new connection by following the guidelines provided in the [Create Connection](#) article.
- Ensure that you select the **SSO** option in the **Authentication** tab.
- For **Identity Provider ID**, use the **ID** defined during the identity provider setup.
- If you leave the **Role** field empty, the system will default to the Role specified in the **Custom scopes** of the Identity provider setup. If **Custom scopes** was not specified, you can manually input different roles here to fine-tune user permissions.

12. Now you can log in through the newly created Federated authentication method in CloudBeaver, using the **Configuration name** you assigned during the setup.



Okta OpenId

Table of contents

[Overview](#)

[Configuration steps](#)

[Step 1: Enabling Okta OpenID Authentication](#)

[Step 2: Adding an Identity Provider](#)

[Step 3: Logging in](#)

Note: This feature is available in [Enterprise](#), [AWS](#) and [Team](#) editions only.

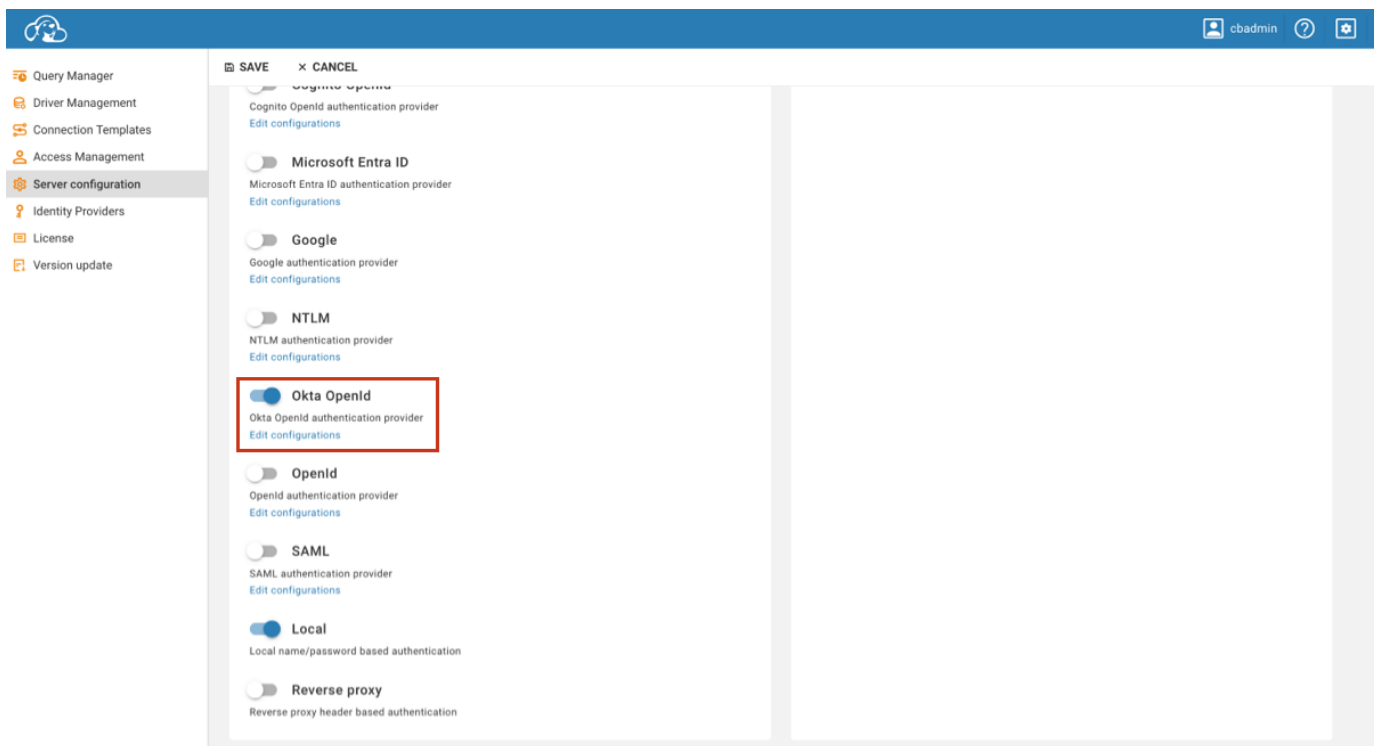
Overview

Okta OpenID Authentication utilizes Okta as an identity provider to authenticate users in applications through OpenID Connect. It simplifies user access control by providing a centralized authentication mechanism, thereby reducing the need for separate account and password management. For detailed setup and configuration instructions for Okta OpenID Connect, refer to the [official Okta documentation](#).

Configuration steps

Step 1: Enabling Okta OpenID Authentication

1. As an administrator, go to **Settings -> Server Configuration**.
2. Find the **Okta OpenID** option in the Authentication Settings section and activate this setting to enable Okta OpenID authentication.



3. Save the changes.

Step 2: Adding an Identity Provider

1. As an administrator, navigate to **Settings -> Identity Providers**.
2. Click on the **+ Add** button.
3. Fill in the following fields:

Configuration creation

OPTIONS

Provider * ID *

Okta OpenId okta

Configuration name *

Okta OpenID

Description

Icon URL

☐ Disabled

OPENID

Client ID *

898EXAMPLEEdjsokfjdsf8

Client Secret

Okta Domain *

example-98209384092.okta.com

☐ Read Okta group information

☐ Read user info

AWS

Name of a AWS role claim

aws-role

CANCEL CREATE

Field	Description
Provider	Select <code>Okta OpenID</code> from the dropdown menu.
ID	Enter a unique identifier for the configuration.
Configuration name	Enter a descriptive name for this configuration.
Description	Provide a brief description of this identity provider configuration.
Icon URL	Enter the URL of an icon to represent this provider.
Disabled	Leave unchecked to enable this identity provider.
Client ID	The client identifier provided by the OpenID Connect provider.
Client Secret	A secret key associated with the client ID for authentication.
Okta Domain	Organization domain in Okta.
Read Okta group information	If checked than Active Directory user group information will be claimed. May be required for Okta permissions integration.
Read user info	Read user profile data, using <code>userinfo</code> endpoint URL.
Name of AWS role claim	Name of AWS role claim that contains the name of the AWS role.

4. Click on the **Create** button.

5. Copy Redirect and Sign out Links:

1. Enter the newly created identity provider.
2. Copy the **Redirect** link and the **Sign out** link.

The screenshot shows the 'Okta OpenId' configuration page in the Okta Admin Console. The left sidebar contains navigation links: Query Manager, Driver Management, Connection Templates, Preferences, Access Management, Server configuration, AWS Settings, Identity Providers (selected), AI Settings, License, and Version update. The main content area has a table with columns: CONFIGURATION NAME, PROVIDER, DESCRIPTION, and DISABLED. Below the table is the 'OPTIONS' section with four tabs: Provider, OPENID, AWS, and LINKS. The 'Provider' tab is active, showing fields for Provider (Okta OpenId), ID (okta), Configuration name (Okta OpenId), Description, and Icon URL. The 'OPENID' tab shows Client ID (0oafEXAMPLEd7), Client Secret, Okta Domain (example-98902843902384.okta.com), and checkboxes for 'Read Okta group information' and 'Read user info'. The 'AWS' tab shows 'Name of a AWS role claim' (aws_role). The 'LINKS' tab shows 'Sign in', 'Sign out', and 'Redirect' links, with the 'Sign out' and 'Redirect' links highlighted by a red box.

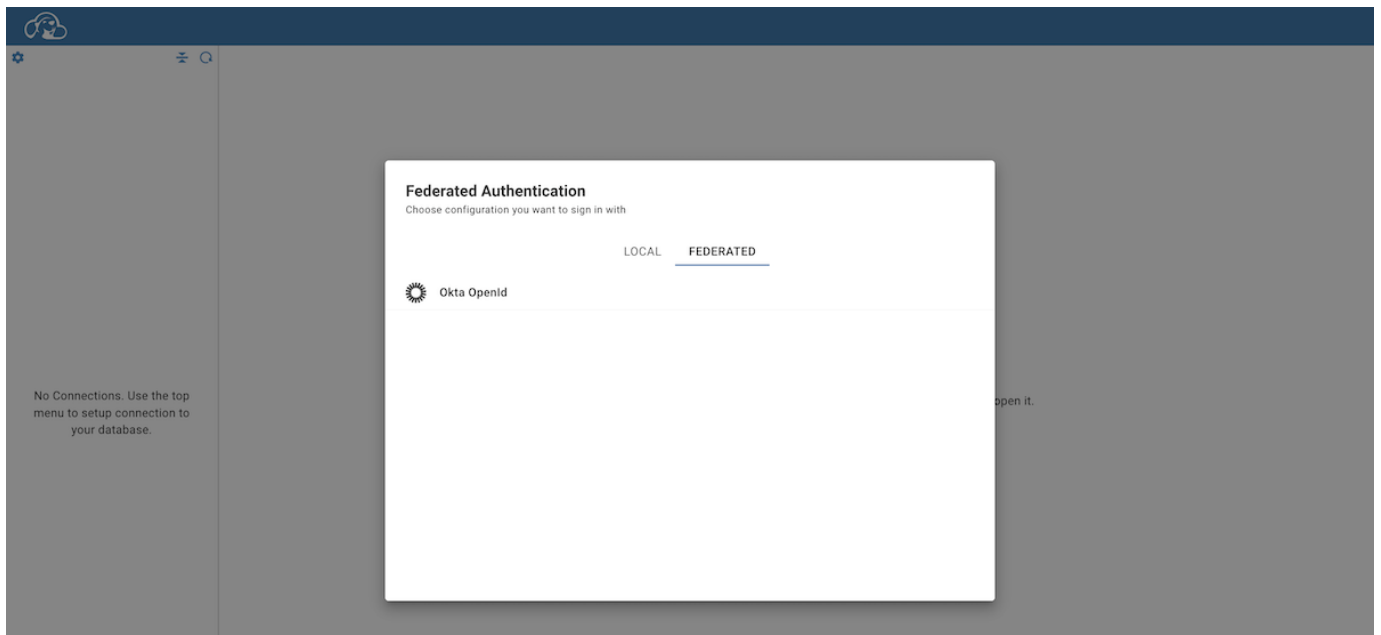
6. Update Redirect URIs in Okta:

1. In your Okta application, navigate to **General -> Login**.
2. Under **Sign-in redirect URIs**, paste the copied **Redirect** link.
3. In the same section, locate **Sign-out redirect URIs** and paste the **Sign out** link there.
4. Click **Save** in Okta to finalize these configurations.

The screenshot shows the 'Grant type' configuration page in the Okta Admin Console. The left sidebar contains navigation links: Dashboard, Directory, Customizations, Applications, Security, Workflow, Reports, and Settings. The main content area has a search bar and a 'Grant type' section with options for 'Client acting on behalf of itself' and 'Client acting on behalf of a user'. The 'Sign-in redirect URIs' section is highlighted, showing a list of URIs and a '+ Add URI' button. The 'Sign-out redirect URIs (Optional)' section is also highlighted, showing a list of URIs and a '+ Add URI' button.

Step 3: Logging in

1. With the Okta OpenID configuration now established, proceed to the login screen.
2. Select the Federated authentication method, labeled with the **Configuration name** you specified.



3. Clicking on this authentication method will redirect you to the Okta page.
4. After filling your username and password of the Okta account, you will be automatically redirected and logged into the CloudBeaver.

Cognito OpenId

Table of contents

[Overview](#)

[Configuration steps](#)

[Step 1: Enabling Cognito OpenID Authentication](#)

[Step 2: Adding an Identity Provider](#)

[Step 3: Logging in](#)

Note: This feature is available in [Enterprise](#), [AWS](#) and [Team](#) editions only.

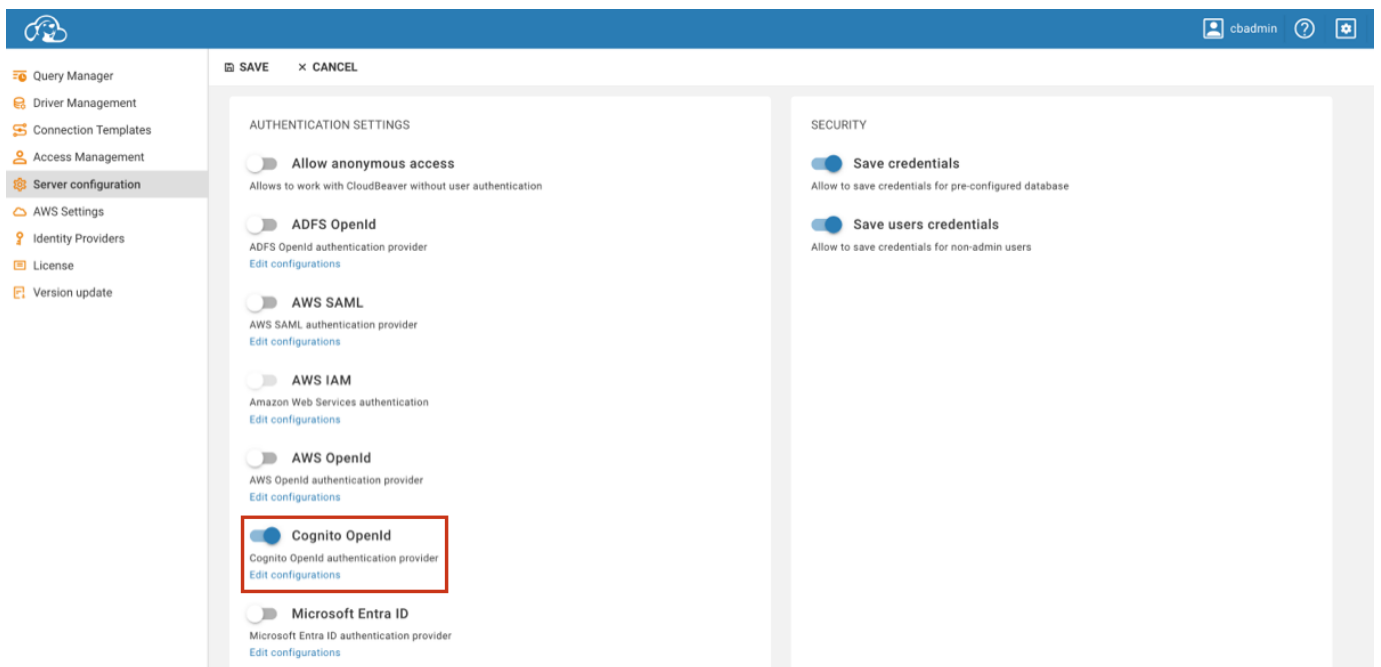
Overview

Cognito OpenID Authentication provides a robust solution for managing user authentication and access in applications. By integrating with Amazon Cognito, it utilizes the OpenID Connect protocol to offer a secure and scalable user management system. This method simplifies the authentication process. For detailed instructions on setting up Cognito OpenID, it is advisable to consult the [official Amazon Cognito documentation](#).

Configuration steps

Step 1: Enabling Cognito OpenID Authentication

1. As an administrator, go to **Settings -> Server Configuration**.
2. Find the **AWS** option (in the Configuration section) and **Cognito OpenID** (in the Authentication Settings section). Activate this setting to enable Cognito OpenID authentication.



3. Save the changes.

Step 2: Adding an Identity Provider

1. As an administrator, navigate to **Settings -> Identity Providers**.
2. Click on the **+ Add** button.
3. Fill in the following fields:

Field	Description

Provider	Select Cognito OpenID from the dropdown menu.
ID	Enter a unique identifier for the configuration.
Configuration name	Enter a descriptive name for this configuration.
Description	Provide a brief description of this identity provider configuration.
Icon URL	Enter the URL of an icon to represent this provider.
Disabled	Leave unchecked to enable this identity provider.
Client ID	The client identifier provided by the OpenID Connect provider.
Client Secret	A secret key associated with the client ID for authentication.
IDP auth endpoint URL	The endpoint for initiating the authentication process.
IDP token endpoint URL	The endpoint for obtaining access and refresh tokens.
IDP userinfo endpoint URL	Fill in with the endpoint URL found in the Amazon Cognito User Pool under "App integration".
Region	Specify the AWS region of your Cognito User Pool.
Identity pool ID	Enter the ID of your Cognito Identity Pool.
Custom role ARN	Provide ARN of the role that will be used by the users during authorization.

4. Click on the **Create** button.

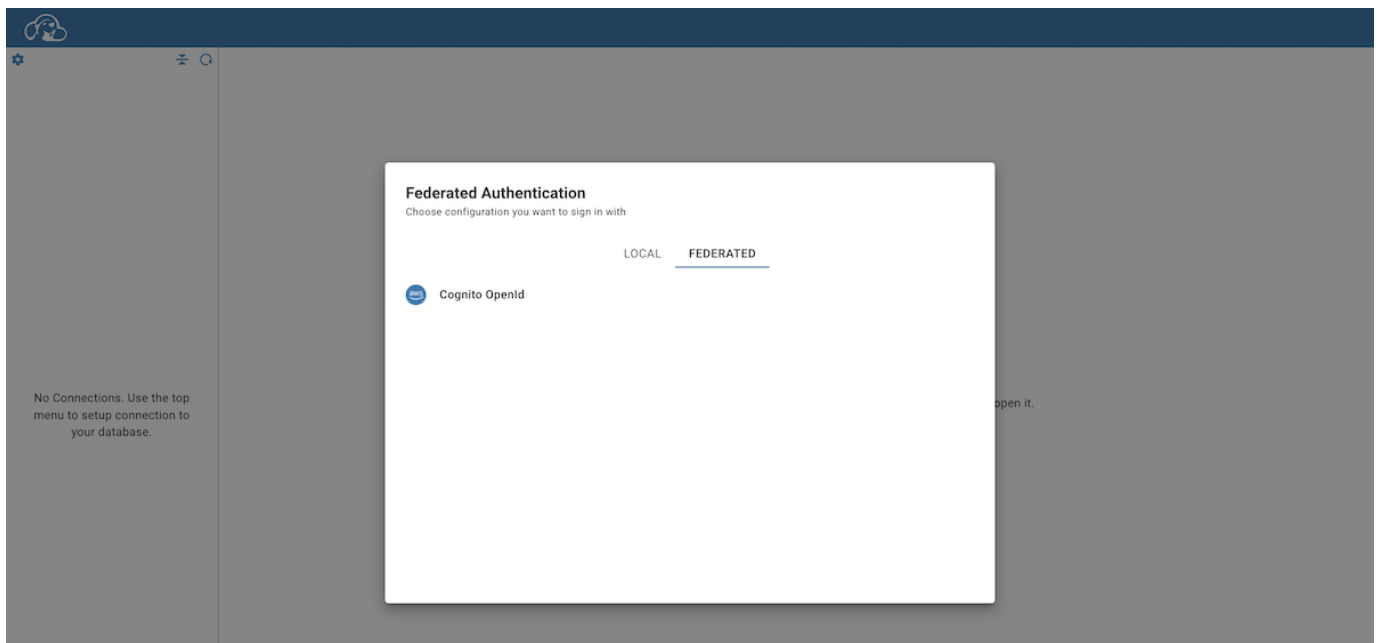
5. Copy Redirect and Sign out Links:

1. Enter the newly created identity provider.
2. Copy the **Redirect** link and the **Sign out** link.

6. Update Redirect URIs in Amazon Cognito.

Step 3: Logging in

1. With the Cognito OpenID configuration now established, proceed to the login screen.
2. Select the Federated authentication method, labeled with the **Configuration name** you specified.



3. Clicking on this authentication method will redirect you to the **Sign in with Google** page.
4. After selecting the necessary account, you will be automatically redirected and logged into the CloudBeaver.

JWT authentication

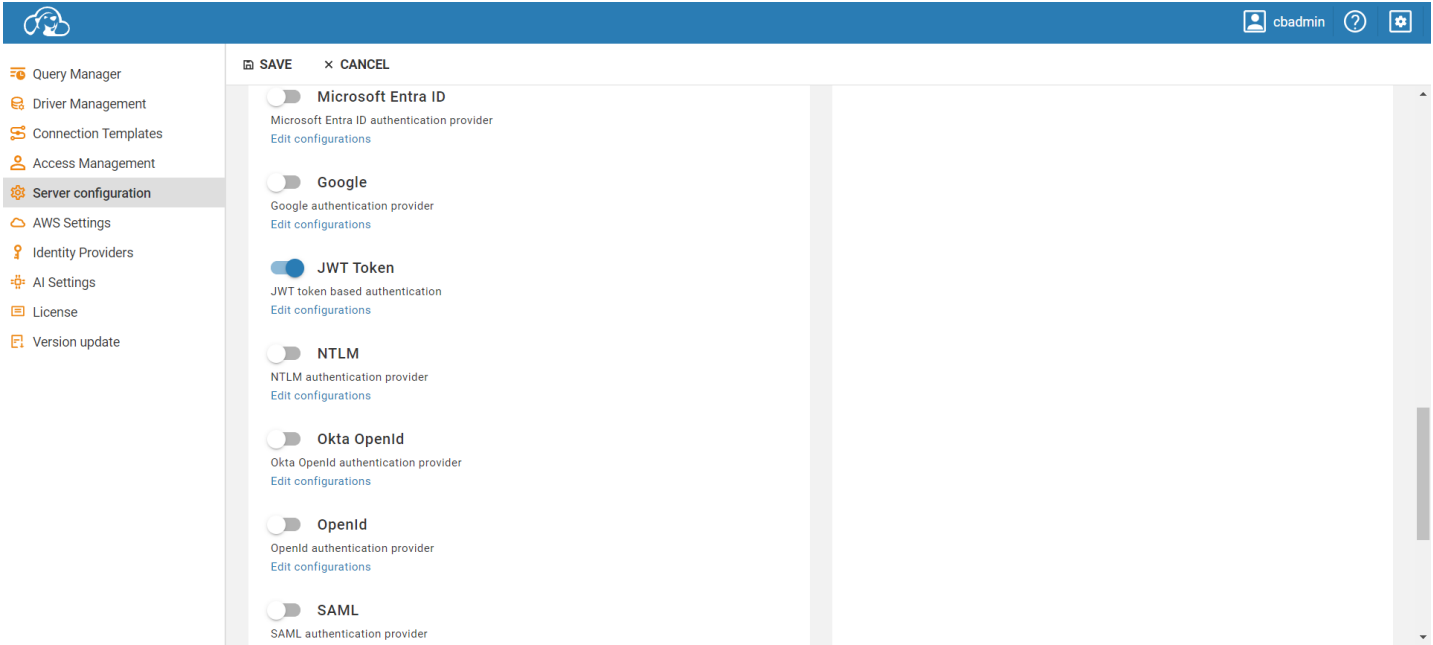
Table of contents

- [JWT Token configuration](#)
 - [Enabling JWT Token authentication](#)
 - [Configuring identity provider](#)
 - [Testing JWTTOKEN authentication](#)
 - [instruction on what parameters to record in fields in case of encoded token](#)

JWT Token configuration

Enabling JWT Token authentication

Go to the Administration menu and enable **JWT Token** in the Server configuration tab.



Configuring identity provider

1. Create your jwt token necessarily with a valid and secure secret (big enough)

You can use our template for the payload:

```
{"email": "example@gmail.com", "firstName": "Test", "lastName": "User"}
```

In this template you can change/add the attributes as you want, but you will need to set them in point 2

2. Go to the Identity Providers tab and create a new configuration using the JWT Token details.

The screenshot shows a web form for configuring an Identity Provider. It is divided into three main sections: General, JWT, and DECRYPTION. The General section includes fields for Provider (JWT), ID (jwt), Configuration name (jwt), Description, Icon URL, and a Disabled checkbox. The JWT section includes fields for Cookie name (jwtToken), Email attribute, Team attribute, First name attribute, and Last name attribute. The DECRYPTION section includes fields for Public key and Secret key.

General	JWT	DECRYPTION
Provider * JWT	JWT	
ID * jwt	Cookie name jwtToken	Public key
Configuration name * jwt	Email attribute	Secret key
Description	Team attribute	
Icon URL	First name attribute	
<input type="checkbox"/> Disabled	Last name attribute	

- **Cookie name** - is the name of the cookie that you will pass to your browser
- **Email attribute** - is the name of the email attribute in your jwtToken, 'email' by default
- **Team attribute** - is the name of the team attribute in your jwtToken, 'team' by default
- **First name attribute** - is the name of the first name attribute in your jwtToken, 'firstName' by default
- **Last name attribute** - is the name of the last name attribute in your jwtToken, 'lastName' by default
- **Public key** - must be specified if you have encrypted your jwt token (carefully it is not a Secret key).
- **Secret key** - parameter, specified when creating a jwt token, must be large enough and secure, otherwise it will be considered invalid (256 bit), e.g. on jwt.io:

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
    
) ☐ secret base64 encoded
```

Testing JWTToken authentication

In order to test your authentication, you'll need to:

1. Create a jwt token on jwt.io or any other method

Encrypt it if you need to do so

instruction on what parameters to record in fields in case of encoded token

- 1.1. Private Key - in the case of encryption we write in the Secret key The private key must

begin with -----BEGIN PRIVATE KEY-----

end -----END PRIVATE KEY-----

you can insert the key with or without these captions.

- 1.2. Public Key - in the case of coding, we write in the Public key

The public key must

begin with -----BEGIN PUBLIC KEY-----

end -----END PUBLIC KEY-----

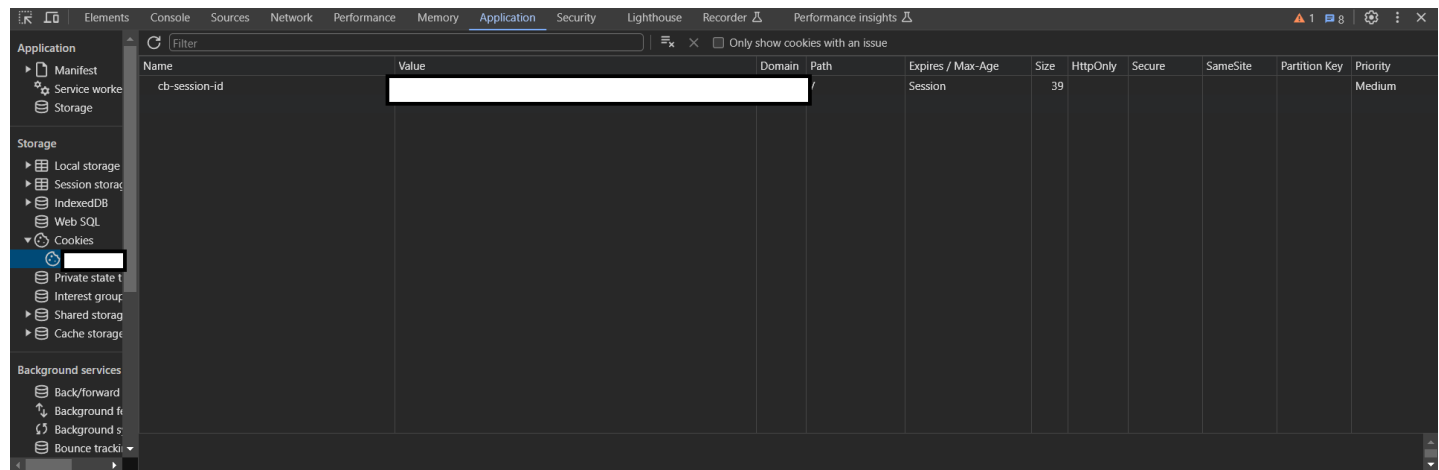
you can insert the key with or without these captions.

You only need to specify the keys when encrypting

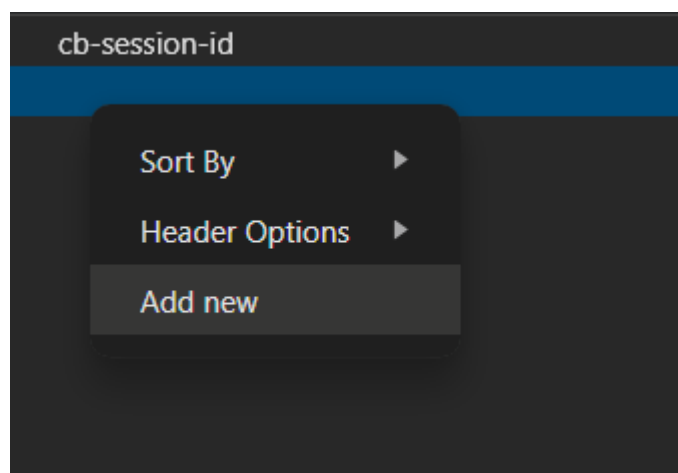
2. Insert your jwtToken into your browser cookie, for example for google chrome:

2.1. Press F12 on Windows or CTRL + SHIFT + C on Linux, also you can right click on the page and click inspect 2.2.

Choose **Application** -> **Cookies** on your website



2.3. Add your cookie with your cookie name



2.4. Insert your cookie name as name, and your jwt token as value, example:



3. Refresh your page

TestUser

USER INFO

CANCEL

USER INFO

Userid *

example@email.com

Display Name *

TestUser

First Name

Test

Last Name

User

AWS Role ARN

Microsoft Entra ID User ID

AUTH TOKENS

JWT Token

Authenticated with JWT Token provider

1/4/2024, 3:24:02 PM

LOGOUT

If you initially go in with a customized JWT and passed the jwt Token via cookie, you don't need to refresh the page, it will already use it, in my example setting up a manual jwt Token substitution

NTLM

Table of contents

[Overview](#)

[Configuration steps](#)

[Step 1: Enabling NTLM Authentication](#)

[Step 2: Adding an Identity Provider](#)

[Step 3: Logging in](#)

Note: This feature is available in [Enterprise](#), [AWS](#) and [Team](#) editions only.

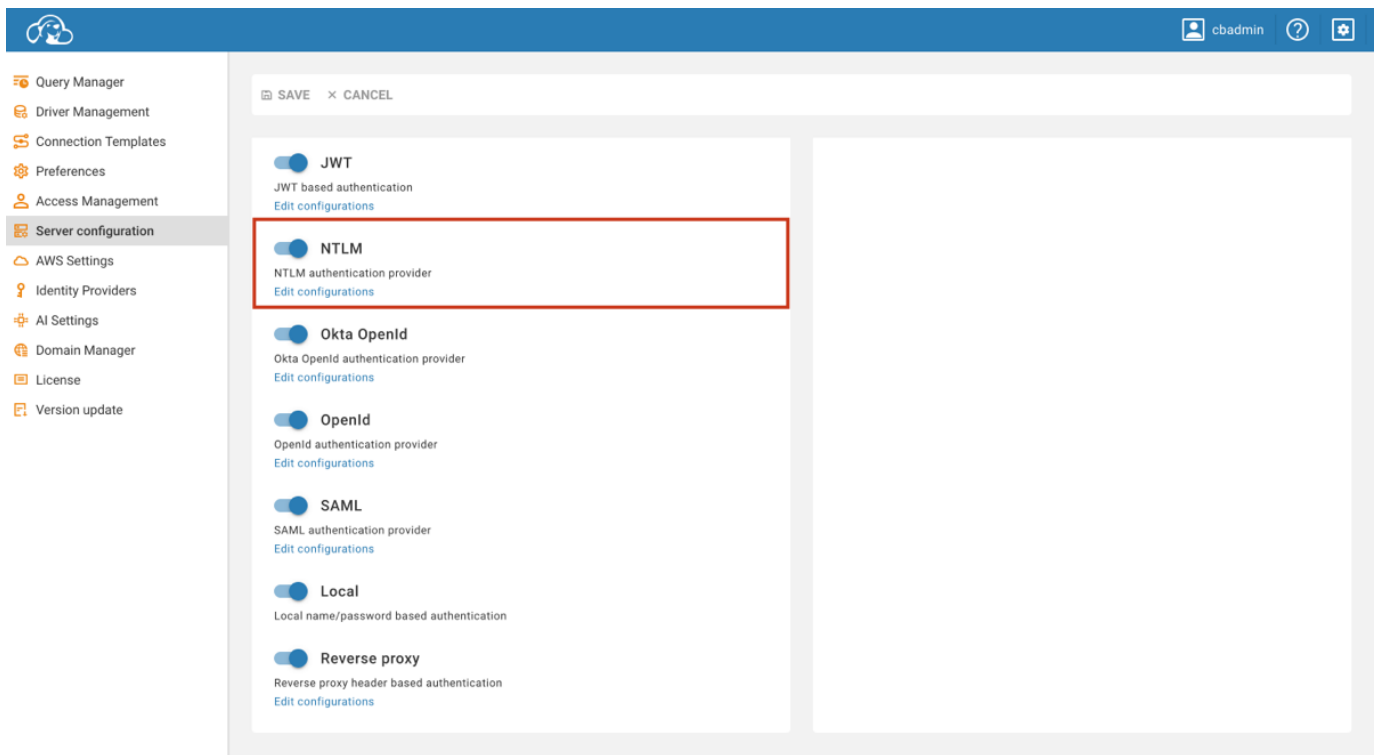
Overview

NTLM (NT LAN Manager) Authentication is a protocol used by Microsoft to authenticate users and provide secure network communications. NTLM uses a challenge-response mechanism for authentication, where the user credentials are never sent over the network directly. For detailed setup and configuration of NTLM, refer to the [official NTLM documentation](#).

Configuration steps

Step 1: Enabling NTLM Authentication

1. As an administrator, go to **Settings -> Server Configuration**.
2. Find and activate the **NTLM** option in the Configuration section.



3. Save the changes.

Step 2: Adding an Identity Provider

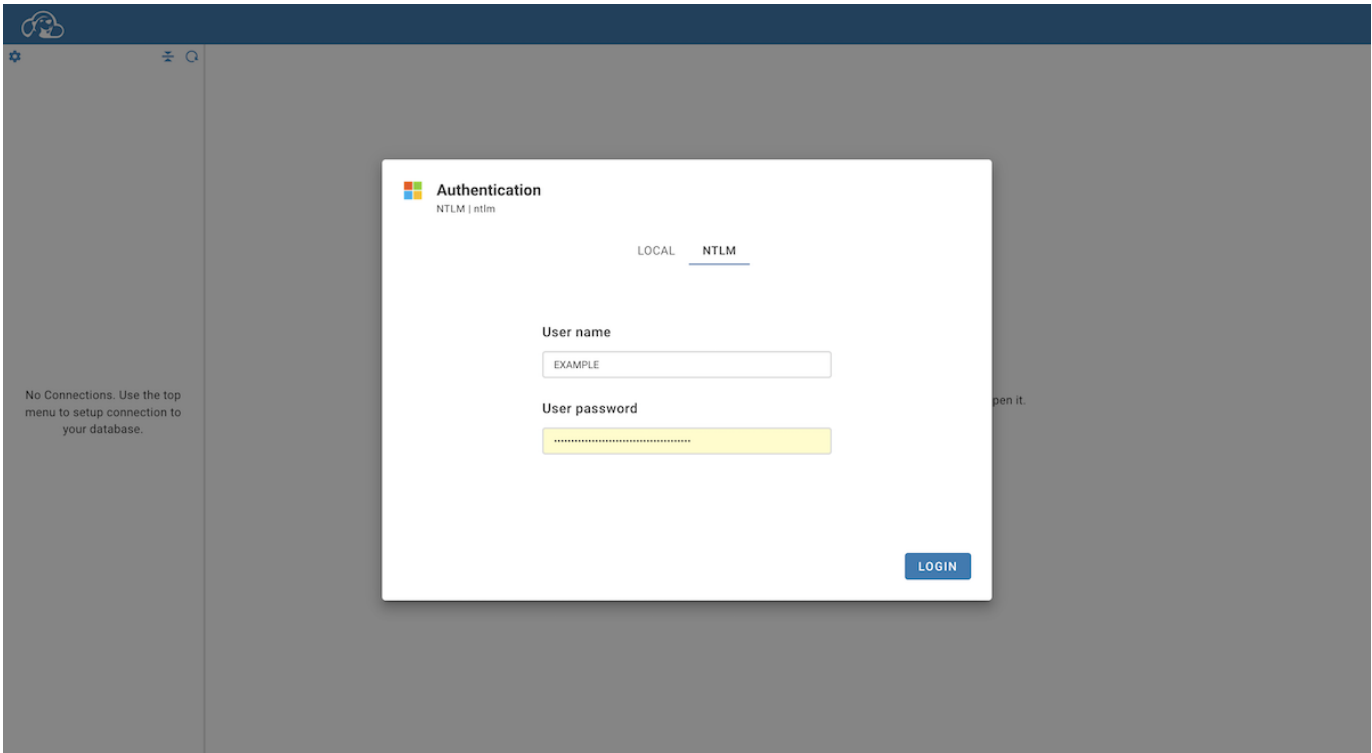
1. As an administrator, navigate to **Settings -> Identity Providers**.
2. Click on the **+ Add** button.
3. Fill in the following fields:

	Field	Description
	Provider	Select NTLM from the dropdown menu.
	ID	Enter a unique identifier for the configuration.
	Configuration name	Enter a descriptive name for this configuration.
	Description	Provide a brief description of this identity provider configuration.
	Icon URL	Enter the URL of an icon to represent this provider.
	Disabled	Leave unchecked to enable this identity provider.
	Host	Enter the server hostname or IP address where NTLM authentication is handled.
	Port	Specify the port number used for NTLM communication.
	Domain	Enter the Windows domain name against which the NTLM server authenticates.

4. Click on the **Create** button.

Step 3: Logging in

1. With the **NTLM** configuration now established, proceed to the login screen.
2. Select the **NTLM** authentication method, labeled with the **Configuration name** you specified.



3. Fill in your **User name** and **User password** to log in.

Azure AD authentication

Table of contents

- [Overview](#)
- [Enabling Azure AD authentication provider](#)
- [Azure Active Directory Configuration](#)
 - [Optional configuration](#)
- [Cloudb Beaver Configuration](#)
 - [Create Identity Provider](#)
 - [Configure Identity Provider](#)
 - [Configure the Redirect link](#)
- [Testing the Azure AD authentication](#)
- [Configure SQL Server databases access](#)

Overview

CloudBeaver supports authorization through Azure AD.

To do this, you must have:

- An active Azure account.
- A configured application in Azure AD.

You will need the following settings for your application from CloudBeaver:

Name	Description
Redirect Url	Url to which Azure AD will send you a response about the authorization attempt request which is taken from the identity provider in CloudBeaver.

- Configured **Azure AD** identity provider in CloudBeaver.

You will need the following settings:

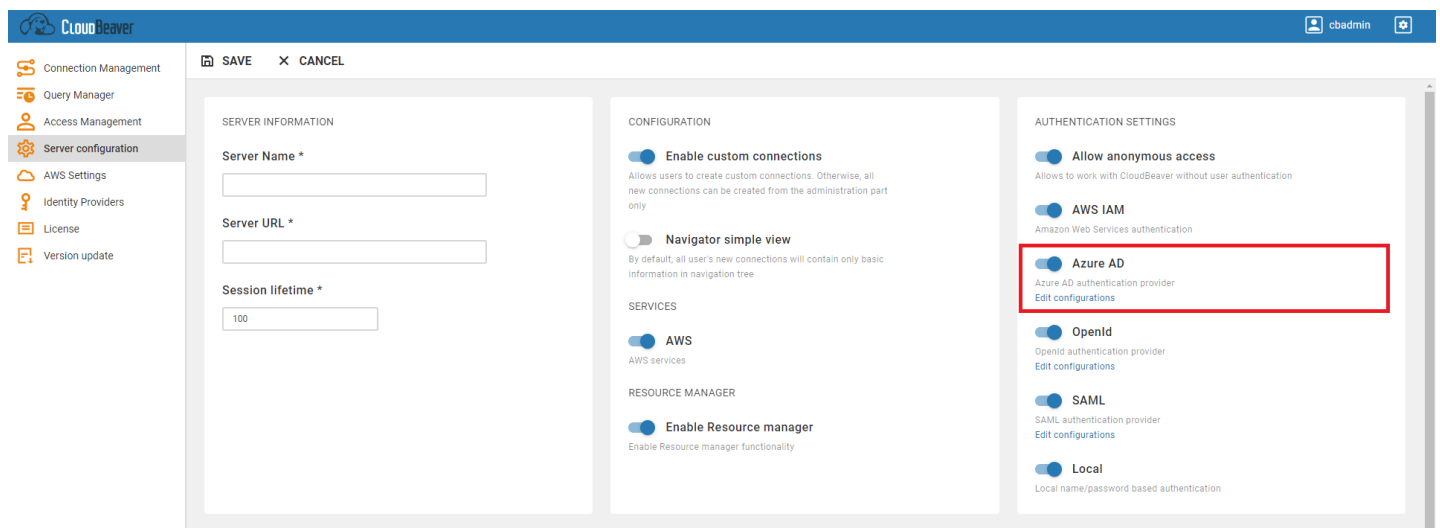
--	--

Name	Description
Domain / Tenant ID	The organization's domain or Tenant ID in Azure
An Application (client) ID	The ID of Azure AD application
A Secret Key	A Secret key from Azure AD application

Enabling Azure AD authentication provider

This step is required for users to be able to use the authorization through Azure AD. However, it might not work immediately as you will need to configure the provider.

1. Log into CloudBeaver as an administrator
2. Go to the Administration menu and enable **Azure AD** in the Server configuration tab.



Azure Active Directory Configuration

Authorization to the Microsoft platform is only possible using registered applications, so we need to create an application in the Azure AD, if it does not exist, and configure it.

1. Register a new Enterprise Application in Azure AD according to the [official Microsoft documentation](#).

2. Cloudbeaver uses the OpenId protocol for authorization in Azure Active Directory.

For this it is necessary to configure the application secrets - more information on how to do this can be found at [official Microsoft documentation](#).

Do not forget to record the value of the secret key because it can only be obtained once. If you do not do this you will have to repeat this step.

Optional configuration

Cloudbeaver supports the ability to read and display information about the user's first and last name from the OpenID token. If you want to support this feature you need to add the **family_name** and **given_name** fields to the response token. More information on how to do this can be found at [official Microsoft documentation](#)

Home > CloudBeaver

CloudBeaver | Token configuration

Search (Ctrl+/) << Got feedback?

- Overview
- Quickstart
- Integration assistant
- Manage
 - Branding & properties
 - Authentication
 - Certificates & secrets
 - Token configuration**
 - API permissions

Optional claims

Optional claims are used to configure additional information which is returned in one or more tokens. [Learn more](#)

+ Add optional claim + Add groups claim

Claim ↑↓	Description	Token type ↑↓	Optional settings
family_name	Provides the last name, surname, or family nam...	ID	- ...
given_name	Provides the first or "given" name of the user, as...	ID	- ...

Cloudbeaver Configuration

Create Identity Provider

To allow users to choose Azure AD as an authorization method, a new identity provider must be created:

1. Go to the Identity Providers tab and create a new configuration using the Azure AD details.

CloudBeaver

cbadmin

Connection Management

Query Manager

Access Management

Server configuration

Identity Providers

License

Version update

+ ADD REFRESH DELETE

OPTIONS

CANCEL CREATE

Provider * ID *

Azure AD

Configuration name *

Description

Icon URL

☐ Disabled

AZURE AD

Domain / Tenant ID

Application (client) ID

Secret Key

Configure Identity Provider

1. Set Domain / Tenant ID

Open Azure Active Directory/Your Directory/Overview page and copy the **Tenant ID** or **Primary domain** (these values are equivalent) value into the Cloudbeaver Azure AD provider **Tenant ID** field. How to get **Tenant ID** value in other ways you can read [here](#).

Home >

Default Directory | Overview

Azure Active Directory

Overview

Preview features

Diagnose and solve problems

Manage

Users

Groups

External Identities

Roles and administrators

Administrative units

Enterprise applications

Devices

App registrations

+ Add Manage tenants What's new Preview features Got feedback?

Overview Monitoring Properties Tutorials

Search your tenant

Basic information

Name	Default Directory	Users	View
Tenant ID	a1b2c3d4-e5f6-7890-b1c2-d3e4f5g6h7i8	Groups	View
Primary domain	a1b2c3d4-e5f6-7890-b1c2-d3e4f5g6h7i8	Applications	View
License	Azure AD Free	Devices	View
Alerts			

2. Set Application (client) ID

Open the application page [registered in this step point 1](#) and copy the value into the Cloudbeaver Azure AD provider **Application (client) ID** field.

Search (Ctrl+ /)

Delete Endpoints Preview features

Overview

Quickstart

Integration assistant

Manage

Branding & properties

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Essentials

Display name

[CloudBeaver](#)

Application (client) ID

00000000-0000-0000-0000-000000000000

Object ID

00000000-0000-0000-0000-000000000000

Directory (tenant) ID

00000000-0000-0000-0000-000000000000

Supported account types

[My organization only](#)

Client credentials

[0 certificate, 1 secret](#)

Redirect URIs

[2 web, 0 spa, 0 public client](#)

Application ID URI

[Add an Application ID URI](#)

Managed application in local directory

[CloudBeaver](#)

[Get Started](#)

[Documentation](#)

3. Set **Secret Key**

Copy the value of the secret key [created in this step point 2](#) into the **Secret Key** field.

4. Save the Identity Provider configuration

Configure the Redirect link

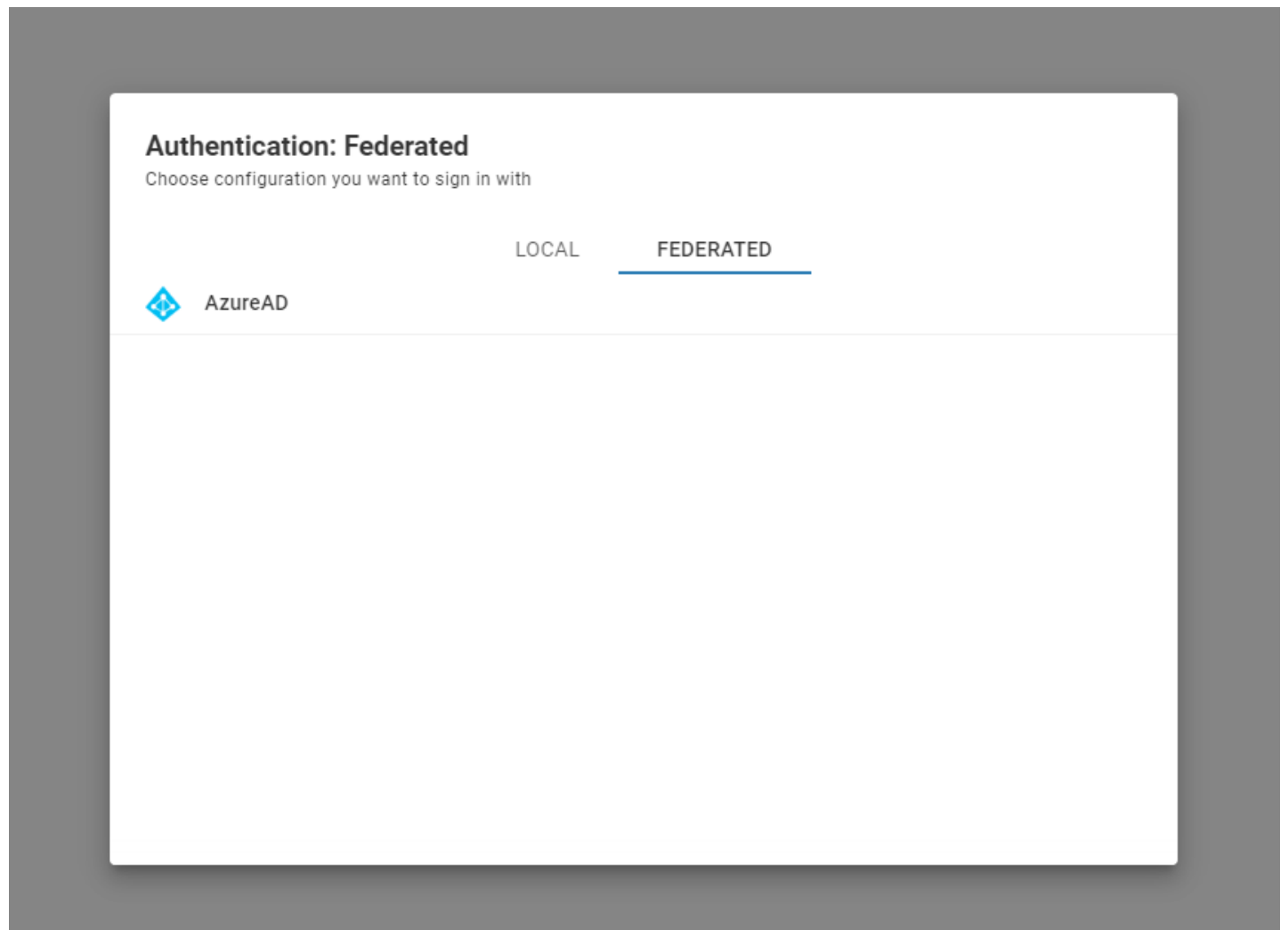
1. Open **Azure AD provider configuration** in Cloudbeaver and copy the **Redirect** link

The screenshot shows the CloudBeaver interface with the 'Identity Providers' section selected in the left sidebar. The main panel displays the 'Azure AD' configuration. The 'Provider' is set to 'Azure AD'. The 'ID' is 'AzureAD'. The 'Configuration name' is 'AzureAD'. The 'Description' is empty. The 'Icon URL' is empty. The 'Domain / Tenant ID' is empty. The 'Application (client) ID' is empty. The 'Secret Key' is empty. The 'Links' section shows 'Sign in' and 'Sign out' links. The 'Redirect' link is highlighted with a red box and contains the URL 'https://cloudbeaver.io/AzureAD/ca...'. The 'Cancel' and 'Save' buttons are visible at the top right of the configuration panel.

2. Add a redirect link to the Azure AD application (select **Web** as platform) - [official Microsoft documentation](#)

Testing the Azure AD authentication

The new Federated tab becomes available after creating the configuration in the CloudBeaver authentication dialog. The user can select the configuration and thereafter login to the application using SSO.



Configure SQL Server databases access

You can use Azure AD authentication to gain access to SQL Server deployed in Azure Cloud.

1. On your Enterprise Application page, click API Permissions tab and add permission `user_impersonation` in `Azure SQL Database` API

2. In your SQL Server you need to map Azure AD users into database users. See Microsoft documentation:

[https://docs.microsoft.com/en-us/azure/azure-sql/database/authentication-aad-configure?
view=azuresql&tabs=azure-powershell#create-contained-users-mapped-to-azure-ad-identities](https://docs.microsoft.com/en-us/azure/azure-sql/database/authentication-aad-configure?view=azuresql&tabs=azure-powershell#create-contained-users-mapped-to-azure-ad-identities)

Google authentication

Table of contents

[Google configuration](#)

[Enabling Google authentication](#)

[Configuring identity provider](#)

[Configuring Google OAuth 2.0 application](#)

[Testing Google authentication](#)

[GCP configuration](#)

[Enable GCP Services](#)

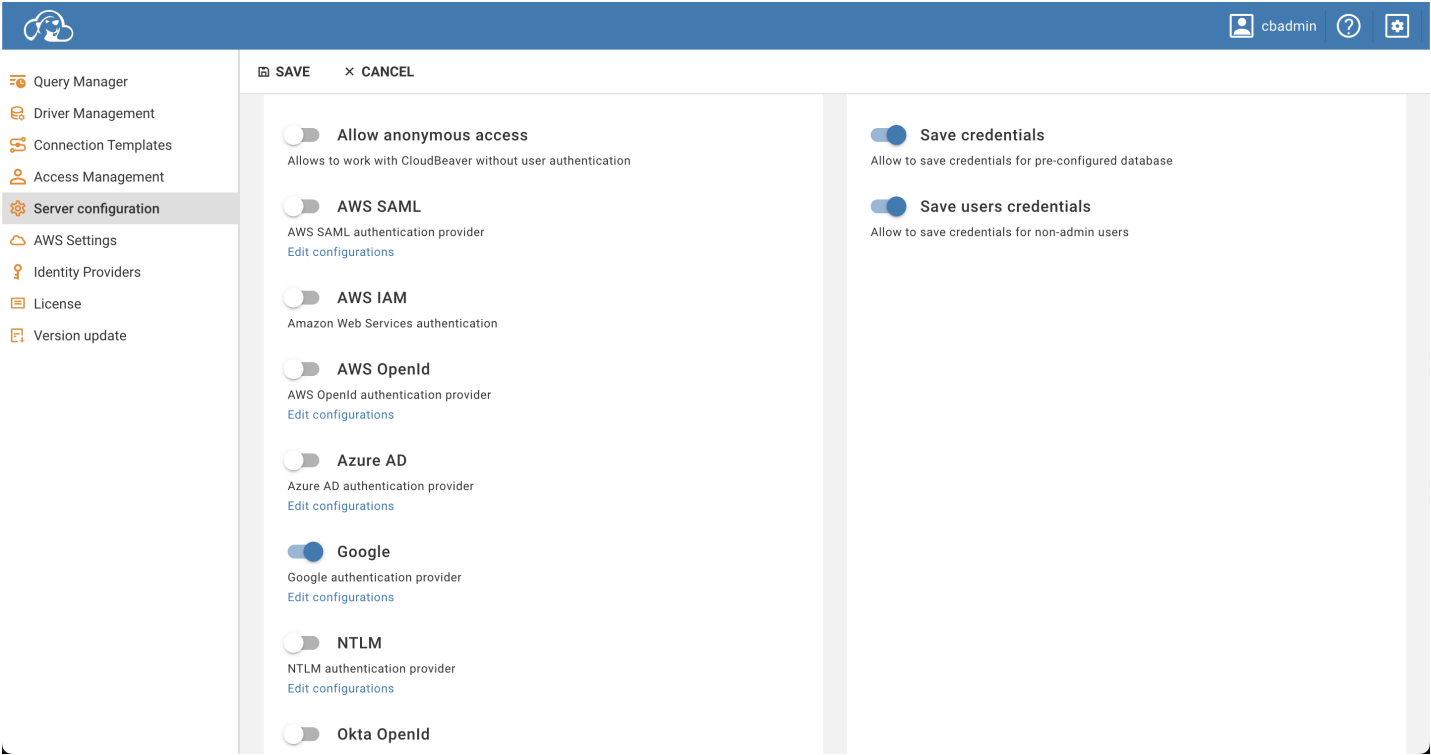
[Add custom scopes to the Google Identity Provider](#)

[Testing](#)

Google configuration

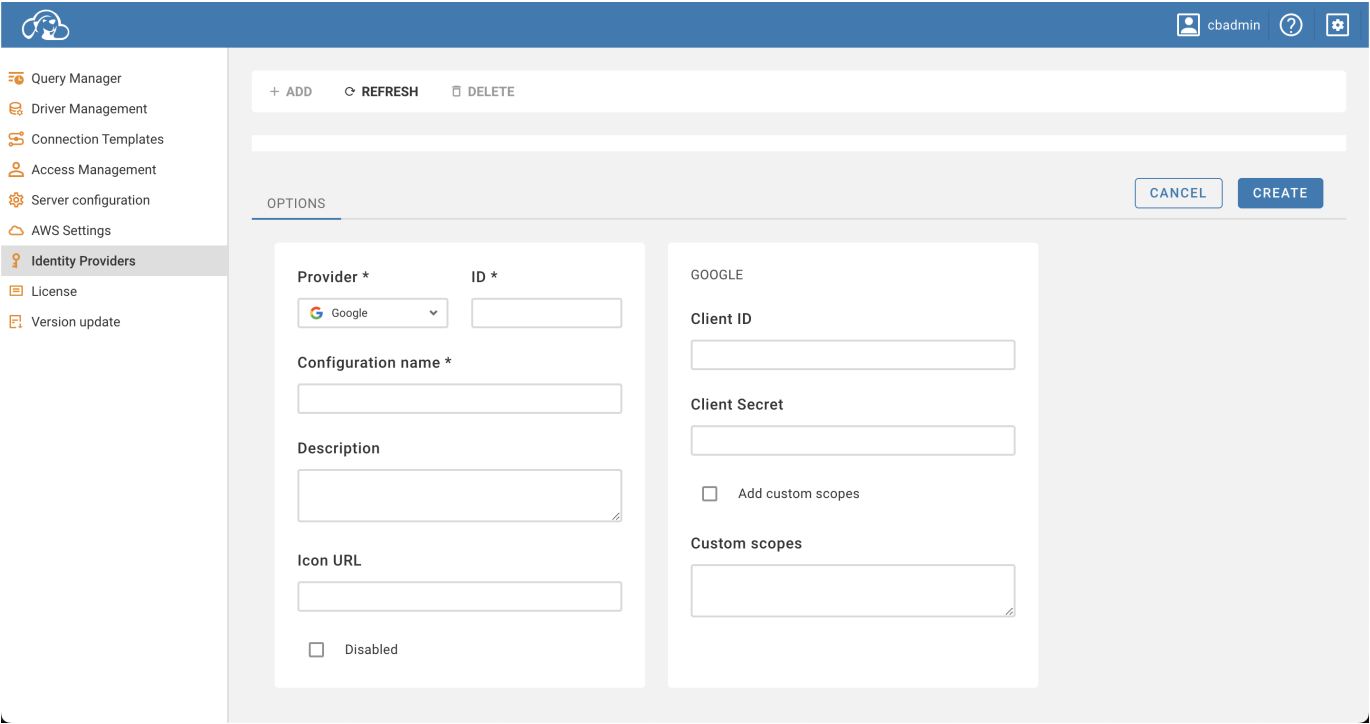
Enabling Google authentication

Go to the Administration menu and enable **Google** in the Server configuration tab.



Configuring identity provider

1. Go to the Identity Providers tab and create a new configuration using the OpenID details.



Client ID - Client ID from your Google OAuth 2.0 application

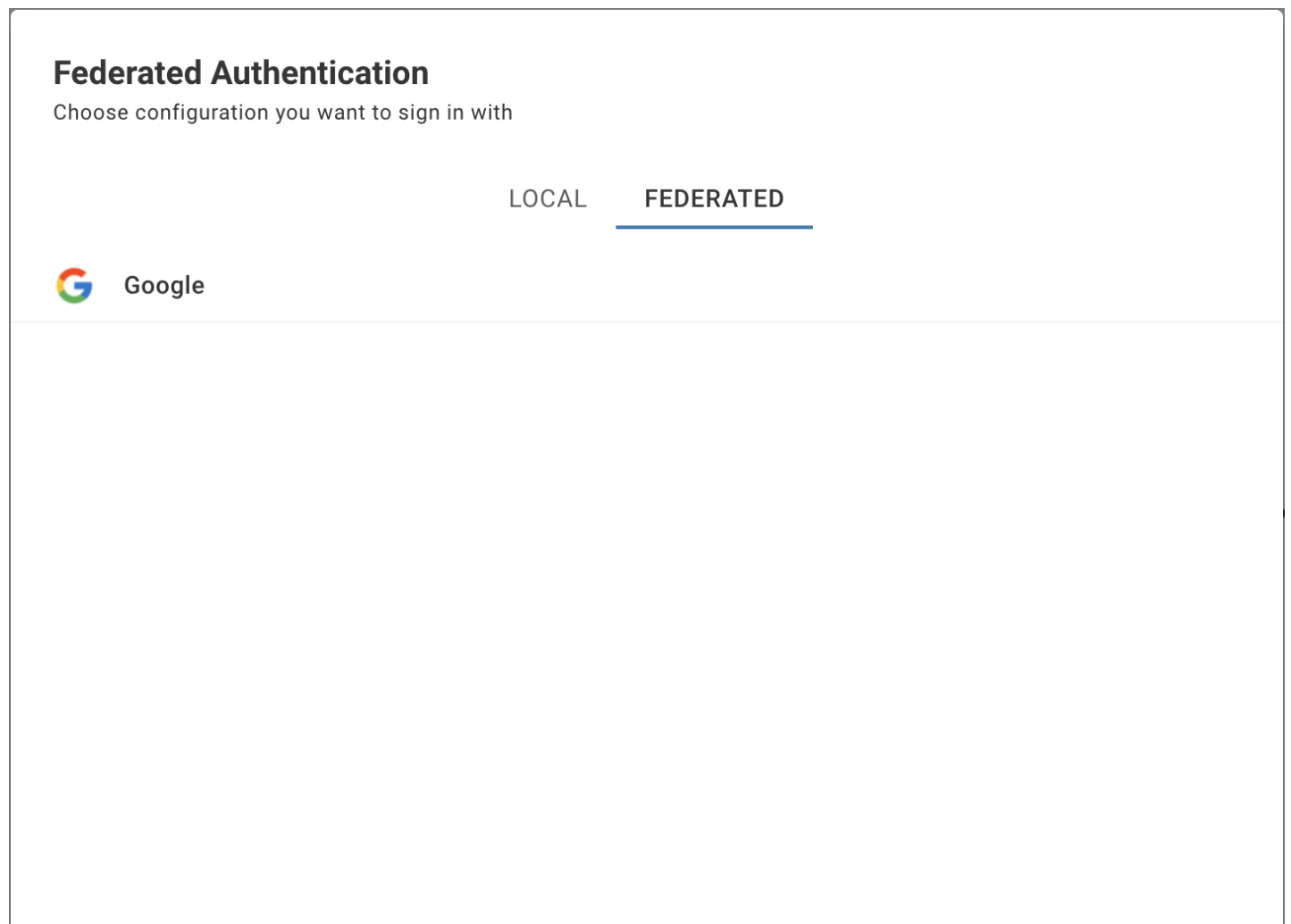
Client Secret - Client Secret from your Google OAuth 2.0 application

Configuring Google OAuth 2.0 application

Open identity provider in CloudBeaver, copy `Redirect url` , and [set up redirect in Google application](#)

Testing Google authentication

The new Federated tab becomes available after creating the configuration in the CloudBeaver authentication dialog. The user can select the configuration and thereafter login into the application using Google SSO.



GCP configuration

In the CloudBeaver, it's possible to view and work with databases hosted in GCP, without the need to manually configure each connection to the database. To do this, you need to configure the integration of Google Identity provider with GCP.

Enable GCP Services

Go to the Administration menu and enable **GCP** services in the Server configuration tab.

Add custom scopes to the Google Identity Provider

- Go to the Identity Providers tab and open your existing Google provider.
- Enable **Add custom scopes** checkbox - this will allow you to specify additional scopes for your identity provider, and expand its capabilities
- Specify the following scopes:


<https://www.googleapis.com/auth/spanner.admin>; <https://www.googleapis.com/auth/bigquery>; <https://www.googleapis.com/auth/cloud-platform>

Scope	Description
https://www.googleapis.com/auth/spanner.admin	Allow to administrate Spanner databases
https://www.googleapis.com/auth/bigquery	Allow to view and manage your data in Google BigQuery
https://www.googleapis.com/auth/cloud-platform	Gives access to the GCP and is



needed to read a list of databases


https://www.googleapis.com/auth/devstorage.full_control


Allow to manage BigQuery data





cbadmin





 Query Manager


 Driver Management


 Connection Templates


 Access Management

 Server configuration

 AWS Settings

 Identity Providers




 License

 Version update

+ ADD

REFRESH

DELETE

	CONFIGURATION NAME	PROVIDER	DESCRIPTION	DISABLED	
<input type="checkbox"/>	<div><div>Provider *</div><div><div>Google</div></div><div>ID *</div><div>google-openid</div></div> <div><div>Configuration name *</div><div>Google</div></div> <div><div>Description</div><div></div></div> <div><div>Icon URL</div><div></div></div> <div><input type="checkbox"/> Disabled</div>	GOOGLE	<div>Client ID</div> <div>*****</div> <div>Client Secret</div> <div>*****</div> <div><input checked="" type="checkbox"/> Add custom scopes</div> <div>Custom scopes</div> <div>https://www.googleapis.com/auth/spanner.admin:h https://www.googleapis.com/auth/bigquery;https:// www.googleapis.com/auth/cloud- platform;https://www.googleapis.com/auth/devstor age.full_control</div>	<div>LINKS</div> <div>Sign in</div> <div>https://cloudbeaver_url/api/google-openid/go... </div> <div>Sign out</div> <div>https://cloudbeaver_url/api/google-openid/go... </div> <div>Redirect</div> <div>https://cloudbeaver_url/api/google-openid/go... </div>	

Testing

After setting up the provider, you need to re-login. Now you can open **Cloud connections** in the connection creation menu, if everything is configured correctly - you will see your project and databases in it

Cloud Connection

Search elements...

⌕ ⚙ ↺

☐ ▾ 📁 DBeaverGCP

☐ > 🗄 SQL

☐ > 🔗 Spanner

☐ > 🔍 BigQuery

CANCEL

CREATE

User credentials storage

Table of contents

[Overview](#)

[Credentials storage](#)

Overview

It is possible to configure CloudBeaver to save database credentials (user names and passwords) in CloudBeaver storage.

In this case, users won't need to enter database credentials every time they connect to a database.

However, the most secure way is to disable this option. See options "Save credentials" and "Save user credentials" in administrator console, page "Server configuration".

Credentials storage

There are two types of database connections: global and user.

Global connections are managed by CloudBeaver administrators, user connections are managed by users themselves.

Global database configuration is stored in workspace sub-folder `GlobalConfiguration` / `.dbeaver`.

Database configurations are stored in the file `data-sources.json`, database credentials are stored in the file `credentials-config.json`. File `credentials-config.json` is encrypted by a special key which is stored in CloudBeaver distribution.

User configuration are stored in workspace sub-folders `user-projects` / `USER_NAME` / `.dbeaver`.

Potentially, if an intruder/malware software will get access to CloudBeaver server filesystem, then it may get access to all stored user credentials. To increase security it is recommended to configure the server to keep workspace on a shared encrypted network folder (e.g. S3, see [S3 Server-side encryption](#)).

- [How to configure S3 encrypted file system on Ubuntu](#)

Cloud Explorer

Table of contents

- [Overview](#)
- [Supported databases](#)
- [Cloud configuration](#)
- [Identity provider configuration](#)
- [Cloud Explorer](#)
- [Database cloud information](#)

Note: This feature is available in [Enterprise](#), [AWS](#) and [Team](#) editions only.

Overview

Cloud Explorer offers deep integration with popular cloud service providers. As of the latest version, it supports Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Entra ID (Azure).

With **Cloud Explorer**, you can set up your cloud access once and then browse, connect, and manage your cloud databases. It saves you from manually configuring each database connection, as it reads all database endpoint information directly from the cloud provider.

Authentication is centralized. You can use your cloud account to access your cloud databases. Before you start using **Cloud Explorer**, you need to [set up your identity provider](#) access. This setup includes access credentials, availability zones for database search, and other cloud-specific settings.

Supported databases

The **Cloud Explorer** allows you to easily connect with and manage various databases. Here is the list of databases it supports:

Providers	Databases
Amazon Web Services	PostgreSQL

	MySQL
	Oracle
	Amazon Redshift
	Amazon Athena
	Amazon DocumentDB
	Amazon DynamoDB
	Amazon Keyspaces
	Amazon ElastiCache (Redis)
	Amazon Timestream
	Amazon Neptune
Google Cloud Platform	AlloyDB for PostgreSQL
	Microsoft SQL Server
	PostgreSQL
	MySQL
	Spanner
	Firestore
	BigQuery
	Bigtable
Microsoft Entra ID (Azure)	Microsoft SQL Server
	PostgreSQL
	MySQL
	CosmosDB (MongoDB)
	CosmosDB (Cassandra)

Cloud configuration

To begin the configuration process, access the Administration panel by navigating to **Settings** -> **Administration** -> **Server Configuration**. In the **Configuration** section, enable the checkbox for the appropriate cloud service provider (**AWS**, **GCP**, **Microsoft Entra ID (Azure)**) that you wish to configure.

Note: In the CloudBeaver AWS Edition, the **AWS** option is enabled by default. Additionally, the **GCP** and **Microsoft Entra ID (Azure)** configurations are not available in this edition.

Identity provider configuration

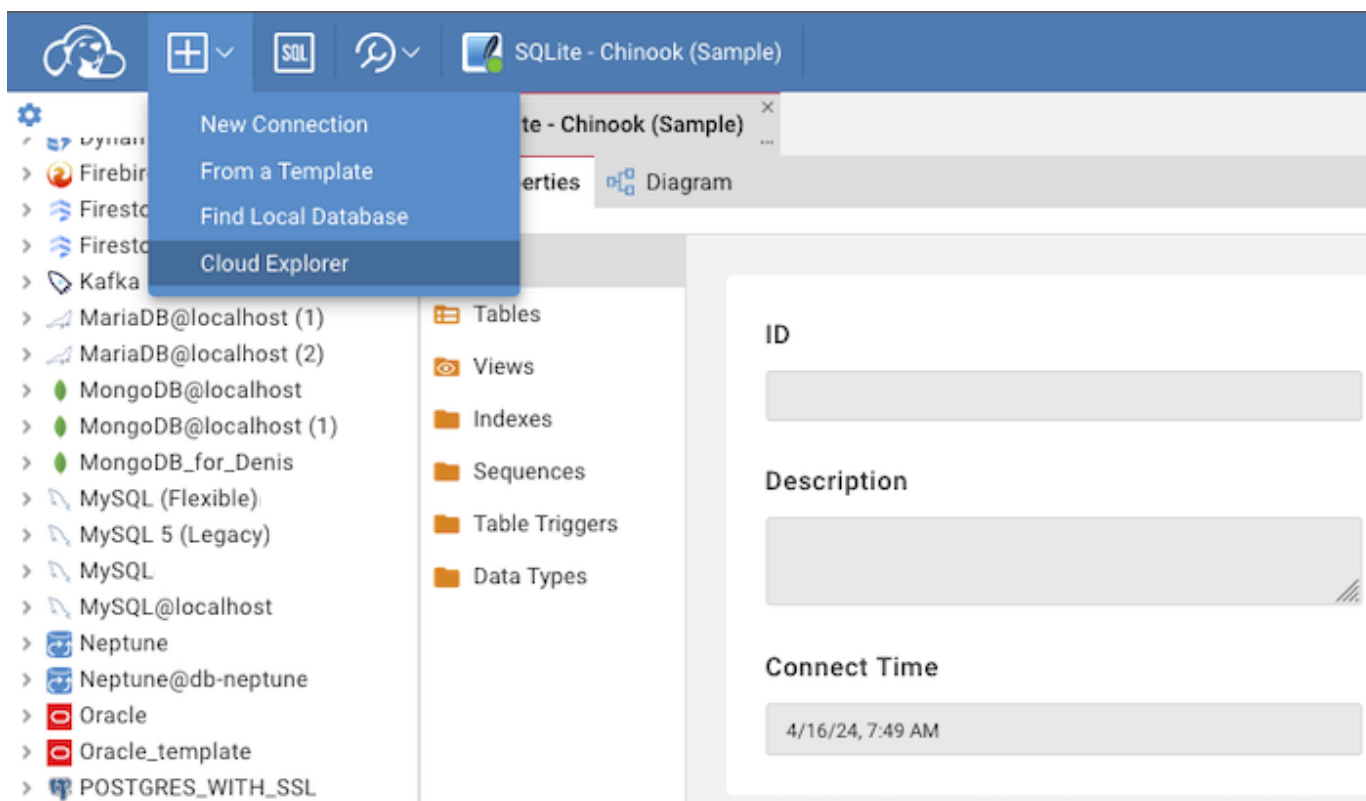
Each cloud service provider requires specific settings for the identity provider. Below is a table outlining the basic guides on configuring these settings.

Provider	Additional article
AWS	You can authenticate in AWS using several methods. Learn more .
GCP	Google setup
Azure	Microsoft Entra ID (Azure) setup

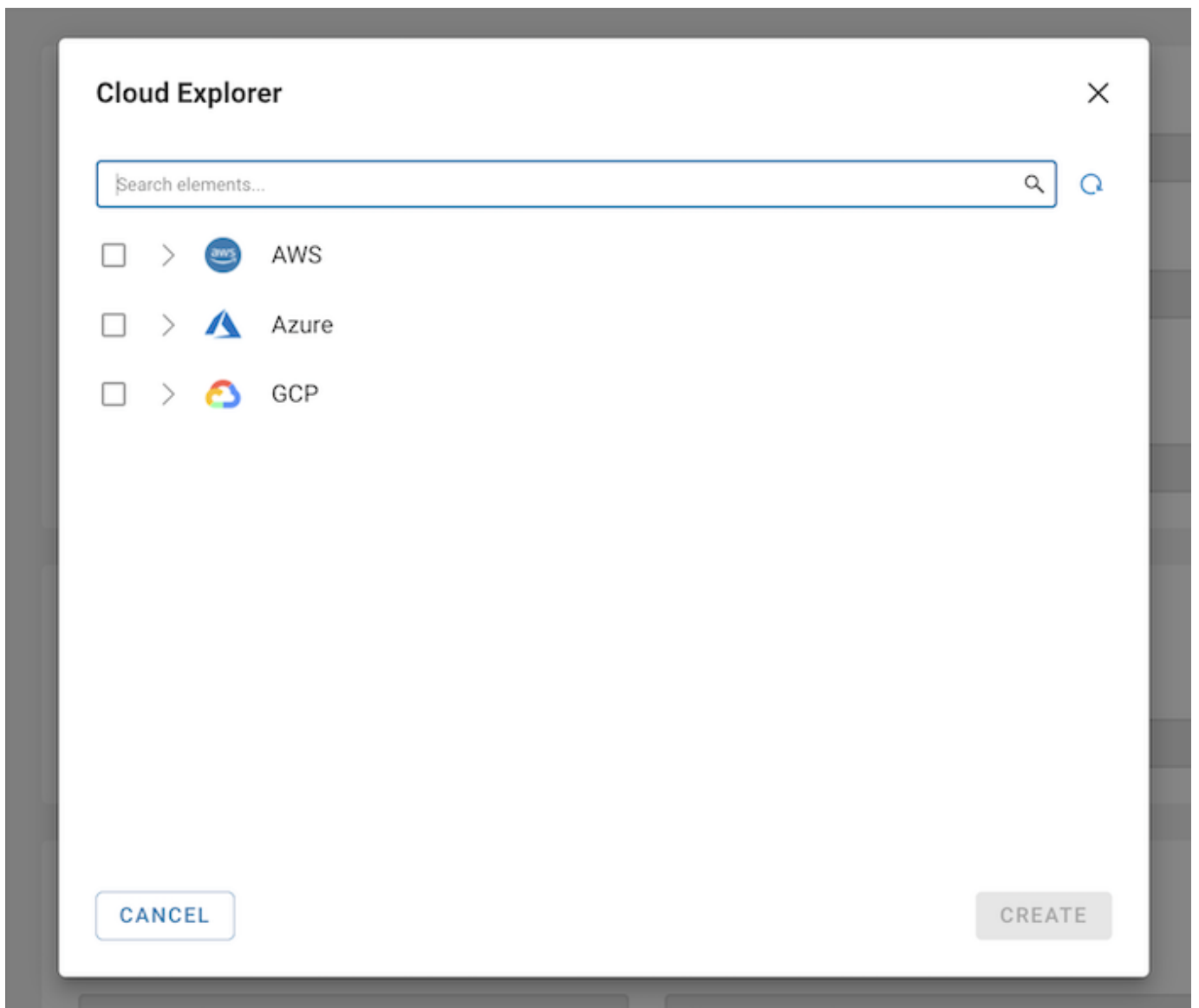
Cloud Explorer

Once you have configured your identity provider, open the **Cloud Explorer** window to add database connections to your [Database Navigator](#).


1. Navigate to **Connection** -> **Cloud Explorer**.



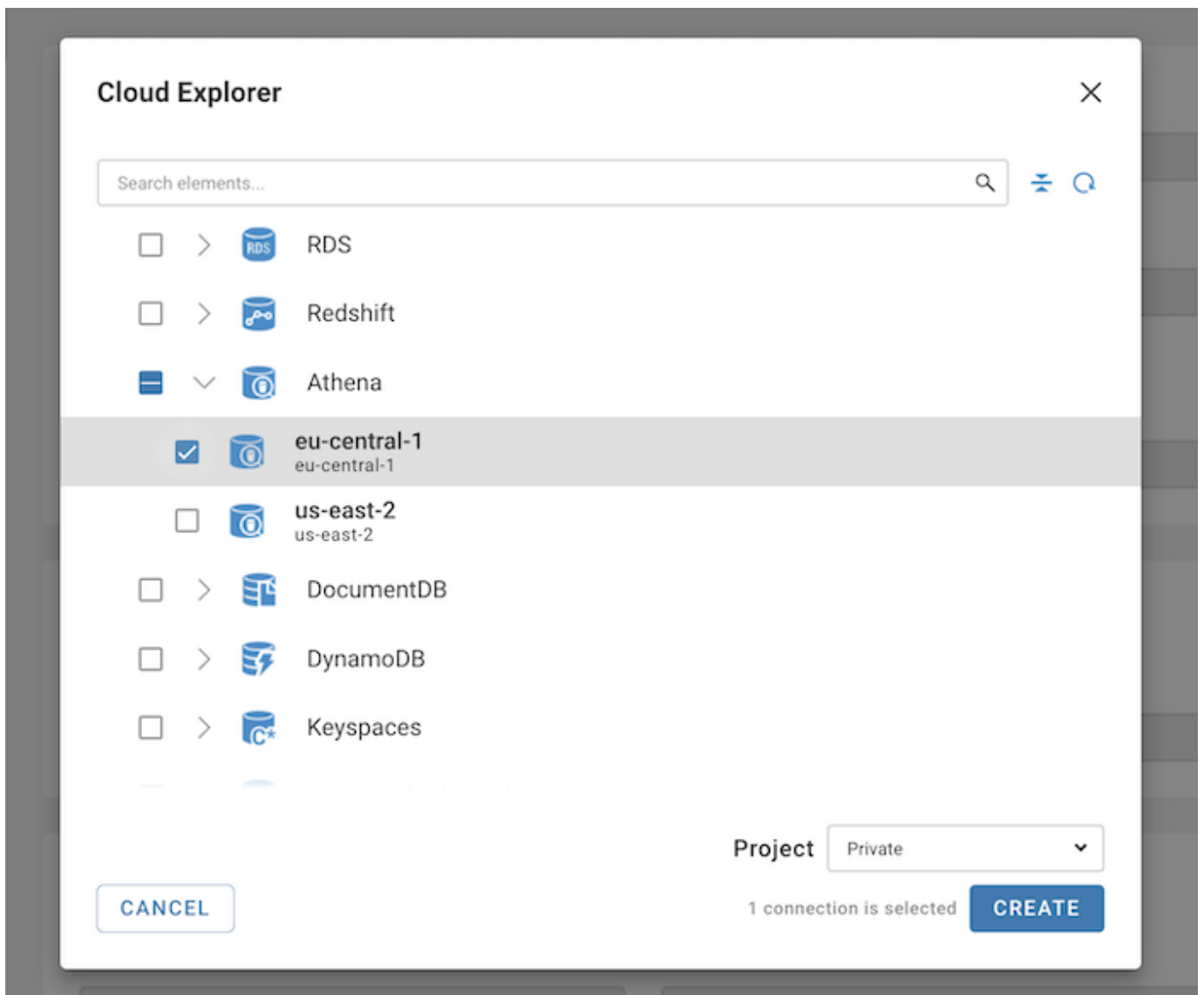
2. In the center of the dialog, you will see cloud databases displayed in a hierarchical view. All databases are grouped by database/service type. When you expand one of the top elements, DBeaver will search for cloud databases in the configured availability zones/regions.



3. Select the cloud service provider. Depending on the type of authentication configured, you may need to log in using the appropriate identity provider in the opened window.

ID	Name
<div><div> Additional Authentication ×</div><div>AWS IAM AWS IAM</div><div><div>AWS IAMFEDERATED</div><div>Static access keys ▾</div><div>Access Key</div><div>EXAMPLE_KEY</div><div>Secret Key</div><div>.....</div><div>LOGIN</div></div></div>	

4. After authentication, select the database you want to add.



Tip: If you have a large number of databases in your cloud, you can search for them using the search bar located above the Cloud Explorer.

5. Click on the **Create** button located in the bottom right corner of the **Cloud Explorer** dialog. Your database will then appear in the Database Navigator.

Database cloud information

You can access your cloud database configuration directly from the Database Navigator. To do this, click on the menu button next to the database you are interested in and select **Edit Connection**. This action opens a special tab in the connection settings dialog. The information displayed in this tab is specific to the cloud and database type of the selected database.

Learn more

Table of contents

[Overview](#)

[Configure the Cloud storage](#)

[Work with SQL scripts](#)

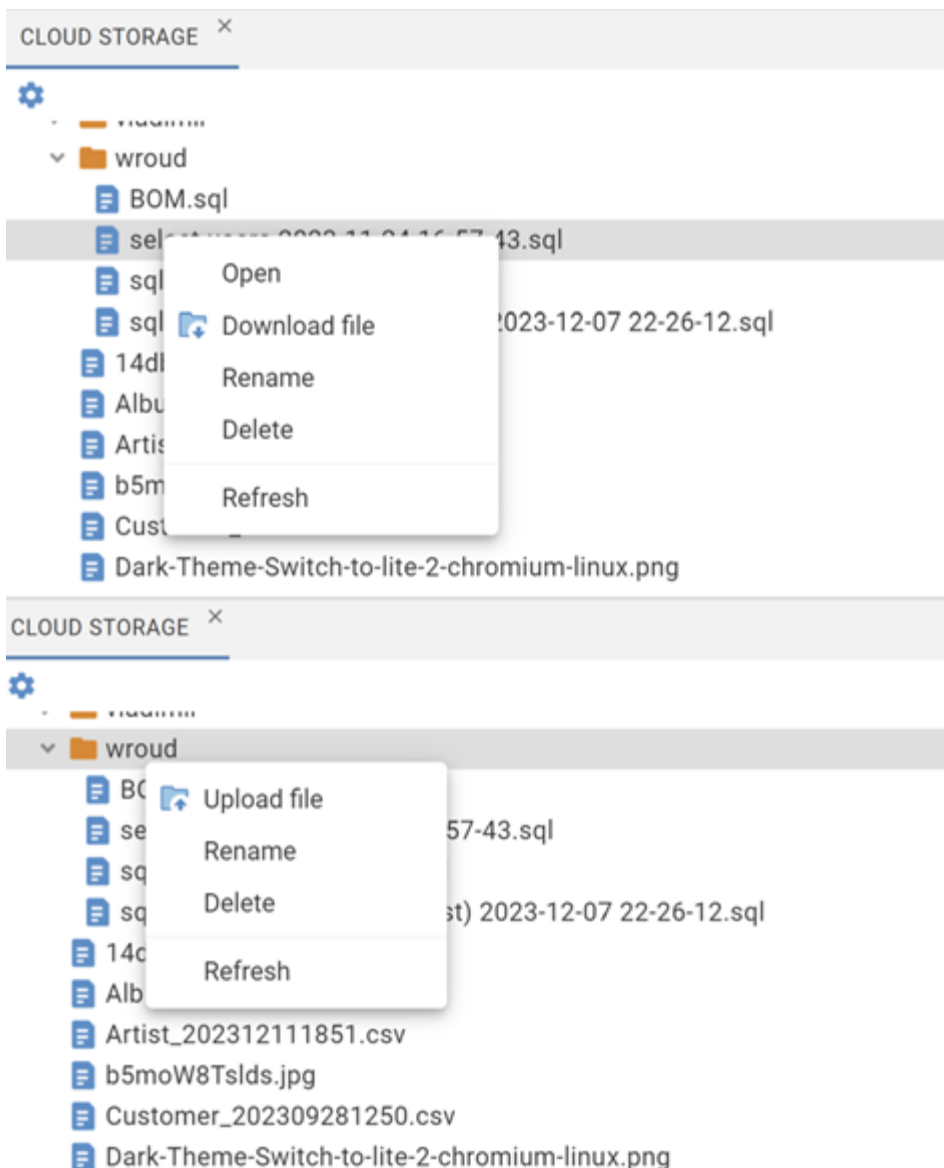
Note: This feature is available in [Enterprise](#) and [Team Edition](#) editions only.

Overview

CloudBeaver provides the ability to use your cloud storage services through a web interface. Currently, we support Amazon Simple Storage Service and Google Cloud Storage.

Within the file browser, you can interact with it like a regular file system, allowing you to:

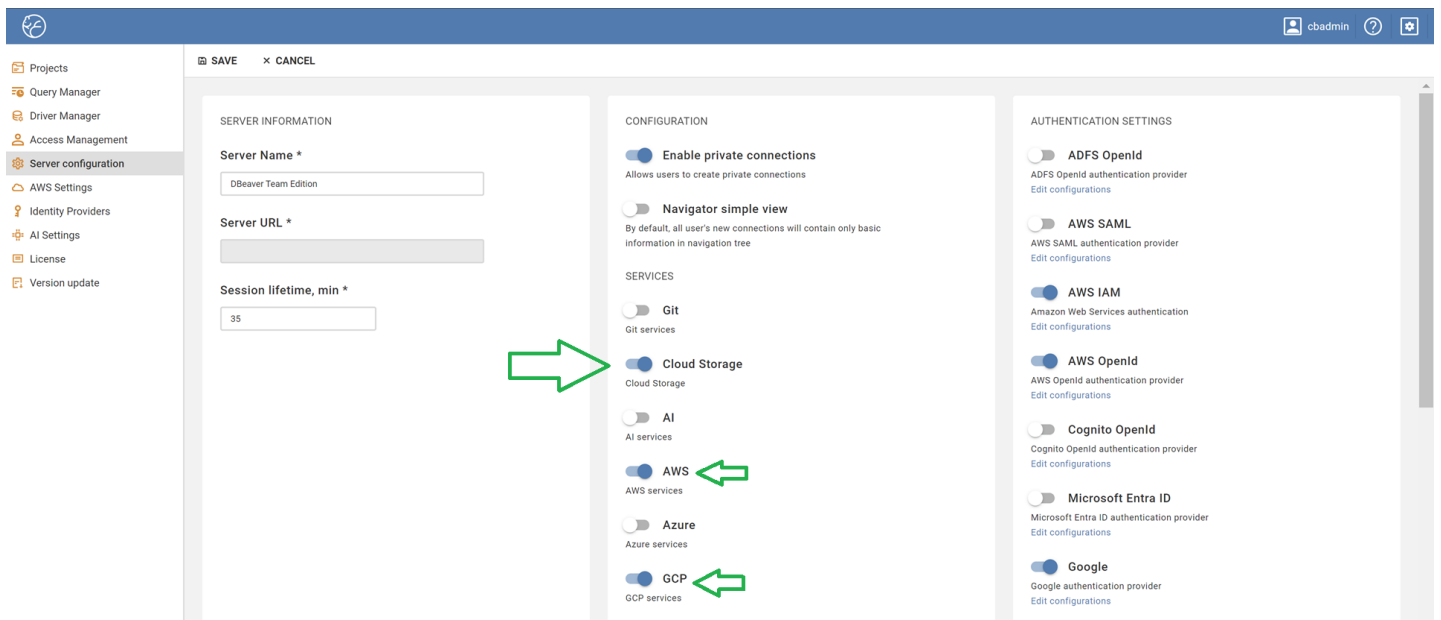
- upload new files to the bucket from your local storage;
- download files to your local storage;
- delete and rename files;
- drag-and-drop files between buckets and different file systems.



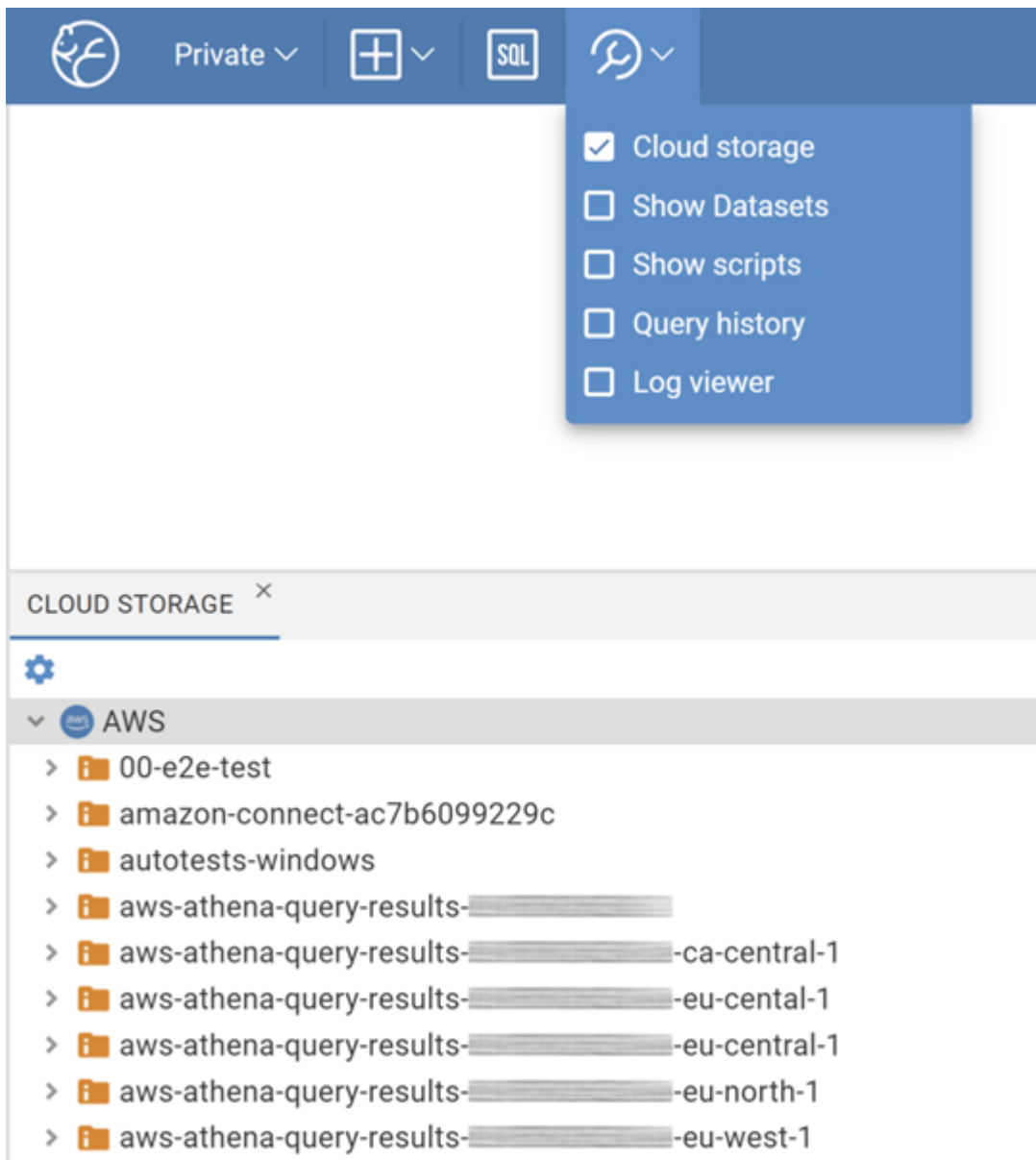
Configure the Cloud storage

To gain access to your storage, you should create a cloud configuration - [AWS](#) or [GCP](#).

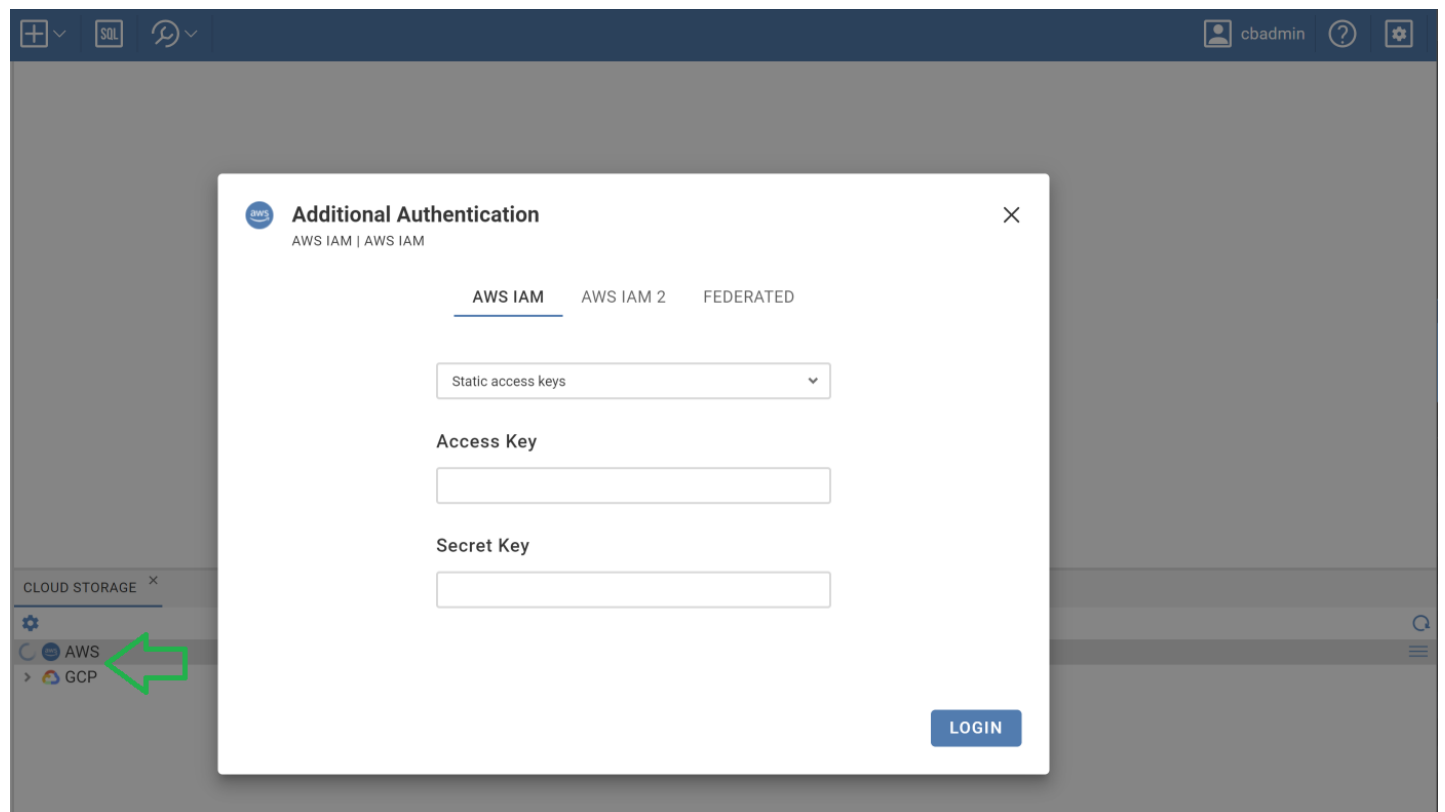
Then enable 'Cloud storage' in the Server configuration. Ensure that the relevant service (AWS, GCP) is also enabled.



Navigate to the Tools menu on the main page of the app. Check the 'Cloud storage' option.



And finally, to view your buckets, expand the file system tree. If you are not logged in under the required provider, a login window will appear. Use the preconfigured authentication.



Work with SQL scripts

The file type is recognized automatically. Only SQL scripts (.sql) could be opened in the app. Double-click on the file to open it in SQL editor. You can easily execute statements or make changes and save them directly to the cloud storage.

select users

1

```
select * from users u where age > 18;
```

CLOUD STORAGE

wroud

BOM.sql

select users 2023-11-24 16-57-43.sql

sql-1 (share) 2023-10-10.sql

sql-2 (PostgreSQL@localhost) 2023-12-07 22-26-12.sql

14dbeaver.pdf

Album_202312111259.csv

Artist_202312111851.csv

b5moW8Tslds.jpg

Customer_202309281250.csv

CloudBeaver User Guide 24.1.ea. Page 125 of 365.

Query Manager

Table of contents

[Overview](#)

[Query Manager Options](#)

[Query types](#)

[Object types](#)

[Filter by date](#)

[Filter by users](#)

[Filter by drivers](#)

[Filter by Projects](#)

[Sorting and Settings](#)

[Auto Refresh](#)

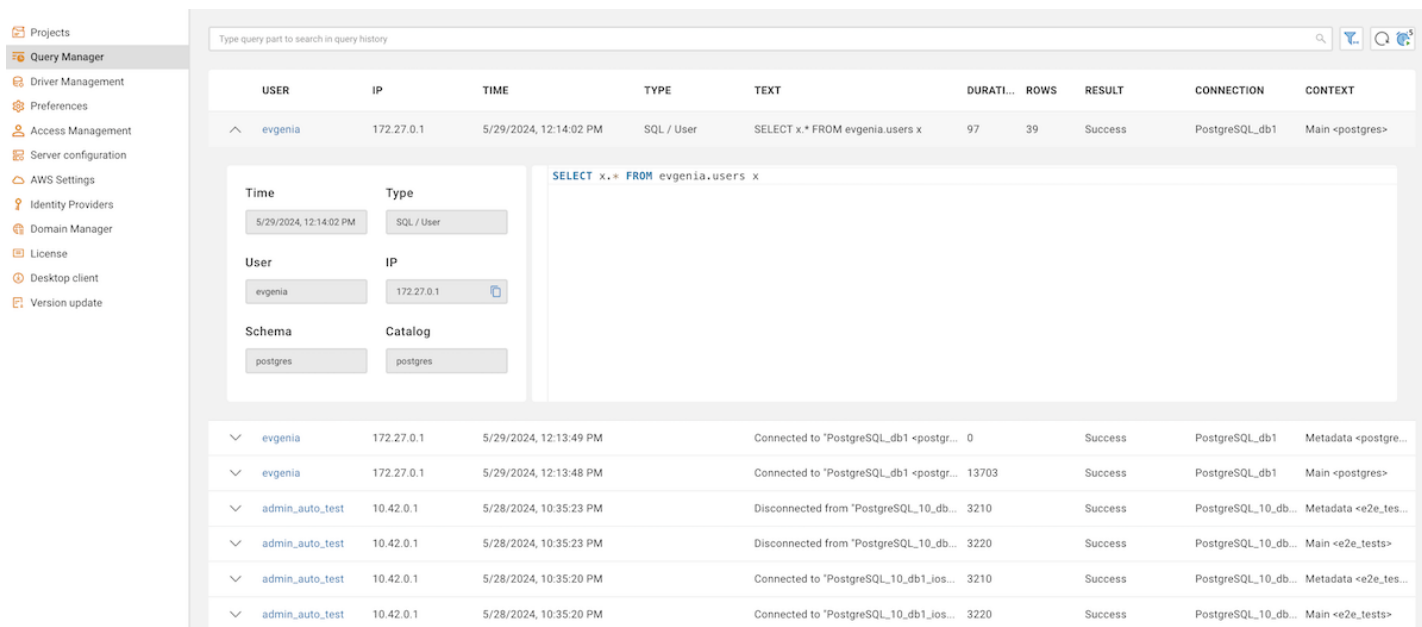
[Restore defaults](#)

Note: This feature is available in [Enterprise](#) and [Team Edition](#) editions only.

Overview

The Query Manager in CloudBeaver is an administrative tool that allows only administrators to monitor and manage all queries executed within the system. It provides comprehensive details on the execution statistics, such as duration, number of rows affected, and the outcome of each query.

To access the **Query Manager**, navigate to **Settings -> Administration** and select the **Query Manager** tab.





The following table describes each field in the Query Manager:


Field	Description
User	Displays the name of the person who executed the query.
IP	Indicates the IP address from where the query was initiated.
Time	Shows when the query was executed.
Type	Identifies whether the query was initiated by a user or generated by the system.
Text	Contains the SQL code that was executed.
Duration	Measures how long the query took to complete, in milliseconds.
Rows	Counts how many rows were affected or returned by the query.
Result	Describes the outcome of the query, such as "Success" or any errors that occurred.
Connection	Specifies which database connection was used to execute the query.
Context	Provides additional information about the environment or session in which the query was executed.

Tip: In the Team Edition, supervisors also have the ability to use the Query Manager to monitor their team's queries. For more detailed information on the supervisory functions within the Query Manager, refer to the article on [Teams in Team Edition](#).

The Query Manager offers several features to enhance usability:

- **Search Functionality:** Use the **Search** function to locate specific queries by entering SQL text in the Search field above the table.
- **Refresh Query Manager:** Manually update the data displayed in the Query Manager by clicking the **Refresh** button  located in the toolbar.
- **Auto Refresh:** The Query Manager can automatically update the query list. You can disable this feature by toggling the **Auto-Refresh** button  in the toolbar. This setting is adjustable, for information on how to customize it, see the section on [Auto Refresh](#).

Query Manager Options

The Query Manager offers a range of customizable settings that allow administrators to tailor the view and behavior according to their specific needs. To access these settings, click the **Query Manager Options** button .

Here is a detailed breakdown of each configurable option available through this feature:

Query types

You can filter which types of queries are displayed in the Query Manager by selecting or deselecting the following options:

Option	Description
User queries	Show all queries directly executed by users.
Filtered user queries	Show user queries that meet specific criteria.
User scripts	Show batches of queries executed as scripts.
Utility functions	Show system-level utility function calls.
Metadata read	Show queries that read database metadata.
Metadata write (DDL)	Show queries that modify database structure.
Query status	Allows to filter queries based on their execution status. The available options are:
	All: Displays all queries regardless of their execution outcome.
	Fail: Shows only the queries that have failed.

	Success: Displays only the queries that have executed successfully.
--	--

Object types

Control visibility of different system object types involved in the queries:

Option	Description
Sessions	Include queries related to user sessions.
Transactions	Include queries that are part of transactions.
Queries	Include individual query executions.

Filter by date

Specify the date range to view queries:

- **From:** Start date and time for the filter.
- **To:** End date and time for the filter.

Filter by users

You can filter the displayed queries by specific users. Use the search field to find and select the desired users to tailor the display of queries according to your needs.

Filter by drivers

This feature enables you to filter queries based on the database drivers. Search for and select the drivers that you are interested in to narrow down the results of displayed queries.

Filter by Projects

Note: This feature is available in the [Team Edition](#) only.

You can filter the displayed queries by projects. This option allows you to search for and select the projects that are relevant to the queries you are interested in.

Sorting and Settings

Adjust how query results are sorted and displayed:

Option	Description
Sorting by	Choose the attribute to sort the queries by (available options: User , Date , Driver , Query text).
Desc	Check this to sort in descending order.
Row Count	Set the number of queries to display per page.

Auto Refresh

Configure automatic refresh of query information:

Option	Description
Enabled	Check this to enable auto refresh every few seconds.
Interval (seconds)	Set how often the Query Manager updates.
Stop on error	Check this to halt auto refresh if an error occurs.

Restore defaults

To restore the default settings, press the **Restore Defaults** button at the bottom of the Query Manager Options window.

Learn more

Table of contents

[Drivers Management](#)

[Creating a Custom Driver](#)

[Uploading Binary Files](#)

[Editing an Existing Driver](#)

[Filtering Drivers](#)

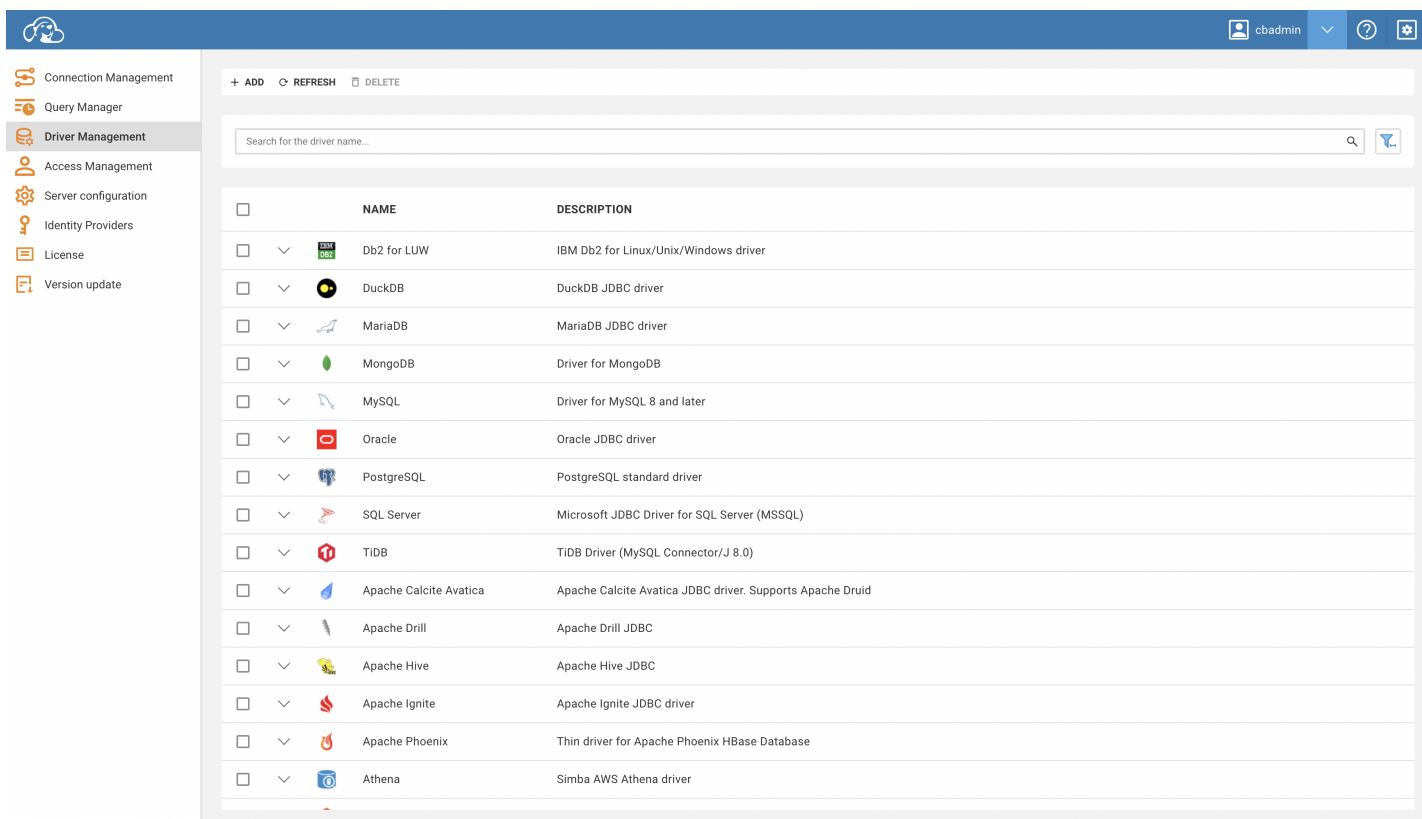
Note: This feature is available in [Enterprise](#) and [Team Edition](#) only.

Drivers Management

This wiki guide provides a step-by-step instruction for managing drivers using the product interface, including creating a new custom driver and editing an existing one.

Before you start managing your drivers, it is important to note that administrative privileges are required. Only users with administrator rights can perform actions such as creating, editing, or deleting drivers.

Creating a Custom Driver

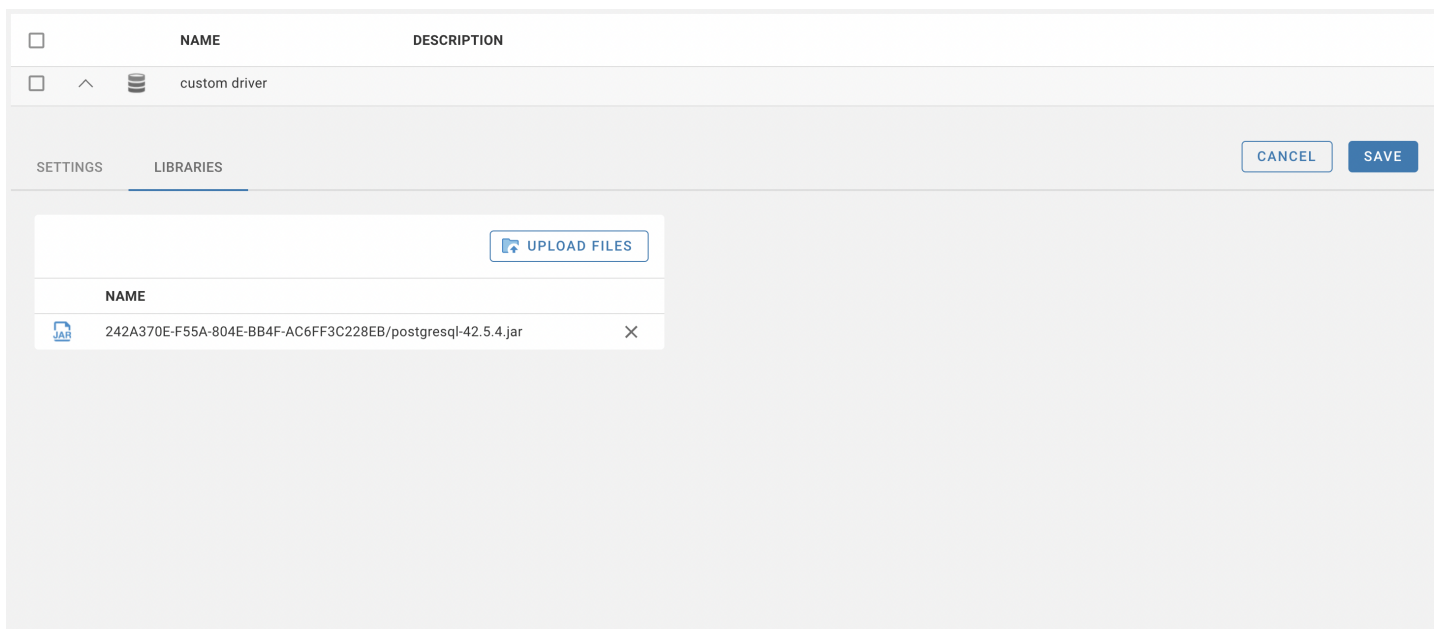


Follow the steps below to create a new custom driver:

1. Navigate to the **Administration** page of the interface.
2. Select the **Driver Management** tab.
3. Click on the **Add** button.
4. A form will appear in which you will have to fill in the necessary fields with the appropriate information.
5. Once completed, click **Create**.

Your new custom driver should now be successfully created.

Uploading Binary Files



In order to make your new driver functional, you will need to upload the `.jar` binary files. This is how you can do it:

1. Locate your newly created driver on the `Driver Management` table.
2. You can filter for your custom drivers by checking the relevant checkbox in the table filters.
3. Click on your custom driver to open it.
4. Navigate to the `Libraries` tab.
5. Click `Upload Files`, then select your `.jar` files.
6. Confirm that your files have appeared in the table.
7. Click `Save`.

Once the binary files have been uploaded, your custom driver will be ready to use. Whenever you need to access it, it can be selected from the driver list.

Editing an Existing Driver

	NAME	DESCRIPTION
<input type="checkbox"/>	Oracle	Oracle JDBC driver

SETTINGS

LIBRARIES

Driver Name *

Driver Type *

Oracle

Class Name *

URL Template

Description

DEFAULTS

Default Port

Default database

CANCEL

SAVE

You can also edit any existing driver. By following these the steps:

1. Open the desired driver from the **Driver Management** table.
2. Modify the fields as required.
3. Click **Save** to preserve your changes.

Your driver is now updated and can be continued to be used with the modified settings.



Please note: Ensure you have the appropriate permissions and are aware of the potential impact before proceeding with these modifications.

Filtering Drivers

Drivers have 3 available filters:

- Search
- Custom/default driver
- Driver state (enabled/disabled/all)

In order to see all filters, you must click a filter button:

cbadmin ? 

Query Manager

Driver Management

Connection Templates

Access Management

Server configuration

AWS Settings

Identity Providers

License

Version update

+ ADD






REFRESH

DELETE

Search for the driver name...

Enabled

Custom drivers

		NAME	DESCRIPTION	ENABLED
<input type="checkbox"/>				
<input type="checkbox"/>	▼	 Db2 for LUW	IBM Db2 for Linux/Unix/Windows driver	<input checked="" type="checkbox"/>
<input type="checkbox"/>	▼	 MariaDB	MariaDB JDBC driver	<input checked="" type="checkbox"/>
<input type="checkbox"/>	▼	 MongoDB	Driver for MongoDB	<input checked="" type="checkbox"/>
<input type="checkbox"/>	▼	 MySQL	Driver for MySQL 8 and later	<input checked="" type="checkbox"/>
<input type="checkbox"/>	▼	 ODBC	Modern JDBC-ODBC bridge	<input checked="" type="checkbox"/>

Drivers can be disabled in the **Server configuration** tab

Supported databases

Table of contents

[Supported databases:](#)

Supported databases:

- Apache Kyuubi
- ClickHouse
- Db2 iSeries/AS 400 for IBM i
- Db2 LUW
- DuckDB
- Firebird
- H2 Embedded
- MariaDB
- MySQL
- Oracle
- PostgreSQL
- SQL Server
- SQLite
- Trino

CloudBeaver PRO products

Include all databases from the list above and the following databases:

- Altibase ★
- Apache Arrow ★
- Apache Calcite Avatica ★
- Apache Drill ★
- Apache Hive ★
- Apache Ignite ★
- Athena ★
- Azure Databricks ★
- Azure SQL Server ★
- Babelfish ★
- Cache ★
- Cassandra ★
- CloudSQL - MySQL ★
- CloudSQL - PostgreSQL ★
- CloudSQL - SQL Server ★
- CockroachDB ★
- CosmosDB (Cassandra) ★
- CosmosDB (MongoDB) ★
- Couchbase ★
- Couchbase 5+ ★
- CouchDB ★

- CrateDB ★
- CUBRID ★
- Dameng ★
- Db2 for z/OS ★
- DocumentDB ★
- Dremio ★
- DynamoDB ★
- EDB ★
- Elasticsearch ★
- Exasol ★
- Firestore ★
- Fujitsu ★
- Google AlloyDB ★
- Google BigQuery ★
- Google Cloud Bigtable ★
- Google Cloud Spanner ★
- Greenplum ★
- HANA ★
- HSQL Sever ★
- InfluxDB ★
- InfluxDB 2 ★
- InfluxDB 3 ★

- Informix ★
- Ingres ★
- InterSystems IRIS ★
- Kafka (ksqlDB) ★
- Keyspaces ★
- Materialize ★
- MaxDB ★
- MongoDB ★
- NDB Cluster ★
- Neo4j ★
- Neptune ★
- Netezza ★
- NetSuite ★
- OceanBase ★
- OpenSearch ★
- Raima ★
- Redis ★
- Redshift ★
- RisingWave ★
- Salesforce ★
- Salesforce CDP ★
- SAP ASE jConnect ★

- ScyllaDB ★
- SingleStore ★
- Snowflake ★
- StarRocks ★
- Sybase ★
- TDengine ★
- TDengine Cloud ★
- Teradata ★
- TiDB ★
- TimescaleDB ★
- Timestream ★
- Vertica ★
- Yellowbrick ★
- Yugabyte CQL ★
- YugabyteDB ★

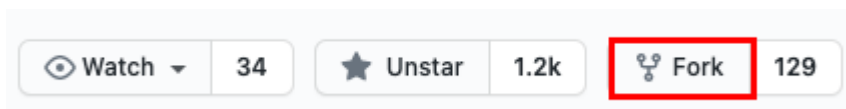
Note: you can add other drivers manually using [the instruction](#).

Localization

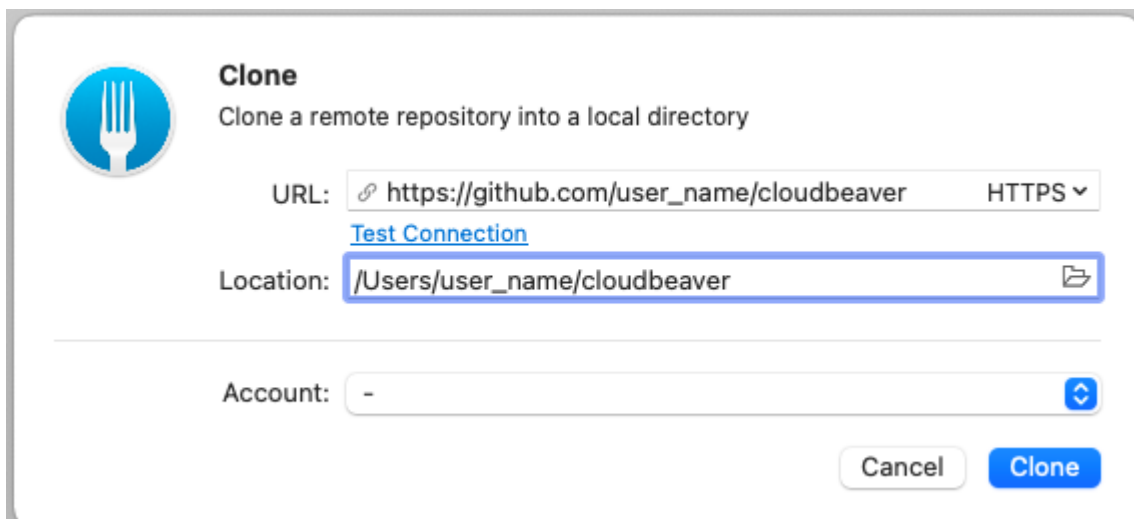
Not all localization comes from the Cloudbeaver source code. Some of it comes from the backend side. So, if you want to, for example, to change or create localization for database objects, you need to do it in another repository, at <https://github.com/dbeaver/dbeaver>. Here is a guide to how you can contribute to DBeaver's localization <https://github.com/dbeaver/dbeaver/wiki/Localization>. To create or improve localization for the Cloudbeaver interface, follow these steps:

Note: The images below are taken from [Fork](#). You can use another application to create pull-requests.

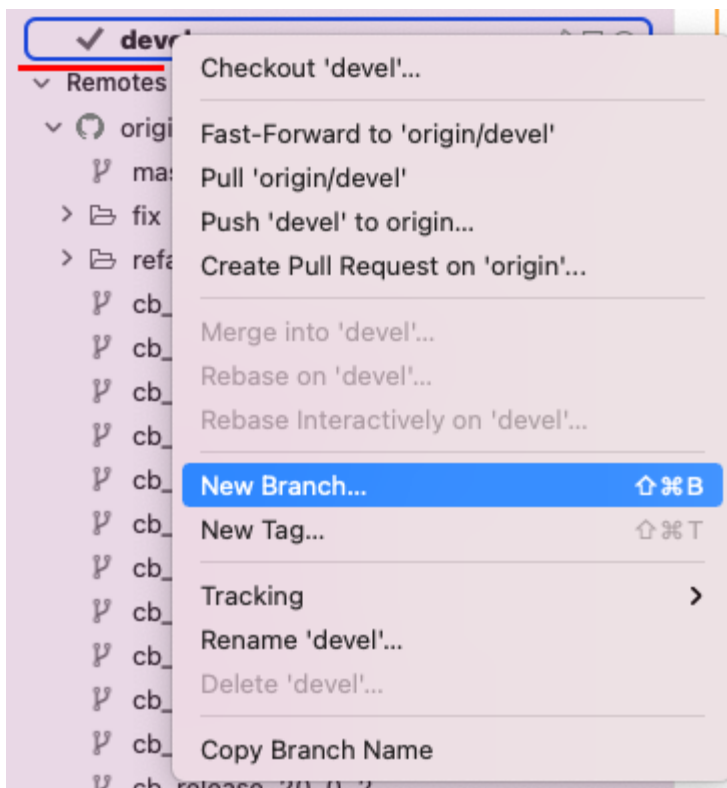
1. Open <https://github.com/dbeaver/cloudbeaver> and fork repository



2. Clone the forked repository <https://github.com/.../cloudbeaver> to your local system.



3. Create a new branch from *devel* branch (name it, for example, *italian-localization*).



4. Go to the local repository and find the localization files you want to translate.

In Cloudbeaver, all translatable resources are located in the locales folder. The path to the folder is

```
[package-name]\src\locales\[locale-code].ts
```

Create a copy of `en.ts` file in the package you want to change the localization in and name it `[locale-code].ts` (e.g. `it.ts` for Italian)

Example: `../repository_name/webapp/packages/core-localization/src/locales`

5. Open the created file and translate the tokens to your language. Change `EN` on `[locale-code]` in the top of the file `export const defaultENLocale => export const defaultITLocale`.

Here is the structure of the language tokens: `['token-name', 'token-value']`. You only need to change the second part: `'token-value'`. For example, if you wanted to translate the `Loading...` token, which is

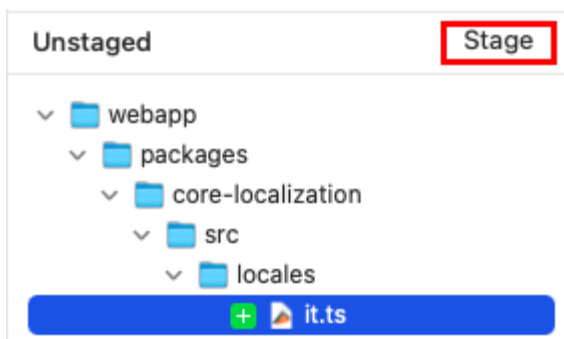
```
['ui_processing_loading', 'Loading...'],
```

it would look like this:

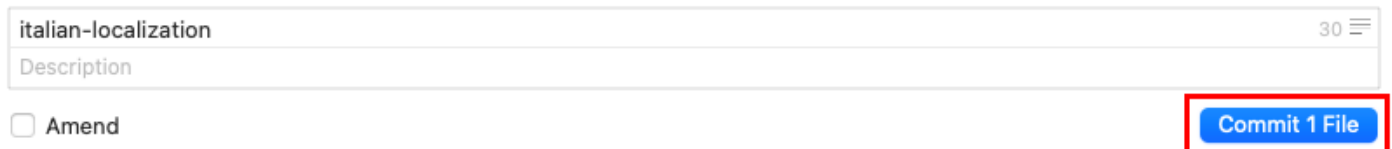
```
['ui_processing_loading', 'Caricamento in corso...']
```

6. Find localization service (`LocalizationService.ts` or similar service name) in same package you just created localization file. Then register new locale analogously to other ones in the file. Use `[locale-code]` for this switch case

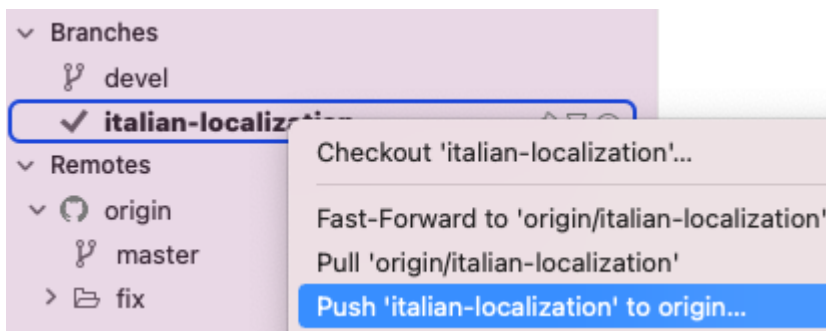
7. Open the branch changes and `stage` them



8. Commit the changes



9. Push to origin



10. Go to Github and press *Compare & pull request* in your repository

11. Write a description and create pull request

Here is [Github instruction](#). You can use different IDEs to create pull requests.

Create Connection

Table of contents

[How to Create a New Connection](#)

[Connection Access Management](#)

[Creating a Template in the Administration Section](#)

[Prerequisites:](#)

[Steps to Create a Template:](#)



[Shared vs Private Projects: Differences](#)

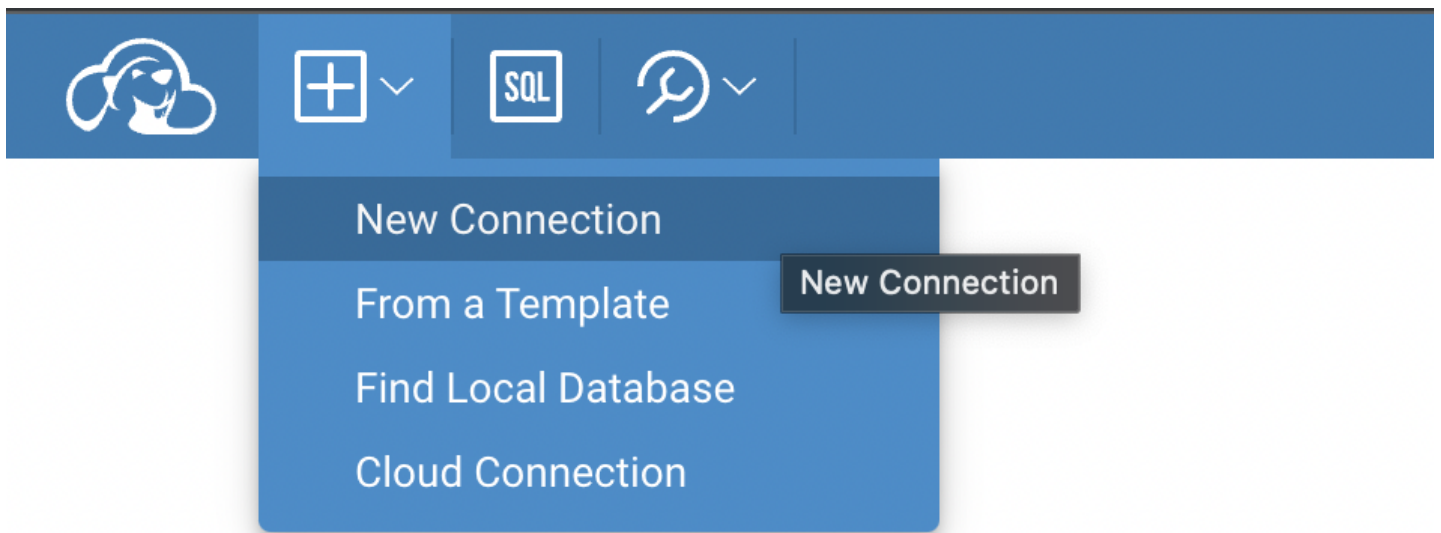
[Create shared/private project](#)

How to Create a New Connection

Follow the steps below to establish a new connection from the public part of the app:

1. Initiate New Connection:

- Click on the  button located in the top menu.
- Select  from the dropdown options.



2. Choose a Driver:

- From the available options, select the appropriate driver.

3. Fill in Connection Details:

- A connection form will be presented.
- Input all necessary details into the form fields.

A screenshot of the CloudBeaver 'New Connection' form. The form is divided into several sections. At the top, there is a blue header bar with icons and a user profile 'cbadmin'. Below the header, there are tabs: 'OPTIONS', 'DRIVER PROPERTIES', 'SSH TUNNEL', and 'SSL'. The 'DRIVER PROPERTIES' tab is active. The form contains the following fields:

- Driver:** A dropdown menu with 'PostgreSQL' selected.
- Configuration:** Two radio buttons, 'Manual' (selected) and 'URL'.
- Host *:** A text input field with 'localhost'.
- Port:** A text input field with '5432'.
- Database:** A text input field with 'postgres'.
- Authentication:** A dropdown menu with 'Database Native' selected.
- User name:** A text input field.
- User password:** A text input field with a toggle icon.
- Save credentials:** A checkbox.
- Connection name *:** A text input field with 'PostgreSQL@localhost'.
- Project:** A dropdown menu with 'Private' selected.
- Description:** A large text area.
- Settings:** Three checkboxes: 'Show all databases', 'Show template databases', and 'Show unavailable databases'.
- User role:** A text input field.

At the top right of the form, there are three buttons: 'CANCEL', 'TEST', and 'CREATE'. A vertical sidebar on the right side of the form contains a back arrow icon.

4. Set Up Advanced Settings (Optional):

- Additional settings such as SSH, SSL, and others can be adjusted.
- Navigate to the relevant tabs located at the top of the connection form to access these settings.

5. Test the Connection (Optional):

- Before finalizing the connection, you have the option to test it.
- Click the **Test** button to ensure the connection parameters are correct.

6. Finalize and Create the Connection:

- Once all details are in place, click the **Create** button.

7. Access and Modify Connection:

- The newly created connection will be visible in the connection navigator menu.
- To edit the connection details:
 - Click on the context button next to the connection name.
 - Select **Edit Connection** from the dropdown menu.

By following these steps, users can successfully create and modify connections in the app.

Connection Access Management

If you have **admin permissions**, you'll have the capability to manage connection's permissions. When working with a shared connection, you will notice an **Access** tab. This is where you can manage who has access to the connection.

Creating a Template in the Administration Section

Before proceeding with the creation of a template, ensure that you have the necessary administrative permissions.

Prerequisites:

- You must have **admin permissions** to access this feature.

Steps to Create a Template:

The steps are similar to connection creation; the only difference is that it must be done in the administration part, under the "Connection Templates" tab. You can learn more about creating connection templates [here](#).

Shared vs Private Projects: Differences

In our product you can create Private and Shared project

- Private projects are exclusive spaces owned by an individual user. They operate as personal projects, safeguarding sensitive information and allowing for undisturbed individual work.
- Shared projects are collaborative hubs where multiple users have access to contribute and edit content. These projects foster teamwork.

The choice between shared and private projects hinges on the project's nature and objectives. Whether opting for the autonomy of a private project or embracing the teamwork in a shared project, understanding these differences empowers individuals and teams to align their project management strategies with their specific needs.

Create shared/private project

If you want to be able to create private projects, your administrator must enable this feature in the **server configuration** panel:

CONFIGURATION



Enable private connections

Allows users to create private connections



Navigator simple view

By default, all user's new connections will contain only basic information in navigation tree

RESOURCE MANAGER



Enable Resource Manager

Enable Resource Manager functionality

When you create new connection, you can choose **Private** or **Shared** project, depending on the purpose for which you need to create:

MAINDRIVER PROPERTIESSSH TUNNELSSL

CANCELTESTCREATE

Driver

PostgreSQL

Configuration

☒ Manual☐ URL

Host *

localhost

Port

5432

Database

postgres

Connection name *

PostgreSQL@localhost

Project

Private

Private

Shared

AUTHENTICATION

Database Native

User name

User password

☐ Save credentials

MISC

User role

DATABASE LIST

SQL

Database navigator

Table of contents

[Overview](#)

[Description](#)

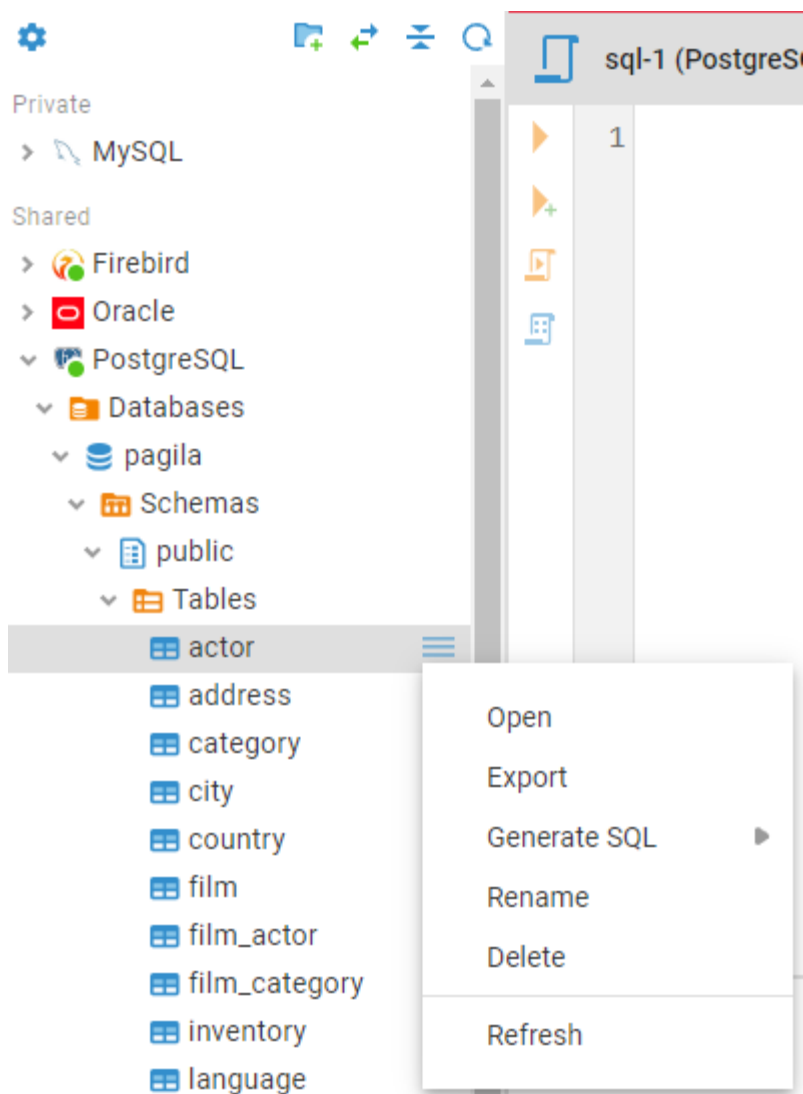
[Structure](#)

[Shortcuts](#)

Overview

Description

The Database Navigator is the main tool to work with the structure and content of databases. It is located on the left-hand side of the page and has [the toolbar](#), [the Settings menu](#) and a tree of objects.



Structure

The Database Navigator contains a tree of objects. Each object in the tree has its own context menu. The tree can contain the following objects:

- Folders
- Database connections
- Database objects - various depending on the database type, such as Tables, Views, Indexes, etc.

To open the context menu for an object, hover your cursor over the object in the Database Navigator and click the sandwich button to the right of the object.

The following table summarizes the context menu items for all types of objects that may appear in the tree. Note that the presence or absence of the context menu items for an object depends on the database and object types.

--	--

Menu item	Description
Open	Opens an object in a separate Document viewer
SQL Editor	Opens a new SQL Editor for the connection
Connection view	Changes the view of the Database Navigator
Edit Connection	Opens the Connection Configuration window that allows the configuration of connection settings
Disconnect	Disconnects from the database
Delete	Deletes an object. WARNING! The Delete menu item removes the object not only from the tree but from the database itself or the file system, and this action is irreversible.
Refresh	Refreshes the object's subnodes – mostly used for refreshing schemas
Rename	Opens the Rename dialog box for an object
Generate SQL	Opens a submenu on which you can select the type of SQL query to generate: <ul style="list-style-type: none"> - SELECT - INSERT - UPDATE - DELETE - MERGE - DDL Clicking one of the items (for example INSERT) generates a relevant query in a separate window
Export	Opens the Export window for table data export

To establish a database connection, do one of the following:

- open the connection;
- expand the connection in the Database Navigator;
- open the SQL Editor for the connection.

Users can store connections in [folders](#).

Shortcuts

Shortcut	Description
Ctrl + F	Filtering elements in tree
Ctrl + Shift + ,	Link with Editor
Ctrl + Shift + /	Collapse all

DB Navigator toolbar

The DB Navigator contains various tools on the top toolbar.



The table below contains a detailed description of each tool.

Tool	Description
Refresh	refreshes all objects in the Navigator tree.
New folder	creates a folder in the Navigator tree to group connections.
Collapse all	collapses all expanded objects in the Navigator tree. It becomes visible when at least one object is expanded.
Link with editor	synchronises the active SQL Editor, Data Editor or Metadata Editor with the element in the Database Navigator. It becomes visible when an Editor for a connection is opened.
Settings	opens the Settings menu with additional tools.

DB Navigator Settings menu

Table of contents

[DB Navigator Settings menu](#)

DB Navigator Settings menu

The Settings menu gives access to additional tools of the Navigator tree. To open it, press the Settings button in the upper left corner of the DB Navigator.



The menu contains following tools:

- SETTINGS
- ☐ Filter
 - ☐ Show collapsed
 - ☒ Save tree state
 - ☐ Folders
 - ☒ Group by Project

The table below provides a description of each tool:

Tool	Description
Filter	Objects' search in the Navigator tree. Enable the Filter switch to see the filter field and enter the object name to see relevant objects in the Navigator. The search goes among visible objects only.
Show collapsed	The additional setting for the Filter. The search will also go through collapsed folders if the user has previously expanded them.
Save tree state	Keep the Navigator tree view after refreshing the browser page. It's enabled by default.

Folders	Displays only one level of folders. The full path to the folder is shown as breadcrumbs. When a user double-clicks on an object, the path to it is displayed in the Navigator tree.
Group by Project	Removes project names from the Navigator tree view.

DB Navigator folders

Table of contents

[Database Navigator folders](#)

Database Navigator folders

Users can conveniently use folders to group and store connections in the Navigator tree.

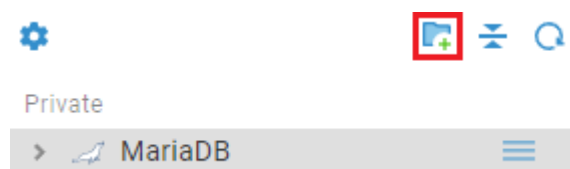
Any logged user can create folders for their connections in the Private project.

Administrators can also create folders for shared connections in the Shared project. Folders for connections will only be available in the Navigator tree. The Connection Management view in the Administration will remain unchanged.


Both, Private and Shared, projects are completely independent and it is not possible to move connections and folders between them. Any number of connections and other folders can be placed in one folder.

To create a folder in the Database Navigator:

1. Click a connection in the Navigator tree, then press the New folder button on the top Navigator toolbar.



2. Enter a folder name in the pop up dialog box and press the Create button.

 **New folder** ×

Name:


CANCEL CREATE

Note: a folder name can contain letters, numbers and the following symbols: _-\$.()@. The folder name must be unique in the project. But a subfolder can have the same name as a folder at a level above. If a user uses unsupported characters or a non-unique name for the folder, an error message will appear and the user will need to change the name in order to create or rename the folder.


The new empty folder will appear in the Navigator tree. You can then place connections in this new folder. Just drag and drop the connection into it.

Private

▼

 analytics

>

 MariaDB

Once a connection is placed into a folder, the information about it appears in the connection window.

Connection name *

Project

Private ▼


Folder

PostgreSQL


To move a connection out of a folder, drag and drop it above the list of Navigator objects.

Private

▼

 analytics

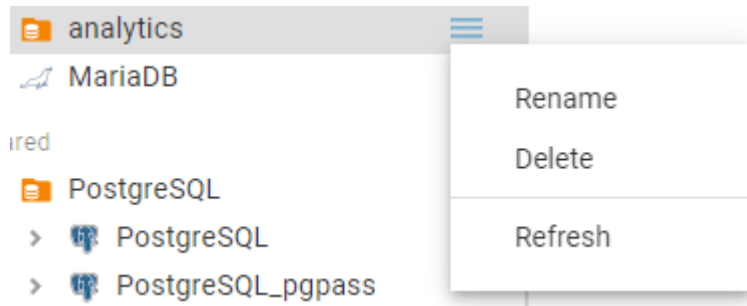
>

 MariaDB

If you want to place a subfolder in a folder: drag and drop a folder into another folder or click on a folder and create a new one. The created folder will be placed in the selected folder.

If you want to move a subfolder out of a folder, drag and drop it above the list of Navigator objects or into another folder.

If you want to rename a folder, select Rename from the folder context menu and change the name in the pop up dialog box.



To delete a folder, select Delete from the folder context menu and confirm the action in the pop up dialog box.

Note: only the folder and subfolders in it will be deleted. All the folder's connections remain in the Navigator tree.

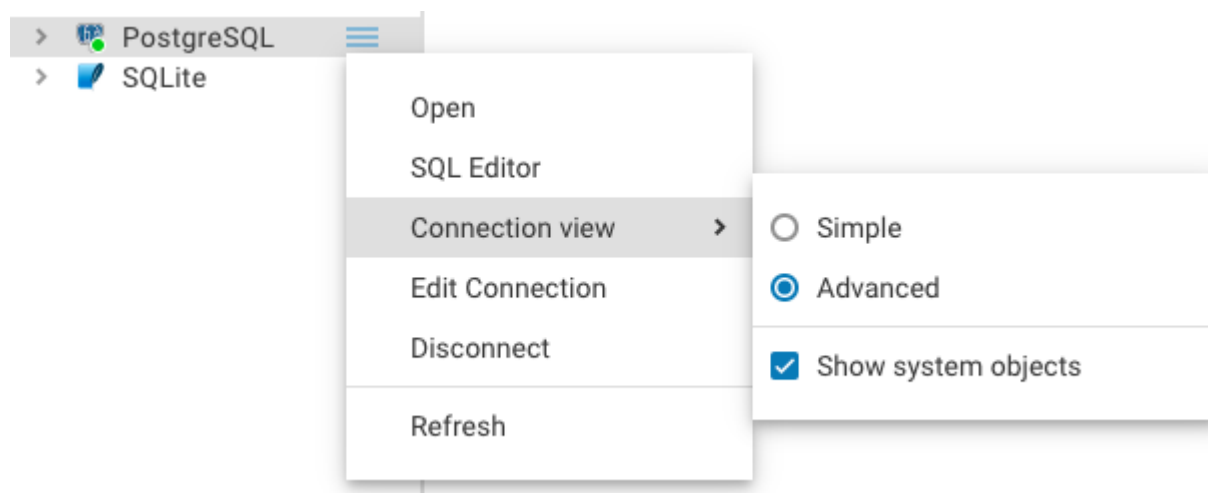
If you want to update a folder's content, select Refresh from the folder context menu.

Simple and Advanced View

Simple and Advanced Connection view defines what [the Database navigator](#) and the Metadata Editor structures will look like:

- **Advanced view** shows all database objects. It is enabled by default.
- **Simple view** shows only the schemas and tables. It can be enabled in the Easy configuration or the Administration menu by the admin for all new connections.

Users can also change the view of any connection in the connection context menu in the Database Navigator.



- **Show system objects** mode makes system schemas and tables available in the Database navigator and the Metadata Editor. It can be enabled in the connection context menu in the Database Navigator.

Tip: Default view could be set by the admin in the Server configuration. This will only affect newly created connections.

Learn more

Table of contents

[Features](#)

[Shortcuts](#)

Features

- [Shortcuts](#)
- [Data Filters](#)
- [Data Ordering](#)
- [Value Panel](#)
- [Grouping panel](#)
- [JSON and Document View](#)
- [Working with spatial/GIS data](#)


Shortcuts

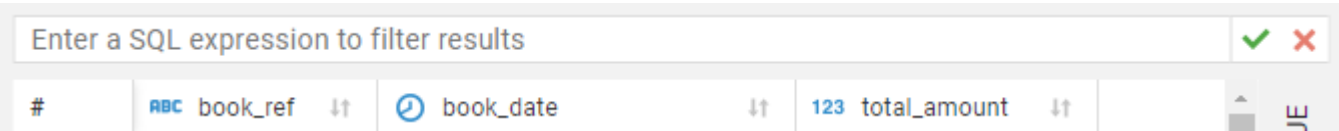
Shortcut	Description
Enter Backspace a-Z and 0-9 keys	Start inline editing
Alt+Insert	Add a new row
Ctrl+Alt+Insert	Duplicate row
Delete	Delete row
Escape	Revert changes

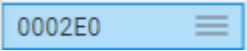
Ctrl+V	Past value to cell from clipboard
Ctrl+C	Copy data to clipboard

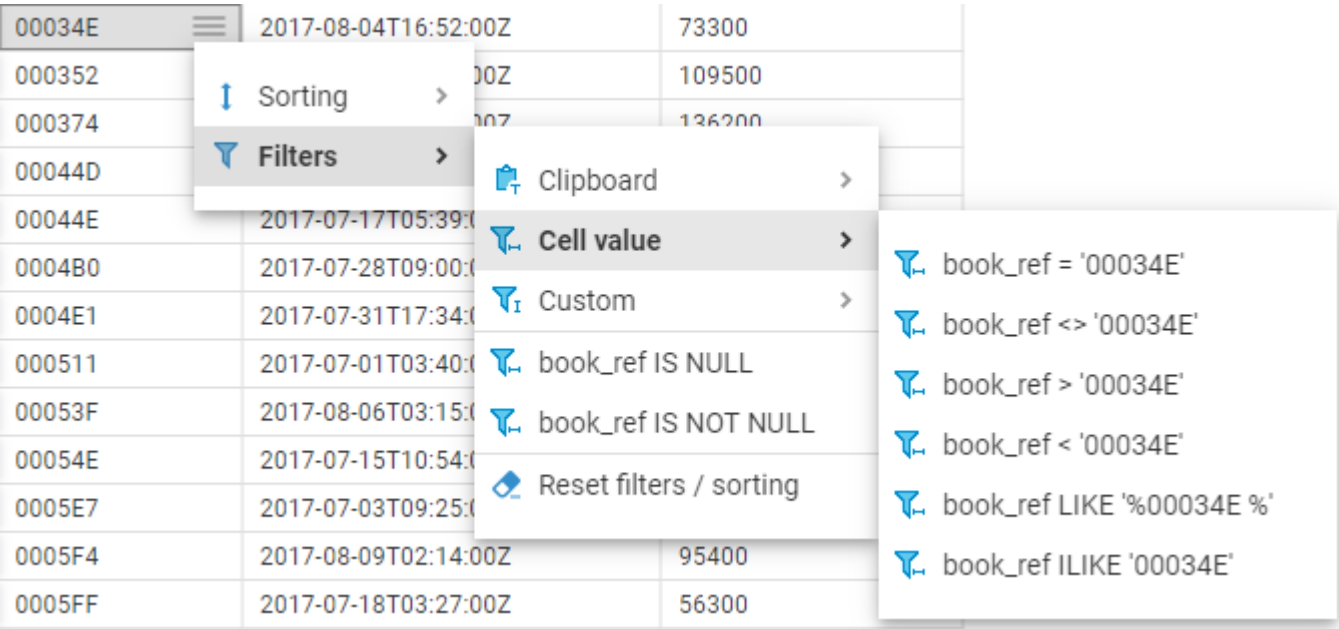
Data Filters


You can apply custom filters to table contents or query results. There are several ways in which you can filter data in a table.








One of the ways is to use the filter field above the table next to the top toolbar. To filter data, enter an SQL expression into the field and click the Apply filter criteria button  next to the field or press `Enter`.



You can apply ready-to-use SQL expressions or SQL expression templates via the context menu. To select a ready SQL expression, select or focus cell and press the Cell context button , then click **Filters** -> **Cell value** in the context menu and choose one of the expressions.








The data updates dynamically. To remove a filter, click **Filters** -> **Delete filter for ...**. If you want to delete all filters, click **Filters** -> **Reset all filters**. You can also delete filters by clicking the Reset filter criteria button  in the top toolbar.

-  Clipboard >
-  Cell value >
-  Custom >
-  book_ref IS NULL
-  book_ref IS NOT NULL
-  Delete filter for book_ref
- Reset all filters
-  Reset filters / sorting




Data Ordering

You can order data in columns in one of the following ways:

1. Click the ordering icon  in the header of the column.

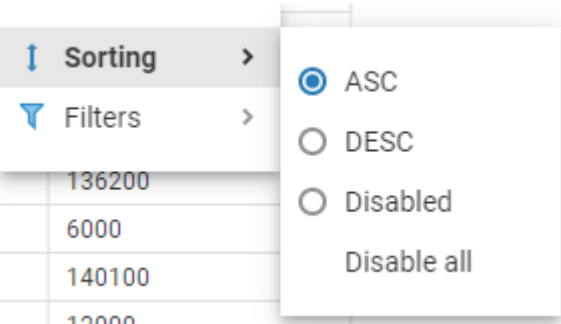
#	ABC book_ref 	 book_date 	123 total_amount 
---	--	---	--

The icon has three states:

- Clicking once establishes an ascending order 
- Clicking a second time changes the order to descending 
- Clicking a third time removes the ordering from the column 

To order data by multiple columns, go column by column, holding the `ctrl` (windows) or `cmd` (mac) button, setting the order with the Ordering icon, starting from the column by which you want to order the data first.

2. Open the context menu, click **Sorting**. Choose from the ordering states we have mentioned before. When we choose an ordering state from the context menu, there is no need to hold the `ctrl` or `cmd` button to order data by multiple columns.



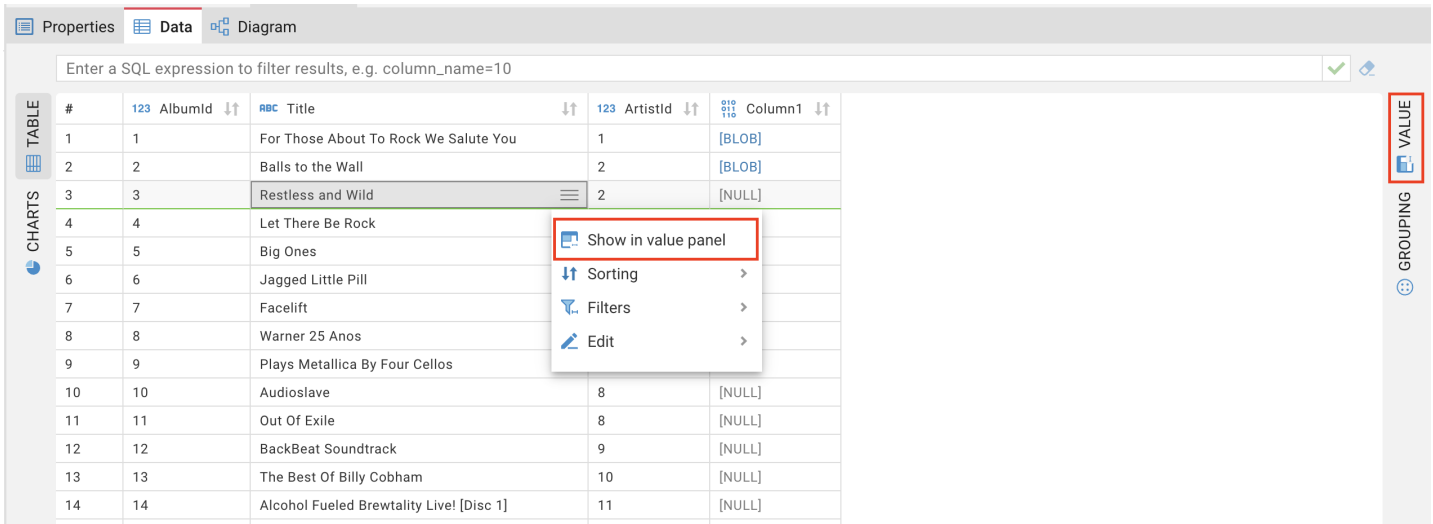
To reset the column data ordering to its initial state, open the Cell context menu and click **Sorting -> Disable**. If you want to reset all data ordering, open the Cell context menu, then click **Sorting -> Disable all**. You can also click the Ordering icon in the header of the any column to reset all data ordering.

Value Panel

The Value Panel provides additional space in the **Data editor** in which you can manipulate data. The panel is handy if you work with complex types (structures, arrays), long text data or BLOBs.

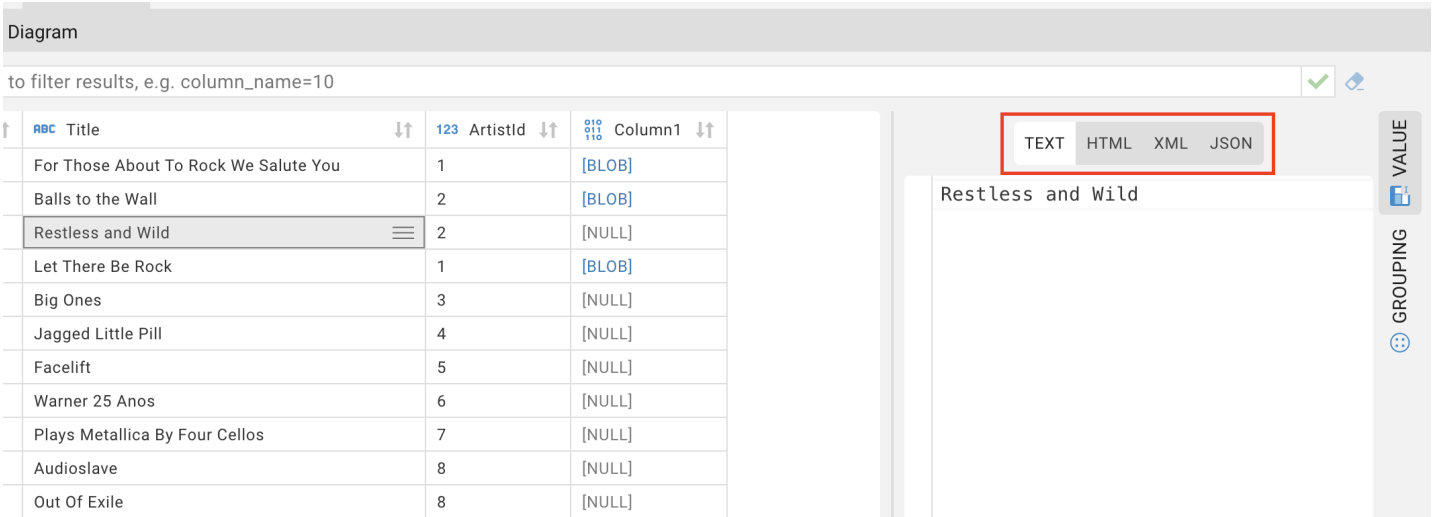
To open the panel, click the **Value** button on the right hand side of the **Data tab**. Alternatively, you can open the **Value** panel by clicking **Show the in value panel** in the cell context menu.

To close the panel, click the **Value** button again.



The **Value** Panel displays just one value that is currently selected or in focus and allows editing.

At the top of the **Value** Panel, you can find several tabs. The tabs depend on the current value type. For example, if your current value is a string, you will find 4 tabs (Plain text, HTML, XML, JSON), each representing a format the string can be shown in.



There are 3 available tabs for BLOB type of data:

- Text
- HEX
- Base64

↑

Column1

↓

[BLOB]
[BLOB]
[NULL]
[BLOB]
[NULL]
[NULL]
[NULL]
[NULL]

IMAGE

TEXT

TEXT

HEX

BASE64

iVBORw0KGgoAAAANSUhEUgAAAG4AAABuCAIAA

VALUE

GROUPING

Grouping Panel

Table of contents

[Overview](#)

[Grouping panel](#)

[Function management](#)

[Adding a function](#)

[Removing a function](#)

[Column management](#)

[Adding a column](#)

[Removing a column](#)

[Filtering](#)

[Default sorting](#)

[Show duplicates only](#)

[Charts](#)

[Export grouped data](#)

Overview

The **Grouping panel** provides tools to calculate statistics based on a table or a custom SQL query. It uses `GROUP BY` queries to extract unique values for `COUNT` (default), `SUM`, `AVG`, `MIN`, `MAX`, and other analytics functions, displaying the results in dedicated columns.

Grouping panel

To obtain the grouping results for one or more columns of a data table, open the **Grouping Panel**. Then, in the result table, place the cursor on the table header and drag-and-drop the column into the panel:

actor

Properties Data Diagram

Enter a SQL expression to filter results, e.g. column_name=10

#	123 actor_id	first_name	last_name	last_update	123 age
1	1	PENELOPE	GUINNESS	2006-02-15 09:34:33.000	[NULL]
2	2	NICK	WAHLBERG	2006-02-15 09:34:33.000	[NULL]
3	3	ED	CHASE	2006-02-15 09:34:33.000	[NULL]
4	4	JENNIFER	DAVIS	2006-02-15 09:34:33.000	[NULL]
5	5	JOHNNY	LOLLOBRIGIDA	2006-02-15 09:34:33.000	[NULL]
6	6	BETTE	NICHOLSON	2006-02-15 09:34:33.000	[NULL]
7	7	GRACE	MOSTEL	2006-02-15 09:34:33.000	[NULL]
8	8	MATTHEW	JOHANSSON	2006-02-15 09:34:33.000	[NULL]
9	9	JOE	SWANK	2006-02-15 09:34:33.000	[NULL]
10	10	CHRISTIAN	GABLE	2006-02-15 09:34:33.000	[NULL]
11	11	ZERO	CAGE	2006-02-15 09:34:33.000	[NULL]
12	12	KARL	BERRY	2006-02-15 09:34:33.000	[NULL]
13	13	UMA	WOOD	2006-02-15 09:34:33.000	[NULL]
14	14	VIVIEN	BERGEN	2006-02-15 09:34:33.000	[NULL]
15	15	CUBA	OLIVIER	2006-02-15 09:34:33.000	[NULL]
16	16	FRED	COSTNER	2006-02-15 09:34:33.000	[NULL]
17	17	HELEN	VOIGHT	2006-02-15 09:34:33.000	[NULL]
18	18	DAN	TORN	2006-02-15 09:34:33.000	[NULL]
19	19	BOB	FAWCETT	2006-02-15 09:34:33.000	[NULL]
20	20	LUCILLE	TRACY	2006-02-15 09:34:33.000	[NULL]
21	21	KIRSTEN	PALTROW	2006-02-15 09:34:33.000	[NULL]
22	22	ELVIS	MARX	2006-02-15 09:34:33.000	[NULL]
23	23	SANDRA	KILMER	2006-02-15 09:34:33.000	[NULL]
24	24	CAMERON	STREEP	2006-02-15 09:34:33.000	[NULL]
25	25	KEVIN	BLOOM	2006-02-15 09:34:33.000	[NULL]
26	26	RIP	CRAWFORD	2006-02-15 09:34:33.000	[NULL]
27	27	JULIA	MCQUEEN	2006-02-15 09:34:33.000	[NULL]
28	28	WOODY	HOFFMAN	2006-02-15 09:34:33.000	[NULL]

There is no data to show.
Drag and drop a column from the result viewer to group values.

200 row(s) fetched - 46ms

If you add several columns to the panel, CloudBeaver groups the data in the order the columns are added and calculates statistics based on the grouping.

actor

Properties Data Diagram

Enter a SQL expression to filter results, e.g. column_name=10

#	123 actor_id	first_name	last_name	last_update	123 age
1	1	PENELOPE	GUINNESS	2006-02-15 09:34:33.000	[NULL]
2	2	NICK	WAHLBERG	2006-02-15 09:34:33.000	[NULL]
3	3	ED	CHASE	2006-02-15 09:34:33.000	[NULL]
4	4	JENNIFER	DAVIS	2006-02-15 09:34:33.000	[NULL]
5	5	JOHNNY	LOLLOBRIGIDA	2006-02-15 09:34:33.000	[NULL]
6	6	BETTE	NICHOLSON	2006-02-15 09:34:33.000	[NULL]
7	7	GRACE	MOSTEL	2006-02-15 09:34:33.000	[NULL]
8	8	MATTHEW	JOHANSSON	2006-02-15 09:34:33.000	[NULL]
9	9	JOE	SWANK	2006-02-15 09:34:33.000	[NULL]
10	10	CHRISTIAN	GABLE	2006-02-15 09:34:33.000	[NULL]
11	11	ZERO	CAGE	2006-02-15 09:34:33.000	[NULL]
12	12	KARL	BERRY	2006-02-15 09:34:33.000	[NULL]
13	13	UMA	WOOD	2006-02-15 09:34:33.000	[NULL]
14	14	VIVIEN	BERGEN	2006-02-15 09:34:33.000	[NULL]
15	15	CUBA	OLIVIER	2006-02-15 09:34:33.000	[NULL]
16	16	FRED	COSTNER	2006-02-15 09:34:33.000	[NULL]
17	17	HELEN	VOIGHT	2006-02-15 09:34:33.000	[NULL]
18	18	DAN	TORN	2006-02-15 09:34:33.000	[NULL]
19	19	BOB	FAWCETT	2006-02-15 09:34:33.000	[NULL]
20	20	LUCILLE	TRACY	2006-02-15 09:34:33.000	[NULL]
21	21	KIRSTEN	PALTROW	2006-02-15 09:34:33.000	[NULL]
22	22	ELVIS	MARX	2006-02-15 09:34:33.000	[NULL]
23	23	SANDRA	KILMER	2006-02-15 09:34:33.000	[NULL]
24	24	CAMERON	STREEP	2006-02-15 09:34:33.000	[NULL]
25	25	KEVIN	BLOOM	2006-02-15 09:34:33.000	[NULL]
26	26	RIP	CRAWFORD	2006-02-15 09:34:33.000	[NULL]
27	27	JULIA	MCQUEEN	2006-02-15 09:34:33.000	[NULL]
28	28	WOODY	HOFFMAN	2006-02-15 09:34:33.000	[NULL]
29	29	ALEC	WAYNE	2006-02-15 09:34:33.000	[NULL]

Enter a SQL expression to filter results, e.g. column_name=10

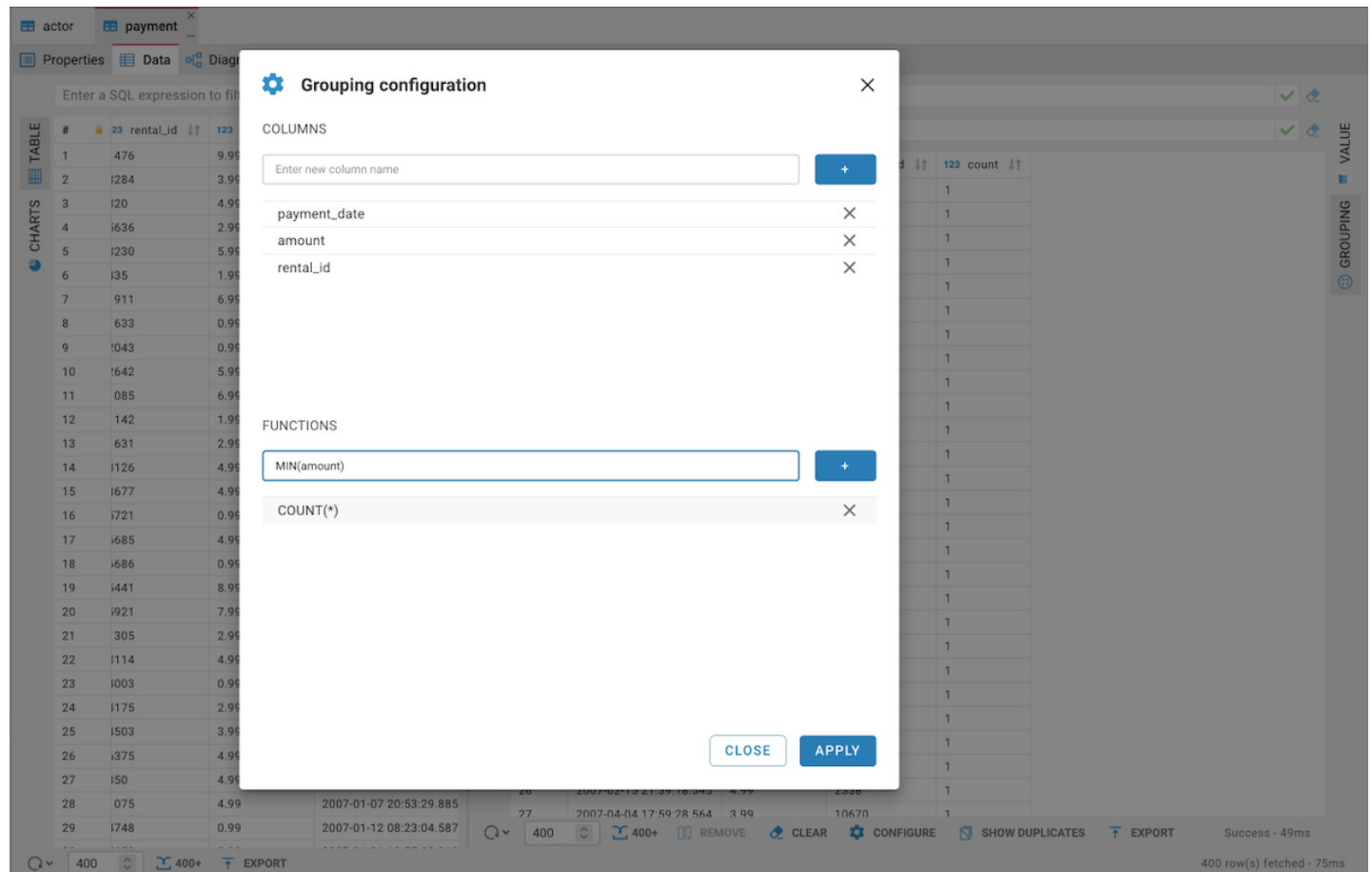
#	last_name	last_update	123 count
1	DEGENERES	2006-02-15 09:34:33.000	3
2	ALLEN	2006-02-15 09:34:33.000	3
3	HURT	2006-02-15 09:34:33.000	1
4	MCKELLEN	2006-02-15 09:34:33.000	2
5	BIRCH	2006-02-15 09:34:33.000	1
6	PINKETT	2006-02-15 09:34:33.000	1
7	SWANK	2006-02-15 09:34:33.000	1
8	NEESON	2006-02-15 09:34:33.000	2
9	HUDSON	2006-02-15 09:34:33.000	1
10	CAGE	2006-02-15 09:34:33.000	2
11	PESCI	2006-02-15 09:34:33.000	1
12	WOOD	2006-02-15 09:34:33.000	2
13	TRACY	2006-02-15 09:34:33.000	2
14	LOLLOBRIGIDA	2006-02-15 09:34:33.000	1
15	JOLIE	2006-02-15 09:34:33.000	1
16	MOSTEL	2006-02-15 09:34:33.000	2
17	GARLAND	2006-02-15 09:34:33.000	3
18	BENING	2006-02-15 09:34:33.000	2
19	BARRYMORE	2006-02-15 09:34:33.000	1
20	BASINGER	2006-02-15 09:34:33.000	1
21	DUKAKIS	2006-02-15 09:34:33.000	2
22	DERN	2006-02-15 09:34:33.000	1
23	AKROYD	2006-02-15 09:34:33.000	3
24	HUNT	2006-02-15 09:34:33.000	1
25	CARREY	2006-02-15 09:34:33.000	1
26	WEST	2006-02-15 09:34:33.000	2
27	DFPP	2006-02-15 09:34:33.000	2

200 row(s) fetched - 46ms


Function management

Adding a function

By default, the **COUNT** function is used. You can add other functions as well.



To add a function, follow these steps:

1. Click the **Configure** button  **CONFIGURE** on the Grouping panel's toolbar.
2. In the **Grouping Configuration** window, locate the **Functions** area and enter the function into the row.

Note: **COUNT** is the only function that allows the use of ***** instead of specifying a column name.

3. To complete the process, click **Apply**.

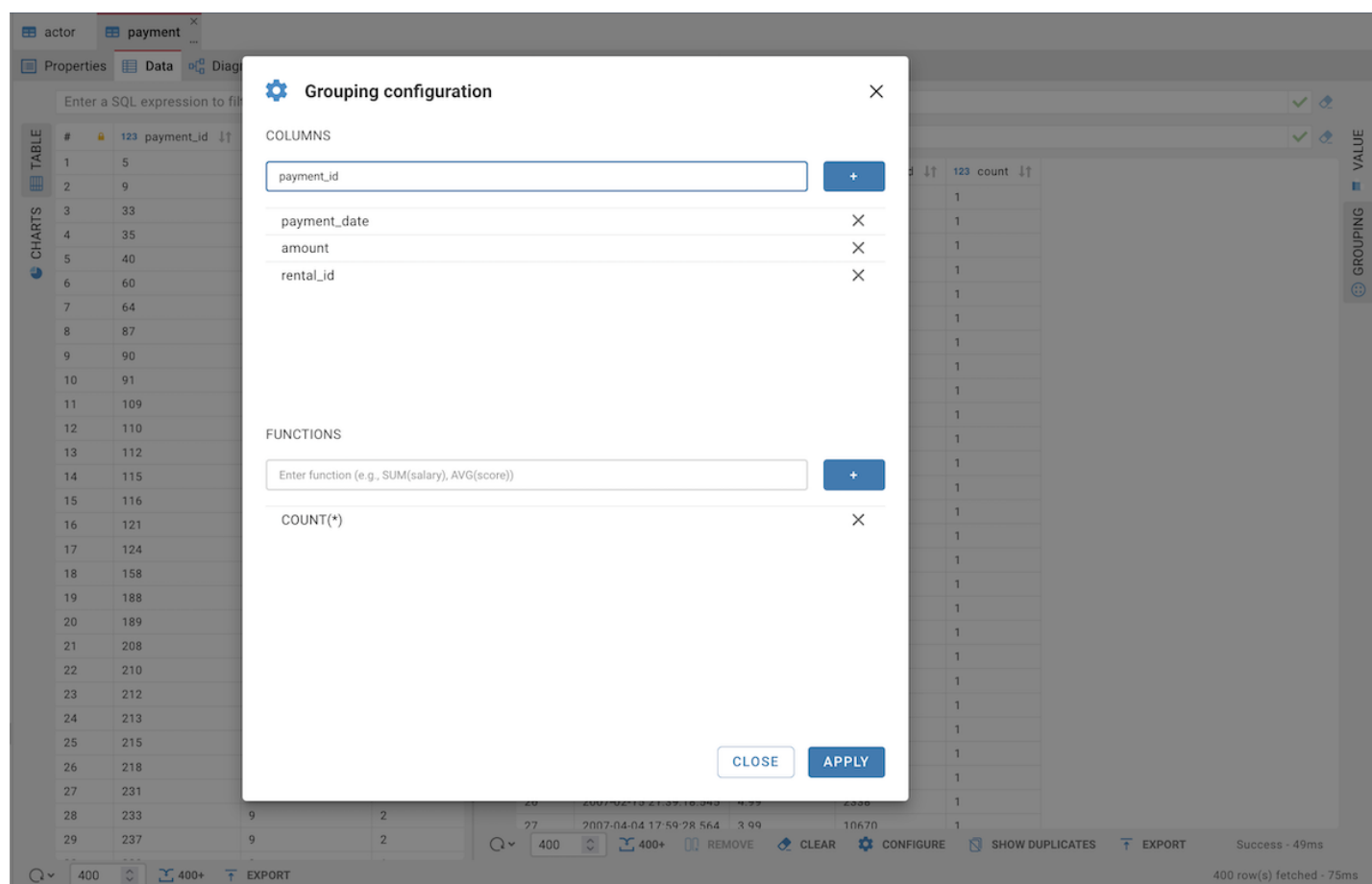
Removing a function

To remove a function, navigate to the **Grouping Configuration** window, click the **Remove** button, and then click **Apply**.


Column management

Adding a column

You can also manage columns in the same **Grouping Configuration** window.



To add a column:

1. Click the **Configure** button  **CONFIGURE** on the Grouping panel's toolbar.
2. In the **Grouping Configuration** window, go to the **Columns** area and enter the column name into the row.
3. To complete the process, click **Apply**.

Tip: You can add a column with an expression for MySQL/MariaDB databases. The expression will be calculated in the resulting column:

Grouping configuration

COLUMNS

TOTAL_PRICE / 100 * 20

TOTAL_PRICE

FUNCTIONS

Enter function (e.g., SUM(salary), AVG(score))

count(*)

CLOSE
APPLY

Enter a SQL expression to filter results, e.g. column_name=10

#	123 TOTAL_PRICE	123 TOTAL_PRICE / 100 * 20	123 count_
1	199	39.8	2
2	279	55.8	2
3	289	57.8	2
4	298	59.6	4
5	328	65.6	2
6	330	66	2
7	338	67.6	2
8	360	72	8
9	378	75.6	2
10	400	80	2
11	406	81.2	2
12	410	82	6
13	428	85.6	2
14	459	91.8	2
15	460	92	17
16	500	100	2
17	590	118	2
18	592	118.4	2
19	610	122	2
20	640	128	2
21	650	130	2
22	690	138	12
23	720	144	4
24	722	144.4	2
25	755	151	2
26	760	152	2
27	788	157.6	7

400

86

REMOVE

CLEAR

CONFIGURE


SHOW DUPLICATES

EXPORT

Success - 29ms

272 row(s) fetched - 92ms

Removing a column

- To remove a column, in the same **Grouping Configuration** window, click the cross (X) icon and then click **Apply**.
- Alternatively, you can remove a column by clicking the column name and then clicking the **Remove** button  in the Grouping panel's toolbar.
- You can also remove a column by dragging and dropping it outside the Grouping panel.

To clear all results from the **Grouping Panel**, click the **Clear** button

Filtering

You can filter data in the **Grouping Panel**. For more information, refer to the [Data Filters](#) article.

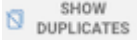
Default sorting

Click the **Sort by Column** button to switch sorting options for the grouped data:

Option	Icon	Description
Unsorted	⇅	Leaves the data in its original order.

Ascending	↑	Sorts the grouped data in ascending order.
Descending	↓	Sorts the grouped data in descending order.

Show duplicates only

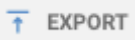
Click the **Show Duplicates** button  to filter the results and display only those rows where the **COUNT** is greater than **1**.

Charts

Note: This feature is available in [Enterprise](#), [AWS](#) and [Team](#) editions only.

You can create and manage various charts directly within the **Grouping Panel**. For more information, refer to the [Managing Charts](#) article.

Export grouped data

You can export grouped data using the Export button  in the **Grouping Panel**. For more details on exporting data, refer to the [Data Export](#) article.

Managing Charts

Table of contents

[Creating Charts In SQL Editor](#)

[Creating Charts In Data Editor](#)

[Creating Charts In Grouping Panel](#)

[Editing Chart Settings](#)

[Setting Axis X](#)

[Setting Axis Y](#)

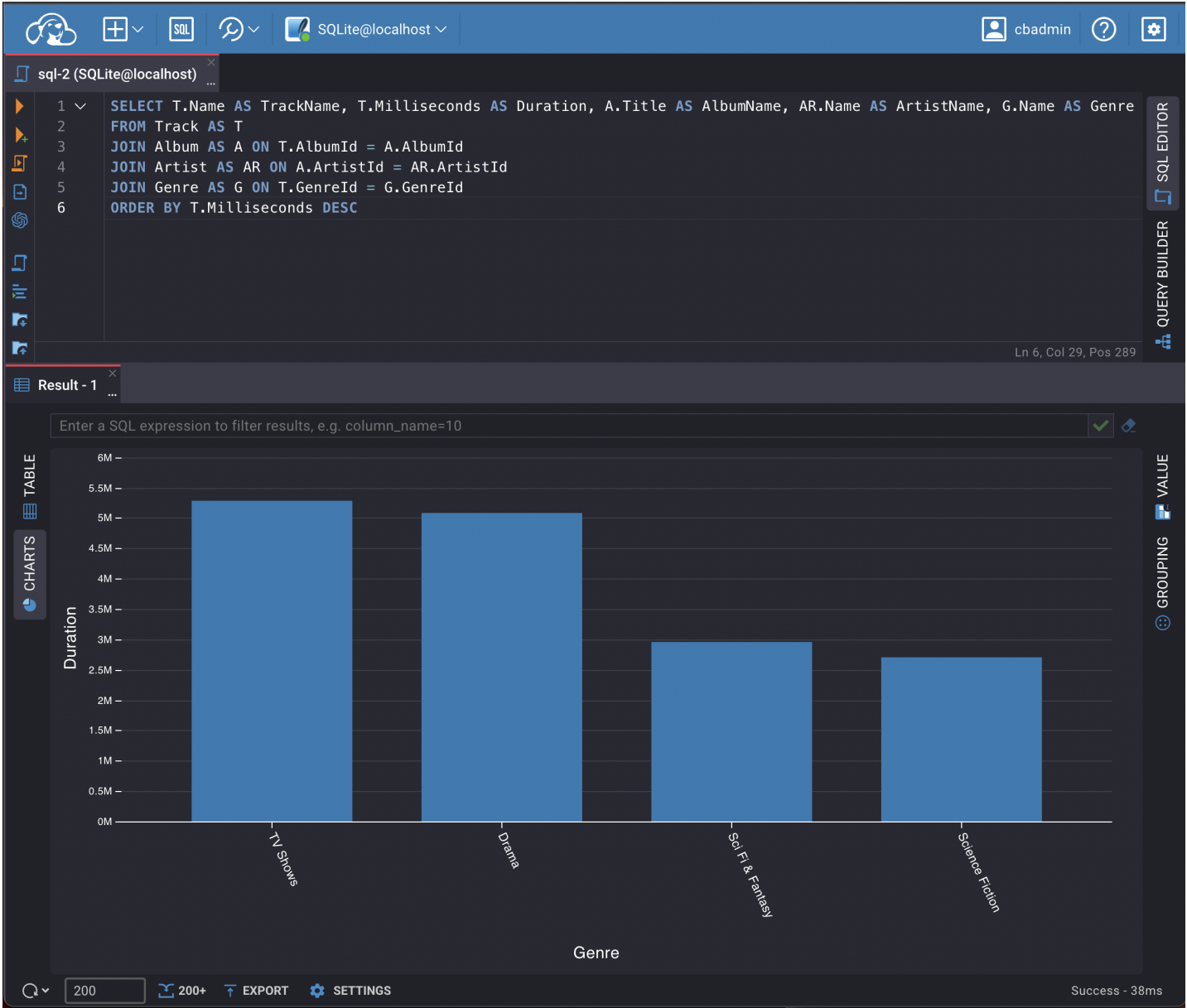
Use CloudBeaver's Charts feature to transform your `SELECT` queries into colorised charts. You can visualize data in **Bar** and **Pie** formats.

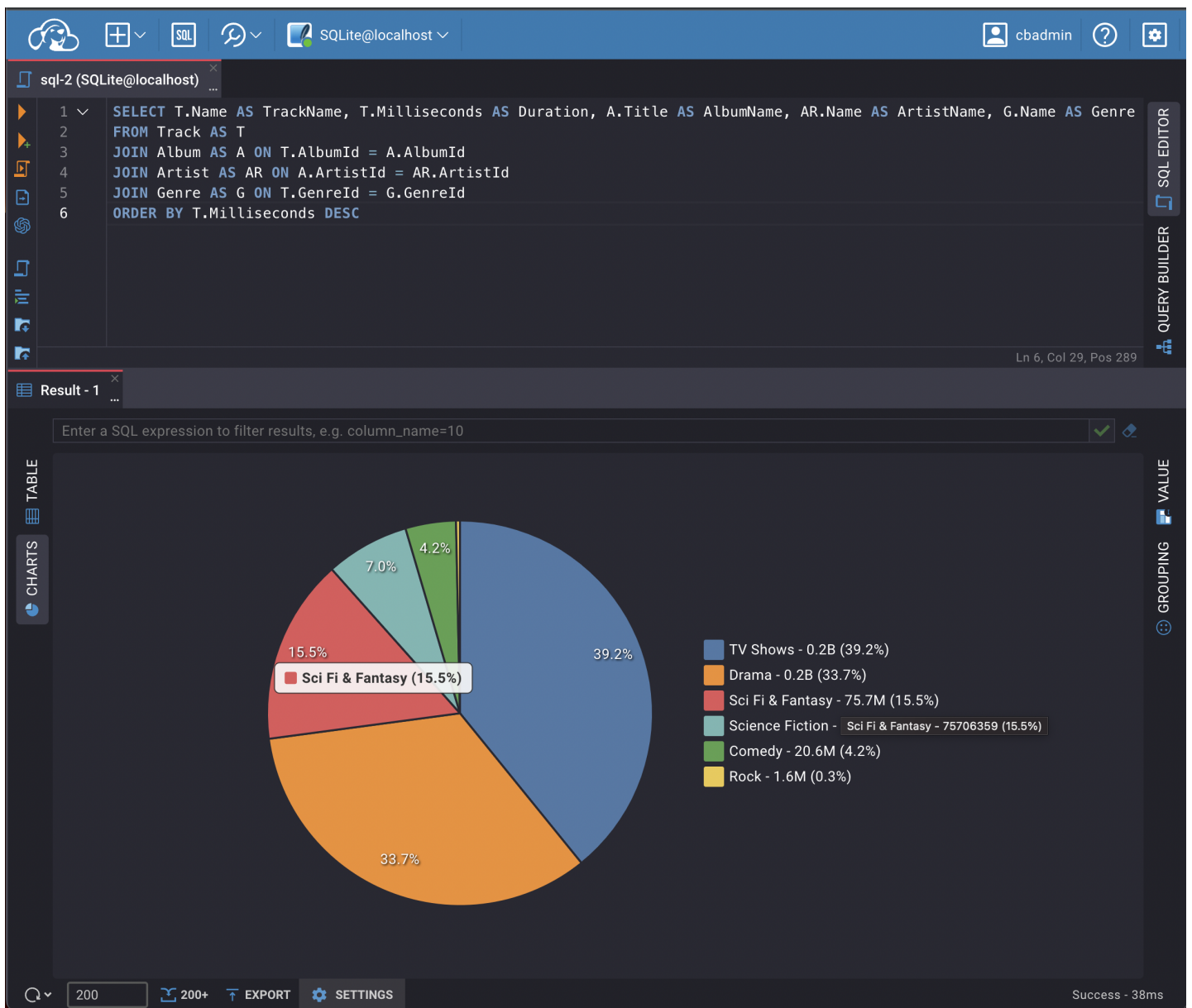
Create a chart directly in the [SQL Editor](#), [Data Editor](#) or Grouping Panel. It's a quick and simple way to bring your data to life.

Creating Charts In SQL Editor

Visualising large data sets makes your analysis faster and more precise. Charts let you grasp important details quickly.

To create a chart in the **SQL Editor**, click the **Charts** button on the left vertical toolbar of the query results area. Your data instantly transforms into a visual representation.





Creating Charts In Data Editor

Charts are a powerful tool for visualising structured analytical data, such as those stored in Views.

If you want to create a chart for a table, it requires some preparation. First, sort and apply various [filters](#) to the table's columns to structure the data. These changes directly affect the appearance of the chart, enabling you to customize the visual representation to fit your needs.

To build a chart using the **Data Editor**, simply press the **Charts** button located in the left vertical toolbar. A chart based on your structured data will be created.

Note: By default, the data for Y-axis is taken from the first column of the table containing numeric values.

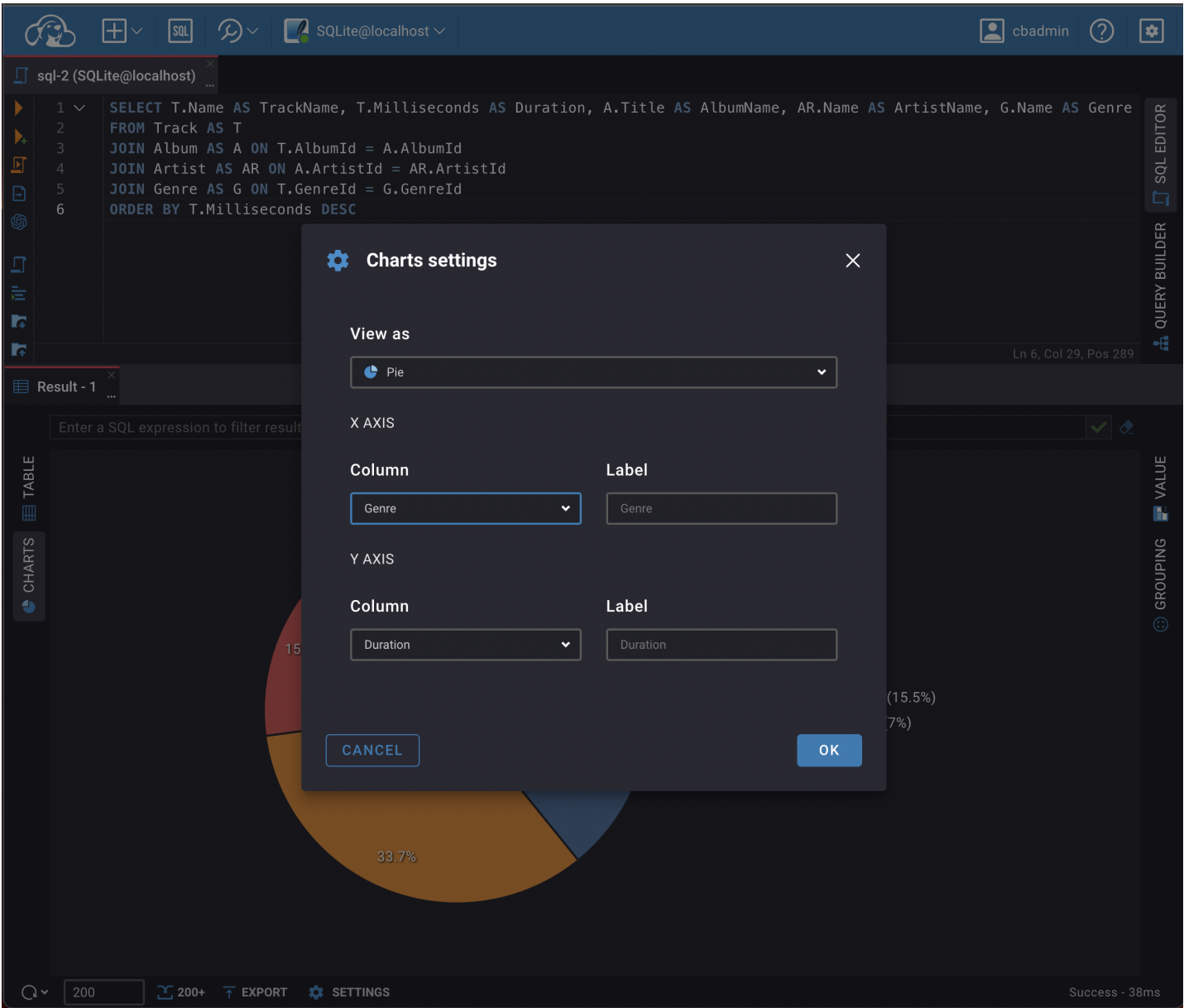
Creating Charts In Grouping Panel

The **Charts** feature also works with analytical tools like the [Grouping Panel](#). When building a chart for a table that contains grouping results for one or more columns, you can easily switch the source data for the X and Y axes by changing the columns in the Charts Editor.

To create a chart in the **Grouping Panel**, press the **Charts** button on the left vertical toolbar.

Editing Chart Settings

To edit the chart settings, click **Settings** button in the bottom menu. The settings dialog will then appear, allowing you to make adjustments.



The following chart settings can be adjusted:

Setting Axis X

- In the **Column** drop down list of available columns select a column whose data will be used on X-axis of the bar chart. Make sure you choose unique columns for X-axis.
- Define a user-friendly axis name in the **Label** text field.

Setting Axis Y

- In the **Column** drop down list of available columns, select a column whose data will be used on Y-axis of the bar chart.
Note, that only numeric and string columns can be used for Y-axis.
- Define a user-friendly axis name in the **Label** text field.

JSON and Document View

Table of contents

[Overview](#)

[Description](#)

[Document selection](#)

[Editing](#)

Overview

Description

JSON data representation is available in several databases such as **DocumentDB**, **DynamoDB** and other **NoSQL databases**. Data is represented as formatted JSON. The headline of the latter is *unique identification* of the document.

Properties Data Diagram

Enter a SQL expression to filter results

year:1933, title:1

```
{
  year: 1933,
  title: "1",
  info: {
    actors: [
      "Fay Wray2",
      "Robert Armstrong",
      "Bruce Cabot"
    ],
    plot: "123 A film crew \n goes to a tropical island for an exotic location shoot and discovers a colossal giant gorilla who takes a shine to their female blonde star.",
    release_date: "1933-03-07T10:00:00Z",
    genres: [
      "Adventure",
      "Fantasy",
      "Horror"
    ],
    image_url: "https://ia.media-imdb.com/images/H/MV5BMTkxOTIxMDU2OV5BMl5BanBnXkFtZTCwNjM5NjQyMg@@_V1_SX400_.jpg",
    directors: [
      "Merian C. Cooper",
      "Ernest B. Schoedsack"
    ],
    rating: 7,
    rank: 3.5515678901234566,
    running_time_secs: 6.34
  }
}
```

year:1933, title:1234567890123456

```
{
  year: 1933,
  title: "1234567890123456",
  info: {
    actors: [
      "Fay Wray2"
    ]
  }
}
```

200 SAVE REVERT EXPORT 200 row(s) fetched - 639ms

Document selection

The document can be selected in order to display all available actions.

The screenshot displays the CloudBeaver interface with the 'Data' tab selected. A SQL filter 'year:1933,title:1' is applied. The main area shows a JSON document for a film from 1933. The left sidebar has icons for TABLE, JSON, and a pencil icon for editing. The bottom toolbar includes buttons for SAVE, REVERT, and EXPORT. The status bar at the bottom indicates '200 row(s) fetched - 639ms'.

```
year:1933,title:1

{
  year: 1933,
  title: "1",
  info: {
    actors: [
      "Fay Wray2",
      "Robert Armstrong",
      "Bruce Cabot"
    ],
    plot: "123 A film crew \n goes to a tropical island for an exotic location shoot and discovers a colossal giant gorilla who takes a shine to their female blonde star.",
    release_date: "1933-03-07T10:00:00Z",
    genres: [
      "Adventure",
      "Fantasy",
      "Horror"
    ],
    image_url: "https://ia.media-imdb.com/images/M/MV5BMTkxOTIxMDU2OV5BMl5BanBnXkFtZTcwNjM5NjQyMg@@._V1_SX400_.jpg",
    directors: [
      "Merian C. Cooper",
      "Ernest B. Schoedsack"
    ],
    rating: 7,
    rank: 3.5515678901234566,
    running_time_secs: 6.34
  }
}
```

```
year:1933,title:1234567890123456

{
  year: 1933,
  title: "1234567890123456",
  info: {
    actors: [
      "Fay Wray2".
    ]
  }
}
```

200 row(s) fetched - 639ms

Editing

To start editing, click on the *pencil icon* in the left panel. You will have your document switched to *the editing mode*. In the editing mode, you can see a toolbar with *apply* and *revert* buttons. If the document has some unsaved changes in it, it will be highlighted with an orange border.

PropertiesDataDiagram

Enter a SQL expression to filter results

TABLEJSON

year:1933,title:1

1 {
2 "year": 1933,
3 "title": "1",
4 "info": {
5 "actors": [
6 "Fay Wray2",
7 "Robert Armstrong",
8 "Bruce Cabot"
9],
10 "plot": "123 A film crew \n goes to a tropical island for an exotic location shoot and discovers a colossal giant gorilla who takes a shine to their female blonde star.",
11 "release_date": "1933-03-07T10:00:00Z",
12 "genres": [
13 "Adventure",
14 "Fantasy",
15 "Horror"
16],
17 "image_url": "https://ia.media-imdb.com/images/M/MV5BMTkxOTIxMDU2OV5BM15BanBnXkFtZTcwNjM5NjQyMg@@._V1_SX400_.jpg",
18 "directors": [
19 "Merian C. Cooper",
20 "Ernest B. Schoedsack"
21],
22 "rating": 7,
23 "rank": 3.5515678901234566,
24 "running_time_secs": 6.34
25 }
26 }

year:1933,title:1234567890123456

{
year: 1933,
title: "1234567890123456",
info: {
actors: [
"Fav Wray2",
}

200

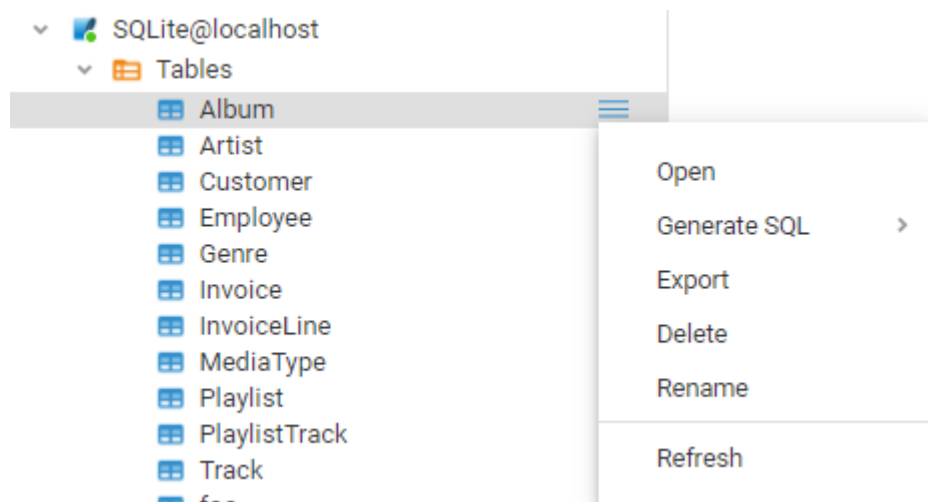
SAVEREVERTEXPORT

200 row(s) fetched - 639ms

Learn more

You can perform data export for database tables:

1. Select a table you want to export in the Database Navigator or the Metadata Editor. In the context menu, choose **Export**. You can also export data from the Data Editor and the ResultSet in the SQL Editor.



2. Choose your export format. CloudBeaver supports different output formats:

- CSV
- DBUnit
- JSON
- Markdown
- Source code
- SQL
- TXT
- XML
- XLSX
- HTML

3. Set the export configuration options. They are specific to the data format you chose.

Export configuration (CSV)



NAME	VALUE	
Characters escape	quotes	▼
Delimiter	,	
File extension	csv	
Format numbers	false	
Header	top	▼
Header format	label	▼
NULL string		
Quote always	disabled	▼
Quote character	"	

BACK

CANCEL

EXPORT

4. Press **Export**. \ Note: avoid changing data in the tables you have selected to be exported while the exporting is in progress.
5. Press **Download** in the pop up dialog.

Learn more

Table of contents

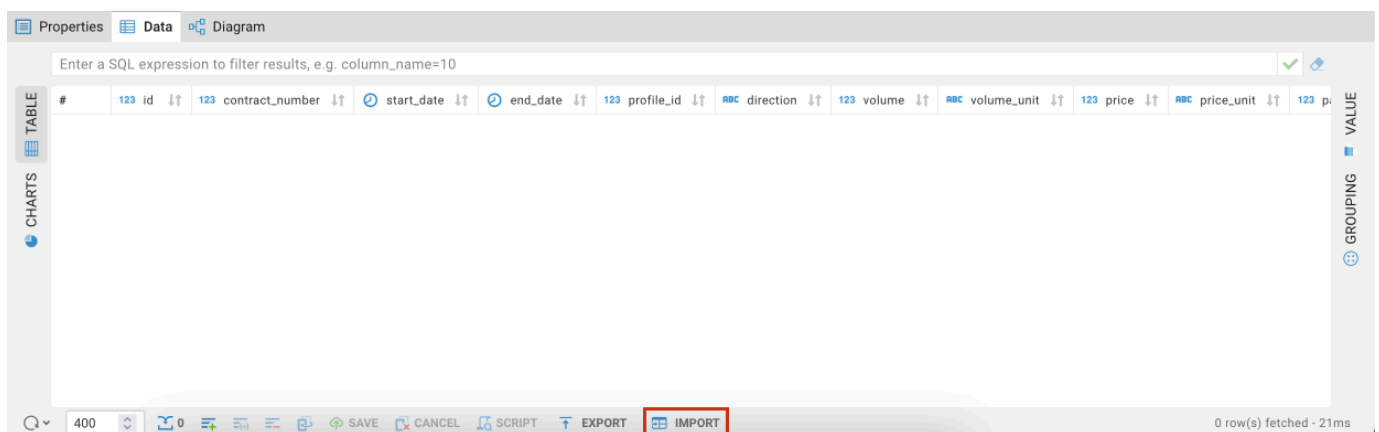
[Import process steps](#)

[Import configuration details](#)


In CloudBeaver, data import is accessible through the [Data Editor](#) when a table is open.




Import process steps

1. Click the **Import** button to open a pop-up window for the import process.




2. Choose the file type for import.


Data import
childdeal
×


 CSV	Import from CSV file(s)
 XLSX	Import from XLSX file(s)
 XML	Import from XML file(s)

CANCEL

3. Choose the file from local storage and press **Import**.

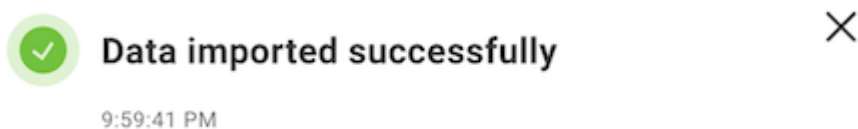

Data import (XML)
childdeal
×

NAME


UPLOAD FILE

CANCEL
BACK
IMPORT

4. Upon successful import, a confirmation message will appear.



Import configuration details

The import process in CloudBeaver includes several key features and limitations:

- The Community Edition (CE) only allows for `CSV` files. Pro versions include `CSV`, `XLSX`, and `XML` files.
- In CloudBeaver Team Edition, data import is available to users with the roles of Editor, Manager, Developer, and Administrator.
- `CSV` files should be comma-delimited.
- The structure (DDL) of the uploaded table must match the existing table, specifically in terms of columns.
- Only unique primary key values are accepted to ensure data integrity and avoid duplicates.
- The import operation does not block the interface, allowing for continued work while the import is processed.
- The system currently supports uploading one file at a time.

Learn more

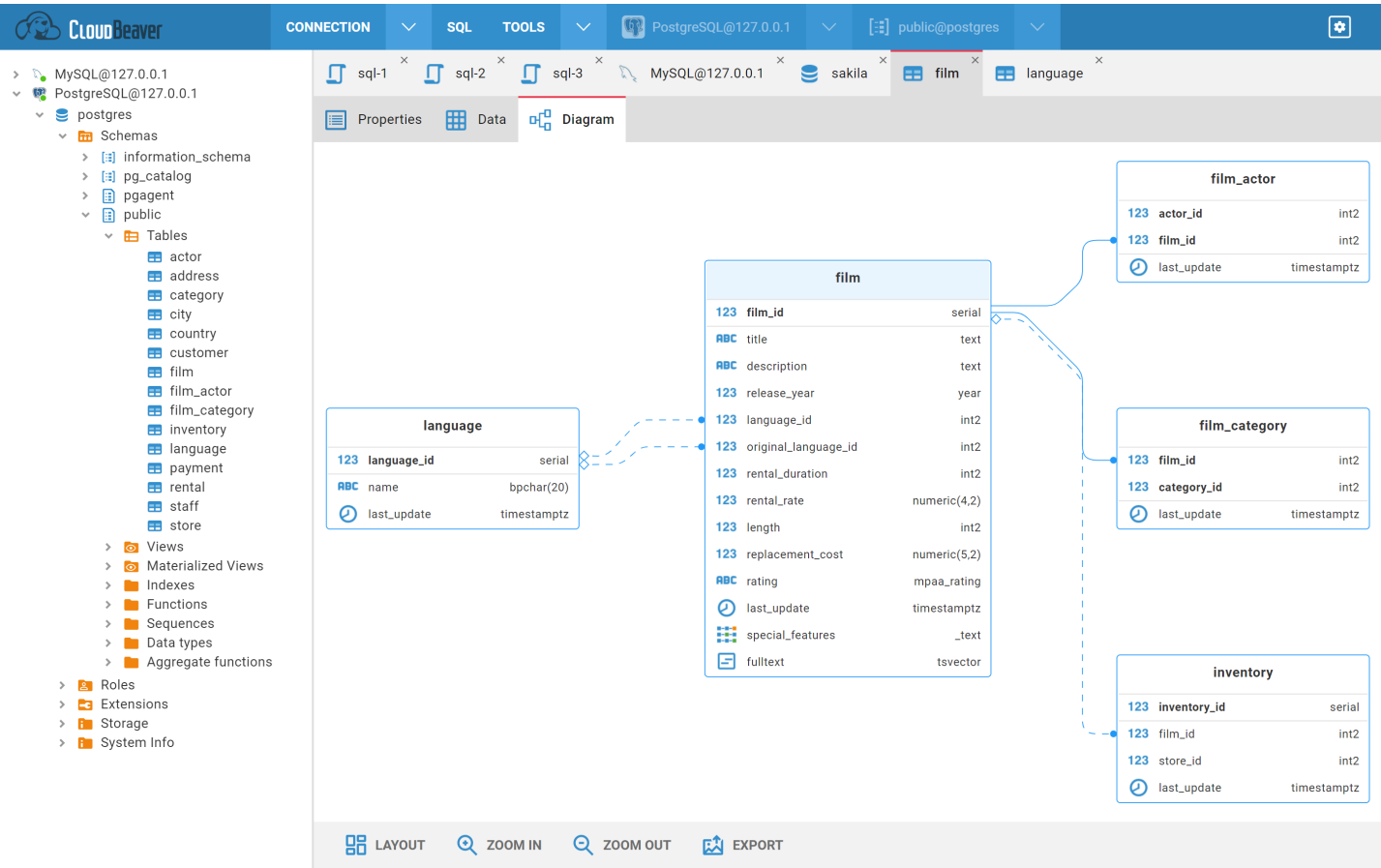
Table of contents

- [Overview](#)
- [Display entities with attributes](#)

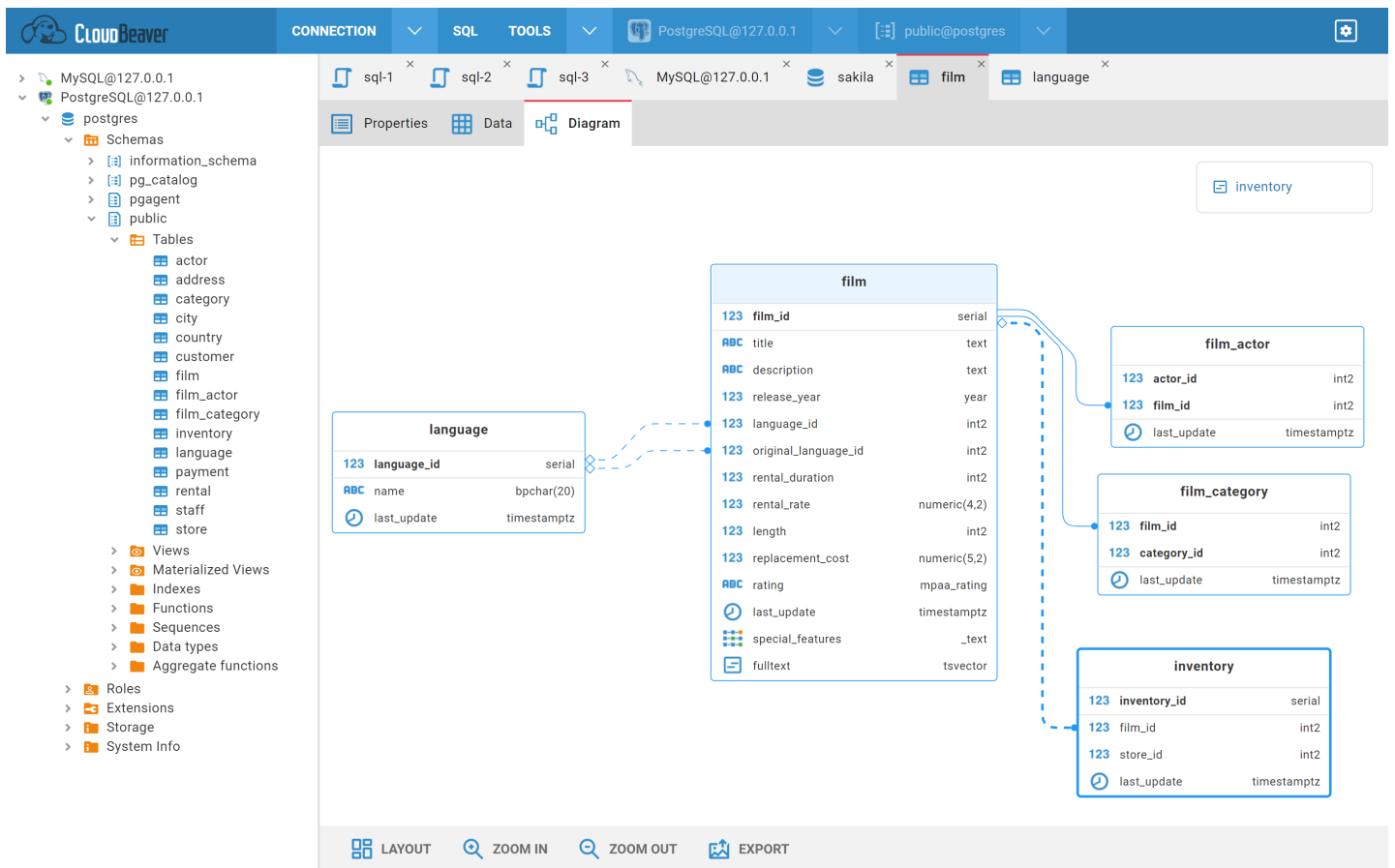
Overview

Display entities with attributes

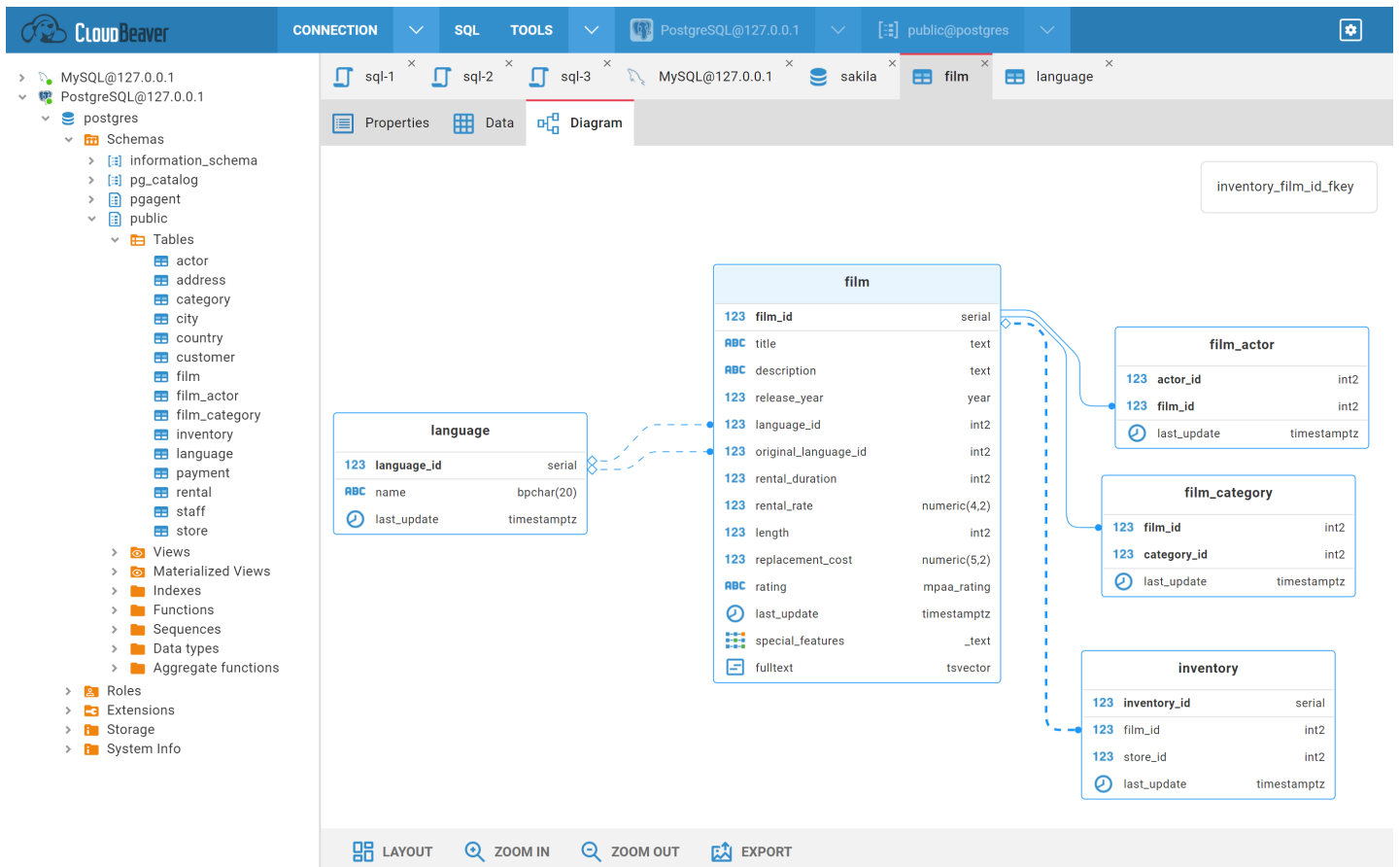
1. Navigate to your connection and open a table or schema
2. Select the "Diagram" tab (if the tab is not presented then the object does not support the diagram presentation)



You can click on an entity to highlight it:



You can click on a relation to get highlight it:

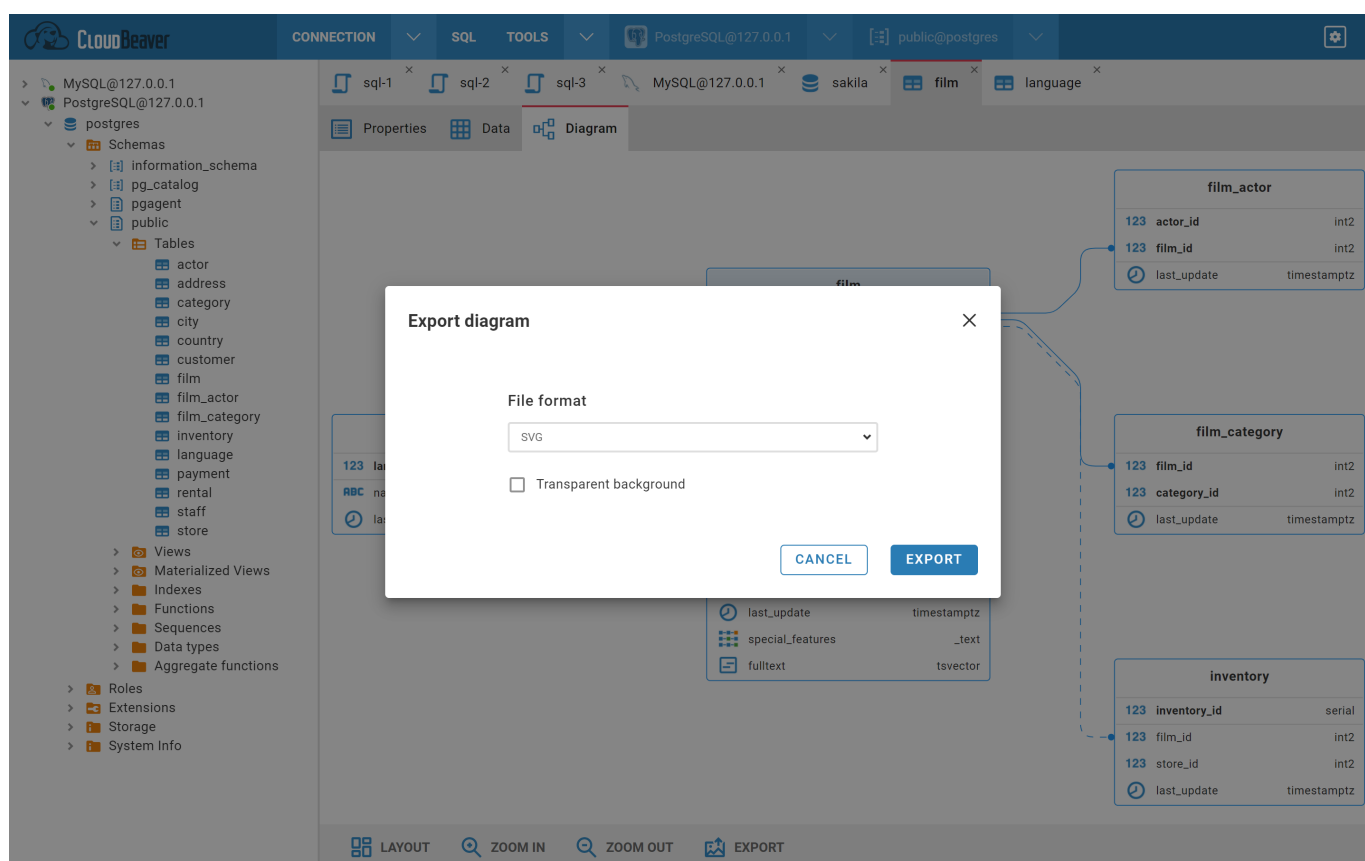


You can open the entity in the metadata editor by:

- double-clicking on the entity
- double-clicking on the entity attribute
- clicking on the link in the entity tooltip

On the bottom toolbar you can find different buttons:

1. Layout - diagram auto layout
2. Zoom in/out
3. Export - export diagram in a **png** or **svg** format



Learn more

Table of contents

- [Overview](#)
- [Shortcuts](#)
- [Statement Execution](#)
- [Script Execution](#)
- [Auto and Manual commit](#)
 - [Switching to Manual commit](#)
 - [Limitations of Rollback capabilities](#)

Overview

SQL Editor supports autocomplete, syntax highlight, statement execution, script execution, and execution plan for some databases.

The screenshot displays the SQL Editor interface. On the left, a sidebar shows a database schema for 'SQLite_del_team1' with tables like Album, Artist, Customer, Employee, foo, Genre, Invoice, InvoiceLine, MediaType, Playlist, PlaylistTrack, Track, EmpView, and test. The main area shows a SQL query: `select * from Album;`. Below the query, a 'Result - 1 <1>' tab displays a table of results. The table has columns: #, AlbumId, Title, ArtistId, and Column1. The results show 9 rows of data, including albums like 'For Those About To Rock We Salute You', 'Balls to the Wall', 'Restless and Wild', 'Let There Be Rock', 'Big Ones', 'Jagged Little Pill', 'Facelift', 'Warner 25 Anos', and 'Plays Metallica Bv Four Cellos'. The bottom status bar indicates 'Success - 24ms'.

#	AlbumId	Title	ArtistId	Column1
1	1	For Those About To Rock We Salute You	1	[BLOB]
2	2	Balls to the Wall	2	[BLOB]
3	3	Restless and Wild	2	[NULL]
4	4	Let There Be Rock	1	[BLOB]
5	5	Big Ones	3	[NULL]
6	6	Jagged Little Pill	4	[NULL]
7	7	Facelift	5	[NULL]
8	8	Warner 25 Anos	6	[NULL]
9	9	Plays Metallica Bv Four Cellos	7	[NULL]

Shortcuts

Shortcut	Description
Ctrl+Enter	Execute SQL statement
Ctrl+ or Ctrl+Shift+Enter	Execute SQL statement in new tab
Alt+X	Execute script
Shift+Ctrl+E	Show execution plan
Alt+T	Open SQL Editor in separate browser tab
Shift+Ctrl+F	Format script
Ctrl+Z or CMD+Z	Undo
Ctrl+Y or Ctrl+Shift+Z or Shift+CMD+Z or CMD+Y	Redo
Esc+Tab	Remove focus from the SQL Editor

Statement Execution

Place the cursor on the line with the statement or select part of the script to execute the statement. Click on the **Run** icon in the left toolbar or use the `Ctrl+Enter` shortcut. The result of the statement execution will be shown under the script editor area. Results will be grouped (`Result - 1 <1>` , `Result - 1 <2>`) if statement execution is finished with more than one result.

Ln 2, Col 24, Pos 44

Statistics - 1 Result - 1 <1> Result - 1 <2>

Enter a SQL expression to filter results, e.g. column_name=10

#	CustomerId	FirstName	LastName	Company	Address	City
1	1	Luis	Gonçalves	Embraer - Empresa Brasileira de Aeronáutica S.A.	Av. Brigadeiro Faria Lima, 2170	São José dos Campos
2	2	Leonie	Köhler	[NULL]	Theodor-Heuss-Straße 34	Stuttgart
3	3	François	Tremblay	[NULL]	1498 rue Bélanger	Montréal
4	4	Björn	Hansen	[NULL]	Ullevålsveien 14	Oslo
5	5	František	Wichterlová	JetBrains s.r.o.	Klanova 9/506	Prague
6	6	Helena	Holý	[NULL]	Rilská 3174/6	Prague
7	7	Astrid	Gruber	[NULL]	Rotenturmstraße 4, 1010 Innere Stadt	Vienne
8	8	Daan	Peeters	[NULL]	Grétrystraat 63	Brussels
9	9	Kara	Nielsen	[NULL]	Sander Boulevard 51	Copenhagen

Success - 31ms

Script Execution

Click on the **Script** icon in the left toolbar or use the **Alt+X** shortcut to execute the script. The summary result will be shown in the **Statistics** tab, and results will be shown in separate **Result** tabs.

Ln 2, Col 24, Pos 44

Statistics - 1 Result - 1 <1> Result - 1 <2>

Executed queries: 2 / 2
Duration: 42 ms
Updated Rows: 0





Auto and Manual commit

By default, all database connections in the SQL Editor operate in **Auto-commit** mode, meaning that changes are automatically committed after each SQL statement is executed. To gain precise control over your transactions, you may switch to **Manual commit** mode.

Switching to Manual commit

In **Manual commit** mode, you manually determine when to commit or rollback transactions. This mode is essential when batch operations need to be treated as a single unit or when you need to inspect changes before making them permanent.

Here is a table outlining the actions and their corresponding icons in **Manual commit** mode:

Action	Icon	Description
Switch to manual commit		Click to open a menu for manual transaction control. You will need to Commit or Rollback changes explicitly.
Commit		Click after executing SQL statements to save the changes to the database.
Rollback		Click to revert changes made by your SQL statements, undoing current transaction changes.
Switch to auto-commit		Click to return to Auto-commit mode, where changes are automatically committed.

Tip: After committing in **Manual commit** mode, performing a **Refresh** is necessary to see the newly added data in the Result tab.

Limitations of Rollback capabilities

In database management, not all commands support rollback operations. It is crucial to understand that Data Definition Language (DDL) commands—such as **CREATE** , **DROP** , or **ALTER** cannot be reversed with transactions for some databases. This means that once these commands are executed, they cannot be reversed even in manual commit mode.


Important: Always check the transaction support for the specific database you are working with to avoid irreversible operations.

Query Execution Plan

Table of contents

[Execution Plan](#)

Execution Plan

If a database driver supports the visualization of the execution plan, you can see the execution plan of the query by pressing `Ctrl+Shift+E` or clicking the **Explain execution plan** button  on the main toolbar. The execution plan command generates a query execution tree as one of the result tabs and is convenient in estimating if the query/script is quick/optimal enough.

You can click the rows of the execution plan to see their details (statistics) in the panel to the right of the plan.

sql-2 (DB)

1

SELECT f.title, a.first_name || ' ' || a.last_name

2

FROM film f, film_actor fa, actor a

3

WHERE f.film_id=fa.film_id AND fa.actor_id=a.actor_id

Execution plan - 1

NODE TYPE	ENTITY	COST	ROWS	TIME	CONDITION	NAME	VALUE
Hash Join		90.44 - 342.13	5363		(fa.actor_id = a.actor_id)	General	
Hash Join		78.55 - 235.84	5363		(fa.film_id = f.film_id)	Node Type	Hash Join
Seq Scan	film_actor	0.00 - 89.63	5363			Entity	
Hash		66.02 - 66.02	1002			Cost	90.44 - 342.13
Seq Scan	film	0.00 - 66.02	1002			Rows	5363
Hash		7.51 - 7.51	351			Time	
Seq Scan	actor	0.00 - 7.51	351			Condition	(fa.actor_id = a.actor_id)
Details							
Parallel-Aware							false
Join-Type							Inner
Startup-Cost							90.44
Total-Cost							342.13
Plan-Rows							5363
Plan-Width							47
Inner-Unique							true
Hash-Cond							(fa.actor_id = a.actor_id)

SELECT f.title, a.first_name || " " || a.last_name
FROM film f, film_actor fa, actor a
WHERE f.film_id=fa.film_id AND fa.actor_id=a.actor_id

Learn more

Table of contents

[Overview](#)

[Creating a Visual Query](#)

[Filtering](#)

[Sorting](#)

[Executing a Visual Query](#)

[Shortcuts](#)

[The Visual Query Builder symbols](#)

[Table symbols](#)

[Join symbols](#)

[Settings](#)

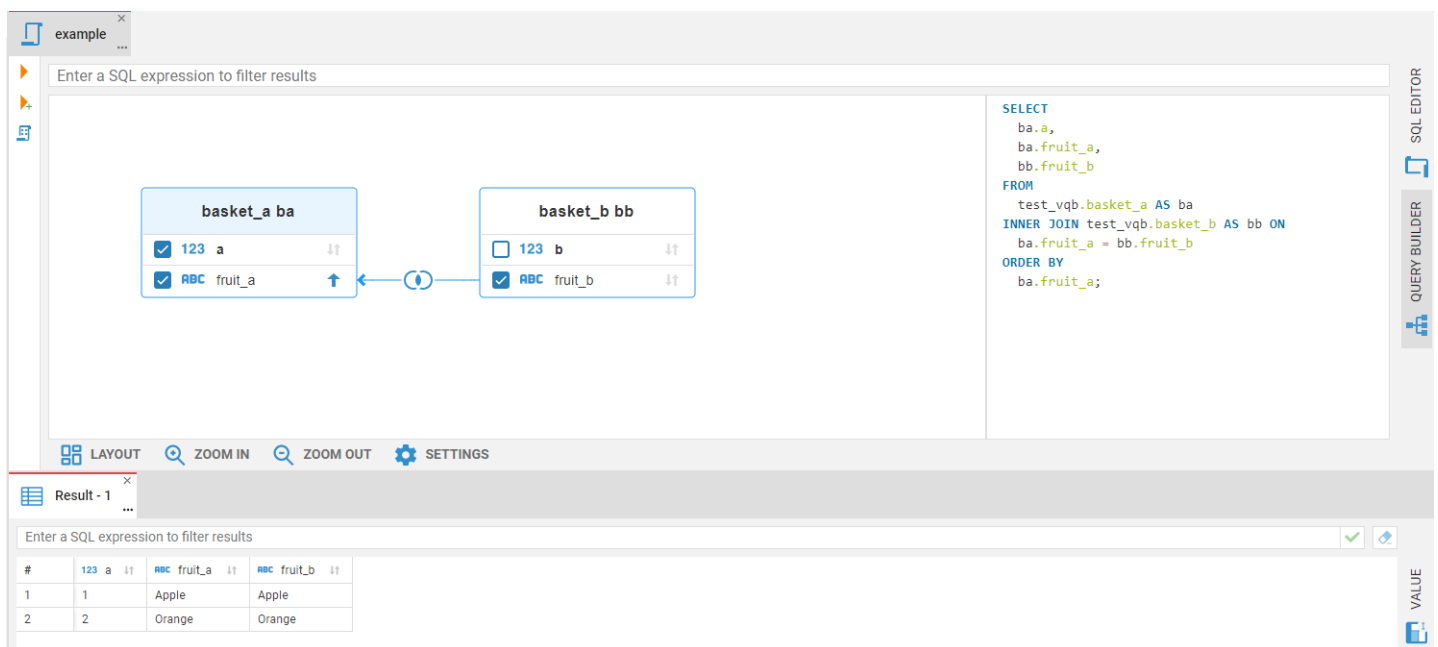
[Visualization of an existing SQL query](#)

Note: This feature is available in Enterprise and Enterprise for AWS editions only.

Overview

The Visual Query Builder is a user-friendly visualization tool that can help you to create queries to the database and see results. You do not need to know SQL language to work in it. The Visual Query Builder may be useful for:

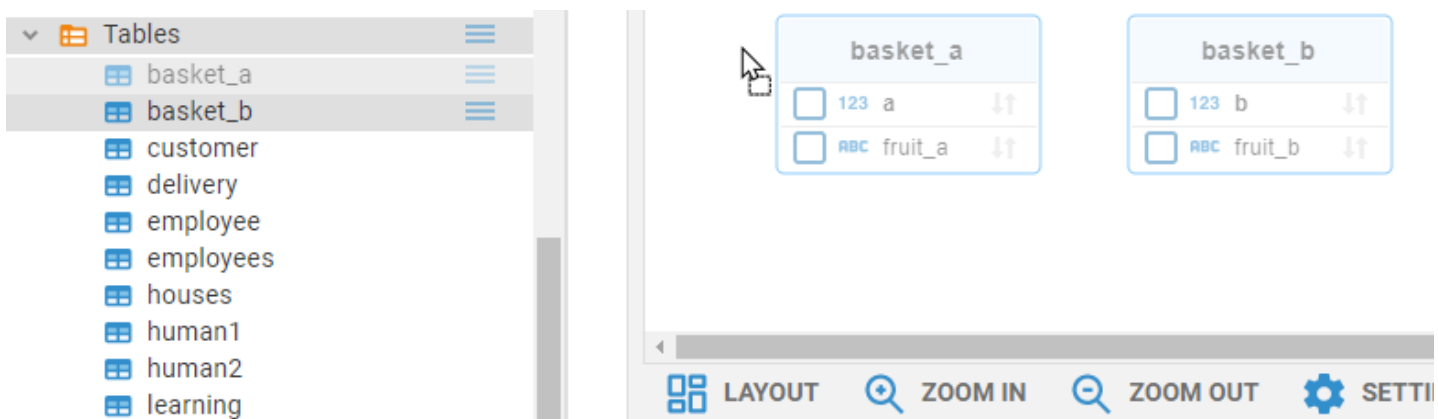
- building queries;
- complex queries analysis;
- easy query editing.



To open the Visual Query Builder, click the Query Builder tab in the SQL Editor right toolbar.

Creating a Visual Query

1. Select tables in the Navigator tree and drag-and-drop them into the Visual Query Builder area. The existing connections between the tables will automatically be displayed. The tables will also be added to the SQL expression which can be found in the field to the right of the diagram.



2. To create a new join between tables, connect their columns holding the left mouse button. The connection between the selected columns of the tables will appear in the diagram and the Inner Join will be added to the SQL script.



3. You can change a join type clicking the join label on the connection line.
4. To remove a join between tables, click on the line, then press the Delete button. The connection will be removed from the diagram and the join will disappear from the SQL script.
5. By default all tables' columns are included in the query. If you only want to see certain columns in your query result, select the checkbox near the column name.



Filtering

6. WHERE condition with the filter value is used for filtering. To add a filter, write it in the top filter field.

Column name	Operation Sign	Value
A table column name. You have to write a table alias before if another column has the same name	The most common signs: =, >, <, <>, LIKE, ILIKE, BETWEEN	A column value, used as a parameter. Text and time values must be rounded by single quotes, numeric values do not need any quotes

Filter example:

e.employeeid>2

customer c

123 customerid

ABC name

payment p

123 paymentid

123 amount

123 customerid

123 employeeid

employees e

123 employeeid

ABC name

```

SELECT
*
FROM
test_vqb.customer AS c
INNER JOIN test_vqb.payment AS p ON
c.customerid = p.customerid
INNER JOIN test_vqb.employees AS e ON
p.employeeid = e.employeeid
WHERE
e.employeeid>2;

```

LAYOUT

ZOOM IN

ZOOM OUT

SETTINGS

Result - 1

...

Enter a SQL expression to filter results



#	123 customerid	ABC name	123 paymentid	123 customerid	123 employeeid	123 amount	123 employeeid	ABC name
1	4	Abid	3	4	3	3000	3	Hassan
2	5	Sia	4	5	4	4000	4	Anna
3	3	Hussain	5	3	5	4000	5	Sau
4	6	Kait	6	6	6	5000	6	Kelsie
5	7	Tony	7	7	7	5000	7	Tory
6	8	Sam	8	8	8	5000	8	Salley

Sorting

- To apply a sorting condition to a column, press the sorting icon next to a column name on the diagram. The column will be sorted in ascending order and the conditional expression ORDER BY will be added to the SQL script. To sort the column in descending order, press the sorting icon again to select the down arrow. If you want to remove a condition, continue to click the sorting icon to deactivate it. Sorting can be applied to multiple columns in different tables. First, apply sorting on the first column you wish to sort, and then on the second, third and so on. You can sort numbers, texts, dates, time and other values.



Executing a Visual Query

Use the Execute SQL statement button  on the left pane to execute a query and get the results in the same tab. If you want to see the result in a new tab, press the Execute SQL statement in a new tab button .

Shortcuts






You can use the same shortcuts as in the SQL Editor to execute the Visual Query.

Key	Description
Ctrl+Enter	Execute the SQL statement
Ctrl+\ or Ctrl+Shift+Enter	Execute the SQL statement in a new tab

The Visual Query Builder symbols




The Visual Query Builder uses the following visual tools to display queries on the diagram:

Table symbols

Symbol	Description
	Table Primary Key is bold and displayed at the top of the table.
	Table Alias is used to shorten your Join Statement.
	Colored table header marks the first table in your Join Statement.
	Colorless header marks a joined table in your Join Statement.
	Line goes from the joined table to the first table.

Join symbols

Available Join types are described in the table below. The Visual Query Builder can show results only for those types of Joins that are supported by your database.

Symbol	Description
	Inner Join
	Left Join
	Left Outer Join

	Right Join
	Right Outer Join
	Full Join
	Full Outer Join
	Cross Join

Settings

You can customize the diagram view using the bottom toolbar to make the work with the diagram easier.



- **Layout** updates the diagram view to display all of its objects in the most optimal way.
- **Zoom in** and **Zoom out** enlarges or shrinks the diagram view.
- **Settings** menu contains additional settings of the Visual Query Builder. Press the Settings button at the bottom toolbar to open it.
 - **Layout on update** enables Auto-layout feature. As soon as you add a new object to the diagram, the diagram view will automatically be updated to display all of its objects in the most optimal way.
 - **Show join type on entities** moves Join labels from lines into headers of joined tables.
 - **Show Type** adds information about column types into entities.
 - **Show Icons** adds icons of column types into entities.
 - **Notation** changes the representation of connection lines. Simple notation is set by default. You can change it to the IDEF1X language type.

Visualization of an existing SQL query

If you write a JOIN statement by yourself and then want to convert it to the diagram view, just switch the SQL Editor with your statement to the Visual Query Builder.

Note: the Visual Query Builder can transform the syntax of your query, but it does not affect the query result in the Result set.

Learn more

Table of contents

[Understanding the AI integration in CloudBeaver](#)

[Initial setup](#)

[Data privacy](#)

[AI settings and customization](#)

[Credentials for OpenAI](#)

[Credentials for Azure AI](#)

[Preferences](#)

[AI smart completion usage](#)

[Accessing prompts history](#)

[Disabling AI features](#)

[Best practices for question formulation](#)

CloudBeaver offers the ability to construct SQL queries using natural language through **AI smart completion** feature. This capability is achieved through integrations with both OpenAI's [GPT-3 language model](#) and [Azure OpenAI](#).

Note: CloudBeaver is not affiliated with OpenAI. Integration is achieved through the public API.

- To utilize this feature, register with OpenAI and obtain a secret key.

Understanding the AI integration in CloudBeaver

With the **AI smart completion** feature, you can type queries in natural language and CloudBeaver will convert them into SQL statements. This tool simplifies writing complex queries by interpreting your input and automatically generating the correct SQL code.

Initial setup

To activate the AI features in CloudBeaver, configure the API token:

1. Navigate to **Administration page** -> **Server Configuration tab**
2. Ensure the **AI Service** option is activated.
3. Navigate to **AI Settings tab** -> **Choose an engine**
4. In the **API token** field, input your AI secret key.
5. Save the changes.

For instructions on utilizing the AI features, visit the [AI Smart completion usage](#).

The screenshot shows the CloudBeaver interface with the 'AI Settings' tab selected in the left sidebar. The main content area is divided into two panels: 'AI SETTINGS' and 'ENGINE SETTINGS'. The 'AI SETTINGS' panel includes a note about metadata transfer to OpenAI and a dropdown menu for selecting an engine, with 'OpenAI' currently selected. The 'ENGINE SETTINGS' panel contains fields for 'API token' (masked with dots), 'Model' (set to 'gpt-3.5-turbo-16k'), and 'Temperature' (set to '0.0'). There is also a checkbox labeled 'Write GPT queries to debug log' which is checked. At the top of the configuration area, there are 'SAVE' and 'CANCEL' buttons.

Data privacy

We prioritize data safety and user privacy. In this section, we outline how data is managed and the measures taken to protect user privacy when using the AI features.

To enable the AI features capabilities, table and column names from the current database schema are transmitted to OpenAI. This step is crucial for accurately translating user requests into SQL queries. Key considerations regarding data privacy are as follows:

- **No Table Data:** Only metadata like table and column names are shared with OpenAI. Actual table data is not transmitted.
- **Log Transparency:** The entire request can be logged for your review. To enable this, navigate to **AI Settings tab** and check the **Write GPT queries to debug log** option.

- **Azure OpenAI privacy:** If you use Azure OpenAI, be aware that it operates under its own [privacy policy](#). It's recommended to review their terms before using.

AI settings and customization

To utilize the AI-enhanced functionalities within CloudBeaver, certain configurations and setup processes are required. This section offers a comprehensive guide on initial setup and customization options to tailor the AI integration according to specific preferences.

Credentials for OpenAI

1. Sign up on the [OpenAI platform](#).
2. Navigate to the [API Keys section](#) and generate a new secret key.
3. Insert this key into CloudBeaver's **Engine Settings**.

Credentials for Azure AI

1. Sign up on the [Azure platform](#).
2. Navigate to the [Azure Portal](#) and create a new AI service under the AI + Machine Learning section.
3. Generate and copy the credentials for the newly created service.
4. Insert these credentials into ClouBeaver's **Engine Settings**.

Preferences

For specific requirements or troubleshooting, you might want to adjust some of the following settings:

- Navigate to **Administration page -> AI Settings -> Engine settings** to access these settings.

Setting	Description
API token	Input your secret key from the OpenAI platform.

Model	Choose the AI model (recommended: gpt-3.5-turbo for SQL).
Temperature	Control AI's creativity from 0.0 (more precise) to 0.9 (more diverse). Note that higher temperature can lead to less predictable results.
Write GPT queries to debug log	Logs your AI requests.

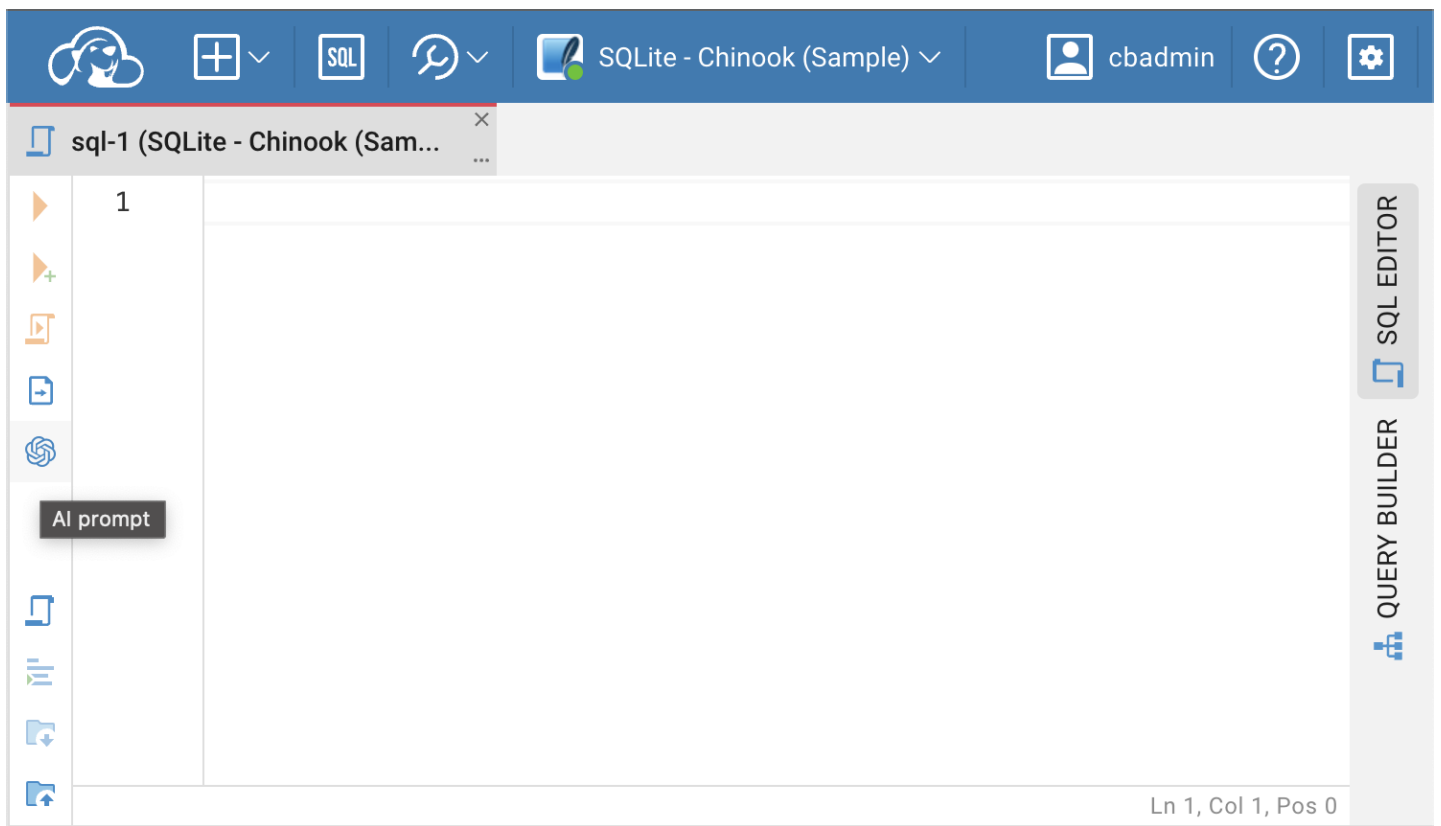
There is also an option to switch the **Engine** from OpenAI to Azure OpenAI. Azure provides a set of distinct settings:

Setting	Description
Endpoint	Configure a custom endpoint URL for Azure OpenAPI interactions.
API version	Select the version of the API you wish to use.
Deployment	Specify the deployment name chosen during model deployment.
Context size	Choose the context size between 2048 and 32768 . A larger number allows the AI to use more data for better answers but may slow down response time. Choose based on your balance of accuracy and speed.

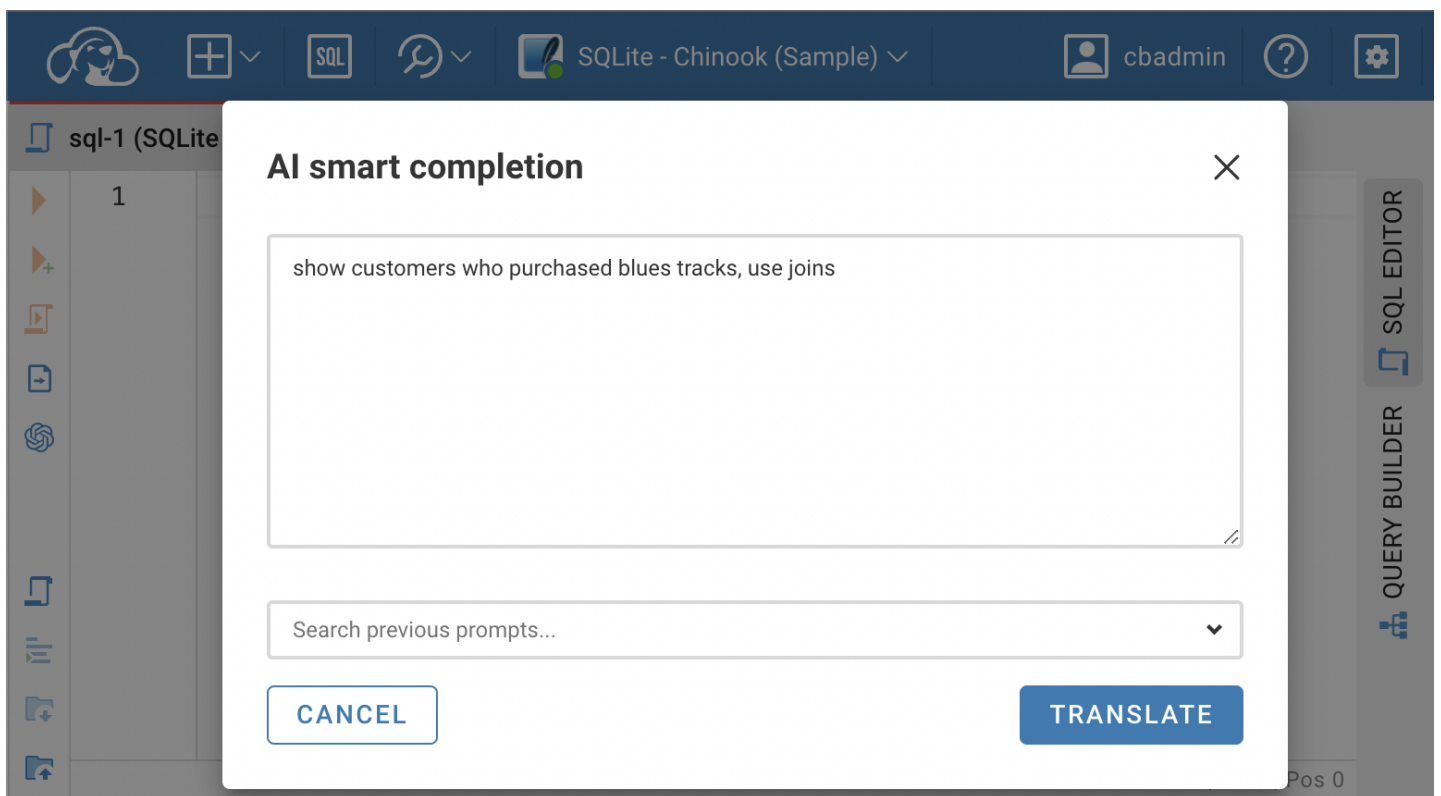
AI smart completion usage

To interact with databases using the **AI Smart completion** feature:

1. Launch the **SQL Editor**.
2. Click on the **AI smart completion** icon  located in the left toolbar of the **SQL Editor**.



3. Input your natural language request in the **AI smart completion** window.
4. Click **Translate** to obtain the SQL query.



Accessing prompts history

Prompts history allows you to review previous prompts in the scope of the session for the chosen tab.

Disabling AI features

To hide the **AI smart completion** icon in the SQL Editor:

- Navigate to **Administration page -> Server Configuration tab -> Services section**.
- Deselect **AI option**.

Best practices for question formulation

When using AI to generate SQL queries, it's essential to provide clear and specific input. Here's how to optimize your questions:

- **Language:** While AI supports multiple languages, it's recommended to use English for best results.
- **Database knowledge:** Familiarity with your database structure enhances the accuracy of generated queries.
- **Explicit details:** If you know certain tables or columns that should be part of the query, include them in your request for better accuracy.

For instance, if you're using the CloudBeaver sample SQLite database, you might phrase your request as:

Example 1: "List all customers from Italy"

Resulting SQL:

```
SELECT *  
FROM customers  
WHERE country = 'Italy';
```

Example 2: "montre les clients de France"

```
SELECT *  
FROM customer  
WHERE country = 'France';
```

Example 3: "show customers who purchased blues tracks, use joins"

```
SELECT c.FirstName, c.LastName, t.Name, g.Name
FROM Customer c
      JOIN Invoice i ON c.CustomerId = i.CustomerId
      JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
      JOIN Track t ON il.TrackId = t.TrackId
      JOIN Genre g ON t.GenreId = g.GenreId

WHERE g.Name = 'Blues'
ORDER BY c.LastName, c.FirstName;
```

Example 4: "get names of customers who purchased blues tracks, use joins"

```
SELECT DISTINCT c.FirstName, c.LastName
FROM Customer c
      JOIN Invoice i ON c.CustomerId = i.CustomerId
      JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
      JOIN Track t ON il.TrackId = t.TrackId
      JOIN Genre g ON t.GenreId = g.GenreId
WHERE g.Name = 'Blues';
```

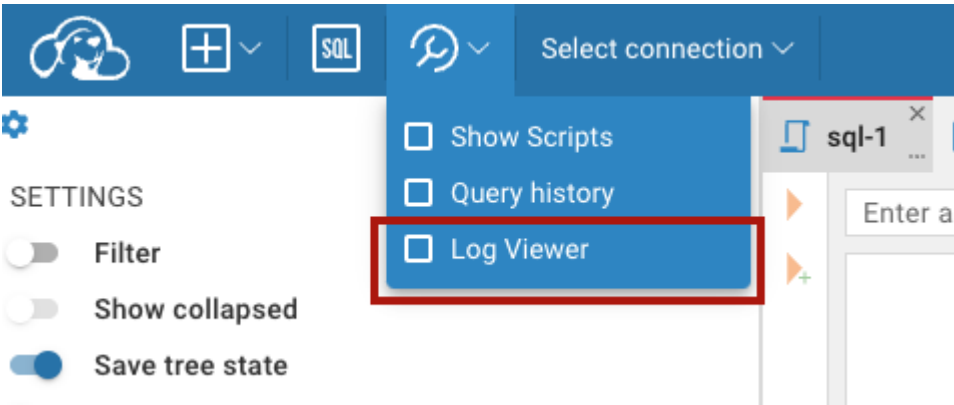
Log Viewer

Table of contents

[Log Viewer](#)

Log Viewer

The Log Viewer shows user logs during a session. It can be opened via the Tools menu on the main page of the application.



The Log Viewer contains the logs' times and descriptions. If you want to see the full information about an error, click the error link. The error details will be displayed in a separate panel on the right.

LOG VIEWER

CLEAR LOGS

TIMESTAMP	MESSAGE
2022-04-12T08:30:41Z	Error executing query: SQL Error [1]: [SQLITE_ERROR] SQL error or missing database (near "some": syntax erro...
2022-04-12T08:30:34Z	Load TableCache
2022-04-12T08:30:23Z	Load Data Types
2022-04-12T08:30:23Z	Load Table Triggers
2022-04-12T08:30:23Z	Load Sequences
2022-04-12T08:30:23Z	Load Indexes
2022-04-12T08:30:23Z	Load Views
2022-04-12T08:30:23Z	Load Tables

COPY

CLOSE

ERROR 2022-04-12T08:30:41Z

Error executing query:
SQL Error [1]: [SQLITE_ERROR] SQL error or missing database (near "some": syntax error)

io.cloudbeaver.DBWebException: Error executing query:
SQL Error [1]: [SQLITE_ERROR] SQL error or missing database (near "some": syntax error)
at io.cloudbeaver.service.sql.WebSQLProcessor.processQuery(WebSQLProcessor.java:199)
at io.cloudbeaver.service.sql.impl.WebServiceSQL\$1.run(WebServiceSQL.java:364)
at io.cloudbeaver.model.session.WebSession\$1.run(WebSession.java:614)
at org.jkiss.dbeaver.model.runtime.AbstractJob.run(AbstractJob.java:105)
at org.eclipse.core.internal.jobs.Worker.run(Worker.java:63)
Caused by: org.jkiss.dbeaver.model.sql.DBSQLException: SQL Error [1]: [SQLITE_ERROR] SQL error or missing database (near "some": syntax error)

Query History

Table of contents

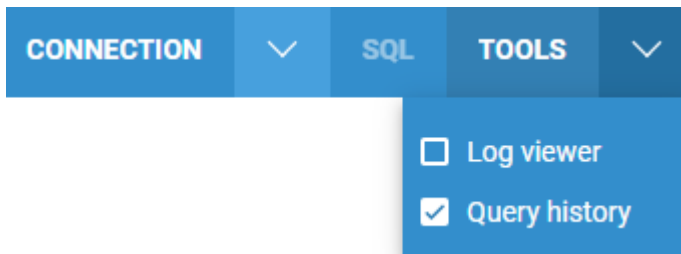
[Query History](#)

[Query History Options](#)

Query History

CloudBeaver EE persists all executed queries in the internal database. A user can see all his own queries in the Query History.

To open the Query History select it in the Tools menu on the main page.



Queries are displayed in the table which contains execution statistics (execution time, duration, number of updated rows, errors, etc). The number of displayed queries cannot exceed 2000 rows. The query history is updated automatically every 5 seconds by default.

CloudBeaver

CONNECTION

SQL

TOOLS

test_user

PostgreSQL

There are no objects to show. Double click on an object in the navigation tree to open it.

QUERY HISTORY

Type query part to search in query history

TIME	TYPE	TEXT	DURATION	ROWS	RESULT	CONNECTION	CONTEXT
26.04.2022, 07:47:20		Connected to "PostgreSQL «postgres»"	0		Success	PostgreSQL	Metadata «postgres»
26.04.2022, 07:47:20		Connected to "PostgreSQL «postgres»"	0		Success	PostgreSQL	Main «postgres»
20.04.2022, 17:17:01	SQL / User	SELECT Name, Title FROM artist LEFT JOIN album ON artist.ArtistId = ...	34	71	Success	SQLite@localhost	Main
20.04.2022, 17:17:01		Connected to "SQLite@localhost"	0		Success	SQLite@localhost	Main
20.04.2022, 16:02:05	SQL / User filtered	SELECT * FROM Album ORDER BY ArtistId	18	200	Success	SQLite@localhost	Main
20.04.2022, 16:02:02	SQL / User	SELECT * FROM Album	42	200	Success	SQLite@localhost	Main
20.04.2022, 16:01:51	SQL / User	drop table test_test	3	0	[SQLITE_ERROR] SQL error or mis...	SQLite@localhost	Main
20.04.2022, 16:01:34	SQL / User	select * from Artist	40	200	Success	SQLite@localhost	Main

If you want to find a query by text, use the Search field above the table.

Query History Options

Press the filter button on the right of the Search field to find more Query History options:

1. Filter conditions can be configured:

- by a query type and an object type;
- by a query status;
- by a date;
- by a driver.

FILTER BY DRIVERS

Search for the driver...

2. Queries can be sorted in a different order:

- by a driver;
- by a query text.

SORTING

Sorting by

☒ Desc

3. The number of loaded queries per one time can be changed in the Row Count field. Once you scroll to the last query of the current portion, the next portion (next N queries) is loaded. Setting the maximum number of rows (2000 queries) can slow down the application.

4. Auto-refresh of the Query History can be turned off in the Query History Options or by pressing the auto-refresh button on the right of the Search field. The additional settings also allow you to:

- set a custom auto-refresh interval;
- stop the auto-refresh mode if new queries aren't received due to an error.

AUTO REFRESH

☐ Enabled

☐ Stop on error

Interval (seconds)

5. To return to the default settings, press the Restore Defaults button at the bottom part of the filter dialog.

Resource Manager

Table of contents

[Overview](#)

[Opening the Resource Manager](#)

[Saving a script](#)

[Opening a script](#)

[Deleting a script](#)

[Script size limit](#)

Overview

The Resource Manager allows users to store and manage scripts in the CloudBeaver server file system. Scripts can be saved in both projects, the Private or the Shared, and available to every user who has access to this project.

Each user has the Private project and can create, edit and delete scripts in it. Other users do not have access to this project.

Administrators can also create and manage scripts in the Shared project. All users with access to this project can see, open and run the project scripts.

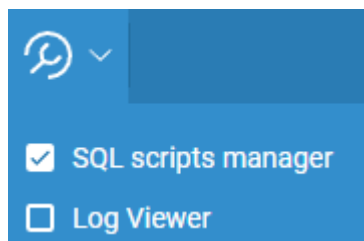
Scripts can be associated with a specific connection. When a user opens such a script, the specified connection is displayed in the SQL Editor top toolbar and the user can immediately execute the script for the connection. **Note:** scripts can only be associated with connections from the same project.

The Resource Manager is only available to logged users. Anonymous users may save scripts on a local machine.

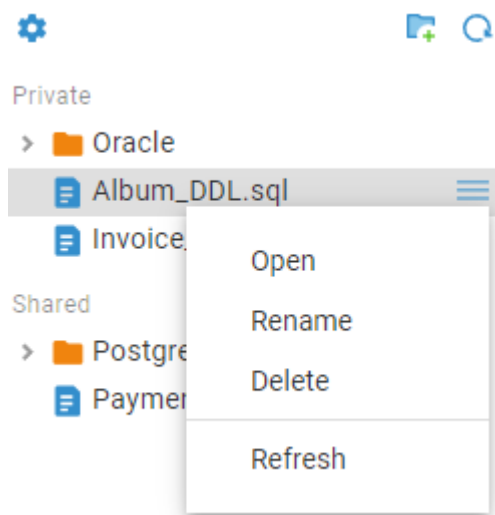
Administrators can deactivate the Resource Manager in the Server Settings in the Administration. The Resource Manager panel disappears from the CloudBeaver interface and users don't have access to scripts on the server.

Opening the Resource Manager


To open the Resource Manager select it in the Tools menu.



The Resource Manager panel will appear to the right of the SQL Editor. The panel contains the list of saved user's scripts in ascending order. Every script has the context menu with tools.



Saving a script

To save an SQL script to the Resource Manager, press the Save button  on the left SQL Editor toolbar:

For users: the script will be saved to the Private project in the CloudBeaver server.

For administrators:

- if a connection for the script is not selected, the script will be saved to the project selected by the administrator in the pop up dialog box.

Save script ×

Name:

Project

Private

Private

Shared

CANCEL

SAVE

- if a connection for the script is selected, the script will be saved to the project where the connection is stored.

The script name will appear in the Resource Manager panel under the project name.

Once a script is saved in the Resource Manager, all future changes will automatically be saved in it:

- each time any changes have been made in it
- every time when the SQL Editor with the script has been closed.

Specifying a connection for a script If necessary, when you want to specify a connection, just select a connection and a schema in the SQL Editor top toolbar for the script. The script will use this connection the next time the user opens it. To change the connection, select another one in the SQL Editor top toolbar. The changes will be saved automatically. To remove the connection, disconnect from the database in the Navigator tree and do not restore the connection for the script.

Opening a script

To open a saved script, double click the script name in the Resource Manager panel or select Open in the script context menu.

Open
Rename
Delete

Refresh

The script will be opened in a separate SQL Editor and its name will be displayed on the SQL Editor tab. If a connection has not been specified for the script, users will need to select a connection name in the SQL Editor top toolbar before execution.

Deleting a script

To delete a script from the Resource Manager, select Delete in the script context menu in the Resource Manager panel. The script will disappear from the panel and will be deleted from the CloudBeaver server. The SQL Editor with the script will also be closed if it has been opened. The deleted script cannot be restored. Scripts from the Shared project can only be deleted by the administrators.

Script size limit

You can configure the size limit for scripts in the Resource Manager. This can be useful if you need to control the amount of space on the CloudBeaver server. The limit also helps to improve the application performance, because opening big scripts can take quite a long time. If a user is saving a large script, they will see a message that the file size has been exceeded and the script will not be saved. [This article describes how to configure the limit.](#)

Installation

Table of contents

[Docker repositories](#)

[Installation](#)

[Running](#)

[Daemon mode](#)

[Accessing databases on the localhost](#)

[Docker parameters explanation](#)

[Run Cloudbeaver server with non-root user](#)

[Offline install](#)

[How to change the base docker image](#)

Docker repositories

CloudBeaver container images are on DockerHub:

Product	Docker repository	URL
CloudBeaver EE	dbeaver/cloudbeaver-ee	https://hub.docker.com/r/dbeaver/cloudbeaver-ee
CloudBeaver AWS	dbeaver/cloudbeaver-aws	https://hub.docker.com/r/dbeaver/cloudbeaver-aws
CloudBeaver Community	dbeaver/cloudbeaver	https://hub.docker.com/r/dbeaver/cloudbeaver

Each image has following tags:

Tag	Description
latest	The latest stable product release
22.1.2, 23.0.1, etc	Exact product version

ea	Early Access version
devel	Development version, unstable

Examples:

- `dbeaver/cloudbeaver-ee:23.0.0` - CloudBeaver EE version 23.0
- `dbeaver/cloudbeaver-aws:ea` - CloudBeaver AWS Early Access version
- `dbeaver/cloudbeaver:latest` - latest community release

Notes:

- We will use repository `cloudbeaver-ee` as an example in the following instructions. Replace it with proper product repository (see above).
- To run docker commands your user must be in proper user group or run it as root (e.g. `sudo docker ps`).

Installation

To install the latest version of CloudBeaver use the following script:

```
docker pull dbeaver/cloudbeaver-ee:latest
```

Running

To run cloudbaver in the terminal:

```
docker run --name cloudbeaver-ee --rm -ti -p 8080:8978 -v /opt/cloudbeaver/workspace dbeaver/cloudbeaver-ee
```

Then switch to the browser and open <http://localhost:8080/>

Daemon mode

Add the following parameters:

```
-d --restart unless-stopped
```

Accessing databases on the localhost

If you need to access the database server on the host machine, add the following parameter in docker run: (on Linux only)

```
--network host
```

Cloudbeaver will work in the host machine network.

If this mode is not suitable for your network environment then you can run the container in the following way:

```
export CB_LOCAL_HOST_ADDR=$(ifconfig | grep -E "([0-9]{1,3}\.){3}[0-9]{1,3}" | grep -v 127.0.0.1 | awk '{
docker run --name cloudbeaver --rm -ti -p 8080:8978 --add-host=host.docker.internal:${CB_LOCAL_HOST_ADDR}
```

or just run script `deploy/docker/run-docker-container.sh`. It passes the IP address of host machine to the container.

Docker parameters explanation

Parameters explanation:

Parameter	Explanation
--name cloudbeaver	Assign container ID (<code>cloudbeaver</code>)
--rm	Removes container on stop
-ti	Enables terminal mode (allows to stop container with <code>CTRL+C</code>)
-p 8080:8978	Maps CloudBeaver public port (8978) to the host machine port (e.g. 8080)
-v /opt/cloudbeaver /workspace	Mounts local folder <code>/var/cloudbeaver/workspace</code> to the server workspace as Docker volume. Required to keep CloudBeaver data after container restart. For Example: <pre>-v /var/cloudbeaver/workspace:/opt/cloudbeaver/workspace</pre> <pre>-v \$HOME/cloudbeaver/workspace:/opt/cloudbeaver/workspace</pre>
--add-host=host. docker.internal:IP address	Adds host name in the container's /etc/hosts file. This may be needed to access the database server deployed on the host machine.
dbeaver/cloudbeaver- ee:latest	Container ID

Run Cloudbeaver server with non-root user

If you want to run CloudBeaver with a non-root user, you have to build your own image with a user inside before the container starts.

Create **Dockerfile** which contains:

```
FROM dbeaver/cloudbeaver-ee:latest
RUN groupadd cloudbeaver
RUN useradd -ms /bin/bash -g cloudbeaver cloudbeaver
RUN chown -R cloudbeaver ./
USER cloudbeaver
```

Run this command to build the image from **Dockerfile**

```
docker build -t my-cloudbeaver .
```

To run CloudBeaver in the terminal:

```
docker run --name cloudbeaver --rm -ti -p 8080:8978 -v /opt/cloudbeaver/workspace my-cloudbeaver
```

Offline install

On a host with no internet access you need to download and archive image:

Note: `<TAG>` is a tag name for docker image (see above). `latest` is the default.

```
docker pull dbeaver/cloudbeaver-ee:<TAG>
docker save dbeaver/cloudbeaver-ee:<TAG> | gzip > cloudbeaver-ee.latest.tar.gz
```

Check that the archive exist:

```
ls -lah
```

Output should looks like:

```
-rw-r--r-- 1 user users 444M may 5 17:32 cloudbeaver-ee.latest.tar.gz
```

Now copy file `cloudbeaver-ee.latest.tar.gz` to some external drive and put to server with running cloudbeaver server.

Load image from archive:

```
docker load < cloudbeaver-ee.latest.tar.gz
```

You will see next output


```
Loaded image: dbeaver/cloudbeaver-ee:<TAG>
```

Upgrade your cloudbeaver-ee server:

```
docker stop cloudbeaver-ee
docker rm cloudbeaver-ee
docker run -d --restart unless-stopped -p 8978:8978 -v /opt/cloudbeaver/workspace dbeaver/cloudbeaver-ee:<TAG>
```

Note: some of docker args may differ from your environment.

How to change the base docker image

Create a new Dockerfile with the following content:

```
FROM alpine:latest

RUN apk update && apk add bash

ENV JAVA_HOME=/opt/java/openjdk
COPY --from=eclipse-temurin:17-alpine $JAVA_HOME $JAVA_HOME
ENV PATH="${JAVA_HOME}/bin:${PATH}"

ENV CLOUDBEAVER_HOME="/opt/cloudbeaver"
COPY --from=dbeaver/cloudbeaver:latest $CLOUDBEAVER_HOME $CLOUDBEAVER_HOME

WORKDIR "/opt/cloudbeaver"

RUN chmod +x run-server.sh

ENTRYPOINT ["/run-server.sh"]
```

The dockerfile above creates an image of the latest CloudBeaver CE based on Alpine.

- To change the OS:
 - Replace the base image on the first line.
 - Adapt the `RUN apk update && apk add bash` line to work with the package manager of your OS.
 - If you're changing the base OS from `musl`-based (like Alpine) to `glibc`-based (like Debian or Fedora), change the tag for `eclipse-temurin` from `17-alpine` to `17`.
- To change CloudBeaver edition or version, change the `dbeaver/cloudbeaver:latest` value to a more appropriate one.

Build a new image with:

```
docker build -t cloudbeaver-<edition>:<your_tag> .
```

Now, you can push the new image to your registry or run the CloudBeaver server.

Version upgrade

Table of contents

[Pull new server version](#)

To upgrade the server, you need to pull a new image from Docker Hub and restart the server.

Before upgrading, ensure you have a [backup](#). In some cases, you won't be able to downgrade if something goes wrong.

Pull new server version

```
docker pull dbeaver/cloudbeaver-ee:${version.number}
docker restart ${containerId}
```

By default, the CloudBeaver EE container has the name `cloudbeaver-ee` (for the AWS server, it is `cloudbeaver-aws`).

If you use a non-default run configuration, run `docker ps` to see all running containers and find your container (it is `dbeaver/cloudbeaver-ee:tag`).

To upgrade to the latest version, use `latest` as `${version.number}`. Otherwise, specify the exact version number, e.g., `23.0.0`.

Workspace backup

By default CloudBeaver keeps all data on disk in a volume `/opt/cloudbeaver/workspace`. By default this volume mounted to host machine folder `/var/cloudbeaver/workspace` or `/var/cloudbeaver-ee/workspace`.

It makes sense to backup this directly from time time and also before product version upgrade.

To archive entire workspace run

```
tar czvf backup.tgz /var/cloudbeaver/workspace
```

or

```
tar czvf backup.tgz /var/cloudbeaver-ee/workspace
```

then move `backup.tgz` to a safe place.

Note: if your server has high load it makes sense to stop it before making a backup. Just run

```
docker stop cloudbeaver-ee
```

 before backup and

```
docker start cloudbeaver-ee
```

 after.

Server configuration

Table of contents

[Main server configuration](#)

[Server parameters:](#)

[Database configuration](#)

[Database connection pool configuration](#)

[Database Initial Data](#)

[Application parameters:](#)

[Resource quotas](#)

[Navigator settings](#)

[Password Policy](#)

[Datasources configuration](#)

[Query manager database configuration](#)

[Using environment variables](#)

[Automatic server configuration](#)

There are several configuration files in CloudBeaver.

Main server configuration

The primary configuration file is `cloudbeaver.conf`. By default, it is placed in the `/etc/cloudbeaver/` folder.

But in most cases, it is redefined for each server by the command line parameter,

```
-web-config <config-file-path> .
```

The server configuration is in the JSONC format (JSON with comments and without redundant quotes). Most of the JSON parsers can parse it in lenient mode.

Additionally, configuration parameters can be specified in the file `workspace/.data/.cloudbeaver.runtime.conf`. It is convenient because the workspace can be deployed as a shared docker volume. `.cloudbeaver.runtime.conf` has the same structure as `cloudbeaver.conf`. However, it has a higher priority than `cloudbeaver.conf`.

Typical configuration:

```

{
  server: {
    serverPort: 8978,
    serverHost: "localhost",
    serverName: "CloudBeaver Sample Server",

    // Paths are absolute or relative to the server root folder
    workspaceLocation: "workspace",
    contentRoot: "web",
    driversLocation: "drivers",

    rootURI: "/",
    serviceURI: "/api/",

    // Webapp configuration file
    productConfiguration: "conf/product.conf",

    expireSessionAfterPeriod: 600000,

    develMode: false,

    sm: {
      enableBruteForceProtection: "${CLOUDBEAVER_BRUTE_FORCE_PROTECTION_ENABLED:true}",
      expiredAuthAttemptInfoTtl: 60,
      maxFailedLogin: "${CLOUDBEAVER_MAX_FAILED_LOGINS:10}",
      minimumLoginTimeout: "${CLOUDBEAVER_MINIMUM_LOGIN_TIMEOUT:1}",
      blockLoginPeriod: "${CLOUDBEAVER_BLOCK_PERIOD:300}",
      passwordPolicy: {
        minLength: "${CLOUDBEAVER_POLICY_MIN_LENGTH:8}",
        requireMixedCase: "${CLOUDBEAVER_POLICY_REQUIRE_MIXED_CASE:true}",
        minNumberCount: "${CLOUDBEAVER_POLICY_MIN_NUMBER_COUNT:1}",
        minSymbolCount: "${CLOUDBEAVER_POLICY_MIN_SYMBOL_COUNT:0}"
      }
    },

    database: {
      url: "jdbc:h2:${workspace}/.data/cb.h2.dat",
      initialDataConfiguration: "conf/initial-data.conf",
      pool: {
        minIdleConnections: 4,
        maxIdleConnections: 10,
        maxConnections: 100,
        validationQuery: "SELECT 1"
      }
    }
  },
  app: {
    anonymousAccessAllowed: true,
    anonymousUserTeam: "user",
    supportsCustomConnections: false,
    publicCredentialsSaveEnabled: true,
    adminCredentialsSaveEnabled: true,
    resourceManagerEnabled: true,
    systemVariablesResolvingEnabled: "${CLOUDBEAVER_SYSTEM_VARIABLES_RESOLVING_ENABLED:false}",

    resourceQuotas: {
      dataExportFileSizeLimit: 10000000,
      resourceManagerFileSizeLimit: 500000,
      sqlMaxRunningQueries: 100,
      sqlResultSetRowsLimit: 100000,
      sqlResultSetMemoryLimit: 2000000,
      sqlTextPreviewMaxLength: 4096,
      sqlBinaryPreviewMaxLength: 261120
    },
    defaultNavigatorSettings: {
      showSystemObjects: true,
      showUtilityObjects: false,
      showOnlyEntities: false,
      mergeEntities: false,
      hideFolders: false,
      hideSchemas: false
    },
    plugins: {

    },
    enabledAuthProviders: [
      "local"
    ],
    enabledDrivers: [

```

```

    ],
    disabledDrivers: [
      "sqlite:sqlite_jdbc",
      "h2:h2_embedded",
      "h2:h2_embedded_v2"
    ]
  }
}

```

All paths can be absolute or are relative to the server start directory (or current directory).

Server parameters:

Name	Description
serverPort	Port CloudBeaver server listens on
serverHost	The network interface CloudBeaver server binds to as an IP address or a hostname. If null or 0.0.0.0, then bind network interface to all available interfaces.
serverURL	Server address (full URL). Used to generate links and for third-party services integration.
workspaceLocation	Root folder for projects
contentRoot	Path to directory with static content
driversLocation	Optional path for driver jar files
rootURI	Web application URI prefix. <code>/</code> by default
serviceURI	Services API URI prefix (relative to rootURI). <code>/api/</code> by default.
productConfiguration	Path to product (web interface) configuration file (json)
develMode	When set to true extra debug, the information is printed in logs and the GraphQL console is enabled on the server.
expireSessionAfterPeriod	Maximum idle time after which the user's session will be closed.

Database configuration

Configures CloudBeaver database where it keeps users, credentials and permission.

In the section `server.database` :

--	--

Name	Description
driver	Database driver (e.g. <code>sqlite</code> , <code>h2_embedded</code> , <code>postgres-jdbc</code> , etc)
url	Database JDBC URL (e.g. <code>jdbc:postgresql://localhost:5432/cb</code>
user	Database user name
password	Database user password
initialDataConfiguration	Path to the initial data file (json) that will be loaded on the first time the server is run

Database connection pool configuration

Configures connection pool to be used by the CloudBeaver database.

In the section `server.database.pool` :

Name	Description
validationQuery	Query that will check the successful connection to the database
minIdleConnections	Minimum number of idle connections that should be kept in the pool
maxIdleConnections	Maximum number of idle connections that should be kept in the pool
maxConnections	Maximum number of idle and active connections that should be kept in the pool

Database Initial Data

Configures initial data containing administrator credentials and a list of teams and their permission.

Stored in a separate file. The path to which is specified in the `server.database.initialDataConfiguration` section.

Name	Description
adminName	Username for administrator
adminPassword	Password for administrator
teams	List of initial teams

Teams schema

Name	Description
------	-------------

subjectId	Id for the team
teamName	Name for the team
description	Team description
permissions	Set of available permissions for the team

Configuration example:

```
{
  adminName: "cbadmin",
  adminPassword: "cbadmin20",
  teams: [
    {
      subjectId: "admin",
      teamName: "Admin",
      description: "Administrative access. Has total and full authority.",
      permission: ["public", "admin"]
    },
    {
      subjectId: "user",
      teamName: "User",
      description: "Standard user",
      permission: ["public"]
    }
  ]
}
```

Application parameters:

In the section `app` :

Name	Description
anonymousAccessEnabled	Allows anonymous access. Anonymous users have a team <code>anonymousUserTeam</code> .
anonymousUserTeam	A team that will be assigned to the anonymous user, <code>user</code> by default.
authenticationEnabled	Enables users' authentication. If disabled, then only anonymous access is allowed.
supportsCustomConnections	Allows users to create custom connections to any databases. Otherwise only the CB administrator can create/edit connections.
publicCredentialsSaveEnabled	Allows you to save user database credentials in a local cache.
adminCredentialsSaveEnabled	Allows you to save global database credentials in a local cache.
redirectOnFederatedAuth	If there is only one federation authentication configuration, then login attempt will automatically be made when the application is opened.

forwardProxy	Identifies the originating IP address and other headers of a client connecting to a web server through an HTTP proxy.
enabledDrivers	List of drivers that are allowed to be used. If the list is empty, all drivers are allowed.
disabledDrivers	List of drivers that are prohibited for use. If the list is empty, all drivers are allowed.
defaultAuthProvider	The provider that will be used for authorization by default.
enabledAuthProviders	List of allowed authorization providers. If the property is absent, all providers are allowed.
defaultNavigatorSettings	Default database navigator settings.
showReadOnlyConnectionInfo	Enables showing the information about a connection if the user has read-only permission for it.
grantConnectionsAccessToAnonymousTeam	Provides access to the predefined shared connections for the "User" team.
systemVariablesResolvingEnabled	Enables the ability to use environment variables in the connection configuration.

Resource quotas

You can configure the following resource quotes in the section `app.resourceQuotas` :

Name	Description
dataExportFileSizeLimit	Maximum file size for data export operation (in bytes)
resourceManagerFileSizeLimit	Maximum file size saved in the resource manager (in bytes)
sqlMaxRunningQueries	Maximum number of simultaneous queries for a single user session. Includes data read queries (i.e. table data view)
sqlResultSetRowsLimit	Maximum number of rows to select from a table or query
sqlTextPreviewMaxLength	Maximum size for text file shown in value panel (in bytes)
sqlQueryTimeout	Maximum time (in seconds) for SQL query execution (including table data read)
sqlBinaryPreviewMaxLength	Maximum size for binary file (also affects JSON in the SQLite) shown in value panel (in bytes)

Navigator settings

You can configure the following properties in the section `app.defaultNavigatorSettings`:

Name	Description
showSystemObjects	Show system objects.
showUtilityObjects	Show "utility" objects.
showOnlyEntities	Only show schemas and tables.
mergeEntities	Show all types of database objects in one list (tables, sequences, etc.).
hideFolders	Hide folders like "Tables", "Schemas", "Columns", etc.
hideSchemas	Do not show schemas (all tables in one list).

Simple view mode properties example:

```
defaultNavigatorSettings: {
  showOnlyEntities: true,
  hideFolders: true,
  hideVirtualModel: true
}
```

Password Policy

You can find information about password policy settings on [this](#) page.

Datasources configuration

You can find a detailed description at [here](#)

Query manager database configuration

You can find a detailed description at [here](#)

Using environment variables

You can use references on environment variables in most server configuration properties. For example:

```
{
  server: {
    serverPort: ${cb.port},
    serverHost: "${cb.host}",
    serverName: "CloudBeaver Server",

    rootURI: "${cb.prefix}",
    serviceURI: "/api/",

    expireSessionAfterPeriod: ${cb.expire-time},
  }
}
```

Automatic server configuration

When you start the CloudBeaver server for the first time, you will see the administrator interface for server configuration.

Sometimes, the server must be configured automatically (e.g., when it is run in the Kubernetes environment).

The following parameters must be specified in the configuration:

Name	Description	Example
CB_SERVER_NAME	Server name	Test Server
CB_SERVER_URL	Server base URL	https://cloudbeaver.domain.com:10000/
CB_ADMIN_NAME	Administrator user name	administrator
CB_ADMIN_PASSWORD	Administrator user password	S0mePazzword

These parameters can be specified in:

- OS environment variables
- The configuration file `.cloudbeaver.auto.conf` must be placed in the exact location as the `cloudbeaver.conf` file.

Query manager database configuration

It is possible to configure a database to store all information from the Query manager.

The configuration for the Query Manager database is located in [the configuration file](#) under the `qm` section.

Additionally, you have the option to configure it using environment variables.

Typical configuration:

```
{
  server: {
    ...
  },
  app: {
    ...
  },
  qm: {
    driver: "${CLOUDBEAVER_QM_DB_DRIVER:h2_embedded_v2}",
    url: "${CLOUDBEAVER_QM_DB_URL:jdbc:h2:${workspace}/.metadata/qmdb/qmdb}",
    user: "${CLOUDBEAVER_QM_DB_USER:''}",
    password: "${CLOUDBEAVER_QM_DB_PASSWORD:''}",
    schema: "${CLOUDBEAVER_QM_DB_SCHEMA:''}"
  }
}
```

Name	Variable name	Description
driver	CLOUDBEAVER_QM_DB_DRIVER	Database driver (e.g. <code>postgres-jdbc</code> , <code>oracle-thin</code> , etc)
url	CLOUDBEAVER_QM_DB_URL	Database JDBC URL (e.g. <code>jdbc:postgresql://localhost:5432/qm</code>)
user	CLOUDBEAVER_QM_DB_USER	Database user name
password	CLOUDBEAVER_QM_DB_PASSWORD	Database user password
schema	CLOUDBEAVER_QM_DB_SCHEMA	Database schema

Configuring server datasources

Table of contents

[Configuring server "predefined" connections](#)

[Overview](#)

[Datasources configuration file](#)

Configuring server "predefined" connections

See [Connection configuration](#) for descriptions of the different connection types.

Overview

The CloudBeaver server may have a set of pre-configured database connections. This configuration is stored on a server and cannot be changed by end-users.

An End-user may choose one of the pre-configured connections on the main CloudBeaver toolbar. Then the user has to provide a username/password in order to connect to the pre-configured datasource. No other parameters are needed.

See [Server configuration](#) for information about the server and workspace configuration.

Datasources configuration file

All project-level configurations are stored in the folder, `${CLOUDBEAVER_WORKSPACE}/GlobalConfiguration/.dbeaver`. Datasources are configured in the file, `data-sources.json`.

It has the same format as [DBeaver](#) datasources configuration file. In version 1.0 CloudBeaver does not support UI for datasources configuration (mostly because it is quite complicated).

You can create this configuration in DBeaver and then copy it to your server configuration folder. Then you can patch the configuration manually by editing the configuration json.

Connection configuration

Table of contents

[Connection types in CB](#)

[Pre-configured connections](#)

[Template connections](#)

[Custom connections](#)

[Cloud connections](#)

[SSH key storage](#)

Connection types in CB

Pre-configured connections

The configuration is located in `${WORKSPACE}/GlobalConfiguration/.dbeaver/data-sources.json`.

Preconfigured connections are always visible in the database navigator. Users cannot delete or change them.

Only the administrator can edit them.

Template connections

Template connections are similar to the provided connections. The main difference is that they are not present in the database navigator by default.

Users can add them to the navigator tree by using the main toolbar Connection->New Connection->From template. Only the administrator can edit the template connections.

Custom connections

Custom connections can be created by users (Note: configuration parameter `supportsCustomConnections` must be turned on).

- Click on the main toolbar->Connection->New Connection->Custom.
- Choose the connection driver
- Fill in the connection parameters
- Click "Create" and the connection will be added in the navigator tree

Cloud connections

Cloud connections cannot explicitly be created or deleted by users. Their configuration is provided by a cloud service provider (e.g. thru AWS API). Once CB will find such connections (by using cloud configuration specified by the server administrator) they will become visible in the navigator tree.

SSH key storage

You can store your SSH key and SSH user in `data-sources.json` :

Variable	Value
configuration.handlers.ssh_tunnel.properties.keyValue	SSH key
connections.configuration.handlers.ssh_tunnel.user	SSH user name

connections.

CloudBeaver and Nginx

Table of contents

[Configuring CloudBeaver with Nginx](#)

[Installing Nginx](#)

[Add proxy config](#)

Configuring CloudBeaver with Nginx

By default CloudBeaver listens to plain http protocol, processes all static content via the Jetty server and is not load balanced.

All these issues can be resolved by putting a real web server in front of CloudBeaver.

We can use Nginx as the most popular web server.

Installing Nginx

```
sudo apt update
sudo apt install nginx
```

Add proxy config

Open the Nginx configuration in your favorite text editor.

The default Nginx config file is `/etc/nginx/sites-enabled/default`.

```
location / {
    proxy_pass          http://localhost:8978;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    Host $http_host;
    proxy_http_version  1.1;
    proxy_set_header    Upgrade $http_upgrade;
    proxy_set_header    Connection "upgrade";
}
```

To identify the information (*real IP address*) of client connected to the web server through NGINX, add the parameter `forwardProxy:true` into your [server configuration](#).

Domain manager

Table of contents

[Overview](#)

[Key features](#)

[Domain manager configuration](#)

[Initial setup](#)

[Configuration fields](#)

[Modifying domain settings](#)

[Important notes](#)

[SSL certification](#)

[Manual configuration](#)

Note: This feature is available in [Enterprise](#) and [Team](#) editions only. For Team Edition, the configuration is accessible exclusively through the web interface.

Overview

The Domain Manager in CloudBeaver enables you to configure custom subdomains for your server. This feature is essential for organizations that manage multiple environments, such as `development`, `testing`, and `production`, or different projects within the same main domain. The ability to create distinct subdomains makes it easier to organize, access, and manage different project stages, thereby enhancing efficiency and clarity in operations.

Key features

- **Automatic configuration:** When deploying using the [Team Edition](#) or [CloudBeaver Enterprise](#) deployment repositories, everything works seamlessly out of the box.
- **Subdomain customization:** Administrators have the ability to set up a specific subdomain within the organization's domain.

- **SSL certificate integration:** During the Domain Manager setup, an SSL certificate is automatically generated via [Let's Encrypt](#), ensuring that the connection to the server remains secure. [Learn more](#)

Domain manager configuration

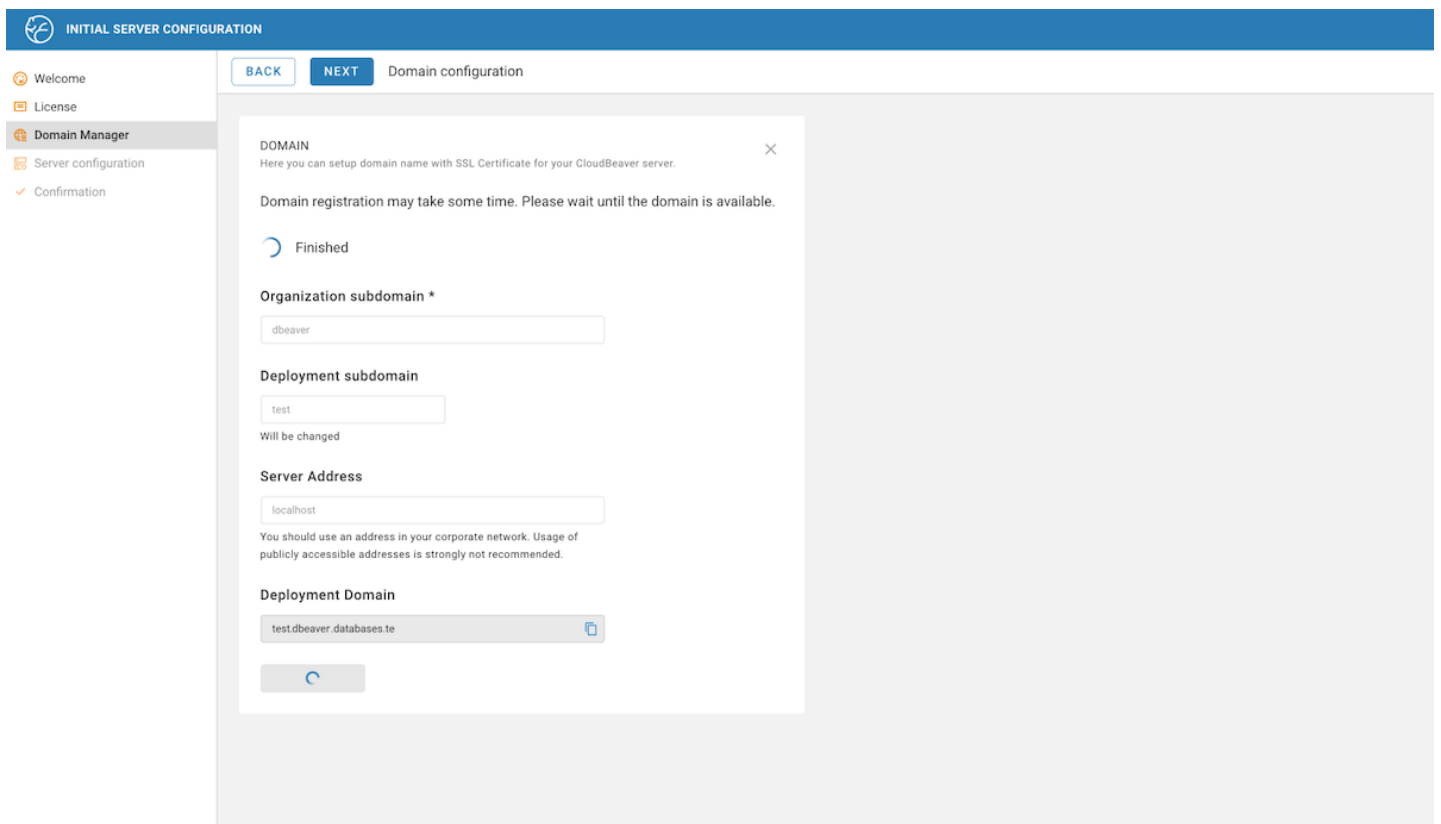
You can access the Domain Manager during the initial server configuration or modify settings after CloudBeaver has been fully set up.

- **Initial setup:** You can specify the subdomain during the initial server configuration. [Learn more](#)
- **Modifying domain settings:** You can adjust the subdomain at any time after the application has been launched. [Learn more](#)

Initial setup

To configure a custom subdomain during the initial setup, follow these steps:

1) Navigate to the **Domain Manager** tab during the initial server configuration. 2) Fill in the necessary fields as detailed in the [table](#) below. 3) Submit your settings and wait for the domain registration to complete. There may be a waiting period for domain registration. 4) Once the domain is active, you will be redirected to your server at the newly constructed domain, e.g., `development.example-corp.databases.com`.



Configuration fields

Field/Button	Description
Organization Subdomain	Specify the Organizational subdomain. Automatically assigned during initial setup based on your licence.
Deployment Subdomain (Optional)	Specify the Deployment subdomain for managing multiple environments, such as <code>development</code> or <code>test</code> .
Server Address	Provide the address where your CloudBeaver server is hosted.
Delete	Delete the Organizational subdomain.

Modifying domain settings

Once CloudBeaver is operational, you can modify the domain settings to better suit evolving needs or correct initial configurations.

To modify your domain settings, follow these steps: 1) Log in as an administrator. 2) Navigate to **Settings -> Administration -> Domain Manager**. 3) Fill in the necessary [fields](#). 4) Submit your settings and wait for the domain registration to complete. There may be a waiting period. 5) Once the domain is active, you will be redirected to your server at the newly constructed domain, e.g., `development.example-corp.databases.com`.

The screenshot shows the CloudBeaver interface with the 'Domain Manager' settings dialog open. The dialog has a title bar with 'SAVE' and 'CANCEL' buttons. The main content area is titled 'DOMAIN' and includes a close button 'X'. Below the title, there is a subtitle: 'Here you can setup domain name with SSL Certificate for your CloudBeaver server.' A warning message states: 'Domain registration may take some time. Please wait until the domain is available.' The settings are organized into sections: 'Organization subdomain *' with a text input field containing 'dbeaver'; 'Deployment subdomain' with a text input field containing 'test' and a 'DELETE' button; 'Server Address' with a text input field containing 'localhost' and a note: 'You should use an address in your corporate network. Usage of publicly accessible addresses is strongly not recommended.'; 'Deployment Domain' with a text input field containing 'test.dbeaver.databases.te' and a copy icon; and 'Valid until' with a date '9/1/2024, 2:00:00 AM' and the text 'Will be renewed automatically'.

Important notes

- The Organization subdomain can be changed once. If it is already set, you must delete it before assigning a new one.
- After deleting a subdomain, the service will be accessible via the Server's Address.
- Changes to the Deployment Subdomain only affect the specified Server Address.
- You can change the Deployment Subdomain up to three times per calendar month.
- The subdomain name can have up to 32 characters and must comply with standard website domain validity criteria.

- After removing or modifying the subdomain, you may lose access to the server, and cloud-based Single Sign-On (SSO) functionality may cease to work. Ensure that one of the following authentication methods is configured and operational: [IAM](#), [NTLM](#), LDAP, or [local access](#).

SSL certification

CloudBeaver uses an automated system to secure your server connections with SSL certificates from Let's Encrypt. This setup is available both during the [initial server configuration](#) and when making adjustments via the [admin panel](#). The system utilizes a pre-configured nginx environment integrated with Certbot, which automatically handles the generation and renewal of SSL certificates. Specify your subdomain during setup, and the system will manage the SSL configuration on your behalf.

Each generated SSL certificate for the subdomain is valid for three months. To ensure continuous security and functionality, the server automatically initiates the certificate renewal process one month before the certificate's expiration.

Manual configuration

If you have specific SSL certificate needs or require a custom configuration, you have to set up it manually. This option involves editing the nginx configuration files yourself to integrate your SSL certificate. It is suitable for users with unique security needs or those who prefer to handle their SSL configurations on their own.

Configuring HTTPS for Jetty server

IMPORTANT: You must replace `{...}` blocks with your own values.

1. Open the terminal and navigate to the workspace directory
2. Type the following commands:

1. `openssl genrsa -des3 -passout pass:1 -out {your domain}.pass.key 2048`
2. `openssl rsa -passin pass:1 -in {your domain}.pass.key -out {your domain}.key`
3. `rm {your domain}.pass.key` (or `del {your domain}.pass.key` on Windows)
4. `openssl req -key {your domain}.key -sha256 -new -out {your domain}.csr`
5. `openssl x509 -req -days 3650 -in {your domain}.csr -signkey {your domain}.key -out {your domain}.p12 -name`
6. `openssl pkcs12 -export -in {your domain}.p12 -inkey {your domain}.key -out {your domain}.p12 -name`
7. `keytool -importkeystore -deststorepass {your password} -destkeypass {your password} -destkeystore`

3. Create a new file called `ssl-config.xml` in the `.data` directory inside the workspace with the following content:

```
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN" "http://www.eclipse.org/jetty/configure_10_0.dtd">
<Configure id="Server" class="io.cloudbeaver.server.jetty.CBJettyServer$JettyServer">
  <New id="httpConfig" class="org.eclipse.jetty.server.HttpConfiguration">
    <Set name="sendServerVersion">false</Set>
    <Set name="sendDateHeader">false</Set>
  </New>

  <Call name="addBean">
    <Arg>
      <New id="sslContextFactory" class="org.eclipse.jetty.util.ssl.SslContextFactory$Server">
        <Set name="keyStorePath">
          {Full path to your keystore. Example: /opt/cloudbeaver/workspace/cb_keys/domain.test.keystore}
        </Set>
        <Set name="keyStorePassword">
          {The password you specified when creating certificates}
        </Set>
        <Set name="trustStorePath">
          {Full path to your keystore example: /opt/cloudbeaver/workspace/cb_keys/domain.test.keystore}
        </Set>
        <Set name="trustStorePassword">
          {The password you specified when creating certificates}
        </Set>
        <Set name="IncludeProtocols">
          <Array type="String">
            <Item>TLSv1.2</Item>
          </Array>
        </Set>
        <Set name="IncludeCipherSuites">
          <Array type="String">
            <Item>TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384</Item>
            <Item>TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384</Item>
          </Array>
        </Set>
        <New id="tlsHttpConfig" class="org.eclipse.jetty.server.HttpConfiguration">
          <Arg>
            <Ref refid="httpConfig" />
          </Arg>
        </New>
      </New>
    </Arg>
  </Call>
</Configure>
```

```

        </Arg>
        <Call name="addCustomizer">
            <Arg>
                <New class="org.eclipse.jetty.server.SecureRequestCustomizer">
                    <Set name="sniHostCheck">false</Set>
                </New>
            </Arg>
        </Call>
    </New>
</New>
</Arg>
</Call>

<Call id="sslConnector" name="addConnector">
    <Arg>
        <New class="org.eclipse.jetty.server.ServerConnector">
            <Arg name="server">
                <Ref refid="Server" />
            </Arg>
            <Arg name="factories">
                <Array type="org.eclipse.jetty.server.ConnectionFactory">
                    <Item>
                        <New class="org.eclipse.jetty.server.SslConnectionFactory">
                            <Arg name="next">http/1.1</Arg>
                            <Arg name="sslContextFactory">
                                <Ref refid="sslContextFactory" />
                            </Arg>
                        </New>
                    </Item>
                    <Item>
                        <New class="org.eclipse.jetty.server.HttpConnectionFactory">
                            <Arg name="config">
                                <Ref refid="tlsHttpConfig" />
                            </Arg>
                        </New>
                    </Item>
                </Array>
            </Arg>
            <Set name="port">
                8978
            </Set>
            <Set name="idleTimeout">
                <Property name="jetty.idleTimeout" default="30000" />
            </Set>
        </New>
    </Arg>
</Call>
</Configure>

```

4. Start the application using the following command:

- In docker:

```
docker run --name={container name} -p 8978:8978 -ti -v {absolute path to workspace}:/opt/cloudbeaver
```

- From sources: `./run-server.sh`

Learn more

Table of contents

[Overview](#)

[Product config structure](#)

[Configuration example](#)

[Shortcuts](#)

[Configuration file locations](#)

[Table of settings](#)

[Explanation](#)

Overview

This guide describes how to set up default CloudBeaver preferences through the configuration files.

Product config structure

Packages are mapped as follows in the configuration file:

- `core-ui` (package name) -> `ui` (name in config)
- `plugin-notifications` (package name) -> `notifications` (name in config)

Here is a structural example:

```
{
  core: {
    [core package name]: {
      [property name]: [property value]
    },
    ...
  },
  plugin: {
    [plugin package name]: {
      [property name]: [property value]
    },
    ...
  }
}
```

Configuration example

The following JSON provides an example of how global properties and plugin-specific settings are defined:

```
{
  // Global properties
  core: {
    authentication: {
      baseAuthProvider: 'local',
      primaryAuthProvider: 'local'
    },
    browser: {
      'cookies.disabled': false
    },
    theming: {
      defaultTheme: 'light'
    },
    localization: {
      defaultLanguage: 'en'
    },
    'navigation-tree': {
      childrenLimit: 500,
      editing: true,
      deleting: true
    }
  },
  plugin: {
    'sql-editor': {
      maxFileSize: 100
    },
    notifications: {
      notificationsPool: 5,
      maxPersistentAllow: 5
    },
    'data-spreadsheet': {
      hidden: false
    },
    'data-viewer': {
      disableEdit: false
    },
    'log-viewer': {
      refreshTimeout: 3000,
      maxLogRecords: 1000,
      logBatchSize: 2000,
      maxFailedRequests: 3
    },
    'data-export': {
      disabled: false
    },
    'erd-viewer': {
      maxColumnsToDisplay: 15000
    }
  }
}
```

Shortcuts

Config below is equivalent to example config for plugins: 'log-viewer', 'data-export', 'erd-viewer'

```
{
  ...
  'plugin.log-viewer.refreshTimeout': 3000,
  'plugin.log-viewer.maxLogRecords': 1000,
  'plugin.log-viewer.logBatchSize': 2000,
  'plugin.log-viewer.maxFailedRequests': 3,
  'plugin.data-export.disabled': false,
  'plugin.erd-viewer.maxColumnsToDisplay': false
}
```

Configuration file locations

The configuration files for the application are located in different directories based on their scope and usage.

Below are the paths where these files can be found, listed in the order of their priority:

1. `workspace/.data/.cloudbeaver.runtime.conf` - Runtime configuration, highest priority.
2. `conf/product.conf` - Server configuration.
3. `webapp/packages/product-default/src/config.json5` - Web application configuration.

Table of settings

Variable	Deprecated	Default value	Description
<code>plugin.notifications.notificationsPool</code>	<code>core_events.notificationsPool</code>	5	Maximum notifications
<code>plugin.notifications.maxPersistentAllow</code>	<code>core_events.maxPersistentAllow</code>	5	Maximum presistent notifications
<code>core.browser.cookies.disabled</code>	<code>core.cookies.disabled</code>	false	Whether an app can use cookies or not

Explanation

If you want to disable the data export functionality and increase the refresh timeout for the [Log Viewer](#) you can modify the settings in the `.cloudbeaver.runtime.conf` file. Follow these steps:

1. Open the `.cloudbeaver.runtime.conf` file
2. Paste the following code:

```
{
  plugin: {
    'log-viewer': {
      refreshTimeout: 7000
    },
    'data-export': {
      disabled: false
    }
  }
}
```

Authentication

[Product configuration parameters](#)

Variable	Deprecated	Value	Description
core.authentication. disableAnonymousAccess	core.authentication. disableAnonymousAccess	false	Disable anonymous access in administration

Theming

[Product configuration parameters](#)

Variable	Deprecated	Value	Description
core.theming. defaultTheme	core.user. defaultTheme	'light'	Default application theme

Localization

[Product configuration parameters](#)

Variable	Deprecated	Value	Description
core.localization. defaultLanguage	core.user. defaultLanguage	'en'	Default application language

Database Navigator

[Product configuration parameters](#)

Variable	Deprecated	Value	Description
core.navigation-tree.childrenLimit	core.app.navigationTree.childrenLimit	100	Sets navigator tree level fetch size
core.navigation-tree.editing	core.app.metadata.editing	true	Enables nodes renaming
core.navigation-tree.deleting	core.app.metadata.deleting	true	Enables nodes deletion

Data Editor

[Product configuration parameters](#)

Variable	Deprecated	Value	Description
plugin.data-viewer.disableEdit	core.app.dataViewer.disableEdit	false	Disables editing for non-admin users
plugin.data-viewer.disableCopyData		false	Disables copying of data for non-admin users
plugin.data-viewer.fetchMin		100	Minimum amount of fetched rows
plugin.data-viewer.fetchMax		5000	Maximum amount of fetched rows
plugin.data-viewer.fetchDefault		200	Default amount of fetched rows
plugin.data-spreadsheet.hidden	plugin_data_spreadsheet_new.hidden	false	Hides data editor from ui

SQL Editor

[Product configuration parameters](#)

Variable	Deprecated	Value	Description
plugin.sql-editor. maxFileSize	core.app.sqlEditor. maxFileSize	10240	Sets the maximum file size for importing in KB

Log Viewer

[Product configuration parameters](#)

Variable	Deprecated	Value	Description
plugin.log-viewer.refreshTimeout	core.app.logViewer.refreshTimeout	3000	Timeout between requests
plugin.log-viewer. maxLogRecords	core.app.logViewer. maxLogRecords	1000	Sets the maximum displayed records count
plugin.log-viewer.logBatchSize	core.app.logViewer.logBatchSize	2000	Sets the maximum loading records count
plugin.log-viewer. maxFailedRequests	core.app.logViewer. maxFailedRequests	3	Sets the maximum retries to load data

Data Export

[Product configuration parameters](#)

Variable	Deprecated	Value	Description
plugin.data-export.disabled	plugin_data_export.disabled	false	Disables data export

ERD

[Product configuration parameters](#)

This parameters only available in Enterprise and AWS products

Variable	Deprecated	Value	Description
plugin.erd-viewer. maxColumnsToDisplay	plugin_erd_viewer. maxColumnsToDisplay	15000	Sets the maximum displayed objects count

Connections

[Product configuration parameters](#)

Variable	Value	Description
plugin.connections. hideConnectionViewForUsers	false	Hides "Connection view" context menu item for non admin users

Command line parameters

Table of contents

[Modify run-server script](#)

[Pass parameters using the environment variable](#)

[Manual](#)

[Docker](#)

CloudBeaver support the same [system parameters](#) as DBeaver.

There are two ways to pass command line parameters to CloudBeaver server

Modify run-server script

- Modify `run-server.sh` script, add extra parameters after `java` command in last line. For example, add parameter `-Xmx2048` in server start:

```
java -Xmx2048M -jar ${launcherJar} -product io.cloudbeaver.product.ce.product -web-config conf/cloudb
```

Note: *to be able to modify run script you must build CloudBeaver from sources. It doesn't make sense to modify the script in docker container because all changes will be reset after container restart.*

Pass parameters using the environment variable

Set variable `JAVA_OPTS` to appropriate parameters value. It works for manual server start and for docker container start.

Manual

```
export JAVA_OPTS=-Xmx2048
./run-server.sh
```

Docker

You can pass JAVA_OPTS variable to docker container by using `-e` docker parameter:

```
sudo docker run -d --restart unless-stopped -p 80:8978 \  
-e JAVA_OPTS=-Xmx2048 \  
-v /var/cloudbeaver/workspace:/opt/cloudbeaver/workspace dbeaver/cloudbeaver:latest`}
```


Local Preferences

Table of contents

[Overview](#)

[Preferences](#)

[Interface](#)

[Data viewer](#)

[SQL Editor](#)

[Tools](#)

Overview

CloudBeaver provides extensive customization and management capabilities through its Preferences settings, allowing both administrators and regular users to tailor the application according to their needs. To access these settings:

- **Administrators:** Navigate to **Settings -> Administration -> Preferences**.
- **Users:**
 - Go to **Settings -> Preferences**.
 - Alternatively, navigate to **Profile -> Preferences**

Preferences

Upon accessing the preference page, you will encounter the following settings for customization:

Interface

Setting	Description	Admin only
	Sets the application's	

Language	language.	No
	Available languages:	
	- English	
	- French	
	- German	
	- Italian	
	- Japanese	
	- Korean	
	- Brazilian Portuguese	
	- Romanian	
	- Russian	
	- Simplified Chinese	
	- Spanish	
	- Traditional Chinese	
	- Ukrainian	
Theme	Application color theme. Available themes:	No
	-Light	
	-Dark	

Data viewer

Setting	Description	Admin only
Disable Table presentation	Disables the table view in Data Viewer.	Yes
Disable Copy	Disables copying of data in Data Viewer.	Yes
Disable Edit	Disables editing of data in Data Viewer.	Yes

Default fetch size	Default number of rows to fetch.	No
Maximum fetch size	Maximum number of rows to fetch.	Yes
Minimum fetch size	Minimum number of rows to fetch.	Yes

SQL Editor

Setting	Description	Admin only
Insert table aliases	Automatically inserts table aliases in the FROM clause of a SQL query. Options:	No
	- Disable : Do not insert any aliases automatically.	
	- my_table mt : Use a short alias for tables.	
	- my_table AS mt : Use the AS keyword with a short alias for tables.	
Use long object names	Always use schema/catalog for object names.	No

Tip: Read more about [SQL Editor](#).

Tools

Setting	Description	Admin only
Disable Tools	Removes Tools menu from the top menu bar.	Yes

Overview

Table of contents

[AWS Marketplace](#)

[IAM/EC2 installation](#)

[Docker image](#)

[License](#)

[Troubleshooting](#)

[Cloud explorer](#)

AWS Marketplace

You can subscribe to CloudBeaver EE in the AWS Marketplace here: <https://aws.amazon.com/marketplace/pp/B08QTY2JTF>.

There is a one-month trial period.

IAM/EC2 installation

After launching EC2 instance based on CloudBeaver IAM, open the page `http://EE2_IP/` where `EE2_IP` is the IP address of your new EC2 machine.

On the first page you will see [server configuration wizard](#).

You can connect to your EC2 instance using SSH. You need to specify the SSH keypair during the EC2 instance launch. You can use the user name `ubuntu` to connect.

Docker image

Besides the AWS marketplace, you can run CloudBeaver AWS from the docker image.

In this case, you will need to specify a license during product configuration. A license can be purchased at <https://cloudbeaver.io> website.

Docker images:

- dbeaver/cloudbeaver-aws:latest - latest release build.
- dbeaver/cloudbeaver-aws:dev - latest developer build.
- dbeaver/cloudbeaver-aws:[version] - image of exact version of CloudBeaver AWS.

To install just run the command:

```
sudo docker pull dbeaver/cloudbeaver-aws:latest
```

To run CloudBeaver in the terminal:

```
sudo docker run --name cloudbeaver --rm -ti -p 8080:8978 -v /var/cloudbeaver/workspace:/opt/cloudbeaver/w
```

License

If you install CloudBeaver AWS from AWS Marketplace, you don't need a license, as it is part of the Marketplace agreement.

If you install it from the docker image, then you need to purchase a license.

Troubleshooting

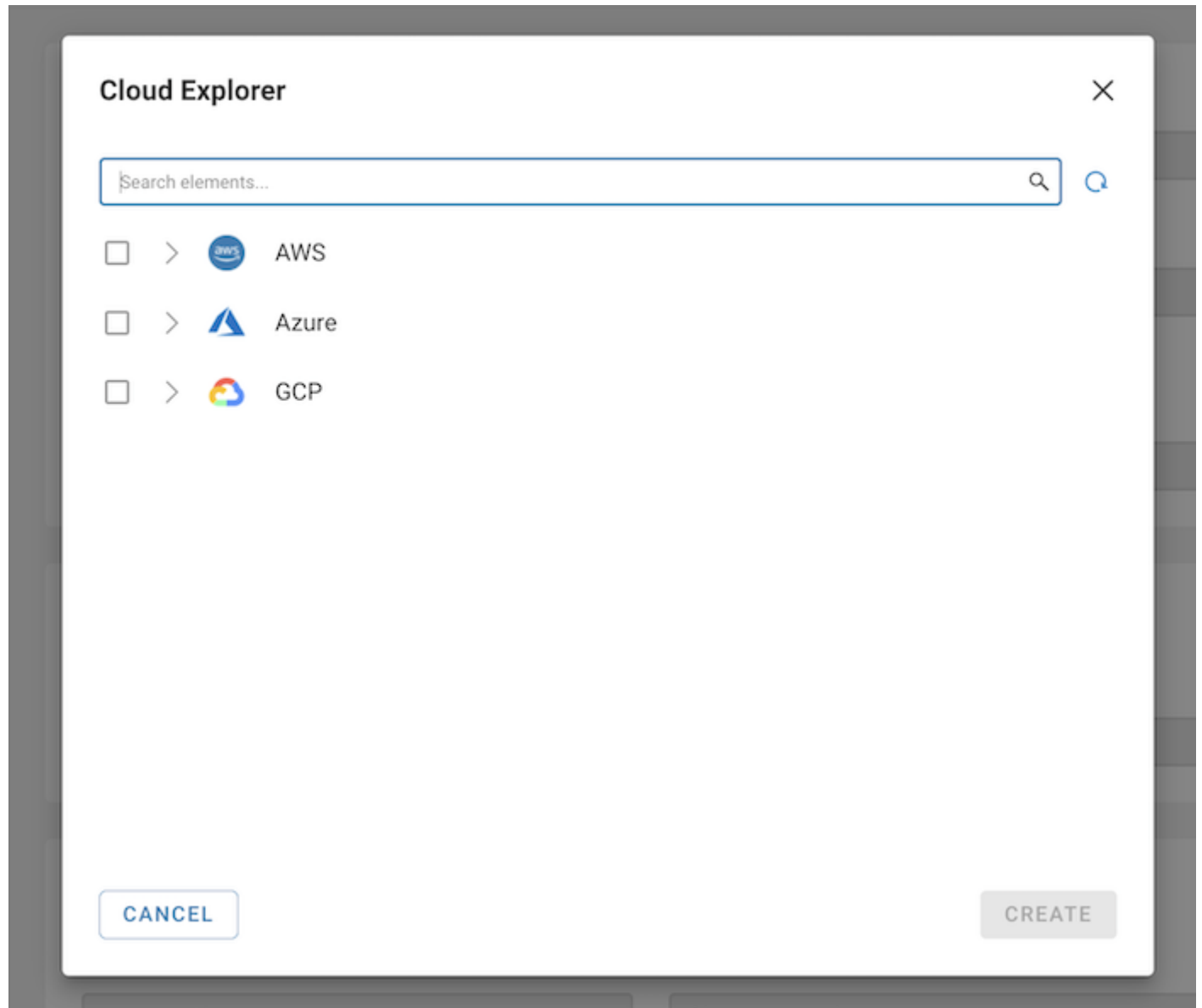
If you installed CloudBeaver AWS from Marketplace but it still asks for a license, then it may be a result of problems with EC2 metadata service availability. This may happen if your AWS policy requires the use of IMDSV2 and restricts the use of IMDSV1.

As CloudBeaver runs in a Docker container, it has a different IP address and can't connect to the EC2 metadata service IMDSV2. Thus, it can't be determined that it was run from the Marketplace installation.

Solution: run docker container with parameter `--network host`. Thus, CloudBeaver will have the same IP address and can access EC2 metadata.

Cloud explorer

You can use an embedded [Cloud Explorer](#) in order to find and add existing AWS databases:



AWS Settings

Table of contents

[Authentication](#)

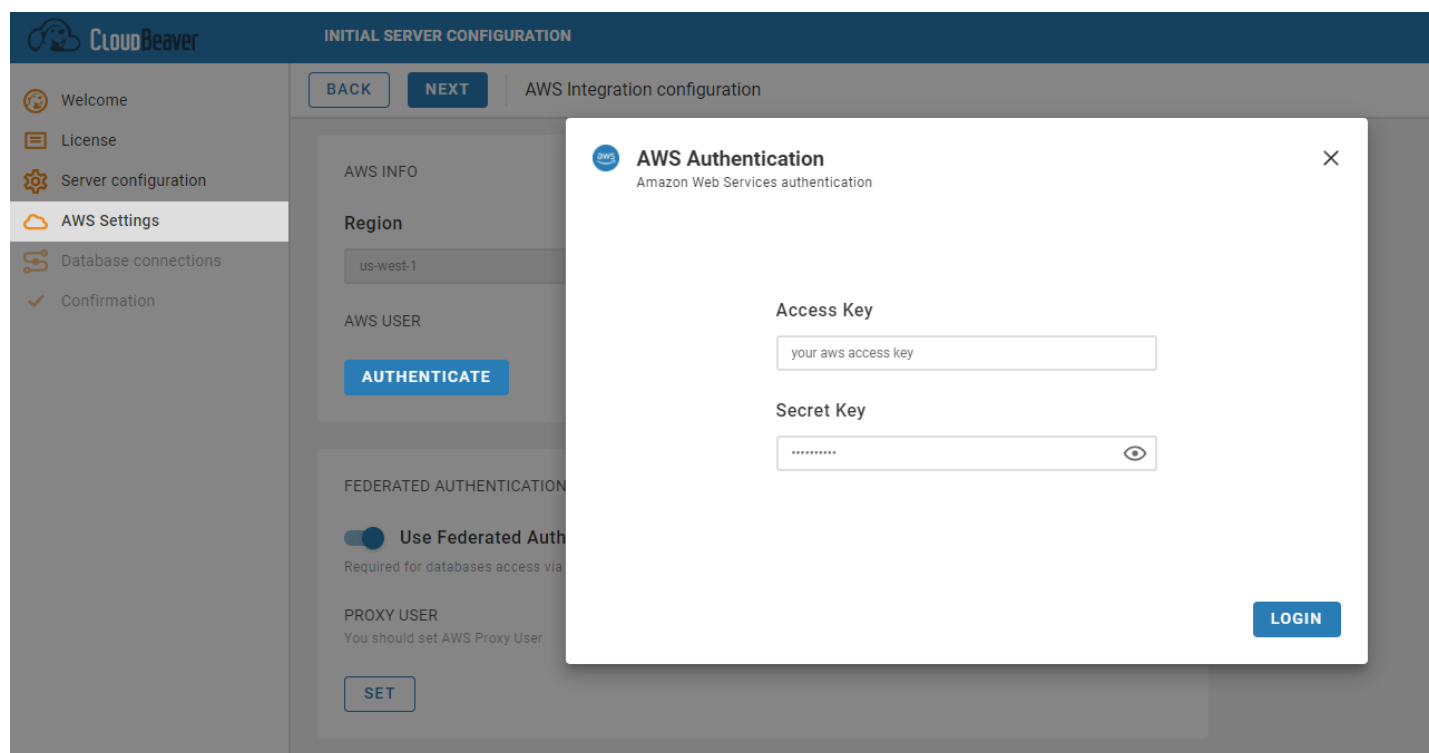
[Federated Authentication](#)

[Regions](#)

Authentication

To get access to the AWS Services from the CloudBeaver you need to authenticate to your AWS Account. All CloudBeaver AWS settings will depend on this AWS account permissions. You will not be able to change entered AWS credentials after the end of configuration process. Also, these credentials will be automatically assigned to the administrator in CloudBeaver. We highly recommend to create a special AWS user with all required permissions for the administrator account.


You can learn more about AWS Authentication [here](#).




The screenshot shows the CloudBeaver 'INITIAL SERVER CONFIGURATION' interface. On the left is a sidebar with navigation links: Welcome, License, Server configuration, AWS Settings (highlighted), Database connections, and Confirmation. The main panel is titled 'AWS Integration configuration' and contains sections for 'AWS INFO' (Region: us-west-1), 'AWS USER' (with an 'AUTHENTICATE' button), 'FEDERATED AUTHENTICATION' (with a toggle for 'Use Federated Auth'), and 'PROXY USER' (with a 'SET' button). A modal window titled 'AWS Authentication' is open in the foreground, showing fields for 'Access Key' (with placeholder text 'your aws access key') and 'Secret Key' (with a toggle for visibility), and a 'LOGIN' button at the bottom right.


Federated Authentication


Here you can setup a proxy user to be able to connect to the AWS Services via SSO. You can learn more about it in [this article](#).


 CloudBeaver


INITIAL SERVER CONFIGURATION


 Welcome

 License

 Server configuration

 AWS Settings

 Database connections

 Confirmation

BACK

NEXT

AWS Integration configuration

FEDERATED AUTHENTICATION

☒

 Use Federated Authentication

Required for databases access via external identity providers







PROXY USER

You should set AWS Proxy User

SET

Regions

When you view your AWS Resources in the CloudBeaver, you see only the resources that are tied to the AWS Regions that you specified in this step. For example, when you search for the AWS databases in [Cloud Explorer](#), you see only the databases that exist in these specific regions. Regions can be configured later in the [Administration](#) section.

-  Welcome
-  License
-  Server configuration
-  **AWS Settings**
-  Database connections
-  Confirmation

[BACK](#)
[NEXT](#)

AWS Integration configuration

REGION LIST

[All regions](#)
[Selected regions](#)

AFRICA

☐ Africa (Cape Town) (af-south-1)

ASIA

☐ Asia Pacific (Hong Kong) (ap-east-1)

☐ Asia Pacific (Osaka) (ap-northeast-3)

☐ Asia Pacific (Singapore) (ap-southeast-1)

☐ Asia Pacific (Tokyo) (ap-northeast-1)

☐ China (Ningxia) (cn-northwest-1)

EUROPE

☒ Europe (Frankfurt) (eu-central-1)

☐ Asia Pacific (Mumbai) (ap-south-1)

☐ Asia Pacific (Seoul) (ap-northeast-2)

☐ Asia Pacific (Sydney) (ap-southeast-2)

☐ China (Beijing) (cn-north-1)

☐ Middle East (Bahrain) (me-south-1)

☐ Europe (Ireland) (eu-west-1)

Learn more

Please note: This article has been updated and its content is now available in a revised form at a new location.

Please refer to the updated article [Cloud Explorer](#) for the most current information.

Overview

Table of contents

[Enterprise database drivers:](#)

CloudBeaver EE is an advanced version of the [CloudBeaver](#) product.

It contains all features of the CloudBeaver Community plus:

- Enterprise database drivers
- [Cloud Authentication support](#)
- [ER diagrams for database schemas and tables](#)

Enterprise database drivers:

- Relational databases
 - Apache Calcite
 - Apache Drill
 - Apache Ignite
 - Apache Phoenix
 - SQL Server
 - Sybase, SAP ASE
 - DB2
 - Snowflake
 - Databricks
 - Vertica

- Netezza
- Hive
- Google Bigquery
- Intersystems Cache
- Clickhouse
- CockroachDB
- Dremio
- DuckDB
- EnterpriseDB
- Informix
- MaxDB
- Neo4j
- Oceanbase
- Opensearch (Elasticsearch)
- SAP HANA
- Teradata
- Timescale
- Trino
- Yellowbrick
- Yugabyte
- NoSQL databases
 - MongoDB

- Cassandra
- InfluxDB
- Couchbase
- CouchDB
- Redis
- AWS databases
 - RDS/Aurora
 - Athena
 - Redshift
 - DynamoDB
 - DocumentDB
 - Keyspaces
 - Timestream

License Management

Table of contents

[CloudBeaver Licenses](#)

[License installation](#)

[License upgrade](#)

[Team Edition license](#)

CloudBeaver Licenses

Following products need a license:

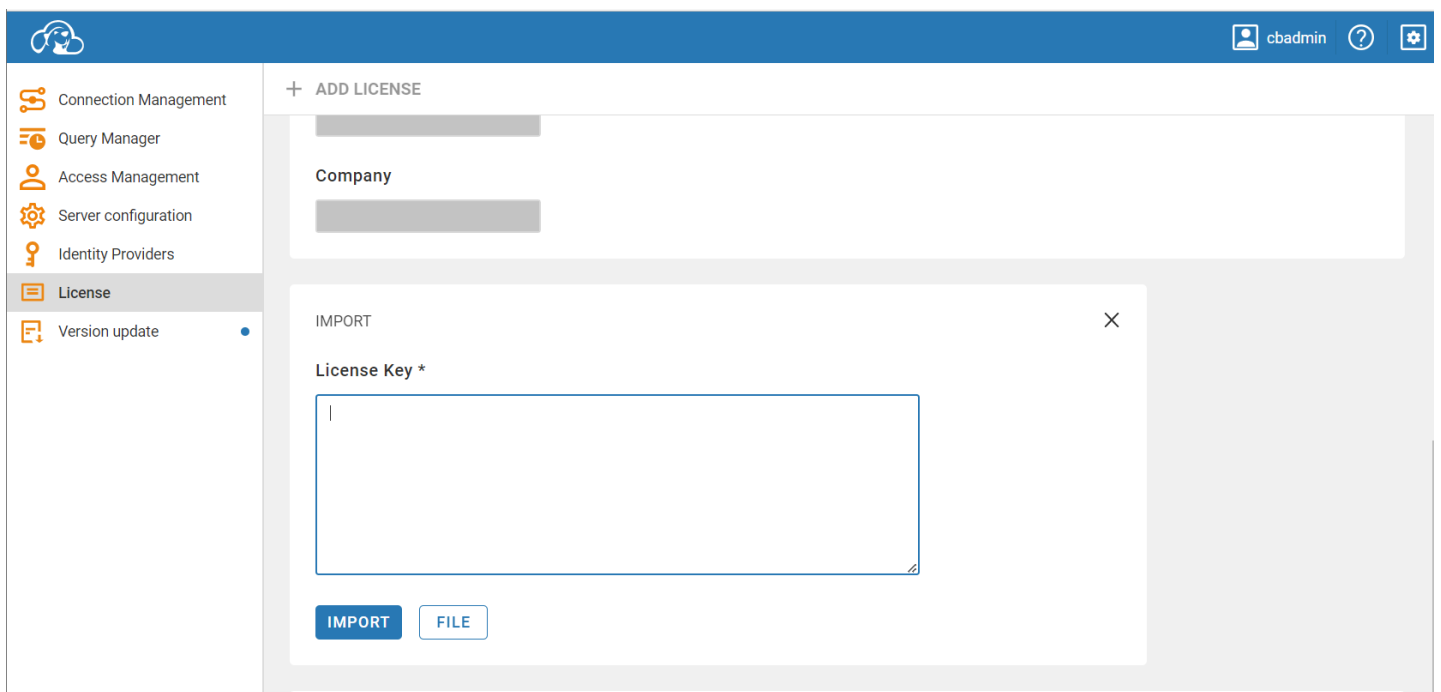
- CloudBeaver EE
- CloudBeaver AWS in a custom host
- CloudBeaver Team Edition

License installation

You need to get a license from your profile on <https://dbeaver.com/cloudbeaver-enterprise/> web site.

You can use a commercial or trial license. All licenses work the same way.

You add your license when you first start CloudBeaver during [Server configuration](#). Also, it's possible to return to the [Administration panel](#) anytime and import the license. This can be done in two ways: by adding the license text or by uploading the license file. In the first case, please note that you need to copy-paste the full license text (not just the license ID). The license text starts with '-' and ends with '==' characters.



Once you have the license text in the filed, click 'Import'.

License upgrade

License has an expiration period. Typically it is 1 year.

If your machine has access to internet then license will be updated automatically after you you upgraded it on dbeaver.com.

Otherwise you will need to go to admin panel again and re-install the license.

Team Edition license

In CloudBeaver Team Edition license has additional parameter: assigned roles and user count for each role.

Thus the license limits number of users who can log-in and use CloudBeaver. CloudBeaver validates these numbers automatically.

Team Edition Overview

DBeaver Team Edition is a distributed application which allows different users in your organization to work altogether on the same shared resources, connect to databases and perform various database-related tasks from a desktop or web-based user interface.

[Product overview](#)

Getting started with Team Edition

Table of contents

[Step 1. Server deployment](#)

[Step 2. Initial server setup](#)

[Step 3. Download and use the desktop client \(optional\)](#)

Team Edition is the most comprehensive DBeaver product for effective team collaboration which consists of three parts:

1. **DBeaver Team Edition server** that handles authentication, provisions projects and enables collaboration features
2. **DBeaver Team Edition web** client which is based on CloudBeaver Enterprise
3. **DBeaver Team Edition desktop** client which is based on DBeaver Ultimate.

To start working with Team Edition, go through the following steps:

Step 1. Server deployment

The server part consists of several docker containers. We offer instructions for the two most common orchestration technologies: Docker Compose and Kubernetes. You can find these instructions [in our public git repository](#). If these instructions don't suit your environment or if you need any other help, please don't hesitate to contact our [technical support](#).

Step 2. Initial server setup

[Initial server setup](#)

Step 3. Download and use the desktop client (optional)

You can download DBeaver Team client from <https://dbeaver.com/download/team-edition/>.

After installation, you need to configure desktop client to connect to your previously deployed server.

By default, the client will try to detect server automatically by getting value from:

- Environment variable `DBEAVER_DOMAIN_CONTROLLER`
- Init parameter `DBEAVER_DOMAIN_CONTROLLER` (it can be set in dbeaver.ini file by adding line `-DDBEAVER_DOMAIN_CONTROLLER=URL` at the end
- Windows registry key `HKEY_CURRENT_USER\Software\DBeaverTeam\DomainControllerURL`
- Windows registry key `HKEY_LOCAL_MACHINE\Software\DBeaverTeam\DomainControllerURL`

If none of these methods succeed (default behavior) then it will ask for DC URL in popup dialog. After that passed URL will be saved in file `%APPDATA%\DBeaverData\team-workspace\.metadata\domain-controller.properties`. You can change saved value in this file manually later.

Team Edition Server Configuration

Table of contents

[Initial login](#)

[License import](#)

[Authentication configuration](#)

[Local users](#)

[Federated authentication](#)

[Azure AD](#)

[Google GSuite](#)

Initial login

Use the default credentials to login:

User name: `cbadmin` User password: `cbadmin20`

License import

You need a license in order to start working with CloudBeaver.

Use the same instruction as for other CloudBeaver products: [License Management](#)

Authentication configuration

You can use different ways to manage and authenticate your users.

Local users

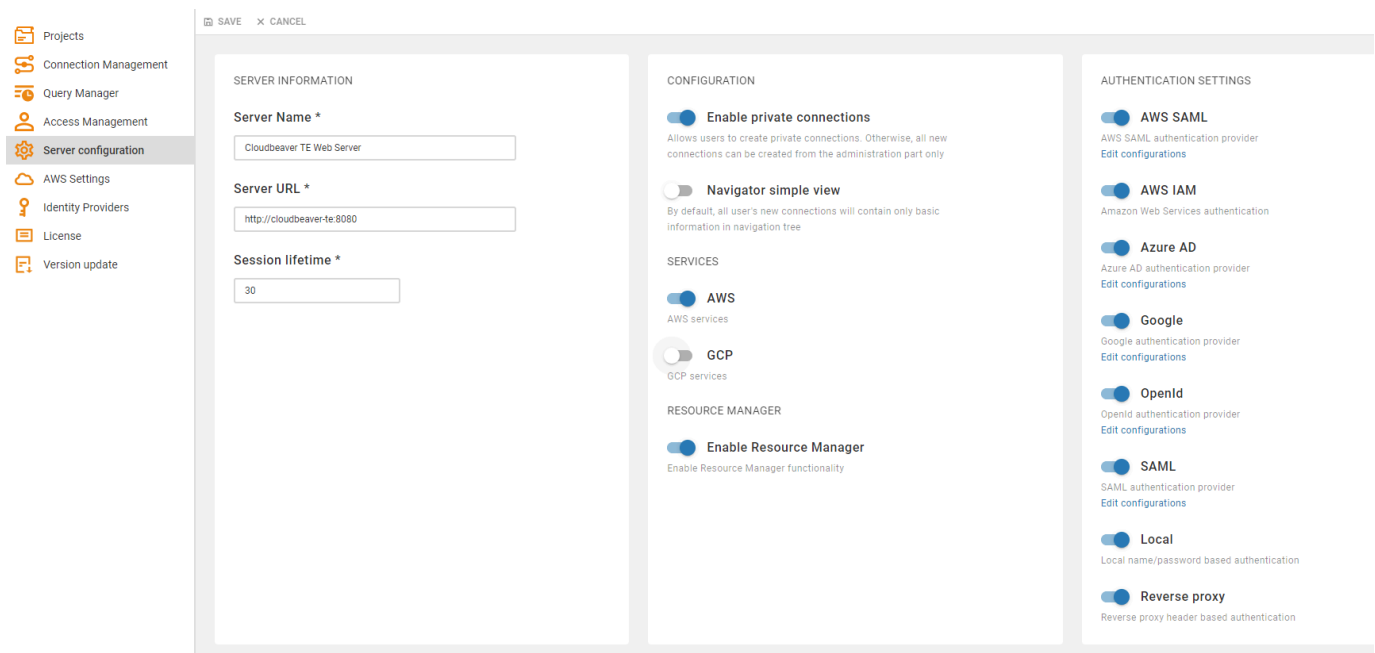
Local users are stored in CloudBeaver internal database. You need to create local users manually and specify username and password for them.

User can change its password later. See "[creating users](#)" for more details.

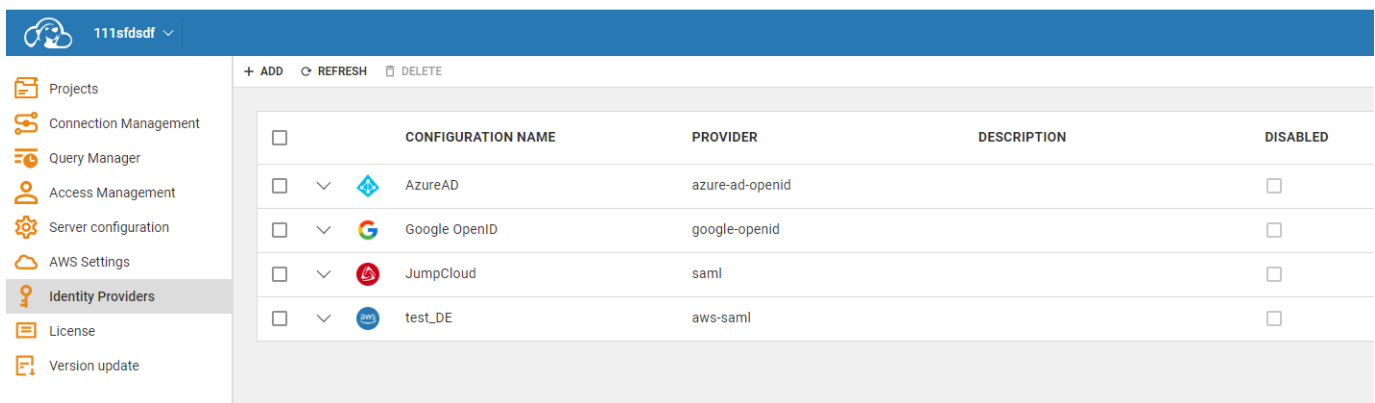
Federated authentication

You can keep third party catalog service like Active Directory to keep your users and manager authentication. To use them in DBeaver/CloudBeaver your need to configure identity provider(s).

1. Go to administration panel, switch to tab "Server Configuration" and enable needed services and authentication methods:



2. Got to "Identity providers" tab and configure federated authentication:



Below is the list of supported identity providers:

Azure AD

You can configure Azure AD integration so users will be stored in AD. There is no need to manage their credentials or teams manually.

Identity Providers

License

Version update

Provider *

Azure AD

ID *

AzureAD

Configuration name *

AzureAD

Description

Icon URL

☐ Disabled

AZURE AD

Domain / Tenant ID

dc218

Application (client) ID

ca21

Secret Key

0tN

☒ Database authentication provider

☒ Read AD group information

AD Group: Administrators

bdr73

AD Group: Developers

LINKS

Sign in

https://dc/api/azure-ad-openid/...

Sign out

https://dc/api/azure-ad-openid/...

Redirect

https://dc/api/azure-ad-openid/...

You need to specify following parameters:

Name	Description
Domain / Tenant ID	Azure AD domain name or tenant ID
Application ID	Azure AD enterprise application ID (the same as client ID in OpenID auth)
Secret Key	Azure AD enterprise application client secret key
Database authentication provider	If checked then access to SQL Server will be claimed along with user information. Open ID scope = <code>https://database.windows.net//.default</code>
Read AD group information	If checked then user AD groups information will be claimed. It is needed to associate AD user with CludBeaver roles. Open ID scope = <code>groups</code> .
AD Group: Administrators	AD group identitiers (coma separated). Users from these groups will be granted with Administrator role
AD Group: Developers	AD group identitiers (coma separated). Users from these groups will be granted with Developer role
AD Group: Managers	AD group identitiers (coma separated). Users from these groups will be granted with Manager role

Note: Users will be created automatically in Team Edition on their first login. The User ID will be the same as the user's email in Azure AD.

Google GSuite

OPTIONS

CANCEL

SAVE

Provider *

OpenId

ID *

google-openid

Configuration name *

google-openid

Description

Icon URL

☐ Disabled

OPENID

Client ID

990ip...

Client Secret

GCzE

IDP auth endpoint URL

https://accounts.google.com/o/oauth2/auth

IDP token endpoint URL

https://oauth2.googleapis.com/token

LINKS

Sign in

https://id/google-o...

Sign out

https://google-o...

Redirect

https://anid/google-o...

You need to specify following parameters:

Name	Description
Client ID	Client ID from GSuite OpenID credentials
Client Secret	Client secret from GSuite OpenID credentials
IDP auth endpoint	<code>https://accounts.google.com/o/oauth2/auth</code>
IDP token endpoint	<code>https://oauth2.googleapis.com/token</code>

Note: Users will be created automatically in Team Edition on their first login. The User ID will be the same as the user's email in GSuite.

Projects in Team Edition

Table of contents

[Overview](#)

[Projects management](#)

[Create new Project](#)

[Edit access](#)

[Delete Project](#)

[How to switch between Projects](#)

[Multiple Projects](#)

[Private Project](#)

Overview

In DBeaver Team Edition, projects serve as organizational units that administrators use to manage and share database resources. These resources include database connections, tasks, SQL scripts, ER diagrams, datasets and bookmarks.

Administrators maintain full visibility over all projects, except for those marked as private. They can view all connections, datasets, and scripts. Administrators are the only role controlling access to each project, allowing specific users and roles to interact with the resources.

The visibility, editing permissions, and other interactions with the project's resources can vary depending on the user's permissions.

Projects management

The user interface (UI) for managing projects in DBeaver Team Edition may differ slightly between the web and desktop versions. However, the functionality remains consistent across both platforms, ensuring a uniform user experience, except for the [Multiple projects](#) feature.

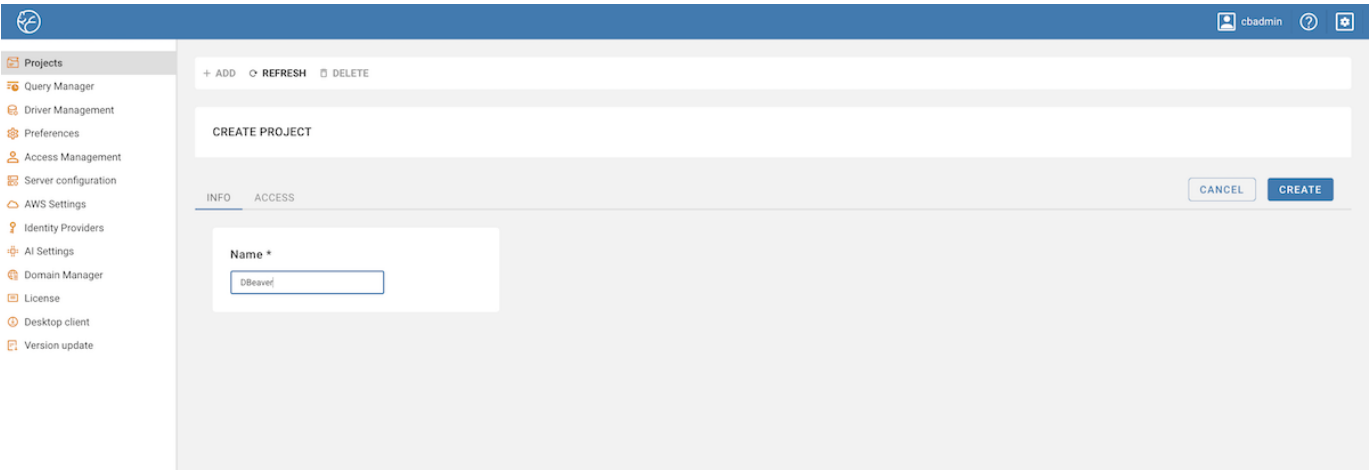


Note: Viewer and Editor roles are not supported in the desktop version of Team Edition. For more information on user roles, refer to [role management](#) article.

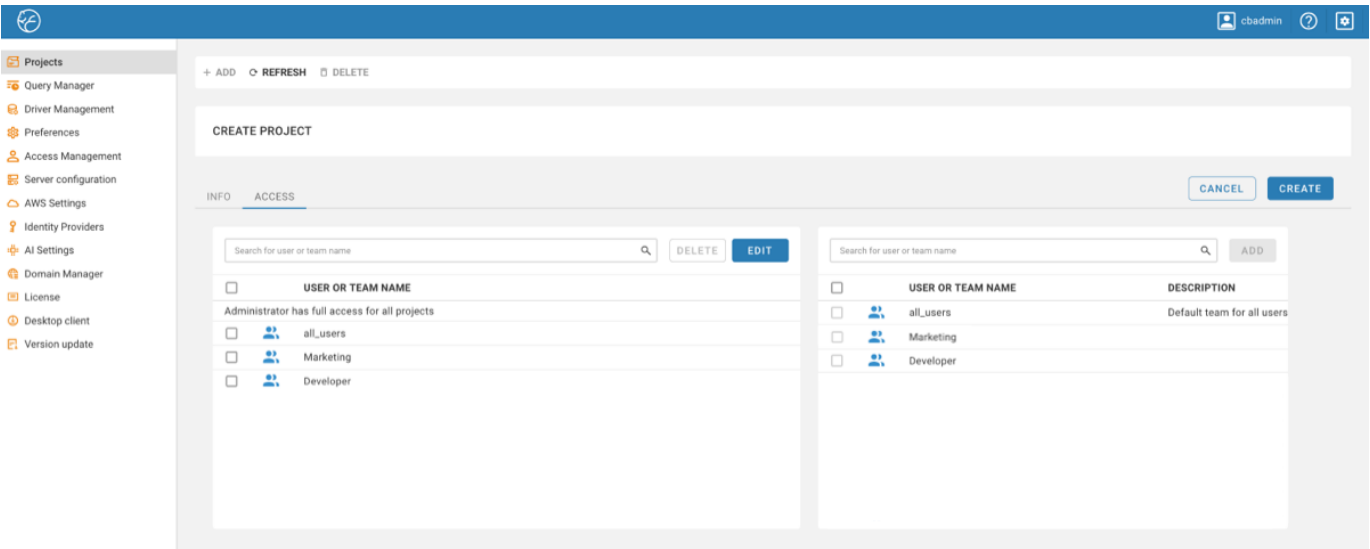
Create new Project

Web version:

- 1. As administrator, go to **Settings -> Administration**.
- 2. Click on the **Projects** tab.
- 3. Click the **+ Add** button and enter the project name.




- 4. For access permissions, go to **Access**, click **Edit**, and select users or teams.
- 5. To add users or teams, check the box next to their names and click the **Add** button.

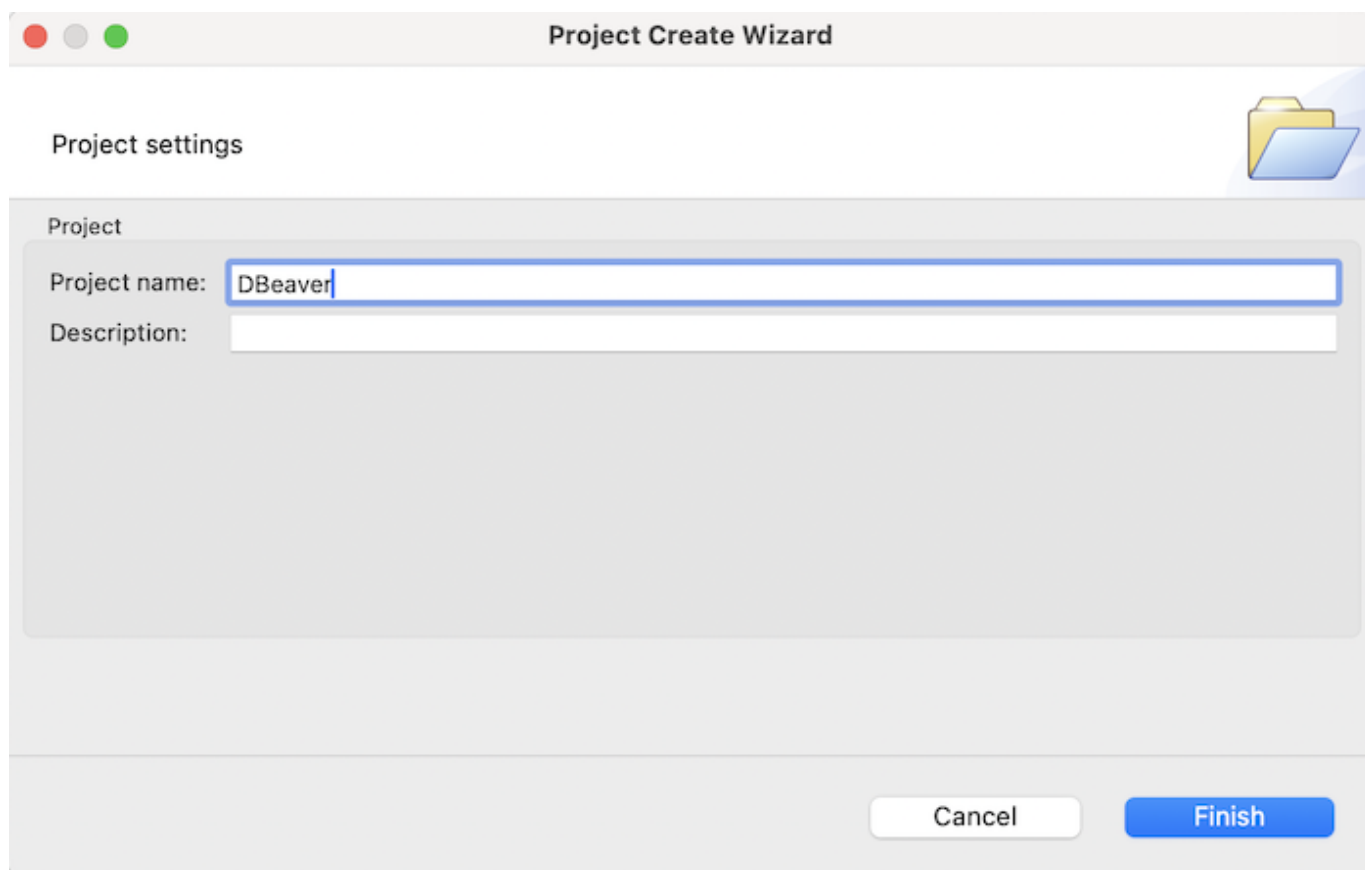


Desktop version:

1. As administrator, navigate to the **Projects** view.

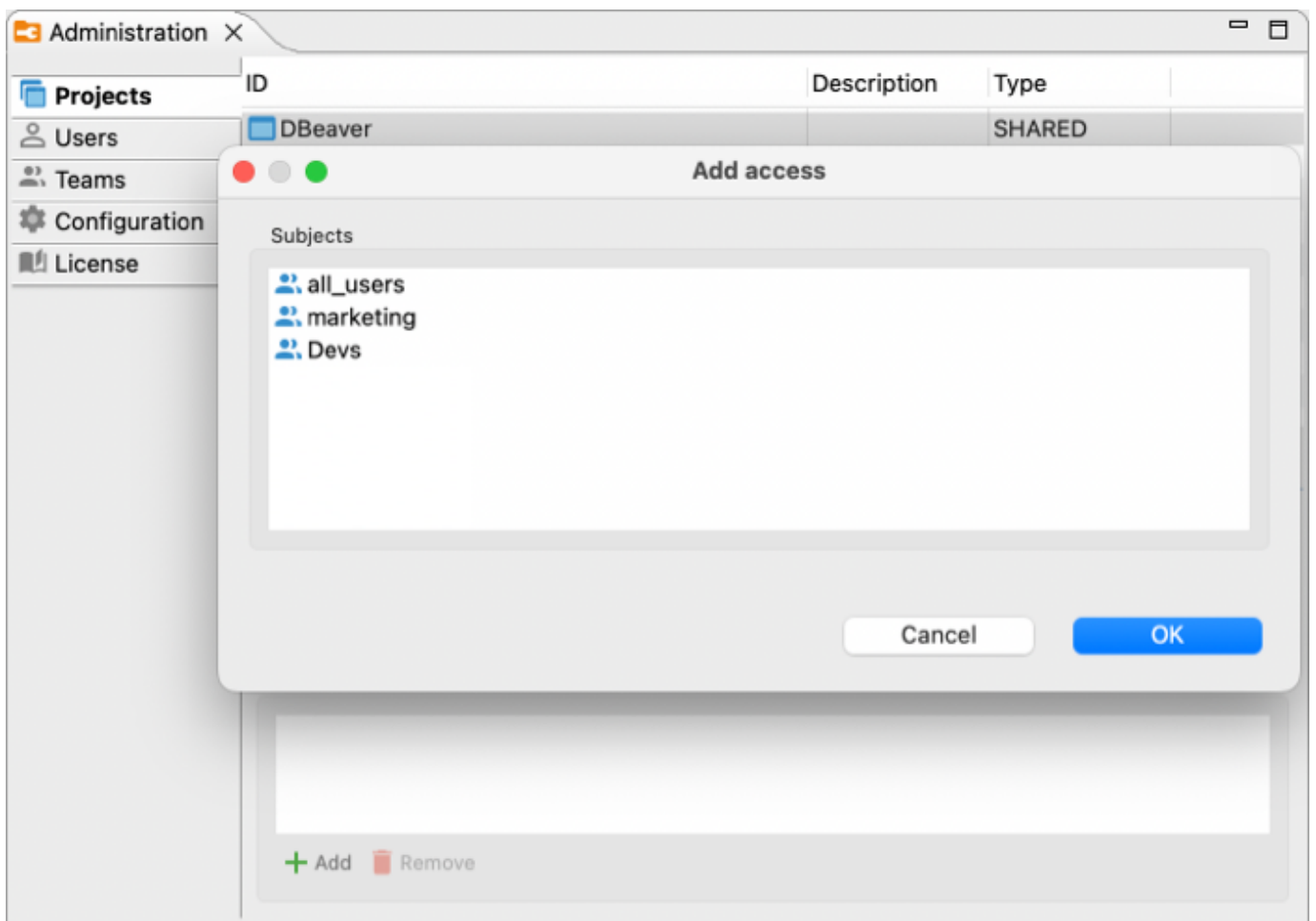
Tip: If the **Projects** view is not open, access it by going to **Window -> Projects**.

2. Click on the **Create Project** button .
3. Enter the project name and provide a description (optional).



The screenshot shows a macOS-style dialog box titled "Project Create Wizard". Inside, there's a section labeled "Project settings" with a folder icon. Below this is a "Project" section containing two text input fields. The first field, labeled "Project name:", has the text "DBeaver" entered. The second field, labeled "Description:", is empty. At the bottom right of the dialog are two buttons: "Cancel" and "Finish".

4. For access permissions, go to **Window -> Show View -> Administration**, and select the newly created project.
5. To add users or teams, press the **+ Add** button, choose one or more users/teams, and click **OK**.



Edit access

Editing access is quite similar to the steps during the creation of a project:

Web version:

1. As administrator, go to **Settings -> Administration -> Projects** and select the project.
2. Select the **Access** tab.
3. To modify access permissions:
 - To add new users or teams, click **Edit**, check the box next to their names in the right window and click the **Add** button.
 - To remove existing users or teams, select them in the left window and click the **Delete** button.

Desktop version:

1. As administrator, navigate to **Window -> Show View -> Administration**.

2. Select the **Projects** tab.
3. To modify access permissions:
 - To add new users or teams, press the **+ Add** button, choose one or more users/teams, and click **OK**.
 - To remove existing users or teams, select them in the **Access** window and click the **Remove** button.

Tip: You can also manage project access directly through the [Users](#) and [Teams](#) configuration interfaces.

Delete Project

To delete a project:

Web version:

1. As administrator, go to the **Settings -> Administration -> Projects**.
2. Select the checkbox next to the project you wish to remove.
3. Click on **Delete** button.
4. Confirm the deletion to permanently remove the project.

Desktop version:

1. As administrator, navigate to the **Projects** view.
2. Select the project you wish to remove, right-click on it, and click **Delete** button.
3. Confirm the deletion to permanently remove the project.

How to switch between Projects

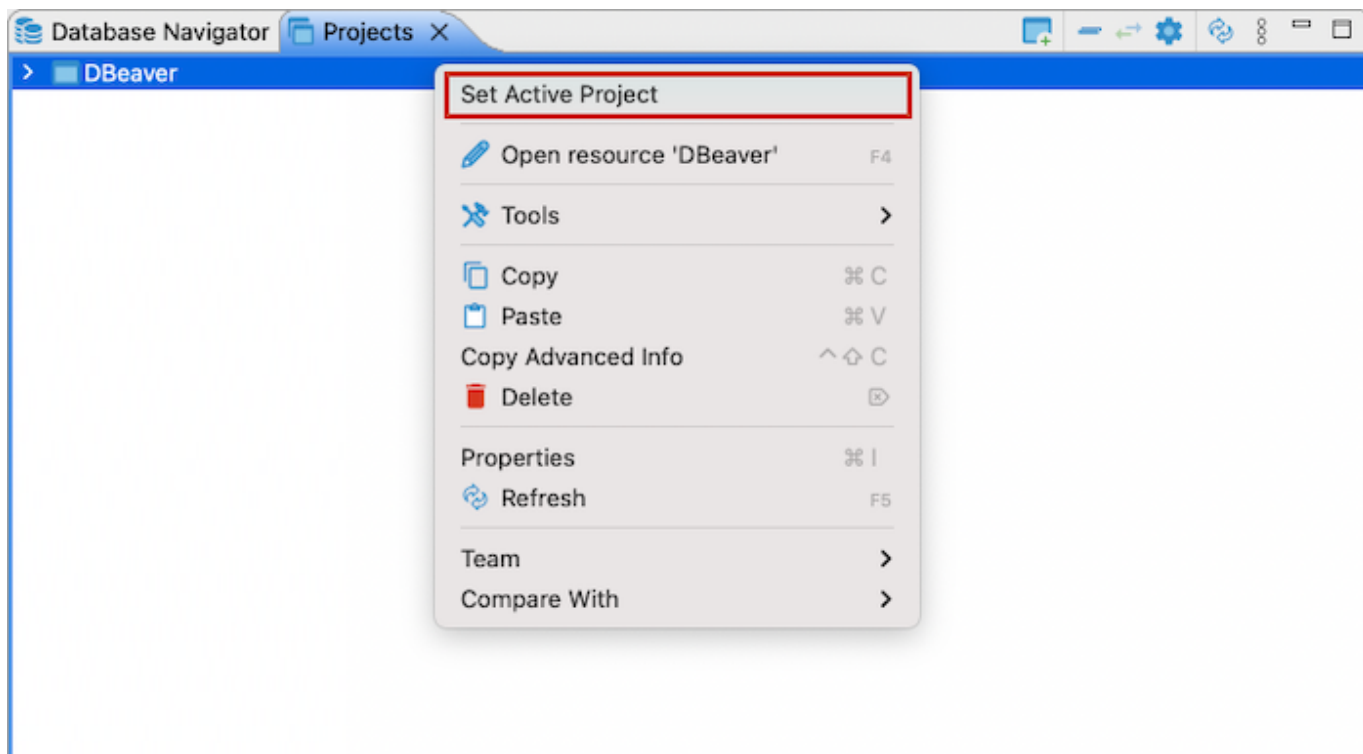
Web version:

- To switch between different projects you have access to, click the **Project Selector** button, located in the **Toolbar**.



Desktop version:

- In the desktop version, access the **Project** view to see all the projects available to you. To open a project and make it active in the **Database Navigator**, right-click on the desired project and select **Set project as active**.



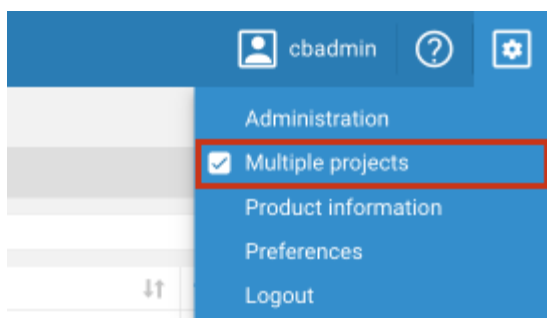
Note: Depending on the selected project, a different set of database connections, tasks, SQL scripts, ER diagrams, datasets and bookmarks will be displayed.

Multiple Projects

You can display multiple projects simultaneously in the **Database Navigator**.

To enable this feature:

1. Use the checkbox **Multiple projects** in **Settings** to allow multiple projects in the **Database Navigator**.



2. Choose the projects you want to see in your Database Navigator. Click the **Project Selector** button, located in the **Toolbar**.



Note: This feature is only available in the web interface of the Team Edition.

Private Project

You have a default project named **Private** where you can create connections, scripts, tasks, diagrams, bookmarks and datasets that are not visible to the Administrator. This project operates as a personal space, safeguarding sensitive information and providing a workspace free from disturbances.

To enable or disable the creation of private projects, an administrator needs to go to **Settings -> Administration -> Server Configuration** and toggle **Enable private connections**.

Web Version:

- To access your private project, select the **Private** project using the **Project Selector** button. If you have enabled the **Multiple projects** feature, you can open several projects simultaneously, including your private one.

Desktop Version:

- To access your private project in the desktop interface, navigate to the **Private** project in the **Project** view, right-click on it and select **Set project as active**.

Teams in Team Edition

Table of contents

[Overview](#)

[Team creation](#)

[Predefined Team types](#)

[Integration with Identity Providers](#)

[Configuration steps](#)

[Automatic membership management](#)

[Updating Team memberships](#)

[User Management](#)

[Supervisor](#)

[Project Management](#)

Overview

The Team Edition provides a comprehensive team management feature tailored for both web and desktop environments. This feature allows administrators to create and manage Teams effectively, grouping users together to facilitate project-based permissions and credential sharing.

In the Team Edition, the process of managing team memberships differs slightly between the web and desktop versions, primarily in the user interface layout.

Team creation

To create a new Team, follow these steps:

Web version:

1. As administrator, go to the **Settings -> Administration -> Access Management -> Teams**.
2. Click on the **+ Add** button.
3. Fill in the necessary details in the provided fields.

Desktop version:

1. As administrator, navigate to **Window -> Show View -> Administration**.
2. Select the **Teams** tab.
3. Click on the **+ Create** button.
4. Specify the name of the Team in the window that appears.

- After creating the Team, select it from the list.
- Fill in the necessary details for the selected Team.

Field Name	Description	Additional Info
Team ID (only in the web version)	A unique identifier for the Team.	
Team Name	The name of the Team.	
Description (only in the web version)	A brief description of the Team and its purpose.	
Parameters	Additional parameters based on the authentication provider.	Read more about Integration with Identity Providers .

Predefined Team types

Team Edition includes `all-users` predefined Team type. This Team includes all users by default.

Integration with Identity Providers

You have the ability to integrate Teams with various identity providers. This integration allows for the utilization of roles and groups defined by your identity provider to manage Team memberships automatically.

Configuration steps

- When creating or editing a Team, navigate to the **Parameters** section.
- Here, depending on your identity provider, you can associate the Team with a specific identity attribute:

Provider	Attribute	Related articles
AWS	<code>AWS Role ARN</code>	AWS OpenID , AWS OpenID via Okta
SAML	<code>SAML Group ID</code>	SAML configuration
Microsoft Entra ID	<code>Microsoft Entra ID Group ID</code>	Microsoft Entra ID

Okta OpenID	OKTA Group ID	Okta OpenID
-------------	---------------	-----------------------------

Automatic membership management

Once the integration is set up, whenever a user authenticated by the configured identity provider logs into Team Edition, the application will check for matching identity attributes. If there is a match with any of the defined parameters within Teams, the user will be automatically assigned to the appropriate Team.

Updating Team memberships

For the changes to take effect, especially in cases where group memberships are updated:

- Users may need to log off and log back in through the Single Sign-On (SSO).
- Alternatively, changes will take effect after the session timeout.

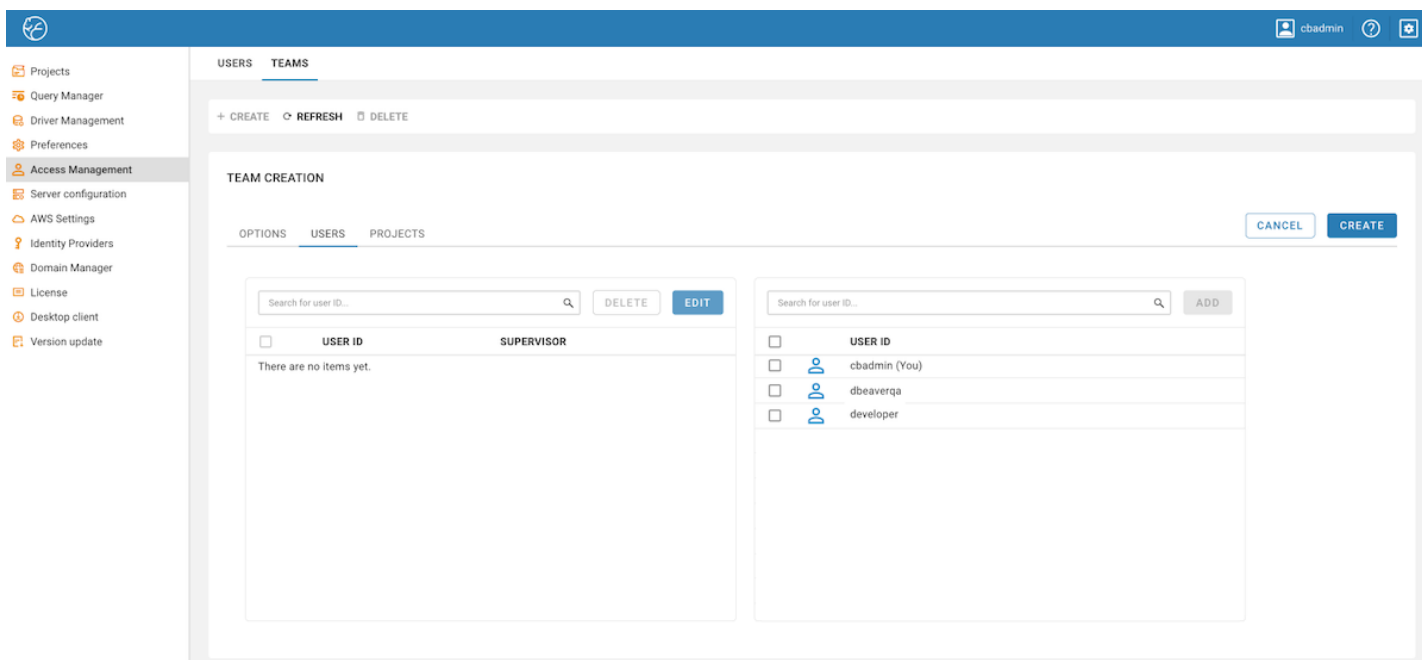
These actions ensure that the updated claims from the identity provider are received by Team Edition, thereby refreshing the Team memberships.

User Management

Web version:

In the **Users** tab, administrators can manage Team memberships:

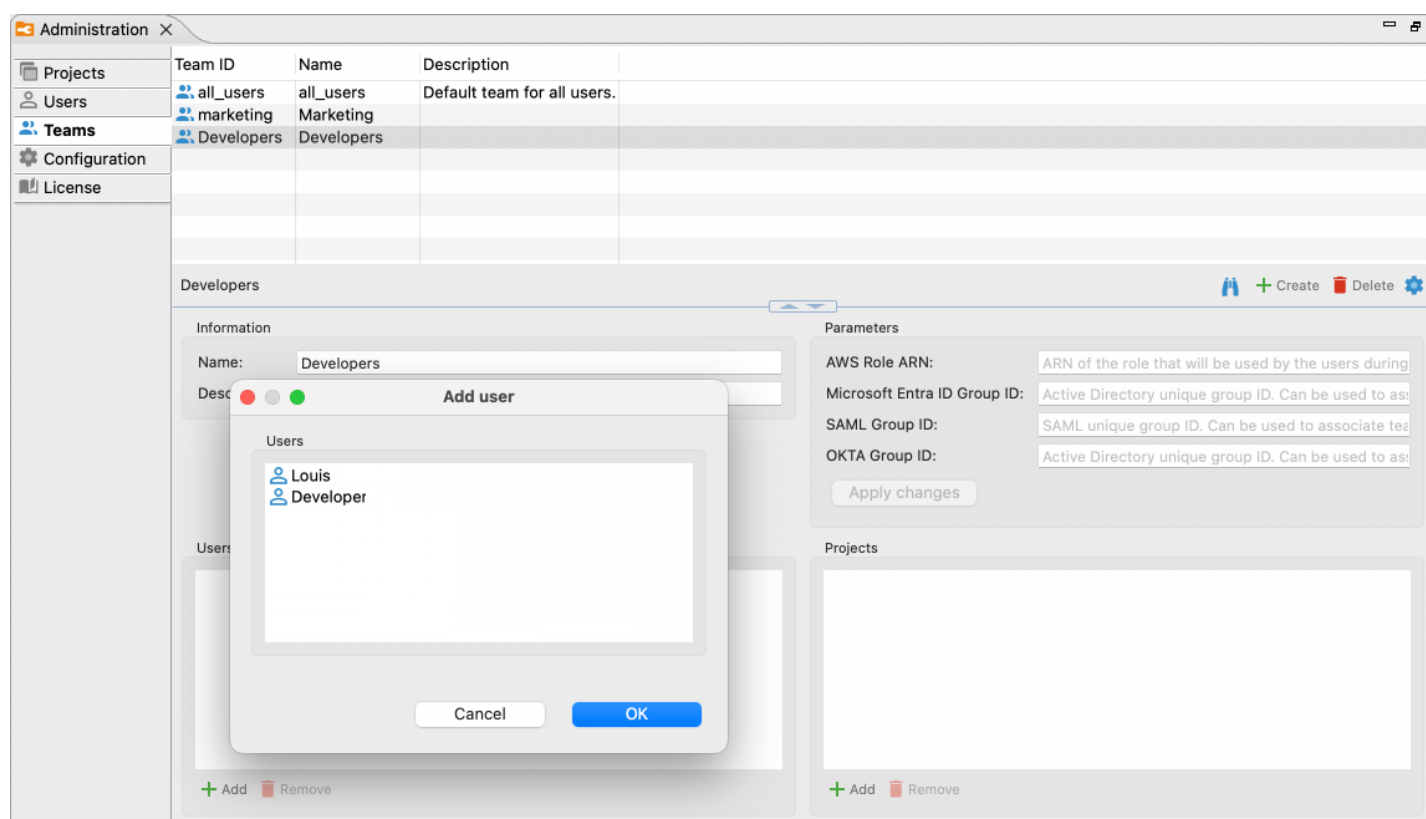
- To add a user to the Team, click **Edit**, select the desired users, and then click **Add**.
- To remove a user from the Team, select the user and click **Delete**.



Desktop version:

In the **Administration** window, administrators can manage Team memberships:

- To add a user to the Team, click **+ Add**, select the desired users, and then click **Ok**.
- To remove a user from the Team, select the user and click **Delete**.



Tip: One user can be a member of multiple teams.

Supervisor

In the **Users** tab, you can use the **Supervisor** checkbox to mark certain users as supervisors. Supervisors can view their team's queries in the [Query Manager](#).

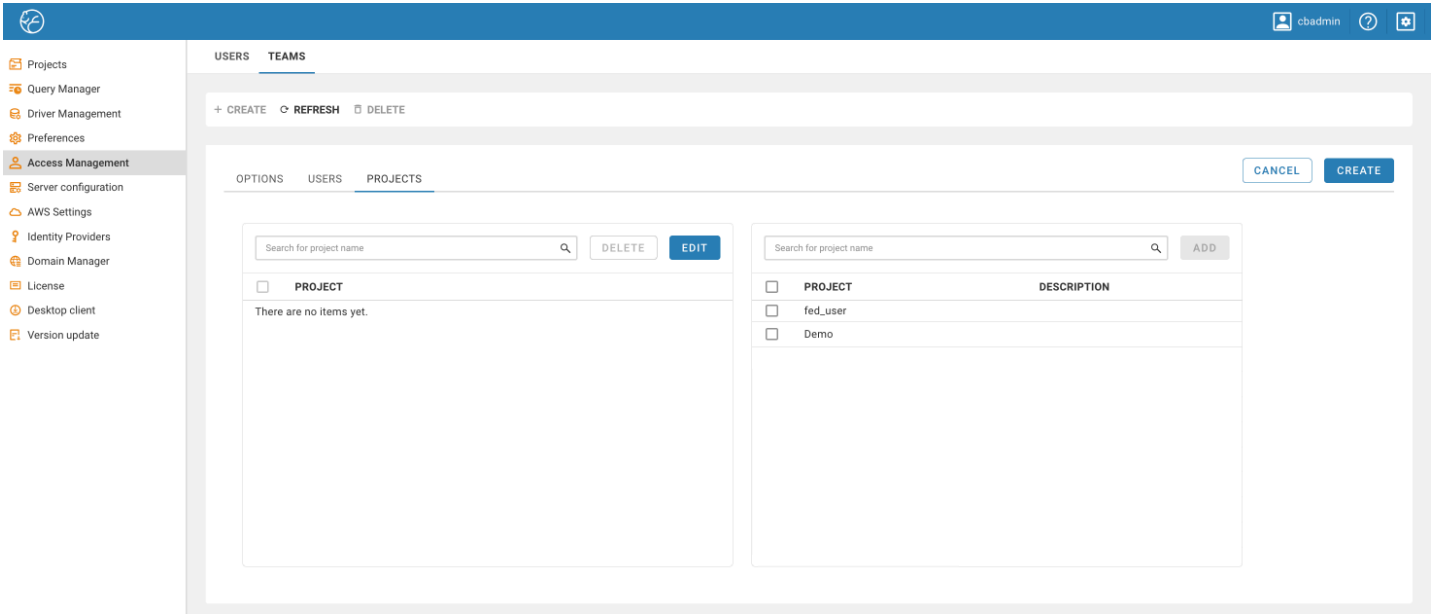
Note: The checkbox for assigning supervisor roles is exclusive to the web interface, but the capabilities granted by this permission are accessible in both the web and desktop versions.

Project Management

Web version:

In the **Projects** tab, administrators can manage which projects are available to the Team:

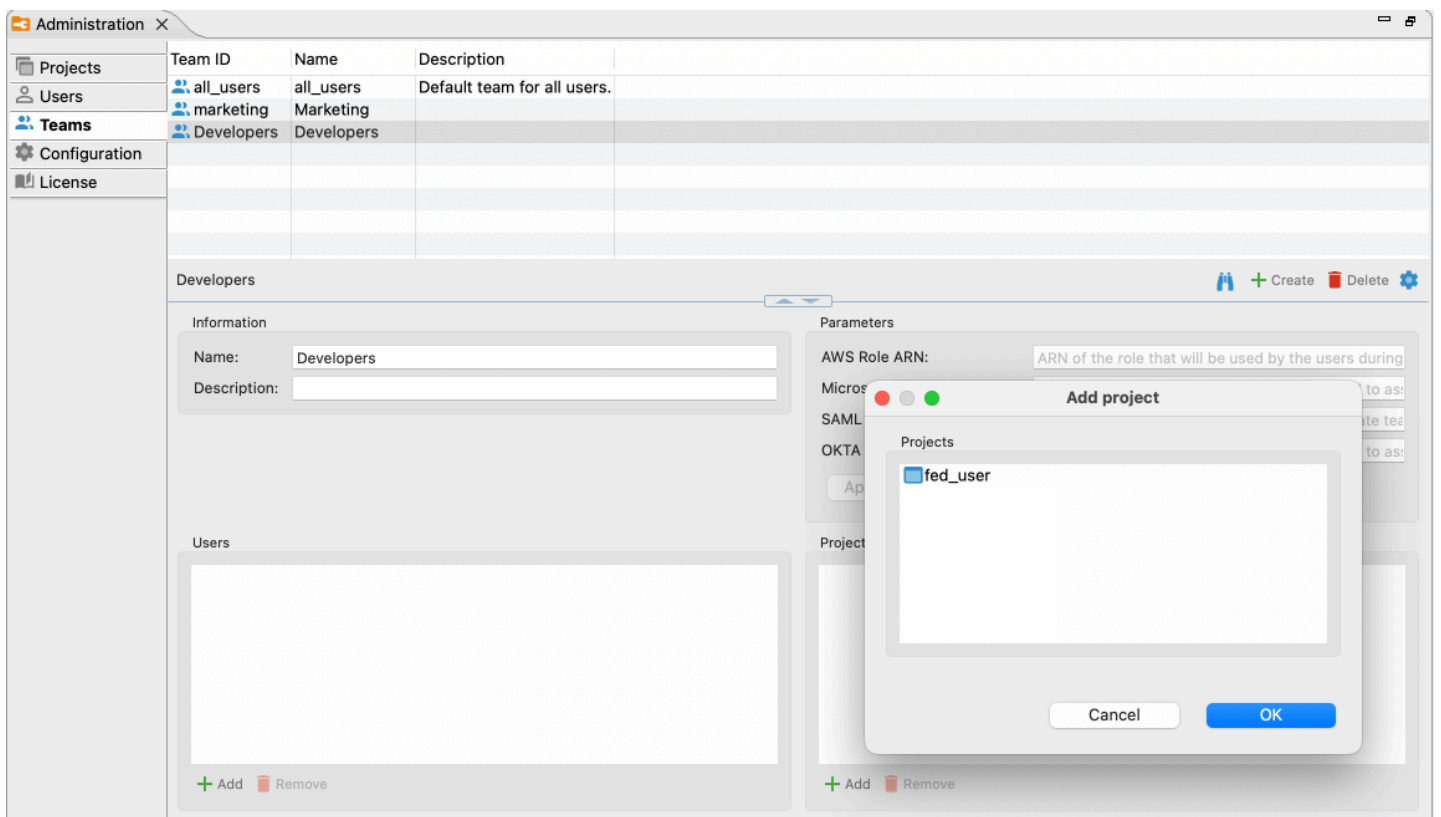
- To add a project to the Team, click **Edit**, choose the desired project, and then click **Add**.
- To remove projects from the Team, select the project and click **Delete**.



Desktop version:

In the **Projects** tab, administrators can manage which projects are available to the Team:

- To add a project to the Team, click **Edit**, choose the desired project, and then click **Add**.
- To remove projects from the Team, select the project and click **Delete**.



Tip: One project can be a part of multiple teams.

Roles in Team Edition

Table of contents

[Overview](#)

[Administrator](#)

[Capabilities](#)

[Developer](#)

[Capabilities](#)

[Manager](#)

[Capabilities](#)

[Editor and Viewer](#)

[Capabilities](#)

[Summary](#)

Overview

Roles in DBeaver Team Edition are designed to manage user access effectively and enhance security. This feature allows administrators to assign specific functionalities and access levels to different users, ensuring each team member has the necessary tools while maintaining tight control over sensitive data.

Roles are structured in a tiered manner, each encompassing the functionalities of more task-specific roles:

Administrator

The Administrator role in DBeaver Team Edition is central to configuring the server and managing settings. This role is essential for ensuring that team members can efficiently access the necessary resources without compromising data security.

Important: At least one of the roles must be an Administrator role. There is no limit on the amount number of Administrators that are allowed.

Capabilities

- **Configuring server and settings:** Responsible for setting up the server and managing settings for other users.
- **Creating and managing shared projects:** Administrators can create shared projects containing necessary database connections for team collaboration. [Learn more.](#)
- **Monitoring user activity:** Using the Query manager in the admin panel, Administrators can view, filter, and analyze users' activities.
- **Full application access:** Post configuration, Administrators have access to all the functionalities of the desktop and web applications.
- **Role assignment:** Determines the functionalities available to other team members based on their assigned roles.

Developer

The Developer role is designed for users who need comprehensive access to the features of DBeaver Team Edition, focusing on database administration and project development.

Capabilities

- **Full feature access:** Complete access to all desktop and web version features of DBeaver Team Edition, including database administration.
- **Connection and script management:** Ability to configure new and edit existing connections, as well as create SQL scripts and resources.

Note: Unlike Administrators, Developers do not manage the server, users, licenses, or track user activity.

Manager

The Manager role is ideal for specialists like data analysts, who are proficient in writing SQL queries but do not partake in software development or connection setup.

Capabilities

- **SQL query execution:** Managers can access connections in shared projects for database data retrieval.
- **Application access:** Can use web and desktop applications to view and create scripts and datasets, with editing capabilities based on permissions.

Tip: For detailed information about datasets in DBeaver, please take a look at the article [Datasets in Team Edition](#).

Editor and Viewer

The Editor and Viewer roles in Team Edition are tailored for users who primarily interact with data through the web application. These roles are crucial for tasks like report generation, data processing, and analysis, leveraging datasets for various needs.

Capabilities

- **Data interaction:** Both roles enable viewing, browsing, filtering, and exporting datasets.
- **Data modification:** Editors can modify data if they are given the correct permissions.

Summary

The following table summarizes the capabilities associated with each role in DBeaver Team Edition:

Capability \ Roles	Administrator	Developer	Manager	Editor	Viewer
Full system administration	+	-	-	-	-
Manage users and licenses	+	-	-	-	-
Create/delete projects	+	-	-	-	-
Edit connections	+	+	-	-	-
Private project	+	+	-	-	-

View connections	+	+	+	-	-
Create/edit/delete resources and scripts	+	+	+	-	-
Create/edit datasets	+	+	+	-	-
Create/edit folders	+	+	+	-	-
Use DBeaver desktop	+	+	+	-	-
In-line data editing	+	+	+	+	-
View, browse, filter, export datasets	+	+	+	+	+

Git integration in Team Edition

Table of contents

[Overview](#)

[Git configuration](#)

[Enable Git services](#)

[General Git settings](#)

[Project Git settings](#)

[Testing](#)

Overview

Team Edition has integration with the Git version control system. You can keep your project scripts, diagrams, datasets, connection configurations, bookmarks, and other data in a Git repository.

Git configuration

Enable Git services

Firstly, in order for you to be able to configure git, you need to enable git services. For this:

- go to the Administration menu
- enable `Git` services in the `Server configuration` tab.

- save server configuration

The screenshot shows the 'Server configuration' page in the CloudBeaver administration interface. The left sidebar contains a menu with items: Projects, Query Manager, Driver Management, Access Management, **Server configuration** (highlighted), Git Settings, Identity Providers, License, and Version update. The main content area is divided into two panels. The left panel, titled 'SERVER INFORMATION', contains three fields: 'Server Name *' with the value 'Cloudbeaver TE Web Server', 'Server URL *' with the value 'http://cloudbeaver', and 'Session lifetime, min *' with the value '30'. The right panel, titled 'CONFIGURATION', contains two toggle switches: 'Enable private connections' (disabled) and 'Navigator simple view' (disabled). Below these is a 'SERVICES' section with four toggle switches: 'Azure' (disabled), 'GCP' (disabled), 'AWS' (disabled), and 'Git' (enabled). The top of the interface has a blue header bar with a 'Project' dropdown, a user profile icon labeled 'cbadmin', and help and settings icons.

General Git settings

Now, you need to set the global git settings and set up the credentials of the technical user of your git service. These credentials will be used to access your repositories.

- open the **Git settings** tab
- enter the technical user's username
- enter the technical user's password
- enter the technical user's email

- save configuration

Project ▾ cbadmin ? +

Projects
Query Manager
Driver Management
Access Management
Server configuration
Git Settings
Identity Providers
License
Version update

SAVE × CANCEL

GIT SETTINGS

Here, you can specify technical account credentials for Git integration. For password field you can use a password or a personal access token. You may optionally specify the email shown in the commit message for the committer.
These credentials will be used for accessing repositories specified within the project's configuration settings. Account must have permissions to clone and push to those repositories.

Username * **Password *** **Email**

committer_username committer@example.com

Project Git settings

Each project has its own git settings and is configured separately. Git can be enabled for existing projects, as well as for new ones To configure the project:

- go to the **Projects** tab
- select an existing project or create a new one
- open **Git** tab in the project settings
- set up git parameters:

Parameter	Description
Enabled	Enable/disable git integration for the project
Repository URL	Link to your remote repository
Branch name (optional)	The name of the branch that will be used to save all changes in the project. If no branch is specified, the default branch specified for the repository will be used. !!!WARNING!!! application does not automatically create branches, so if you want to use a branch other than the default, you must create it by yourself in the remote repository, and only then specify it in the project configuration

Data to sync with git	Select the data you want to store in your repository, non-selected types will be ignored and only stored inside the resource manager file system
-----------------------	--

Project

Projects

Query Manager

Driver Management

Access Management

Server configuration

Git Settings

Identity Providers

License

Version update

+ ADD

REFRESH

DELETE

ID	NAME	DESCRIPTION
<input type="checkbox"/> ^ s_Project	Project	

INFO

ACCESS

GIT

GIT SETTINGS

Enabled

Repository URL *

https://example-host.com/example/example-repository

Branch name

custom_branch_name

DATA TO SYNC WITH GIT

Select the data you want to sync with Git. Elements left unselected will be ignored.

☒ Scripts

☒ Diagrams

☐ Data source credentials

☐ Project metadata

☐ Tasks configuration

☒ Datasets

☐ Bookmarks

☐ Data sources

☐ Project settings

CANCEL

SAVE

When setting up the project for the first time, the project and repository files will be synchronized. **!!!WARNING!!!** When synchronizing, the files in the project are considered to be of higher priority than the files in the repository, so in case of a conflict, the files of the repository will be overwritten. In other cases, the repository will store both files from the project and files that were previously in the repository.

Commits

master

Commits on Sep 4, 2023

CloudBeaver: update .gitignore

cbadmin

 authored and

committer

 committed 4 minutes ago

48028e4

<>

CloudBeaver: project synchronization

Resource Manager Web Server

 authored and

committer

 committed 4 minutes ago

ec2914f

<>

Newer

Older

Seeing something unexpected? Take a look at the [GitHub commits guide](#).

Testing

After all settings, any changes to the project resources will be automatically synchronized with the git. The commit message will indicate the changed resource, author of the commit will be the user who made the changes.

CloudBeaver: update Scripts/sql-script.sql

cloudbeaver_user

 authored and

committer

 committed 1 minute ago

483cd60

<>

Datasets in Team Edition

Table of contents

[Datasets in the web version](#)

[Create using the Data Editor](#)

[Create using the SQL Editor](#)

[Create using the Grouping Panel](#)

[Editing Datasets](#)

[Datasets in the desktop version](#)

[Create using the Data Editor](#)

[Create using the SQL Editor](#)

[Create using the Grouping Panel](#)

[Editing Datasets](#)

Note: This feature is available in [Team Edition](#) edition only.

A Dataset is a query result set presented in the form of a data grid that analytical team members can effectively and safely use with [Editor or Viewer roles](#) in Team Edition.

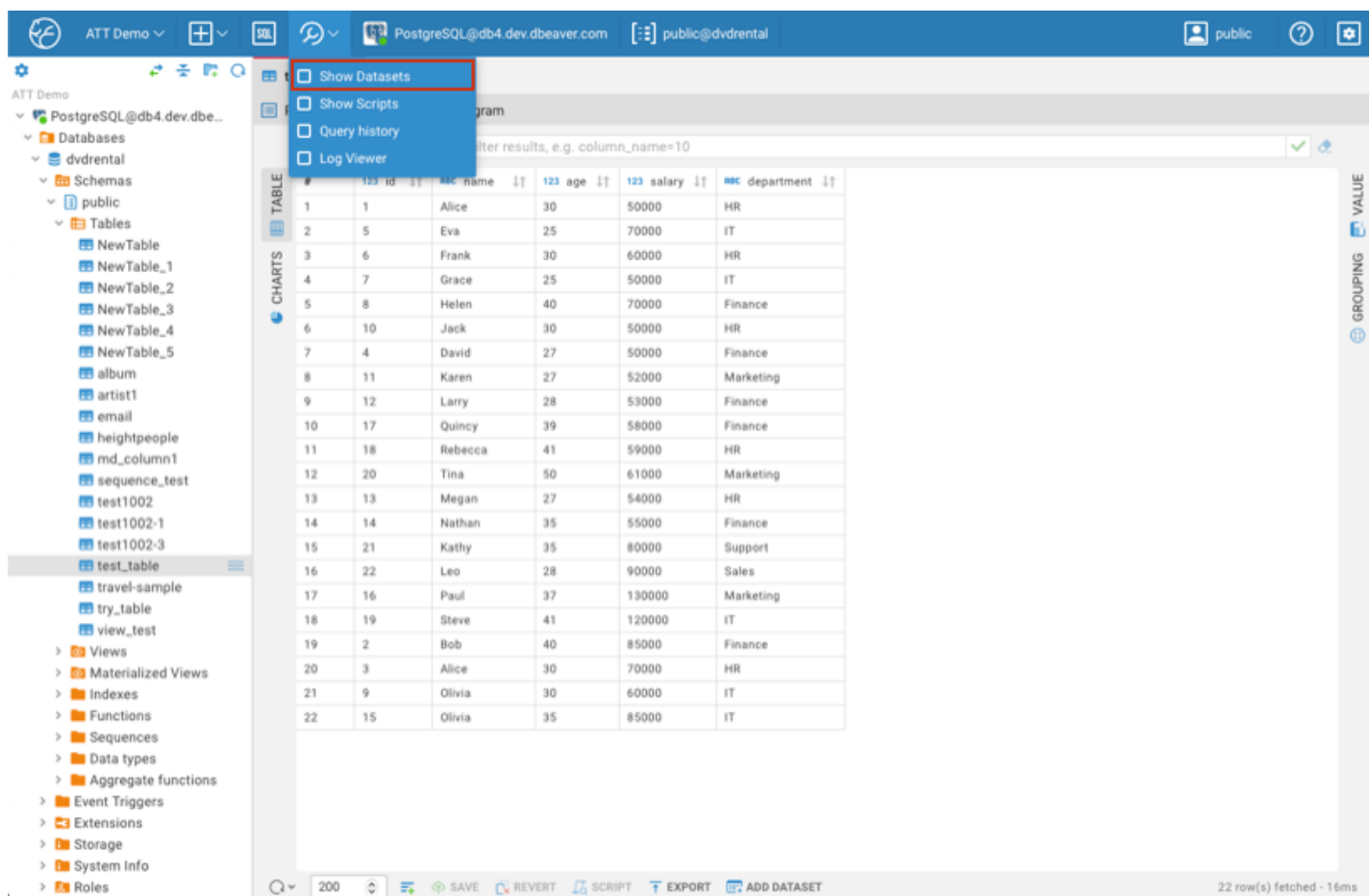
[Administrators, Developers, and Managers](#) create Datasets using existing tables for their non-technical colleagues so that they can conduct data analysis and create reports without going deep into database mechanisms.

Note: Dataset changes are immediately available to colleagues. If you delete or edit a **Dataset** in the web version of **Team Edition**, it will immediately disappear in the desktop version of **Team Edition**, and vice versa.

Datasets are stored in the **Resource Manager server** -> **Team workspace** -> **Project** -> **Datasets** folder.


Datasets in the web version

By checking the **Show Datasets** button, Administrators, Developers, and Managers can open the **Datasets** folder to make their work convenient.



There are several ways to create a new **Dataset** in the web version.

Create using the Data Editor

You can unfold your database connection in the [Database Navigator](#), access the existing table, open it in the main view, and use the **Add Dataset** button  **ADD DATASET** at the bottom.

The screenshot shows the CloudBeaver web interface. On the left, a sidebar lists the database structure: PostgreSQL@db4.dev.dbe..., dvdrental, public, and various tables like NewTable_1 through NewTable_5, album, artist1, email, heightpeople, md_column1, sequence_test, test1002, test1002-1, test1002-3, test_table, travel-sample, try_table, and view_test. The main area displays a table named 'test_table' with the following data:

#	id	name	age	salary	department
1	1	Alice	30	50000	HR
2	5	Eva	25	70000	IT
3	6	Frank	30	60000	HR
4	7	Grace	25	50000	IT
5	8	Helen	40	70000	Finance
6	10	Jack	30	50000	HR
7	4	David	27	50000	Finance
8	11	Karen	27	52000	Marketing
9	12	Larry	28	53000	Finance
10	17	Quincy	39	58000	Finance
11	18	Rebecca	41	59000	HR
12	20	Tina	50	61000	Marketing
13	13	Megan	27	54000	HR
14	14	Nathan	35	55000	Finance
15	21	Kathy	35	80000	Support
16	22	Leo	28	90000	Sales
17	16	Paul	37	130000	Marketing
18	19	Steve	41	120000	IT
19	2	Bob	40	85000	Finance
20	3	Alice	30	70000	HR
21	9	Olivia	30	60000	IT
22	15	Olivia	35	85000	IT

A dialog box at the bottom left indicates 'Dataset successfully added' for 'test_dataset' at 17:01:49. The right sidebar shows a list of datasets, including 'test_dataset'. The bottom status bar shows 'Success - 26ms'.

It will appear on the right panel after clicking the **Add Dataset** button and entering the new **Dataset** name.

Create using the SQL Editor

Another way to create a **Dataset** is to run the query you need in the [SQL Editor](#) and use the **Result tab** as a base. It will appear on the right after clicking the **Add Dataset** button and entering the new **Dataset** name.

SQL Editor

```

SELECT * FROM test_table tt
WHERE id > 10;

```

Statistics - 1

Result - 1 <1>


Enter a SQL expression to filter results, e.g. column_name=10

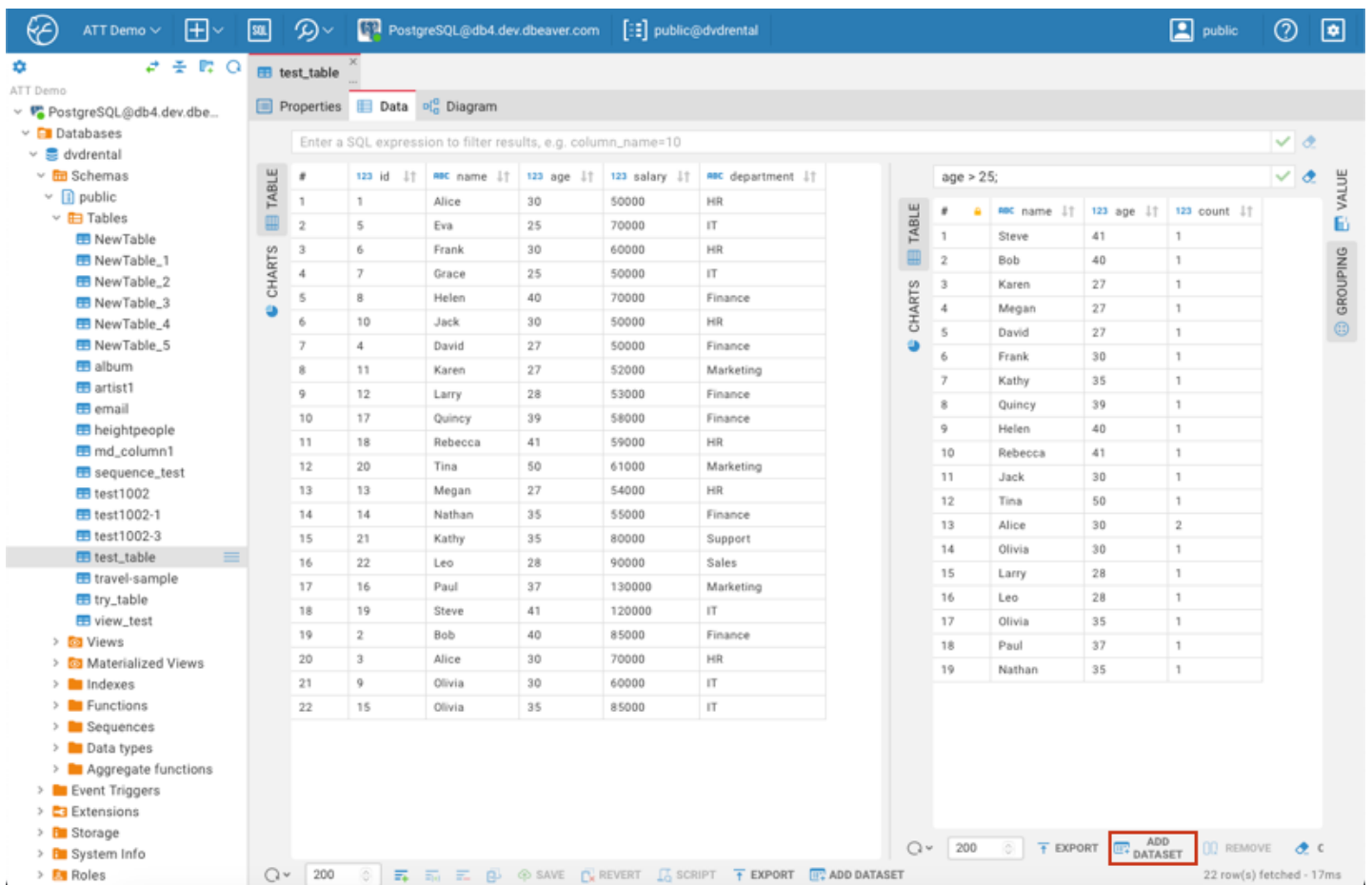
#	id	name	age	salary	department
1	11	Karen	27	52000	Marketing
2	12	Larry	28	53000	Finance
3	17	Quincy	39	58000	Finance
4	18	Rebecca	41	59000	HR
5	20	Tina	50	61000	Marketing
6	13	Megan	27	54000	HR
7	14	Nathan	35	55000	Finance
8	21	Kathy	35	80000	Support
9	22	Leo	28	90000	Sales
10	16	Paul	37	130000	Marketing
11	19	Steve	41	120000	IT
12	15	Olivia	35	85000	IT

ADD DATASET

Success - 25ms

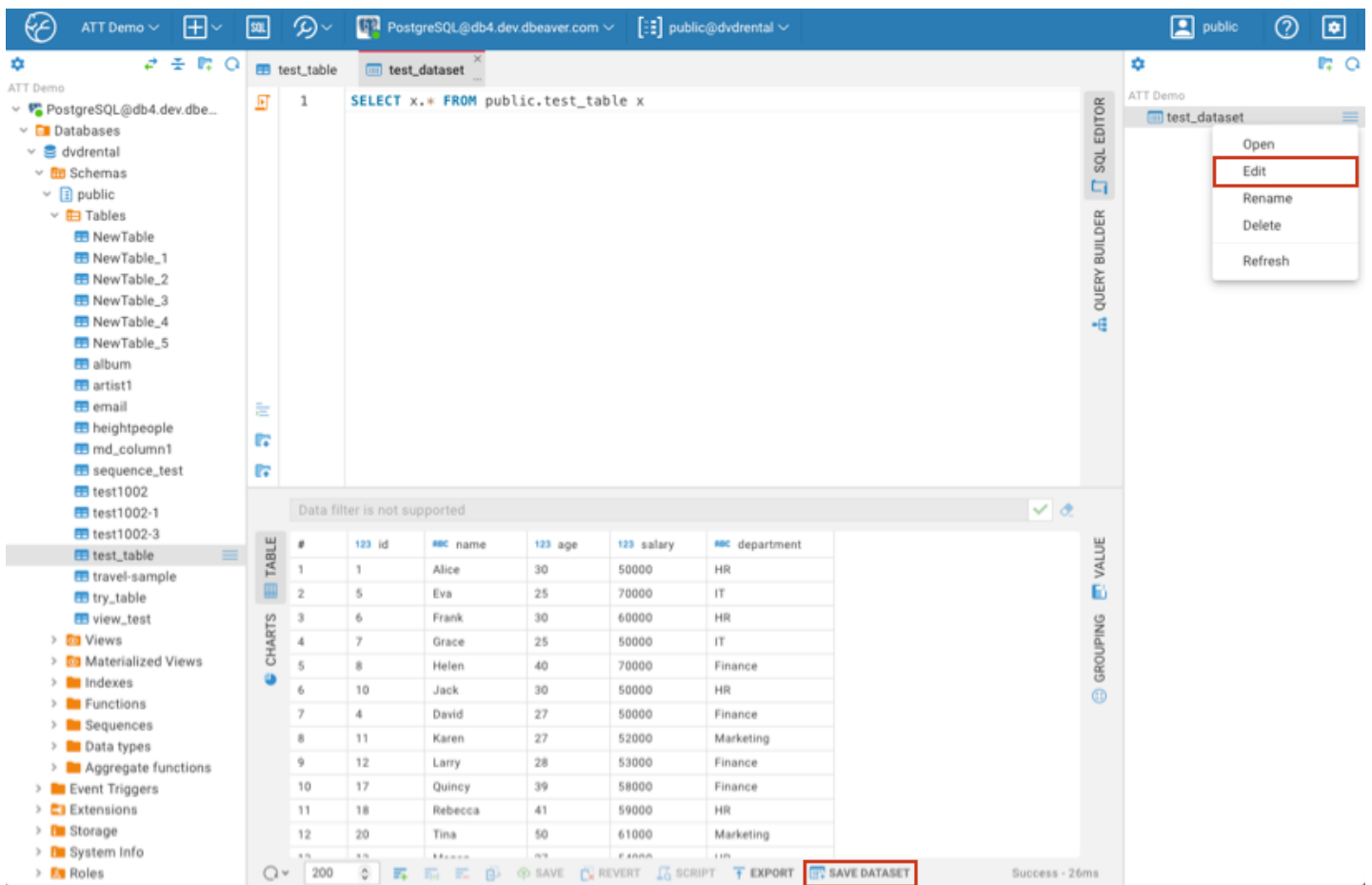
Create using the Grouping Panel

You can also create a **Dataset** from the **Grouping** panel. Open the Grouping window by clicking the corresponding button , move the column/s from the **Data Editor** (enter filters if necessary), and click the **Add Dataset** button.



Editing Datasets

Right-click your **Dataset** on the right panel and get the context menu with the following options: **Open**, **Edit**, **Rename**, **Delete**, and **Refresh**. When you choose **Edit**, the `SELECT` statement used for the **Dataset** creation appears in the main view. You can edit it by specifying additional clauses. Remember to click the **Save Dataset** button at the bottom. The **Refresh** option is useful when you want to get the latest changes from your colleagues who work in another **Team Edition** client.



Datasets in the desktop version

There are also several ways to create a **Dataset** in the desktop version of Team Edition.

Create using the Data Editor

You can access your existing table in the **Database Navigator**, open it in the main view, right-click inside **Data Editor** window, and choose the **Create Dataset** option in the context menu.

Tip: After entering the name and saving, you can access your **Dataset** in the **Datasets** folder in the **Database Navigator**.

The screenshot shows the CloudBeaver SQL Editor interface. The main window displays a table named 'test_table' with the following data:

id	name	age	salary	department
1	Alice	30	50,000	HR
5	Eva	25	70,000	IT
6	Frank	30	60,000	HR
7	Grace	25	50,000	IT
8	Helen	40	70,000	Finance
10	Jack	30	50,000	HR
4	David	27	50,000	Finance
11	Karen	27	52,000	Marketing
12	Larry	28	53,000	Finance
17	Quincy	39	58,000	Finance
18	Rebecca	41	59,000	HR
20	Tina	50	61,000	Marketing
13	Megan	27	54,000	HR
14	Nathan	35	55,000	Finance
21	Kathy	35	80,000	Support
22	Leo	28	90,000	Sales
16	Paul	37	130,000	Marketing
19	Steve	41	120,000	IT
2	Bob	40	85,000	Finance
3	Alice	30	70,000	HR
9	Olivia	30	60,000	IT
15	Olivia	35	85,000	IT

A right-click context menu is open over the table, showing the following options:

- Copy
- Advanced Copy
- Paste
- Advanced Paste ...
- Filter
- Order
- Navigate
- Edit
- View/Format
- Logical structure
- Layout
- Export data
- Open with
- Generate SQL
- Create Dataset** (highlighted)
- Generate Mock Data
- Refresh
- Toggle result panels

Create using the SQL Editor


You can also create a new **Dataset** from the **Result tab** of the **SQL Editor**. Run the query you need, right-click inside **Result tab** window, and choose the **Create Dataset** option.

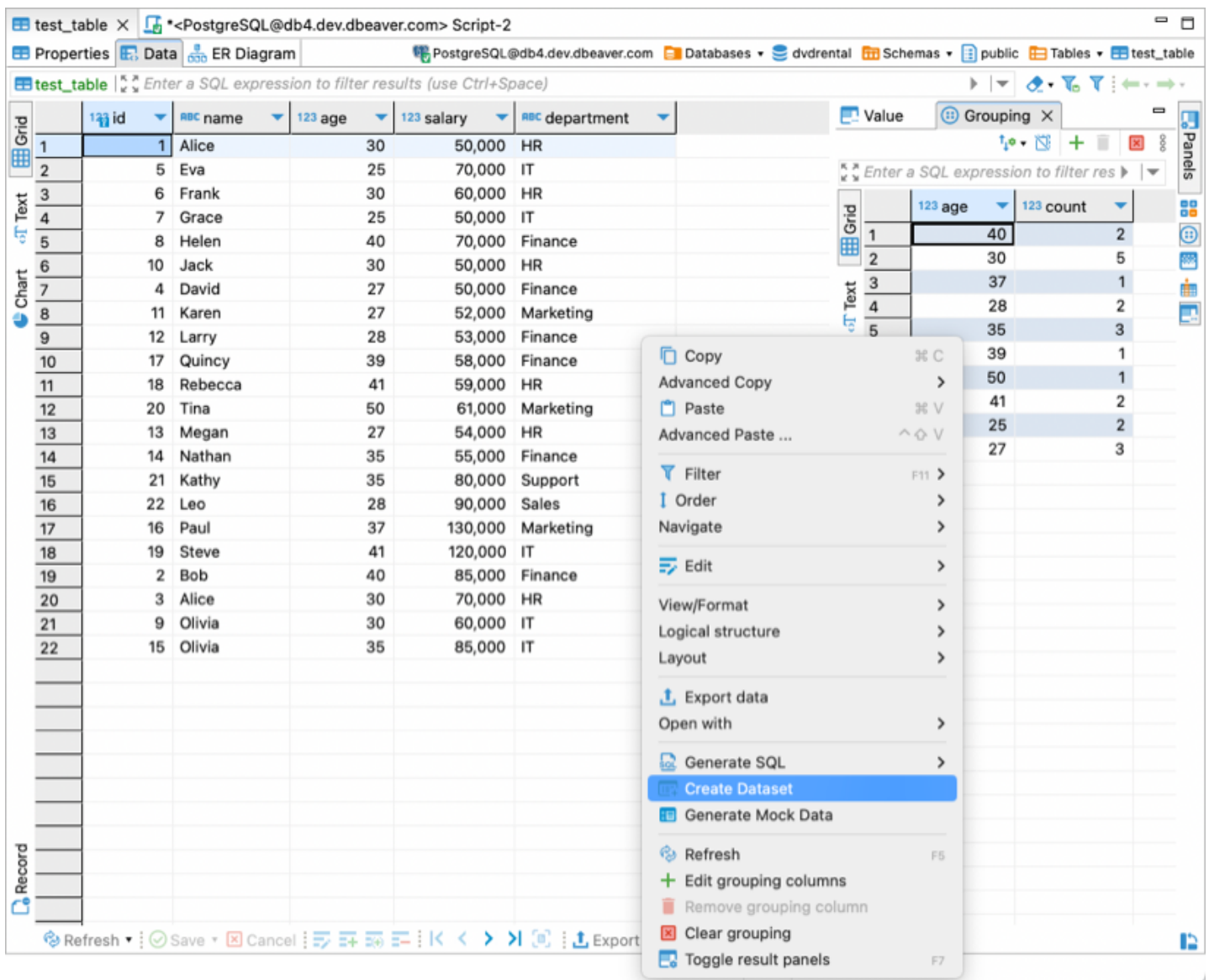
The screenshot shows the DBeaver SQL Editor interface. At the top, the SQL Editor pane contains the query: `select * from test_table tt where age > 30;`. Below it, the 'test_table 1' pane displays the query results in a table grid. A context menu is open over the table, showing various actions like 'Copy row number(s)', 'Advanced Copy', 'Paste', 'Filter', 'Order', 'Navigate', 'Edit', 'Logical structure', 'Layout', 'Export data', 'Open with', and 'Create Dataset' (highlighted in blue). The table data is as follows:

	id	name	age	salary	department
1	8	Helen	40	70,000	Finance
2	17	Quincy	39	58,000	Finance
3	18	Rebecca	41	59,000	HR
4	20	Tina	50	61,000	Marketing
5	14	Nathan	35	55,000	Finance
6	21	Kathy	35	80,000	Support
7	16	Paul	37	130,000	Marketing
8	19	Steve	41	120,000	IT
9	2	Bob	40	85,000	Finance
10	15	Olivia	35	85,000	IT

At the bottom of the interface, there are buttons for 'Refresh', 'Save', 'Cancel', and 'Export data'. A status bar indicates '10 row(s) fetched - 85ms, on 2023-08-22 at 17:16:27'.

Create using the Grouping Panel

You can also create a **Dataset** from the **Grouping panel**. Open the Grouping window by clicking the corresponding button , move the column from the **Data Editor** (enter filters if necessary), and click the **Add Dataset** button.



Editing Datasets

In the desktop edition, **Datasets** are stored in the **Datasets** folder in the **Database Navigator**. You can navigate to this folder, select the desired **Dataset**, and right-click to access the context menu with options to **Open Dataset**, **Delete**, **Rename**, and **Refresh**.

Docker image

Table of contents

[Requirements](#)

[Installation](#)

[Running](#)

[Daemon mode](#)

CloudBeaver is a web application, that requires server deployment.

Requirements

- [Install Docker](#) 20.10 or higher
- amd64 or arm64 system.

Installation

To install the latest version of CloudBeaver use the following script:

```
docker pull dbeaver/cloudbeaver:latest
```

Running

To run CloudBeaver in the terminal:

```
docker run --name cloudbeaver --rm -ti -p 8080:8978 -v /opt/cloudbeaver/workspace dbeaver/cloudbeaver:latest
```

Then switch to the browser and open <http://localhost:8080/>

Daemon mode

Add the following parameters:

```
-d --restart unless-stopped
```

More information can be found on the [Run Docker Container page](#)

AWS Marketplace

Table of contents

[Permissions](#)

[Deployment](#)


Permissions

To use this method, you need an account on Amazon Marketplace with the following permissions:


- [FAWSMarketplaceRead-only](#)
- aws-marketplace:Subscribe on resource

Deployment



1. Log in to [Amazon Marketplace](#), find [CloudBeaver AWS](#), and press **Continue to Subscribe** button.



CloudBeaver Community

By: [DBeaver Corporation](#)  Latest Version: CloudBeaver Community 23.2.0

Database management tool for relational databases

Linux/Unix  [0 AWS reviews](#) | [1 external review](#) 

[Free Trial](#)

[Continue to Subscribe](#)
[Save to List](#)
Typical Total Price
\$0.096/hr
Total pricing per instance for services hosted on t2.medium in US East (N. Virginia). [View Details](#)

2. Accept DBeaver Corporation offer by pressing **Accept Terms** and wait for request processing.

[< Product Detail](#)[Subscribe](#)

Subscribe to this software

To create a subscription, review the pricing information, and accept the terms for this software. You can also create a long term contract on this page.

Terms and Conditions

DBeaver Corporation Offer

By subscribing to this software, you agree to the pricing terms and the seller's [End User License Agreement \(EULA\)](#). You also agree and acknowledge that AWS may, on your behalf, share information about this transaction (including your payment terms) with the respective seller, reseller or underlying provider, as applicable, in accordance with the [AWS Privacy Notice](#). AWS will issue invoices and collect payments from you on behalf of the seller through your AWS account. Your use of AWS services is subject to the [AWS Customer Agreement](#) or other agreement with AWS governing your use of such services. If you are receiving a private offer from a channel partner, you may click [here](#) (for CPPO transaction) or [here](#) (for SPPO transaction) for more information on the channel partner.

[Accept Terms](#)

Then press **Continue to Configuration**.

[Continue to Configuration](#)

Thank you for subscribing to this product! You can now configure your software.

[X](#)[< Product Detail](#)[Subscribe](#)

Subscribe to this software

You're subscribed to this software. Please see the terms and pricing details below or click the button above to configure your software.

Configure your software contract

You can reduce the expense of this software by purchasing a software contract. Select the contract that aligns with your requirements. The fee that you pay will cover the software cost through the contract period. After confirming your contract purchase, the full fee will be charged to your AWS bill. Any software usage beyond the contract limit will receive additional charges based on consumption pricing.

Software contract

Select contract option(s)

Total contract price **\$0**
Due now

[Create Contract](#)

3. Select CloudBeaver version you need in **Software version** field, select region, and press **Continue to Launch**.

4. Select the desired configuration:

- In the **EC2 Instance Type** select t2.medium or more.
- In **Security group settings**, press **Create New Based On Seller Settings**. Then press **Launch**.



CloudBeaver Community

Security Group Settings

A security group acts as a firewall that controls the traffic allowed to reach one or more instances. You can create a new security group based on seller-recommended settings or choose one of your existing groups. [Learn more](#)

Select a security group



Create New Based On Seller Settings

5. Great! CloudBeaver instance is successfully deployed on EC2.



CloudBeaver Community

[< Product Detail](#)

[Subscribe](#)

[Configure](#)

[Launch](#)

Launch this software

Congratulations! An instance of this software is successfully deployed on EC2!

AMI ID: **ami-0692e8c6caa5344d6** ([View Launch Configuration Details](#))

You can view this instance on [EC2 Console](#). You can also view all instances on [Your Software](#). Software and AWS hourly usage fees apply when the instance is running and will appear on your monthly bill.

6. Click **EC2 Console** link, and you will see the list of your instances. The new instance will be the last one, and without a name.

- Click on the empty field to give the instance a name.
- Click on the Instance ID, and you will see the Instance summary.

Instances (5) Info					
<input type="text"/> Find Instance by attribute or tag (case-sensitive)					
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	
<input type="checkbox"/>		i-091a4e23d0f5634d4	Running	t2.medium	
<input type="checkbox"/>		i-03b0007e551373637	Running	t2.medium	
<input type="checkbox"/>		i-08cbcd2bfff33adab	Running	t2.small	
<input type="checkbox"/>		i-04d47727a97a4aefe	Running	t2.small	
<input type="checkbox"/>		i-072491192033263df	Running	t2.medium	

7. Click on the open address link in the **Public IPv4 DNS** section.

[EC2](#) > [Instances](#) > i-072491192033263df

Instance summary for i-072491192033263df (cloubeaver-ce-test) [Info](#)

Updated less than a minute ago

Instance ID i-072491192033263df (cloubeaver-ce-test)	Public IPv4 address 18.193.69.128 open address
IPv6 address -	Instance state Running

7. The page in your default web browser will be opened. You need to agree to use the self-signed certificate and change it later. [How to change certificate](#)

That's all done! CloudBeaver Community instance is ready to use. On the first page, you will see the [server configuration wizard](#).

Table of contents

[Permissions](#)

[Deployment](#)

[Deployment in Google Cloud interface](#)

[Step 1. Import CloudBeaver Community Edition custom image on your GCP account](#)

[Step 2. Create a new GCP Compute Engine instance from the imported image](#)

[Deployment with Google Cloud CLI](#)

[Setup and control options](#)

[CloudBeaver Community Edition server manager](#)

[Version update procedure](#)

Permissions

You need to have a list of permissions like this in your project:

```
compute.images.create
compute.images.list

compute.disks.create
compute.disks.resize
compute.disks.use
compute.disks.list

compute.instances.reset
compute.instances.resume
compute.instances.setMetadata
compute.instances.start
compute.instances.stop
compute.instances.suspend
compute.instances.update
compute.instances.use
compute.instances.setServiceAccount
compute.instances.create
compute.instances.attachDisk

compute.subnetworks.use
compute.subnetworks.useExternalIp
compute.networks.use
compute.networks.useExternalIp
```

If you don't have them, contact your system administrator or project owner.

Deployment

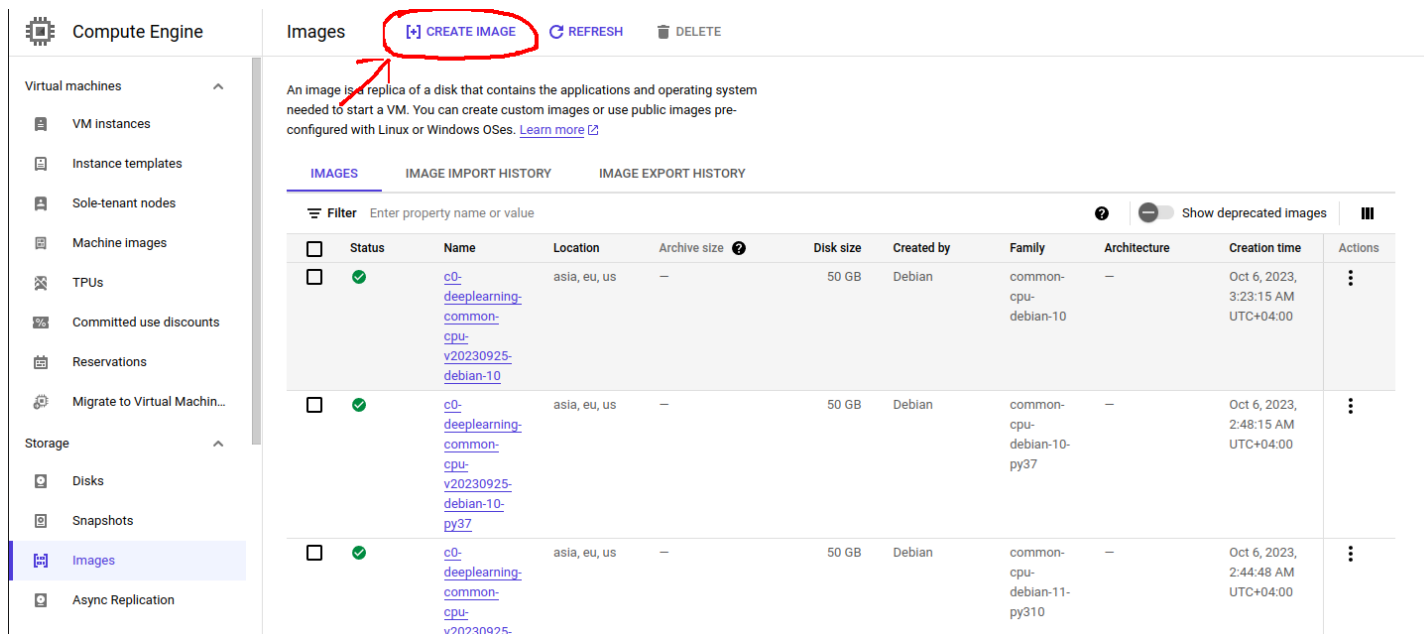
Deployment in Google Cloud interface

To deploy CloudBeaver Community Edition in Google Cloud Console interface, you need to import CloudBeaver Community Edition custom image to your account using the image URI, and then create a new instance from the imported image.

Step 1. Import CloudBeaver Community Edition custom image on your GCP account

1. Log in to your account in [Google Cloud Console](#), navigate to **Compute Engine -> Images**, and click **[+]**

CREATE IMAGE.



The screenshot shows the Google Cloud Console interface for the 'Images' section under 'Compute Engine'. The 'CREATE IMAGE' button is highlighted with a red circle. Below the button, there is a description of what an image is and a link to 'Learn more'. The main part of the interface is a table listing existing images. The table has columns for Status, Name, Location, Archive size, Disk size, Created by, Family, Architecture, Creation time, and Actions. Three images are listed, all with a status of 'OK' and a location of 'asia, eu, us'. The names of the images are truncated in the screenshot.

Status	Name	Location	Archive size	Disk size	Created by	Family	Architecture	Creation time	Actions
OK	c0-deeplearning-common-cpu-v20230925-debian-10	asia, eu, us	—	50 GB	Debian	common-cpu-debian-10	—	Oct 6, 2023, 3:23:15 AM UTC+04:00	⋮
OK	c0-deeplearning-common-cpu-v20230925-debian-10-py37	asia, eu, us	—	50 GB	Debian	common-cpu-debian-10-py37	—	Oct 6, 2023, 2:48:15 AM UTC+04:00	⋮
OK	c0-deeplearning-common-cpu-v20230925-debian-11-py310	asia, eu, us	—	50 GB	Debian	common-cpu-debian-11-py310	—	Oct 6, 2023, 2:44:48 AM UTC+04:00	⋮

2. Fill in the **Create an image** form:

- In the **Name** field write the image name in the following format:

`cloudb Beaver-ce-server-ubuntu/rhel-%version%`

- In the **Source** field select **Virtual disk (VMDK, VHD)**.

- If you are prompted to enable Cloud Build tools and grant permissions, do so.

- Copy the following URI `cloudb Beaver-ce-server/` in the **Virtual disk file** field, click **BROWSE**, and select the version you need.

A screenshot of the CloudBeaver CE Server web interface. The top navigation bar shows a back arrow, the text "cloudbeaver-ce-server" with a dropdown arrow, and icons for a folder with a plus sign and a magnifying glass. The main content area is a table with two rows of virtual machines. Each row has a small square icon with a plus sign in the first column, followed by the VM name in the second column. The first row is "cloudbeaver-ce-rhel-23-2-0.vmdk" and the second row is "cloudbeaver-ce-ubuntu-23-2-0.vmdk".


3. Click **Create**. You may have to wait up to 15 minutes while the CloudBeaver Community Edition server custom image imports to your account.

Step 2. Create a new GCP Compute Engine instance from the imported image


1. Open the tab **Images**, click on the name of the image that you just imported, and click on the **[+] Create instance** button.

←


Images




EDIT



DELETE



CREATE INSTANCE



EXPORT

cloudbeaver-ce-server-ubuntu-23-2-0

Description	Image created by Daisy in workflow "translate" on behalf of root.
Source disk	disk-jv8b0-1
Location	us (United States)
Architecture	—
Labels	<div><div>gce-image-import : true</div><div>gce-image-... : jv8b0</div></div>
Creation time	Oct 30, 2023, 8:05:57 PM UTC+04:00
Encryption type	Google-managed

EQUIVALENT REST

2. Give your instance a name
3. In the **Machine configuration** section, make sure to pick a "Machine type" with recommended memory and cpu (1 CPUs and 4GB RAM) to run CloudBeaver server.


Machine type

Choose a machine type with preset amounts of vCPUs and memory that suit most workloads. Or, you can create a custom machine for your workload's particular needs. [Learn more](#)

PRESET

CUSTOM

e2-medium (2 vCPU, 1 core, 4 GB memory) ▼



vCPU

1-2 vCPU (1 shared core)

Memory

4 GB

▼ ADVANCED CONFIGURATIONS

Availability policies

VM provisioning model

Standard ▼

Choose "Spot" to get a discounted, preemptible VM. Otherwise, stick to "Standard". [Learn more](#)

4. In the "Boot disk" section, click the "Change" button
5. From the "Custom images" tab, select the image that you just imported in the previous steps (cloudbeaver-ce-server-ubuntu/rhel-%version%) from the dropdown menu. Select a disk size of at least 100GB. When you are done, click on **Select**.

Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in [Marketplace](#)

PUBLIC IMAGES


CUSTOM IMAGES

SNAPSHOTS

ARCHIVE SNAPSHOTS

EXISTING DISKS

Source project for images *

 **CHANGE**

☐ Show deprecated images

Image *

cloudbeaver-ce-server-ubuntu-23-2-0

Created on Oct 30, 2023, 8:05:57 PM

Boot disk type *

Balanced persistent disk

[COMPARE DISK TYPES](#)

Size (GB) *

100







Provision between 100 and 65536 GB

[SHOW ADVANCED CONFIGURATION](#)

SELECT

CANCEL

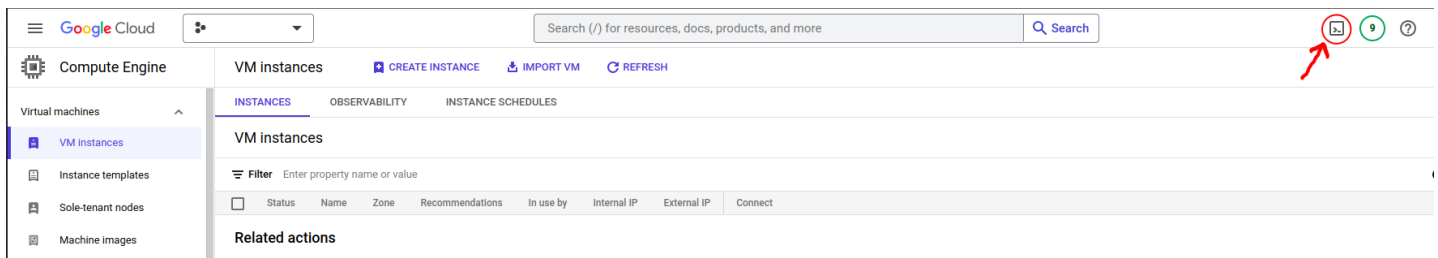
- In the **Firewall** section, make sure to check the **Allow HTTP traffic** and **Allow HTTPS traffic** boxes so that your CloudBeaver server instance can be opened from internet.
- Finally, click the **Create** button. After a few minutes, your CloudBeaver Community Edition server instance should be up and running.

 Compute Engine	VM instances CREATE INSTANCE IMPORT VM REFRESH									
	INSTANCES OBSERVABILITY INSTANCE SCHEDULES									
	Virtual machines ^									
	VM instances									
 Instance templates	 Filter Enter property name or value									
 Sole-tenant nodes	<input type="checkbox"/>	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect	
 Machine images	<input checked="" type="checkbox"/>		cloudbeaver-ce-server-ubuntu-23-2-0	us-central1-a			10.128.0.26 (nic0)	34.136.107.152 (nic0)	SSH	

You can check that your instance is running correctly by copying and pasting the **External IP** address provided by GCP into your browser.

Deployment with Google Cloud CLI

1. Log in to your account in [Google Cloud Console](#), navigate to **Compute Engine** and click on **Activate Cloud Shell**.



2. If you are prompted to authorize, do so.
3. In the terminal that opens, enter the following command:

```
gcloud beta compute instances create cloudbeaiver-ce-server \
--zone=us-central1-a \
--machine-type=e2-medium \
--tags=http-server,https-server \
--image=https://www.googleapis.com/compute/v1/projects/dbeaver-public/global/images/cloudbeaiver-ce-ub
--create-disk=auto-delete=yes \
--boot-disk-size=100GB --boot-disk-device-name=cloudbeaiver-ce-server
```

Where:

- **zone** - Zone of the instances to create. You can choose this from [GCP zones](#)
- **machine-type** - Specifies the machine type used for the instances. (1 CPUs and 4GB RAM resources recommended)
- **tags** - These tags allow network firewall rules and routes to be applied to specified VM instances.
- **image** - Specifies the boot image for the instances. You can choose any of our public images.
- **create-disk=auto-delete=yes** - Creates and attaches persistent disks to the instances. This persistent disk will be automatically deleted when the instance is deleted.
- **boot-disk-size** - The size of the boot disk, is 100GB recommended.
- **boot-disk-device-name** - The name the guest operating system will see for the boot disk.

CloudBeaver Community Edition GCP public image list:

- <https://www.googleapis.com/compute/v1/projects/dbeaver-public/global/images/cloudbeaiver-ce-ubuntu-23->
-

You can change the parameters you need for deployment yourself. For detailed information on working with Google Cloud CLI, you can read the [documentation](#).

Setup and control options

CloudBeaver Community Edition server manager

`cloudbeaver` is a utility to manage a CloudBeaver Community Edition server. Using this manager, you can start or stop the server, as well as update its version.

How to use manager:

1. Connect to your server through the terminal.
 - If you use terminal in browser window:
 - Enter `sudo su - ubuntu` after open terminal if you use Ubuntu version
 - Enter `sudo su - ec2-user` after open terminal if you use RHEL version
2. Enter `cloudbeaver` or `cloudbeaver help` to see the help menu.

Version update procedure

The update occurs with the help of the [manager](#).

1. Connect to your server through the terminal.
2. Enter `cloudbeaver update list`
3. Choose the version you want to update.
4. Run this command: `cloudbeaver update %version%`

Microsoft Azure

Table of contents

[Minimum requirements:](#)

[Deployment](#)

[Setup and control options](#)

[CloudBeaver server manager](#)

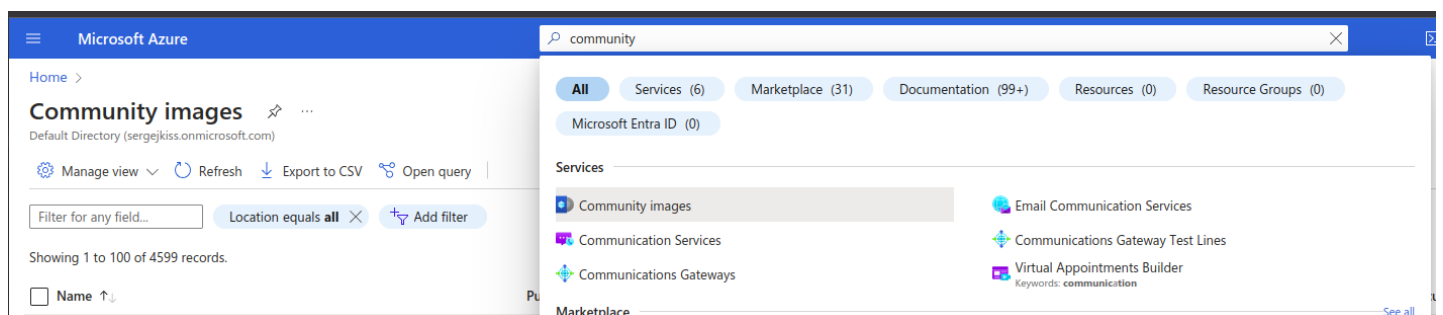
[Version update procedure](#)

Minimum requirements:

- 2 CPUs
- 4GB RAM
- 100GB Storage (SSD recommended)

Deployment

1. Log in to [Microsoft Azure Portal](#) and navigate to **Azure -> Community images**.



2. Enter `cloudb Beaver-ee` in the search field, select location, then the version, and press **Create VM**.

Home > Community images >

Community images

Default Directory (sergeikiss.onmicrosoft.com)

Manage view Refresh

Dbeaver

Name ↑

- cloudbeaver-ce (SwedenCentral/Dbeav...
- cloudbeaver-ee (SwedenCentral/Dbeav...
- dbeaver-te-server (SwedenCentral/Dbe...

SwedenCentral/Dbeaver-1207b371-9147-49ee-9774-27d4dac8b8c4/cloudbeaver-ce

Community image

Search Create VM Create VMSS Refresh Report community image

Overview

Essentials

Resource ID : /CommunityGalleries/Dbeaver-1207b371-9147-49ee-9774-27d4dac8b8c4/... OS type : Linux

Name : cloudbeaver-ce OS state : Generalized

Location : Sweden Central VM Generation : V1

Architecture : x64

Publisher URI : https://dbeaver.com/

Legal agreement URL : -

Public gallery name : dbeaver-1207b371-9147-49ee-9774-27d4dac8b8c4

Versions Properties

Name	Published date	End of life date	Disk size (GB)	Create VM	Create VMSS	Report version
23.2.0	Fri Nov 03 2023	-	100	Create VM	Create VMSS	Report

3. Fill in the required fields:

Home > Community images > SwedenCentral/Dbeaver-1207b371-9147-49ee-9774-27d4dac8b8c4/cloudbeaver-ce >

Create a virtual machine

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Azure subscription 1

Resource group * ⓘ res-group1

[Create new](#)

Instance details

Virtual machine name * ⓘ cloudbeaver-ce-23-2-0 ✓

Region * ⓘ (Europe) Sweden Central

Availability options ⓘ Availability zone

Availability zone * ⓘ Zone 1

Security type ⓘ Standard

Image * ⓘ cloudbeaver-ce/23.2.0 - x64 Gen1

[See all images](#) | [Configure VM generation](#)

VM architecture ⓘ

☐ Arm64

☒ x64

i Arm64 is not supported with the selected image.

- For field **Size** use recommended [Minimum requirements](#) resources for the best experience with this product.
- In the field **Inbound port rules** select 22, 80, and 443 ports.

Select inbound ports *

HTTP (80), HTTPS (443), SSH (22) ✓

☒ HTTP (80)

☒ HTTPS (443)

☒ SSH (22)

- You must configure the SSH user as `ubuntu` proper server management, and enter your SSH key or specify an existing one.

Administrator account

Authentication type ⓘ

- ☒ SSH public key
- ☐ Password

i Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.

Username * ⓘ

ubuntu ✓

SSH public key source

Generate new key pair ✓

Key pair name *

Name the SSH public key

That's all done. The other fields are not required.

Setup and control options

CloudBeaver server manager

`cloudbeaver` is a utility to manage a CloudBeaver server. Using this manager, you can start or stop the server, as well as update its version.

How to use manager:

1. Connect to your server through the terminal.
2. Enter `cloudbeaver` or `cloudbeaver help` to see the help menu.

Version update procedure

The update occurs with the help of the [manager](#).

1. Connect to your server through the terminal.
2. Enter `cloudbeaver update list`
3. Choose the version you want to update.
4. Run this command: `cloudbeaver update %version%`

AWS Marketplace

Table of contents

[Permissions](#)

[Deployment](#)


Permissions

To use this method, you need an account on Amazon Marketplace with the following permissions:

- [FAWSMarketplaceRead-only](#)

Deployment

1. Log in to [Amazon Marketplace](#), find [CloudBeaver AWS](#), and press **Continue to Subscribe** button.



CloudBeaver AWS

By: [DBeaiver Corporation](#) Latest Version: CloudBeaver Enterprise for AWS 23.2.0

Universal database management tool

Linux/Unix ★★★★★ **0 AWS reviews** | **1 external review** ⓘ

Free Trial

Continue to Subscribe

Save to List

Typical Total Price
\$0.583/hr
Total pricing per instance for services hosted on t3.large in US East (N. Virginia). [View Details](#)

Overview


Pricing

Usage

Support

Reviews

Then press **Continue to Configuration**.



CloudBeaver AWS

Continue to Configuration

< Product Detail [Subscribe](#)

Subscribe to this software

You're subscribed to this software. Please see the terms and pricing details below or click the button above to configure your software.

Configure your software contract

You can reduce the expense of this software by purchasing a software contract. Select the contract that aligns with your requirements. The fee that you pay will cover the software cost through the contract period. After

Software contract

Select contract option(s)


Total contract price

\$0

Due now

Create Contract

2. Select CloudBeaver version you need in **Software version** field, select region, and press **Continue to Launch**.
3. Select the desired configuration:
 - In the **EC2 Instance Type** select t2.medium or more.
 - In **Security group settings**, press **Create New Based On Seller Settings**. Then press **Launch**.




CloudBeaver AWS

Subnet Settings

subnet-8d6d7ae6 (eu-central-1a) ▾

↻

IPv4 CIDR block: 172.31.16.0/20

[Create a subnet in EC2](#) 

(Ensure you are in the selected VPC above)

Security Group Settings


A security group acts as a firewall that controls the traffic allowed to reach one or more instances. You can create a new security group based on seller-recommended settings or choose one of your existing groups. [Learn more](#)

default ▾

↻

Create New Based On Seller Settings

4. Great! CloudBeaver instance is successfully deployed on EC2.



CloudBeaver AWS

Congratulations! An instance of this software is successfully deployed on EC2!

AMI ID: ami-02eded04cab94cbd2 ([View Launch Configuration Details](#))

You can view this instance on [EC2 Console](#). You can also view all instances on [Your Software](#). Software and AWS hourly usage fees apply when the instance is running and will appear on your monthly bill.

5. Click **EC2 Console** link, and you will see the list of your instances. The new instance will be the last one and without a name.
 - Click on the empty field to give the instance a name.

- Click on the Instance ID, and you will see the Instance summary.

Instances (1/5)

Info

Find Instance by attribute or tag (case-sensitive)

<1>

⚙

<input checked="" type="checkbox"/>	Name <div>✎</div> <div>▼</div>	Instance ID	Instance state <div>▼</div>	Instance type <div>▼</div>	Status check	Alarm status	Availability Zone <div>▼</div>
<input checked="" type="checkbox"/>	<div></div> <div>✎</div>	i-07818879177b56943	<div>✔ Running</div> <div>🔍🔍</div>	t2.medium	<div>🕒 Initializing</div>	No alarms +	eu-central-1a
<input type="checkbox"/>	<div></div> <div>✎</div>		<div>✔ Running</div> <div>🔍🔍</div>	t2.medium	<div>✔ 2/2 checks passed</div>	No alarms +	eu-central-1b
<input type="checkbox"/>	<div></div> <div>✎</div>		<div>✔ Running</div> <div>🔍🔍</div>	t2.medium	<div>✔ 2/2 checks passed</div>	No alarms +	eu-central-1b
<input type="checkbox"/>	<div></div> <div>✎</div>		<div>✔ Running</div> <div>🔍🔍</div>	t2.small	<div>✔ 2/2 checks passed</div>	No alarms +	eu-central-1b
<input type="checkbox"/>	<div></div> <div>✎</div>		<div>✔ Running</div> <div>🔍🔍</div>	t2.small	<div>✔ 2/2 checks passed</div>	No alarms +	eu-central-1a

6. Click on the open address link in the **Public IPv4 DNS** section.

[EC2](#) > [Instances](#) > i-07818879177b56943

Instance summary for i-07818879177b56943 (cb-aws-test) [Info](#)

Updated less than a minute ago

[Refresh](#) [Connect](#) [Instance state](#) ▼ [Actions](#) ▼

Instance ID 📄 i-07818879177b56943 (cb-aws-test)	Public IPv4 address 📄 3.76.9.38 open address 🔗	Private IPv4 addresses 📄 172.31.26.245
IPv6 address -	Instance state 🟢 Running	Public IPv4 DNS 📄 ec2-3-76-9-38.eu-central-1.compute.amazonaws.com open address 🔗
Hostname type IP name: ip-172-31-26-245.eu-central-1.compute.internal	Private IP DNS name (IPv4 only) 📄 ip-172-31-26-245.eu-central-1.compute.internal	

7. The page in your default web browser will be opened. You need to agree to use the self-signed certificate and change it later. [How to change certificate](#)

That's all done! CloudBeaver AWS instance is ready to use. On the first page, you will see the [server configuration wizard](#).

Docker image

Table of contents

[Requirements](#)

[Installation](#)

[Running](#)

[Daemon mode](#)

CloudBeaver AWS is a web application, that requires server deployment.

Requirements

- [Install Docker](#) 20.10 or higher
- amd64 or arm64 system.

Installation

To install the latest version of CloudBeaver AWS use the following script:

```
docker pull dbeaver/cloudbeaver-aws:latest
```

Running

To run CloudBeaver AWS in the terminal:

```
docker run --name cloudbeaver-aws --rm -ti -p 8080:8978 -v /var/cloudbeaver/workspace:/opt/cloudbeaver/wo
```

Then switch to the browser and open <http://localhost:8080/>

Daemon mode

Add the following parameters:

```
-d --restart unless-stopped
```

Docker image

Table of contents

[Table of contents](#)

[Deploy with a single docker container](#)

[Requirements](#)

[Installation](#)

[Running](#)

[Daemon mode](#)

[Deploy with docker-compose](#)

CloudBeaver Enterprise is a web application that requires server deployment.

You only need a Linux, macOS, or Windows machine with Docker.

Table of contents

- [Deploy with single docker container](#)
- [Deploy with docker-compose](#)

Deploy with a single docker container

Requirements

- [Install Docker](#) 20.10 or higher
- amd64 or arm64 system.

Installation

To install the latest version of CloudBeaver, use the following script:

```
docker pull dbeaver/cloudbeaver-ee:latest
```

Running

To run CloudBeaver Enterprise in the terminal:

```
docker run --name cloudbeaver-ee --rm -ti -p 8080:8978 -v /var/cloudbeaver/workspace:/opt/cloudbeaver/work
```

Then switch to the browser and open <http://localhost:8080/>

Daemon mode

Add the following parameters:

```
-d --restart unless-stopped
```

Deploy with docker-compose

Instructions on how to deploy via Docker-compose with configuration examples can be found here: <https://github.com/dbeaver/cloudbeaver-deploy>

AWS AMI

Table of contents

[Minimum requirements:](#)

[How to deploy AMI in AWS](#)

[Setup and control options](#)

[CloudBeaver server manager](#)

[Version update procedure](#)

Minimum requirements:

- 2 CPUs
- 4 GB RAM
- 100 GB Storage (SSD recommended)

How to deploy AMI in AWS

- Go to [AWS EC2](#) -> AMI Catalog -> Community AMIs
- Find `cloudbeaver-ee`
- Choose version

AMI Catalog

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

AMIs

Create Template with AMI

Launch Instance with AMI

cloudbeaver-ee

X

Quickstart AMIs (0)
Commonly used AMIs

My AMIs (30)
Created by me

AWS Marketplace AMIs (96)
AWS & trusted third-party AMIs

Community AMIs (2)
Published by anyone

Refine results

Clear all filters

Operating system

Linux/Unix

All Linux/Unix

Amazon Linux

CentOS

Debian

Fedora

Gentoo

macOS

openSUSE

Other Linux

Red Hat

SUSE Linux

Ubuntu

cloudbeaver-ee (2 filtered, 2 unfiltered)

< 1 >

Community AMIs

Community AMIs contain all AMIs that are public, therefore anyone can publish an AMI and it will show in this catalog. This catalog can also contain paid products. When using community AMIs it is best practice to ensure you know and trust the publisher before launching an AMI.

cloudbeaver-ee-ubuntu-ee

ami-0c59d6df246b9ade0

CloudBeaver Server ea (Ubuntu)

OwnerAlias: – Platform: Ubuntu Architecture: x86_64 Owner: 859245062624 Publish date: 2023-12-05 Root device type: ebs Virtualization: hvm

ENA enabled: Yes

Select

cloudbeaver-ee-rhel-ee

ami-00a5832a631c869f6

CloudBeaver Server ea (RedHat)

OwnerAlias: – Platform: Red Hat Architecture: x86_64 Owner: 859245062624 Publish date: 2023-12-05 Root device type: ebs Virtualization: hvm

ENA enabled: Yes

Select

- Launch Instance with AMI
- In the **EC2 Instance Type** select `t2.medium` or another instance that meets the [minimum requirements](#).
- In **Security group settings**, select `Allow SSH traffic` , `Allow HTTPS traffic from the internet` , `Allow HTTP traffic from the internet` fields.

Then press **Launch**.

Great! CloudBeaver instance is successfully deployed on EC2.

Setup and control options

CloudBeaver server manager

`cloudbeaver` is a utility to manage a CloudBeaver server. Using this manager, you can start or stop the server, as well as update its version.

How to use manager:

CloudBeaver User Guide 24.1.ea. Page 339 of 365.

1. Connect to your server through the terminal. Use SSH user as `ubuntu` if you use Ubuntu distributive, or `ec2-user` if RHEL distributive.
2. Enter `cloudbeaver` or `cloudbeaver help` to see the help menu.

Version update procedure

The update occurs with the help of the [manager](#).

1. Connect to your server through the terminal.
2. Enter `cloudbeaver update list`
3. Choose the version you want to update.
4. Run this command: `cloudbeaver update %version%`

Google Cloud

Table of contents

[Permissions](#)

[Deployment](#)

[Deployment in Google Cloud interface](#)

[Step 1. Import CloudBeaver Enterprise Edition custom image on your GCP account](#)

[Step 2. Create a new GCP Compute Engine instance from the imported image](#)

[Deployment with Google Cloud CLI](#)

[Setup and control options](#)

[CloudBeaver Enterprise Edition server manager](#)

[Version update procedure](#)

Permissions

You need to have a list of permissions like this in your project:

```
compute.images.create
compute.images.list

compute.disks.create
compute.disks.resize
compute.disks.use
compute.disks.list

compute.instances.reset
compute.instances.resume
compute.instances.setMetadata
compute.instances.start
compute.instances.stop
compute.instances.suspend
compute.instances.update
compute.instances.use
compute.instances.setServiceAccount
compute.instances.create
compute.instances.attachDisk

compute.subnetworks.use
compute.subnetworks.useExternalIp
compute.networks.use
compute.networks.useExternalIp
```

If you don't have them, contact your system administrator or project owner.

Deployment

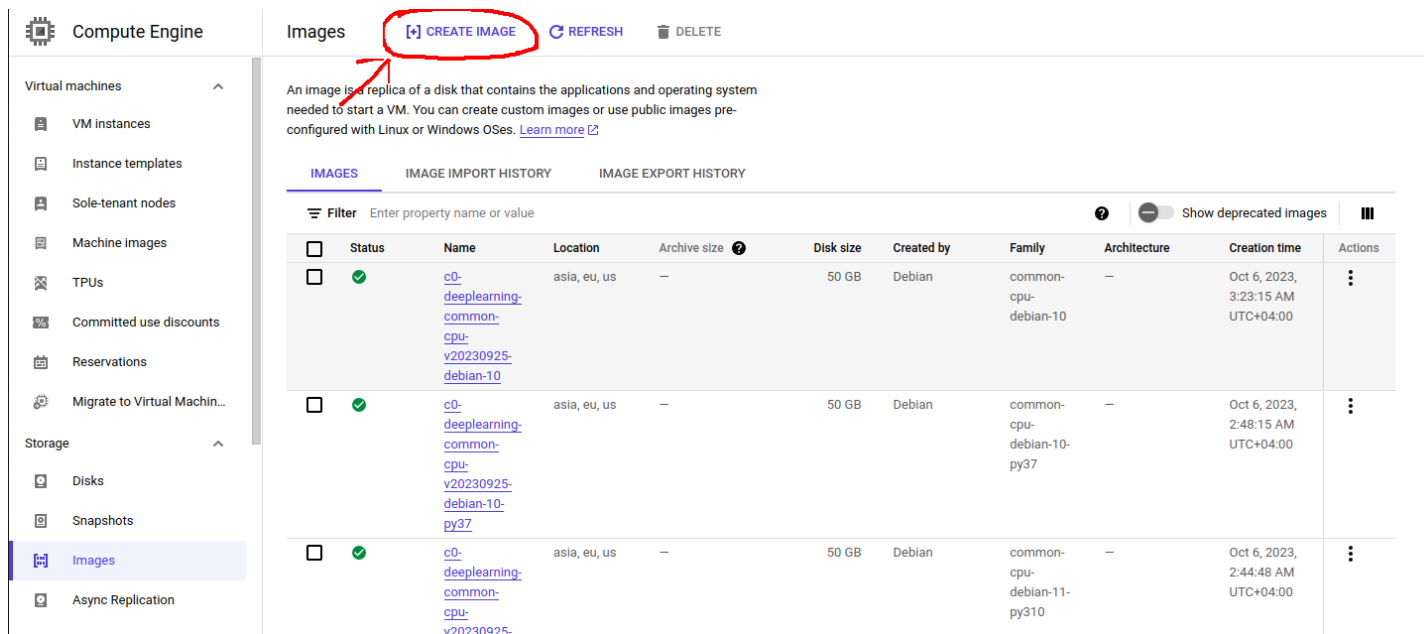
Deployment in Google Cloud interface

To deploy CloudBeaver Enterprise Edition in Google Cloud Console interface, you need to import CloudBeaver Enterprise Edition custom image to your account using the image URI, and then create a new instance from the imported image.

Step 1. Import CloudBeaver Enterprise Edition custom image on your GCP account

1. Log in to your account in [Google Cloud Console](#), navigate to **Compute Engine -> Images**, and click **[+]**

CREATE IMAGE.



The screenshot shows the Google Cloud Console 'Images' page. The 'CREATE IMAGE' button is circled in red. The page displays a list of existing images with the following columns: Status, Name, Location, Archive size, Disk size, Created by, Family, Architecture, Creation time, and Actions. The table contains three rows of images, all with a status of 'OK' and a location of 'asia, eu, us'.

Status	Name	Location	Archive size	Disk size	Created by	Family	Architecture	Creation time	Actions
OK	c0-deeplearning-common-cpu-v20230925-debian-10	asia, eu, us	—	50 GB	Debian	common-cpu-debian-10	—	Oct 6, 2023, 3:23:15 AM UTC+04:00	⋮
OK	c0-deeplearning-common-cpu-v20230925-debian-10-py37	asia, eu, us	—	50 GB	Debian	common-cpu-debian-10-py37	—	Oct 6, 2023, 2:48:15 AM UTC+04:00	⋮
OK	c0-deeplearning-common-cpu-v20230925-py310	asia, eu, us	—	50 GB	Debian	common-cpu-debian-11-py310	—	Oct 6, 2023, 2:44:48 AM UTC+04:00	⋮

2. Fill in the **Create an image** form:

- In the **Name** field write the image name in the following format:

`cloudb Beaver-ee-server-ubuntu/rhel-%version%`

- In the **Source** field select **Virtual disk (VMDK, VHD)**.

- If you are prompted to enable Cloud Build tools and grant permissions, do so.

- Copy the following URI `cloudb Beaver-ee-server/` in the **Virtual disk file** field, click **BROWSE**, and select the version you need.

←

Create an image

Name *

cloudbeaver-ee-server-ubuntu-23-2-0

?

Name is permanent

Source *

Virtual disk (VMDK, VHD)

▼

?

Virtual disk file *

cloudbeaver-ee-server/

?

BROWSE

Object not available. Either it does not exist or you do not have access. Try browsing for the object instead.

Operating system on virtual disk *

Detect operating system.

▼

☒ Install guest packages

Allow Compute Engine to install guest packages to ensure that the image will be bootable.

Family

?

Description

Pricing summary

Your free trial credit will be used for this image. [Google Cloud Free Tier](#)

cloudbeaver-ee-server

cloudbeaver-ee-rhel-23-2-0.vmdk

cloudbeaver-ee-ubuntu-23-2-0.vmdk

That's all done. The other fields are not required.

3. Click **Create**. You may have to wait up to 15 minutes while the CloudBeaver Enterprise Edition server custom image imports to your account.

Images

[+] CREATE IMAGE

REFRESH

DELETE

An image is a replica of a disk that contains the applications and operating system needed to start a VM. You can create custom images or use public images pre-configured with Linux or Windows OSes. [Learn more](#)

IMAGES

IMAGE IMPORT HISTORY

IMAGE EXPORT HISTORY

Filter

Enter property name or value

Status	Cloud Build ID	Image name	Source	Started ↓	Duration
✓	7cc94a23-ee04-4ce0-bf64-40fb606f1583	cloudbeaver-ee-server-ubuntu-23-2-0	gs://cloudbeaver-ee-server/cloudbeaver-ee-ubuntu-23-2-0.vmdk	1 minute ago	13 min 20 sec


Step 2. Create a new GCP Compute Engine instance from the imported image


1. Open the tab **Images**, click on the name of the image that you just imported, and click on the **[+] Create instance** button.


CloudBeaver User Guide 24.1.ea. Page 343 of 365.


←

Images

 EDIT

 DELETE

 CREATE INSTANCE

 EXPORT

cloudbeaver-ee-server-ubuntu-23-2-0

Description	Image created by Daisy in workflow "translate" on behalf of root.
Source disk	disk-03529-1
Location	us (United States)
Architecture	—
Labels	<div><div>gce-image-import : true</div><div>gce-image-... : 03529</div></div>
Creation time	Oct 30, 2023, 8:58:36 PM UTC+04:00
Encryption type	Google-managed

EQUIVALENT REST

2. Give your instance a name
3. In the **Machine configuration** section, make sure to pick a "Machine type" with recommended memory and cpu (1 CPUs and 4GB RAM) to run CloudBeaver server.


Machine type

Choose a machine type with preset amounts of vCPUs and memory that suit most workloads.
Or, you can create a custom machine for your workload's particular needs. [Learn more](#)

PRESET

CUSTOM

e2-medium (2 vCPU, 1 core, 4 GB memory)



vCPU

1-2 vCPU (1 shared core)

Memory

4 GB

ADVANCED CONFIGURATIONS

Availability policies

VM provisioning model

Standard

Choose "Spot" to get a discounted, preemptible VM. Otherwise, stick to "Standard". [Learn more](#)

4. In the "Boot disk" section, click the "Change" button
5. From the "Custom images" tab, select the image that you just imported in the previous steps (cloudbeaver-ee-server-ubuntu/rhel-%version%) from the dropdown menu. Select a disk size of at least 100GB. When you are done, click on **Select**.

Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in [Marketplace](#)

PUBLIC IMAGES

CUSTOM IMAGES

SNAPSHOTS

ARCHIVE SNAPSHOTS

EXISTING DISKS

Source project for images *

?

CHANGE

☐

Show deprecated images

Image *

cloudbeaver-ee-server-ubuntu-23-2-0

Created on Oct 30, 2023, 8:58:36 PM

Boot disk type *

Balanced persistent disk

COMPARE DISK TYPES

Size (GB) *

100

Provision between 100 and 65536 GB

SHOW ADVANCED CONFIGURATION

SELECT

CANCEL

6. In the **Firewall** section, make sure to check the **Allow HTTP traffic** and **Allow HTTPS traffic** boxes so that your CloudBeaver server instance can be opened from internet.
7. Finally, click the **Create** button. After a few minutes, your Team Edition server instance should be up and running.

VM instances

CREATE INSTANCE

IMPORT VM

REFRESH

INSTANCES

OBSERVABILITY

INSTANCE SCHEDULES

VM instances

Filter

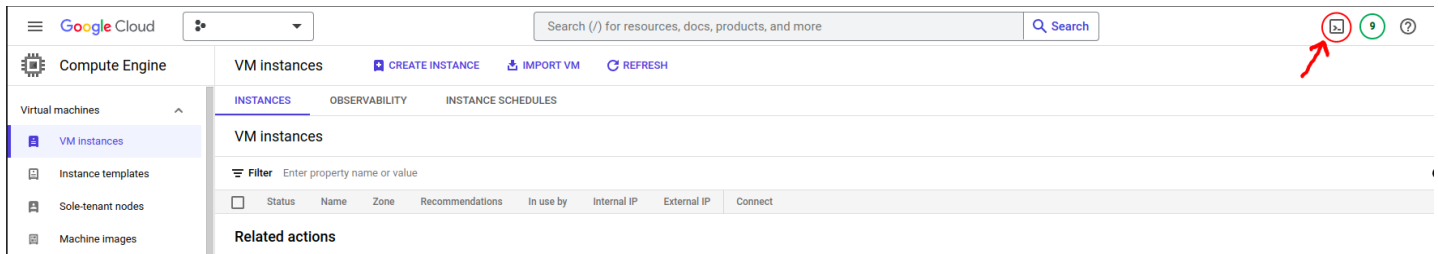
Enter property name or value

<input type="checkbox"/>	Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>	✓	cloudbeaver-ee-server-ubuntu-23-2-0	us-central1-a			10.128.0.30 (nic0)	34.29.72.90 (nic0)	SSH

You can check that your instance is running correctly by copying and pasting the **External IP** address provided by GCP into your browser.

Deployment with Google Cloud CLI

1. Log in to your account in [Google Cloud Console](#), navigate to **Compute Engine** and click on **Activate Cloud Shell**.



2. If you are prompted to authorize, do so.
3. In the terminal that opens, enter the following command:

```
gcloud beta compute instances create cloudbeaver-ee-server \
--zone=us-central1-a \
--machine-type=e2-medium \
--tags=http-server,https-server \
--image=https://www.googleapis.com/compute/v1/projects/db Beaver-public/global/images/cloudbeaver-ee-ubuntu \
--create-disk=auto-delete=yes \
--boot-disk-size=100GB --boot-disk-device-name=cloudbeaver-ee-server
```

Where:

- **zone** - Zone of the instances to create. You can choose this from [GCP zones](#)
- **machine-type** - Specifies the machine type used for the instances. (1 CPUs and 4GB RAM resources recommended)
- **tags** - These tags allow network firewall rules and routes to be applied to specified VM instances.
- **image** - Specifies the boot image for the instances. You can choose any of our public images.
- **create-disk=auto-delete=yes** - Creates and attaches persistent disks to the instances. This persistent disk will be automatically deleted when the instance is deleted.
- **boot-disk-size** - The size of the boot disk, is 100GB recommended.
- **boot-disk-device-name** - The name the guest operating system will see for the boot disk.

CloudBeaver Enterprise Edition GCP public image list:

- [https://www.googleapis.com/compute/v1/projects/dbeaver-public/global/images/cloudbeaver-ee-ubuntu-23-](https://www.googleapis.com/compute/v1/projects/dbeaver-public/global/images/cloudbeaver-ee-ubuntu-23-2-20230920)
- [https://www.googleapis.com/compute/v1/projects/dbeaver-public/global/images/cloudbeaver-ee-rhel-23-2-](https://www.googleapis.com/compute/v1/projects/dbeaver-public/global/images/cloudbeaver-ee-rhel-23-2-20230920)

You can change the parameters you need for deployment yourself. For detailed information on working with Google Cloud CLI, you can read the [documentation](#).

Setup and control options

CloudBeaver Enterprise Edition server manager

`cloudbeaver` is a utility to manage a CloudBeaver Enterprise Edition server. Using this manager, you can start or stop the server, as well as update its version.

How to user manager:

1. Connect to your server through the terminal.
 - If you use terminal in browser window:
Enter `sudo su - ubuntu` after open terminal if you use Ubuntu version
Enter `sudo su - ec2-user` after open terminal if you use RHEL version
2. Enter `cloudbeaver` or `cloudbeaver help` to see the help menu.

Version update procedure

The update occurs with the help of the [manager](#).

1. Connect to your server through the terminal.
2. Enter `cloudbeaver update list`
3. Choose the version you want to update.
4. Run this command: `cloudbeaver update %version%`

Microsoft Azure

Table of contents

[Minimum requirements:](#)

[Deployment](#)

[Setup and control options](#)

[CloudBeaver server manager](#)

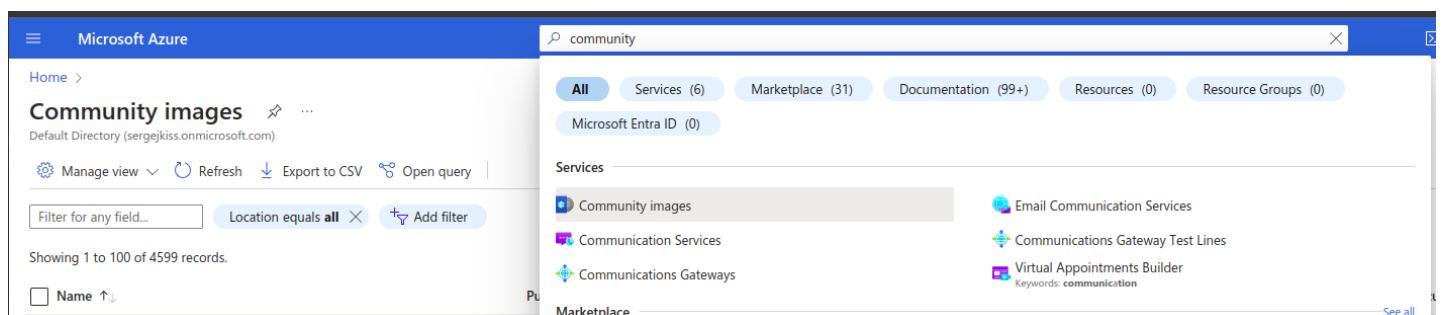
[Version update procedure](#)

Minimum requirements:

- 2 CPUs
- 4GB RAM
- 100GB Storage (SSD recommended)

Deployment

1. Log in to [Microsoft Azure Portal](#) and navigate to **Azure -> Community images**.



2. Enter `cloudb Beaver-ee` in the search field, select location, then the version, and press **Create VM**.

Home > Community images >

Community images
Default Directory (sergejss.onmicrosoft.com)

Manage view
Refresh

Dbeaver
Name ↑

cloudbeaver-ce (SwedenCentral/Dbeaver-...
cloudbeaver-ee (SwedenCentral/Dbeaver-...
cloudbeaver-ee (centralus/Dbeaver-120...
cloudbeaver-ee (eastasia/Dbeaver-1207...
cloudbeaver-ee (eastus/Dbeaver-1207b...
cloudbeaver-ee (northeurope/Dbeaver-...
cloudbeaver-ee (southeastasia/Dbeaver-...
cloudbeaver-ee (westeurope/Dbeaver-...
cloudbeaver-ee (westus/Dbeaver-1207-...

SwedenCentral/Dbeaver-1207b371-9147-49ee-9774-27d4dac8b8c4/cloudbeaver-ee
Community image

Search
Create VM
Create VMSS
Refresh
Report community image

Overview
Essentials

Resource ID : /CommunityGalleries/Dbeaver-1207b371-9147-49ee-9774-27d4dac8b8c4/...
Name : cloudbeaver-ee
Location : Sweden Central
Architecture : x64
Publisher URI : https://dbeaver.com/
Legal agreement URL : -
Public gallery name : dbeaver-1207b371-9147-49ee-9774-27d4dac8b8c4
OS type : Linux
OS state : Generalized
VM Generation : V1

Versions
Properties

Name	Published date	End of life date	Disk size (GB)	Create VM	Create VMSS	Report version
23.2.0	Fri Nov 03 2023	-	100	Create VM	Create VMSS	Report

3. Fill in the required fields:

Create a virtual machine

image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Azure subscription 1

Resource group *

res-group1

Create new

Instance details

Virtual machine name *

cloudbeaver-ee-23-2-0

Region *

(Europe) Sweden Central

Availability options

Availability zone

Availability zone *

Zone 1

Security type

Standard

Image *

cloudbeaver-ee/23.2.0 - x64 Gen1

See all images | Configure VM generation

VM architecture

Arm64

x64

Arm64 is not supported with the selected image.

- For field **Size** use recommended [Minimum requirements](#) resources for the best experience with this product.
- In the field **Inbound port rules** select 22, 80, and 443 ports.

CloudBeaver User Guide 24.1.ea. Page 349 of 365.

Select inbound ports *

HTTP (80), HTTPS (443), SSH (22) ✓

☒ HTTP (80)

☒ HTTPS (443)

☒ SSH (22)

- You must configure the SSH user as `ubuntu` proper server management, and enter your SSH key or specify an existing one.

Administrator account

Authentication type ⓘ

- ☒ SSH public key
- ☐ Password

i Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.

Username * ⓘ

ubuntu ✓

SSH public key source

Generate new key pair ✓

Key pair name *

Name the SSH public key

That's all done. The other fields are not required.

Setup and control options

CloudBeaver server manager

`cloudbeaver` is a utility to manage a CloudBeaver server. Using this manager, you can start or stop the server, as well as update its version.

How to use manager:

1. Connect to your server through the terminal.
2. Enter `cloudbeaver` or `cloudbeaver help` to see the help menu.

Version update procedure

The update occurs with the help of the [manager](#).

1. Connect to your server through the terminal.
2. Enter `cloudbeaver update list`
3. Choose the version you want to update.
4. Run this command: `cloudbeaver update %version%`

SSL certificate configuration

Table of contents

[Proxy and SSL configuration](#)

[Create self-signed certificate](#)

Proxy and SSL configuration

The instance contains an Nginx proxy server, the configuration of which is located at path

```
/etc/nginx/conf.d/cloudbeaver.conf
```

To set up a connection via HTTPS with domain:

- You need to [create](#) or buy a valid TLS certificate for your domain endpoint.
- After you get SSL certificate for your domain you must put it to `/etc/nginx/ssl/fullchain.pem` as certificate and `/etc/nginx/ssl/privkey.pem` as a private key.
- Change `server_name _;` in configuration `/etc/nginx/conf.d/cloudbeaver.conf` to

```
server_name <your-domain>;
```
- Enter in terminal `sudo systemctl reload nginx.service` to reload Nginx proxy
- Now you can open your `CloudBeaver Server` from the browser using your domain address.

Create self-signed certificate

Self-signed certificates are considered insecure for the Internet. Firefox will treat the site as having an invalid certificate, while Chrome will act as if the connection was plain HTTP

You can create self-signed certificate for `<your-domain>` by running the following script in the terminal:

```
SECRET_CERT_CSR="/C=US/ST=NY/L=NYC/O=CloudBeaver /OU=IT Department/CN=<your-domain>"
cd /etc/nginx/
mkdir ssl
cd ssl
sudo openssl req -x509 -sha256 -nodes -days 36500 -subj "$SECRET_CERT_CSR" -newkey rsa:2048 -keyout privkey.pem
```

How to connect CloudBeaver to a database on a separate machine in Azure

Table of contents

[Connect your VM with CloudBeaver to PostgreSQL](#)

[Configure a peering for your networks](#)

[Run CloudBeaver on the Linux VM and connect it to PostgreSQL](#)

[Step 1. Running CloudBeaver](#)

[Step 2. Configure your network to access CloudBeaver](#)

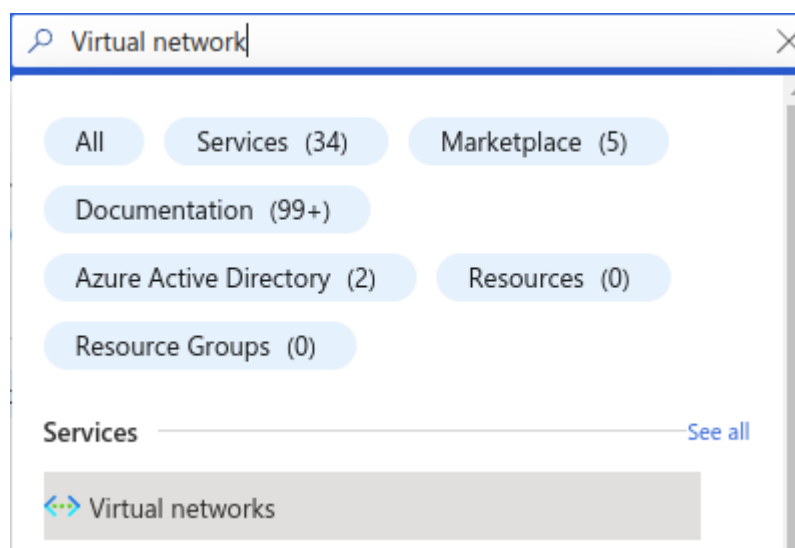
[Step 3. Configure CloudBeaver and connect to PostgreSQL](#)

[Connect DBeaver from Win VM to PostgreSQL](#)

Connect your VM with CloudBeaver to PostgreSQL

Configure a peering for your networks

1. Log in to your account in [Microsoft Azure](#) and go to **Virtual networks**.



2. Select the network that you created during the VM startup process.

Virtual networks

Default Directory

[+](#) Create [Manage view](#) [Refresh](#) [Export to CSV](#) [Open query](#) [Assign tags](#)

Filter for any field...

Subscription equals **all**

[Add filter](#)

[More \(](#)

Showing 1 to 3 of 3 records.

No grouping

List view

<input type="checkbox"/>	Name ↑↓	Resource group ↑↓	Location ↑↓	Subscription ↑↓
<input type="checkbox"/>	SQL-vpc	res-group1	East US	Azure subscription 1
<input type="checkbox"/>	Win-vm-network	res-group1	East US	Azure subscription 1

3. Select **Overview** -> **Capabilities** -> **Peerings** and click **Add peering**.

Win-vm-network
Virtual network

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Address space

Connected devices

Subnets

Bastion

DDoS protection

Firewall

Microsoft Defender for Cloud

Network manager

[Move](#) [Delete](#) [Refresh](#) [Give feedback](#)

Essentials

Resource group (move) : [res-group1](#)

Location (move) : East US

Subscription (move) : [Azure subscription 1](#)

Subscription ID :

Address space : [10.1.0.0/16](#)

DNS servers : [Azure provided DNS](#)

Flow timeout : [Configure](#)

BGP community string : [Configure](#)

Virtual network ID : 9701e2f0-9d1a-46a1

Tags (edit) : [Add tags](#)


Topology

Properties


Capabilities (5)

Recommendations


Tutorials

**DDoS protection**
Configure additional protection from distributed denial of service attacks.

Not configured

**Azure Firewall**
Protect your network with a stateful L3-L7 firewall.

Not configured

**Peerings**
Seamlessly connect two or more virtual networks.

Not configured

4. The **Add peering** window will be opened.

- In the **Peering link name** field add the name for the new peering.
- In the **Virtual network** field select the network where your database is located (**SQL-vpc** in this case).

CloudBeaver User Guide 24.1.ea. Page 354 of 365.

Add peering

Win-vm-network

This virtual network

Peering link name *

win-vm-to-sql-peering ✓

- ☒ Allow 'Win-vm-network' to access the peered virtual network ⓘ
- ☐ Allow 'Win-vm-network' to receive forwarded traffic from the peered virtual network ⓘ
- ☐ Allow gateway in 'Win-vm-network' to forward traffic to the peered virtual network ⓘ
- ☐ Enable 'Win-vm-network' to use the peered virtual networks' remote gateway ⓘ

Remote virtual network

Peering link name *

win-vm-to-sql-peering ✓

Virtual network deployment model ⓘ

- ☒ Resource manager
- ☐ Classic

☐ I know my resource ID ⓘ

Subscription * ⓘ

Azure subscription 1 ▼

Virtual network *

▼

workers-vnet

SQL-vpc

Win-vm-network

☐ Enable the peered virtual network to use 'win-vm-network's remote gateway ⓘ

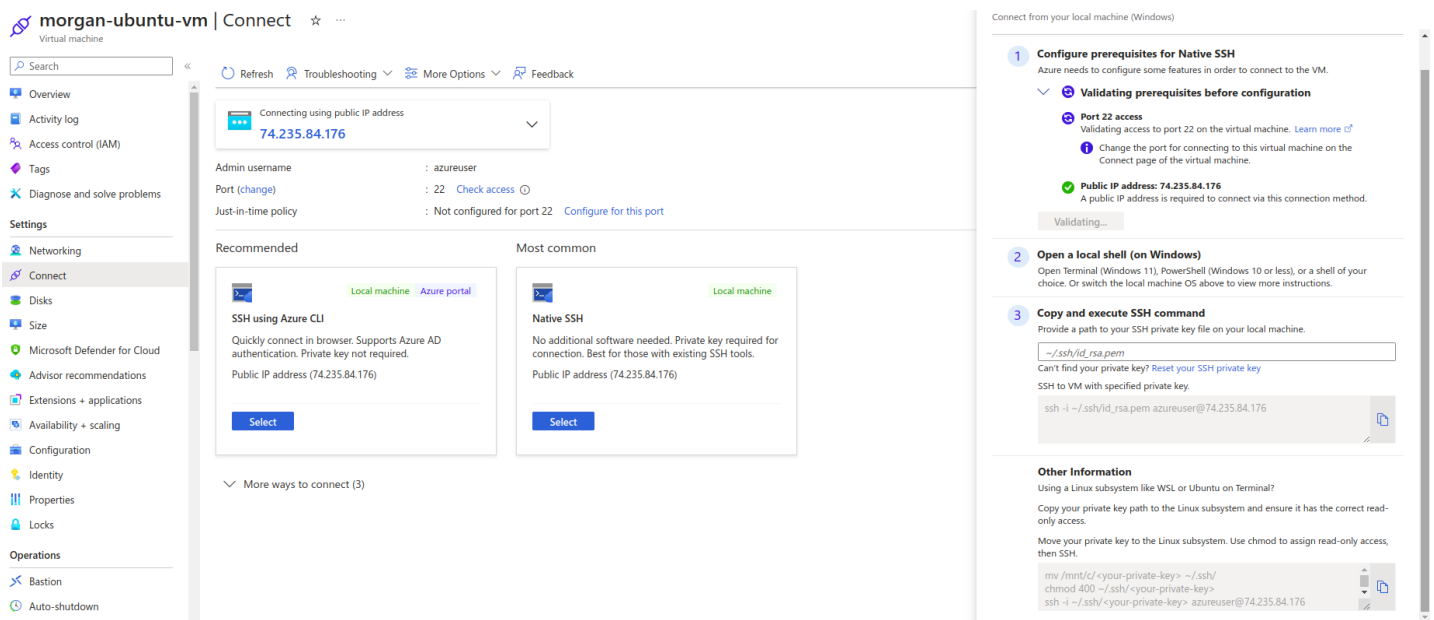
5. Click **Create**. The peering is configured.

Run CloudBeaver on the Linux VM and connect it to PostgreSQL

Step 1. Running CloudBeaver

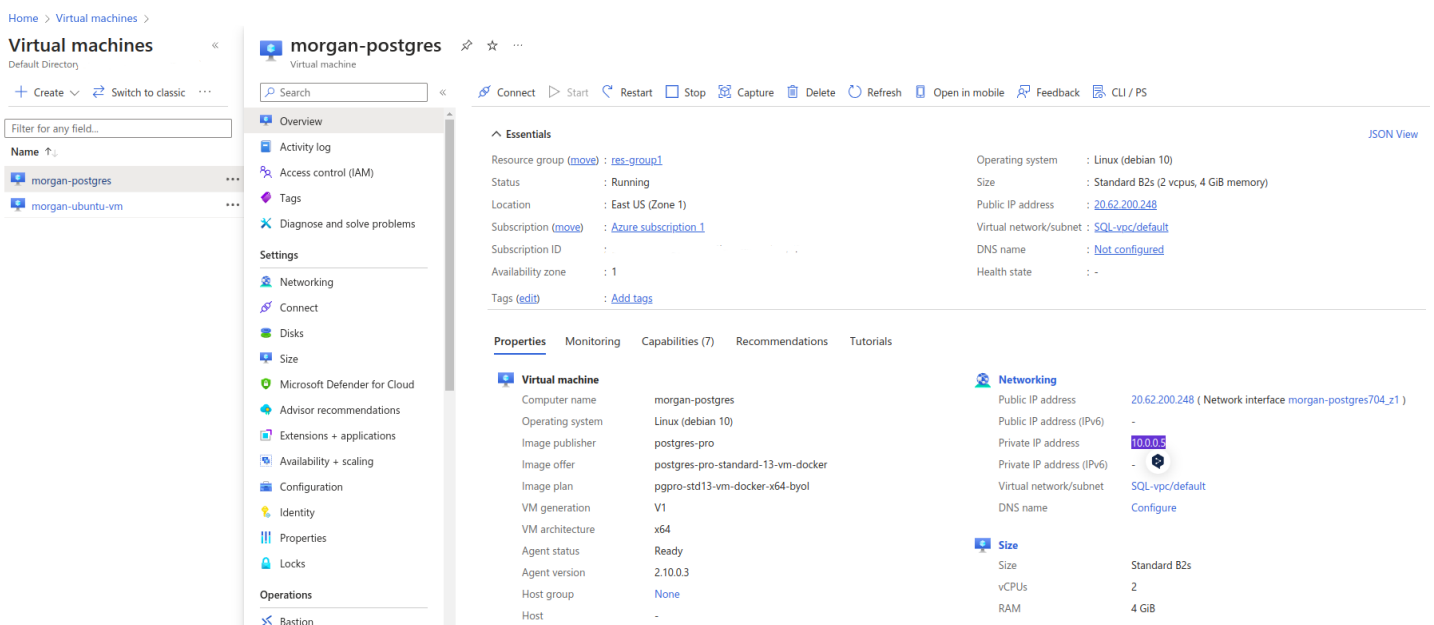
1. Connect to your Linux VM via SSH:

```
ssh -i ~/.ssh/<your_pub_key> <username>@<your_ip>
```



2. Check that your Linux VM has access to your SQL DB private IP.

- Copy the private IP address of your database in **Virtual machines** -> **Your SQL DB** -> **Properties** -> **Networking**.



- Ping this IP in the terminal:

```
root@morgan-postgres:~# ping 10.0.0.5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=0.044 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=0.053 ms
^C
```

If everything was done correctly in the previous steps, you will get the result as in the screenshot above. All that remains is to connect our CloudBeaver server to your database.

3. Install Docker if it is not installed. Check this [documentation](#) for details.

4. Run CloudBeaver with this command:

```
docker run -d --name cloudbeaver-ee --rm -ti -p 8080:8978 -v /var/cloudbeaver/workspace:/opt/cloudbeaver/v
```

```
root@morgan-ubuntu-vm:~# docker run -d --name cloudbeaver-ee --rm -ti -p 8080:8978 -v /var/cloudbeaver/workspace:/opt/cloudbeaver/workspace dbeaver/cloudbeaver-ee:latest
e5227a0cf91a4fd70b62059a964d8eb7d46e6e2243dcd9a823b217d064a3d83f
root@morgan-ubuntu-vm:~# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
e5227a0cf91a   dbeaver/cloudbeaver-ee:latest       "./run-server.sh"       19 seconds ago Up 19 seconds  0.0.0.0:8080->8978/tcp, :::8080->8978/tcp  cloudbeaver-ee
root@morgan-ubuntu-vm:~#
```

You can learn more about CloudBeaver deployment in our [documentation](#).

Step 2. Configure your network to access CloudBeaver

You need to make small changes to the network configuration so that you can access CloudBeaver UI.

1. Go to **Virtual machines** -> **Networking** -> **Inbound port rules** -> **Add inbound port rule**

The screenshot shows the Azure portal interface for a virtual machine named 'morgan-ubuntu-vm'. The 'Networking' section is selected, and the 'Inbound port rules' tab is active. The network interface 'morgan-ubuntu-vm461_z1' is shown with its IP configuration set to 'ipconfig1 (Primary)'. The 'Effective security rules' section displays a table of inbound port rules. The table has columns for Priority, Name, Port, Protocol, Source, Destination, and Action. The rules listed are: SSH (Priority 300, Port 22, TCP, Any source, Any destination, Allow action), AllowVnetInBound (Priority 65000, Any port, Any protocol, VirtualNetwork source, VirtualNetwork destination, Allow action), AllowAzureLoadBalancerInBound (Priority 65001, Any port, Any protocol, AzureLoadBalancer source, Any destination, Allow action), and DenyAllInBound (Priority 65500, Any port, Any protocol, Any source, Any destination, Deny action). An 'Add inbound port rule' button is visible in the top right corner of the rules section.

Priority	Name	Port	Protocol	Source	Destination	Action
300	SSH	22	TCP	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

2. Add access to port **8080**. For security reasons, we recommend specifying your IP or the IP of your proxy in the resource to avoid third-party access.



Add inbound security rule



morgan-ubuntu-vm-nsg

Source ⓘ

My IP address



Source IP addresses/CIDR ranges ⓘ

134.209.249.195

Source port ranges * ⓘ

*

Destination ⓘ

Any



Service ⓘ

Custom



Destination port ranges * ⓘ

8080

Protocol

☒ Any

☐ TCP

☐ UDP

☐ ICMP

Action

☒ Allow

☐ Deny

Priority * ⓘ

310

Name *

AllowUIAccess



Description

Add

Cancel

Give feedback

3. Open in browser `http://<your_vm_ip>:8080`

That's all done! You are in the CloudBeaver interface.

Step 3. Configure CloudBeaver and connect to PostgreSQL

1. Enter your license key and configure your server.

The screenshot shows the 'INITIAL SERVER CONFIGURATION' window. The left sidebar has a menu with 'Welcome', 'License', 'Server configuration' (selected), and 'Confirmation'. The main area has a 'Main server configuration' tab with 'BACK' and 'NEXT' buttons. Below the tabs, there's a message: 'You can configure the main server parameters here. You will be able to add additional services after the server configuration. Administrator is a super user who can configure server, set databases connections, manage other users and much more. Please, remember the entered password. It is not possible to recover administrator password automatically.'

The configuration is divided into several sections:

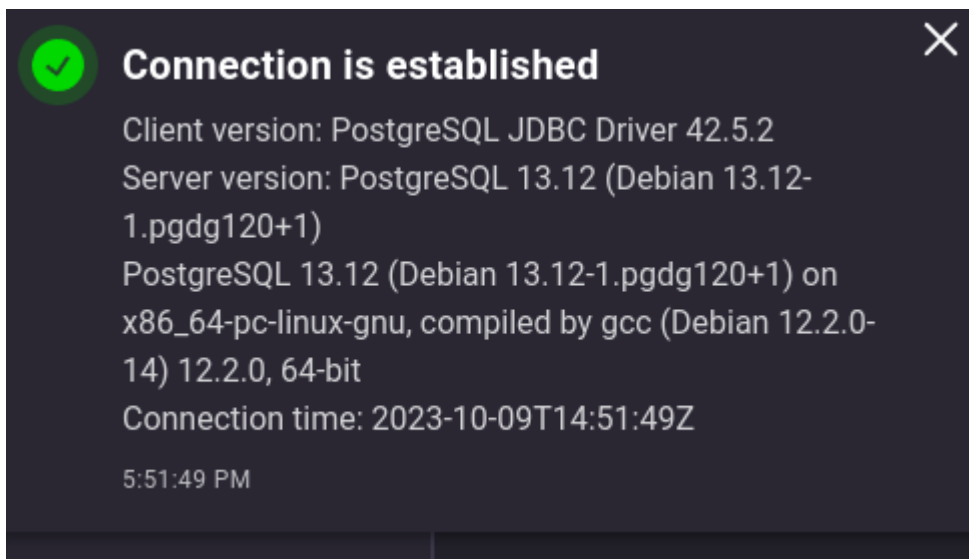
- SERVER INFORMATION:** Includes 'Server Name *' (CloudBeaver EE Web Server), 'Server URL *' (http://74.235.84.176:8080), and 'Session lifetime, min *' (30).
- CONFIGURATION:** Includes 'Enable private connections' (checked), 'Navigator simple view' (checked), and 'Enable Resource Manager' (checked).
- AUTHENTICATION SETTINGS:** Includes 'Allow anonymous access' (unchecked) and 'Local' (checked).
- ADMINISTRATOR CREDENTIALS:** Includes 'Login *' (cadmin).
- SECURITY:** Includes 'Save credentials' (checked).
- DISABLED DRIVERS:** Includes a search bar and a list of drivers: SQLite, H2 Embedded, and H2 Embedded V2.

2. Go to **Connection Templates** , click **Add** , and type your database name (PostgreSQL for example).
3. In the **Host** field enter the IP address of your database you received earlier and indicate the authorization data for your database.

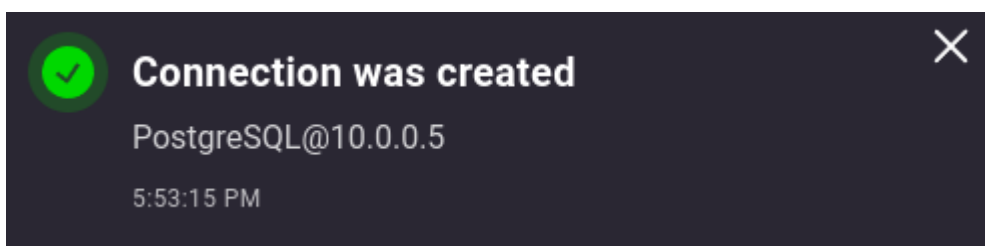
The screenshot shows the 'Connection Properties' dialog for a PostgreSQL connection. The 'Driver' is set to 'PostgreSQL'. The 'Configuration' section has 'Manual' selected. The 'Host *' is '10.0.0.5' and the 'Port' is '5432'. The 'Database' is 'postgres'. The 'Connection name *' is 'PostgreSQL@10.0.0.5'. The 'Description' field is empty. The 'Authentication' section has 'Database Native' selected. The 'User name' is 'postgres' and the 'User password' is masked. The 'Save credentials' checkbox is unchecked. The 'ADVANCED SETTINGS' section has 'MISC' expanded, showing 'User role' (empty), 'DATABASE LIST', and 'SQL'.

At the bottom, there's a table with columns 'CONNECTION NAME' and 'ADDRESS'.

4. Click **TEST** to check that your configuration works.



5. Perfect! Now click **CREATE** in the top right. Connection to your database is added to CloudBeaver.

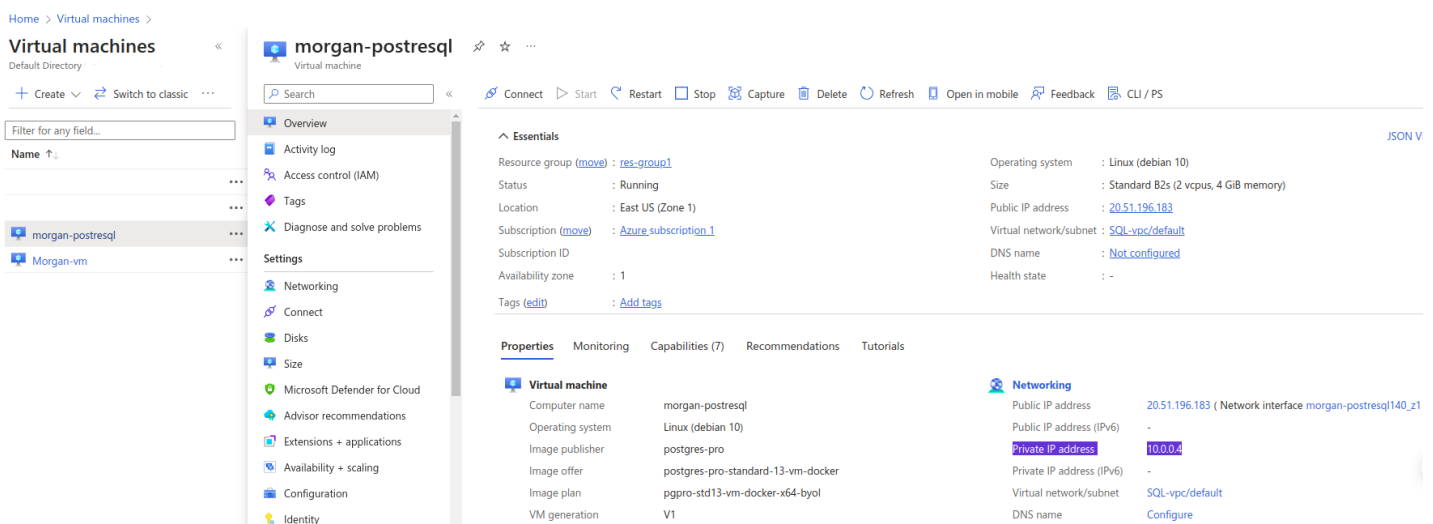


Connect DBeaver from Win VM to PostgreSQL

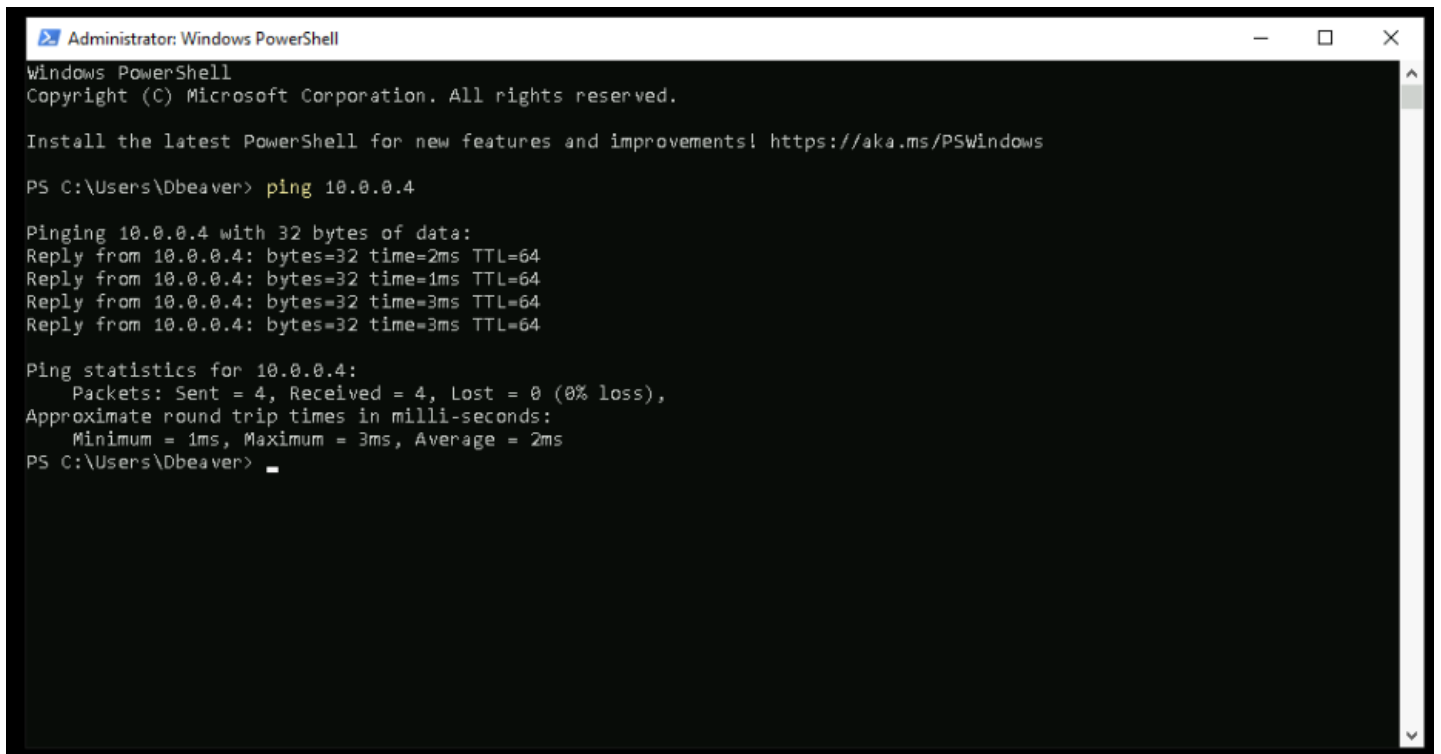
1. Connect to your Windows VM by using RDP.

2. Check that your Linux VM has access to your SQL DB private IP.

● Copy the private IP address of your database in **Virtual machines** -> **Your SQL DB** -> **Properties** -> **Networking**.



- Ping this IP in the terminal:

A screenshot of a Windows PowerShell terminal window titled "Administrator: Windows PowerShell". The window has a black background with white text. The text shows the PowerShell version, copyright information, and a link to update PowerShell. The user has entered the command "ping 10.0.0.4". The output shows four successful replies from 10.0.0.4 with varying times and TTL values. Finally, it shows ping statistics: 4 packets sent, 4 received, 0% loss, with minimum, maximum, and average round trip times.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

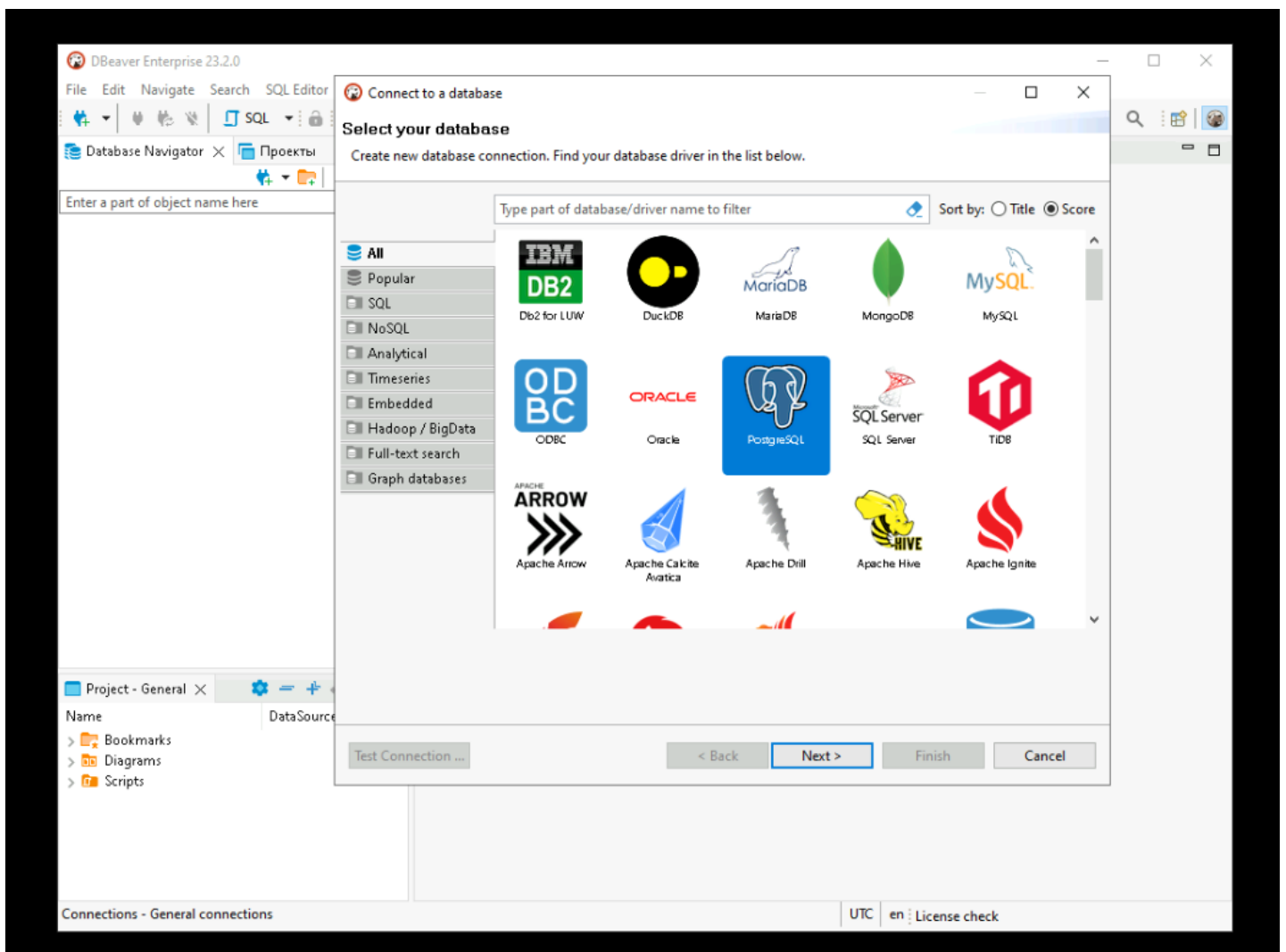
PS C:\Users\Dbeaver> ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=2ms TTL=64
Reply from 10.0.0.4: bytes=32 time=1ms TTL=64
Reply from 10.0.0.4: bytes=32 time=3ms TTL=64
Reply from 10.0.0.4: bytes=32 time=3ms TTL=64

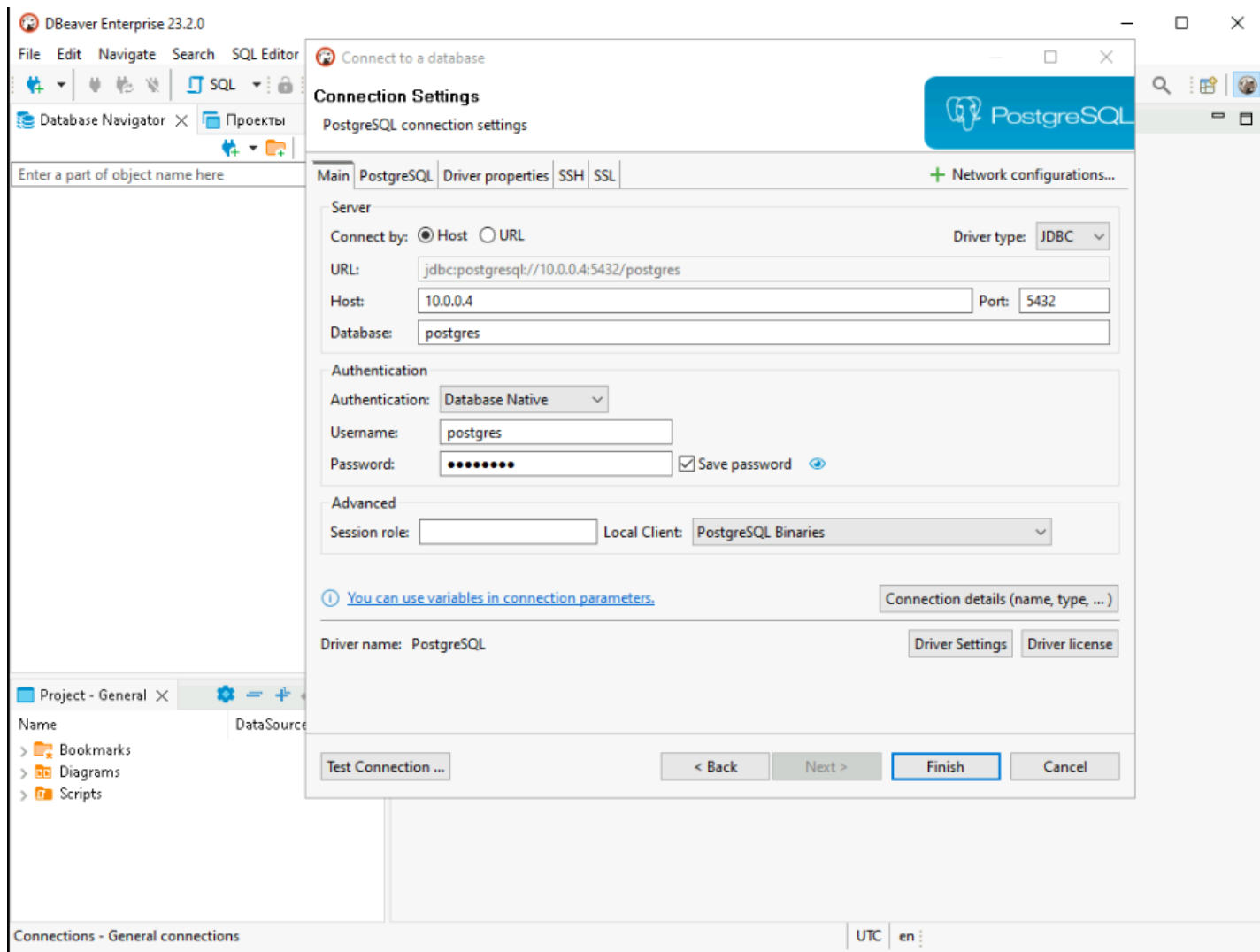
Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 3ms, Average = 2ms
PS C:\Users\Dbeaver> _
```

If everything was done correctly in the previous steps, you will get the result as in the screenshot above. All that remains is to connect our DBeaver to your database.

3. Add your license when starting the application.
4. Create a new connection and choose your database (for example, PostgreSQL).



5. In the **Host** field, specify the Private IP address of your database and fill in other fields.



Perfect! Your connection is ready.

