



# CloudBeaver user guide

# Table of contents

[Application overview](#)

[Administration](#)

[Overview](#)

[Server configuration](#)

[Connection Management](#)

[Access Management](#)

[Users](#)

[Roles](#)

[Authentication methods](#)

[User credentials storage](#)

[Supported databases](#)

[Features](#)

[Database navigator](#)

[Simple and Advanced View](#)

[Data editor](#)

[Data Filters](#)

[Data Ordering](#)

[Value Panel](#)

[JSON and Document View](#)

[Working with spatial/GIS data](#)

[Data export](#)

[Entity Diagrams](#)

[Document viewer](#)

[SQL Editor](#)

[Query Execution Plan](#)

[Cloud Explorer](#)

[Installation](#)

[Running from Docker](#)

[Build and deploy](#)

[Demo Server](#)

[Configuration](#)

[Server configuration](#)

[Configuring server datasources](#)

[Connection configuration](#)

[CloudBeaver and Nginx](#)

[Product configuration parameters](#)

[Command line parameters](#)

[CloudBeaver EE for AWS](#)

[Overview](#)

[AWS Settings](#)

[Cloud Explorer](#)

[Authentication / IAM](#)

[Authentication / AWS SSO](#)

[CloudBeaver Enterprise Edition](#)

[Overview](#)

[SSO](#)

[SAML authentication](#)

[OpenID authentication](#)

[Deployment](#)

# Application overview

## Features

- Open connection to remote database
- Support for preconfigured connections described in deploy config file
- Navigate and observe database structure: schemas, tables, views indexes, etc
- View properties of database objects
- View and edit table data
- Execute SQL queries

## Overview

### Open connection to remote database

1. Select *Connection > Manual > Custom* in a main menu
2. Select database
3. Set connection options and press **Connect** button

#### New connection

Type driver name...	
DB2 LUW	DB2 for Linux/Unix/Windows driver
MariaDB	MariaDB JDBC driver
MongoDB	Driver for MongoDB
MySQL	Driver for MySQL 8 and later
Oracle	Oracle JDBC driver
PostgreSQL	PostgreSQL standard driver
SQL Server	Microsoft JDBC Driver for SQL Server (MSSQL)
SQLite	SQLite JDBC driver
Apache Hive	Apache Hive JDBC
Athena	Simba AWS Athena driver
BigTable	Google BigTable Driver

OPTIONS    DRIVER PROPERTIES    SSH TUNNEL

CANCEL    TEST CONNECTION    CREATE

**Driver** PostgreSQL    **Connection name \*** PostgreSQL@localhost

**Host \*** localhost

**Port** 5432

**Database** postgres

**Description**

**AUTHENTICATION**

**User name**  **User password**

Save credentials

**SETTINGS**

Show all databases  
 Show template databases  
 Show unavailable databases

**User role**

## Navigate database structure

After the connection has been established, the left application panel contains a navigation tree with a database structure

1. Click element in the tree to open/close nested elements
2. Double-click to open object details on the right application panel
3. Hover element and click on the burger icon to open a context menu of the element
  - the context menu of a connection allows you to close the connection **Disconnect**) or open the SQL editor (**SQL**)

**CloudBeaver**

**CONNECTION** ▾

- PostgreSQL (local)
- postgres
  - Schemas
    - public
    - test
      - Tables
        - actor
        - actor123
      - Views
      - Materialized Views
      - Indexes
      - Functions
      - Sequences
      - Data types
      - Aggregate functions
  - Roles
  - Administer
  - Extensions

**Properties**

#	ACTOR_
123	ACTOR_

## View properties of database object

1. Open a new connection
2. Select a desired object in the navigation tree, then double-click on it
3. Object property viewer is opened on the right panel

The screenshot shows the CloudBeaver interface with the 'Properties' tab selected. In the navigation tree on the left, 'film' is selected. The main panel displays the following properties for the 'film' table:

Table Name	film
Object ID	16448
Owner	postgres
Tablespace	pg_default
Row Count Estimate	1000
Has Oids	<input type="checkbox"/>
Partitions	<input type="checkbox"/>

## View and edit table data

1. Open a new connection
2. Select a table element in the navigation tree, then double-click on it
3. Table viewer is opened in the right panel

The screenshot shows the CloudBeaver interface with the 'Data' tab selected. In the navigation tree on the left, 'film' is selected. The main panel displays the data from the 'film' table in a grid format:

#	FILM_ID	TITLE	DESCRIPTION
1	6	AGENT TRUMAN	A Intrepid Panorama of a Robot And a...
2	7	AIRPLANE SIERRA	A Touching Saga of a Hunter And a B...
3	9	ALABAMA DEVIL	A Thoughtful Panorama of a Database...
4	12	ALASKA PHANTOM	A Fanciful Saga of a Hunter And a Pas...
5	213	DATE SPEED	A Touching Saga of a Composer And ...
6	13	ALI FOREVER	A Action-Packed Drama of a Dentist A...
7	14	ALICE FANTASIA	A Emotional Drama of a A Shark And ...
8	15	ALIEN CENTER	A Brilliant Drama of a Cat And a Mad ...
9	16	ALLEY EVOLUTION	A Fast-Paced Drama of a Robot And a...

## Edit cell

Double-click cell to edit it. You can submit changes or cancel editing.

**Note** Submitted changes will be immediately applied to the database

address

Properties Data Diagram

Enter a SQL expression to filter results

#	123 address_id	RBC address	RBC add
1	10	1795 Santiago de Compostela Way	
2	23	1417 Lancaster Avenue	
3	24	1688 Okara Way	
4	25	262 A Corua (La Corua) Parkway	

### Copy cells

- Use drag-n-drop to select a range of cells
- Press Ctrl to add a new selection to the existing one
- Click the index cell to select a row
- Click the column cell to select a column

**Note** If the data table is big and only part of it is downloaded, only the loaded cells will be selected

- Press **Ctrl-C** to copy the selected cells to the clipboard in CSV format

### Execute SQL query

1. Open a new connection
2. Click **SQL** in the main menu
3. SQL editor with results panel is opened
4. Write SQL query and press **Ctrl-Enter** or click the **Execute** icon
5. Query results will be shown in a tab on the SQL results panel

actor    film    sql-1 (PostgreSQL (local))

1 SELECT \* from film;

Result

#	123 FILM_ID	RBC TITLE	RBC DESCRIPTION
1	6	AGENT TRUMAN	A Intrepid Panorama of a Rob...
2	7	AIRPLANE SIERRA	A Touching Saga of a Hunter ...

# Administration

## Features

- Initial server configuration
- Connection management
- User management
- Server configuration

## Overview

### Initial server configuration

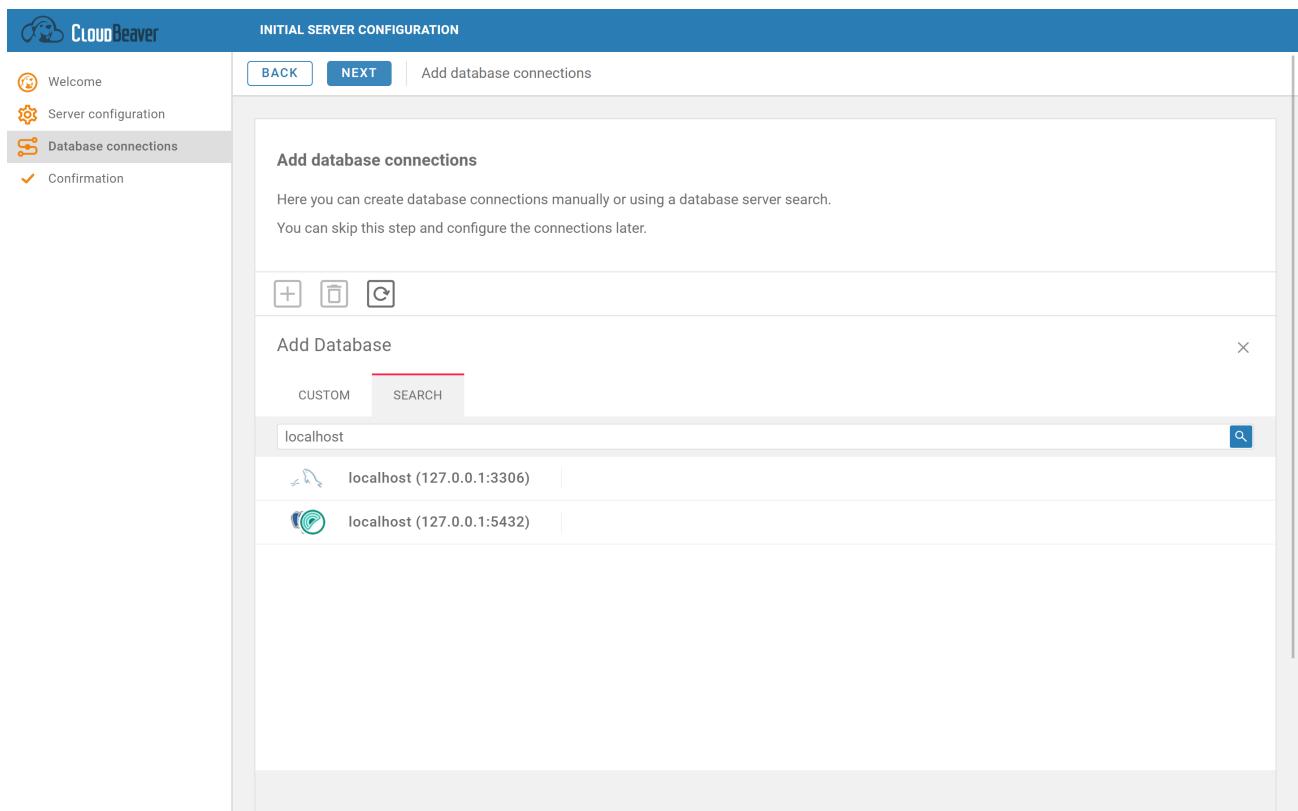
The first time you run the application, the application will require an initial configuration. On the initial configuration you will need to configure some things:

1. Server configuration - base application settings such as: administrator account, server info and base application behavior

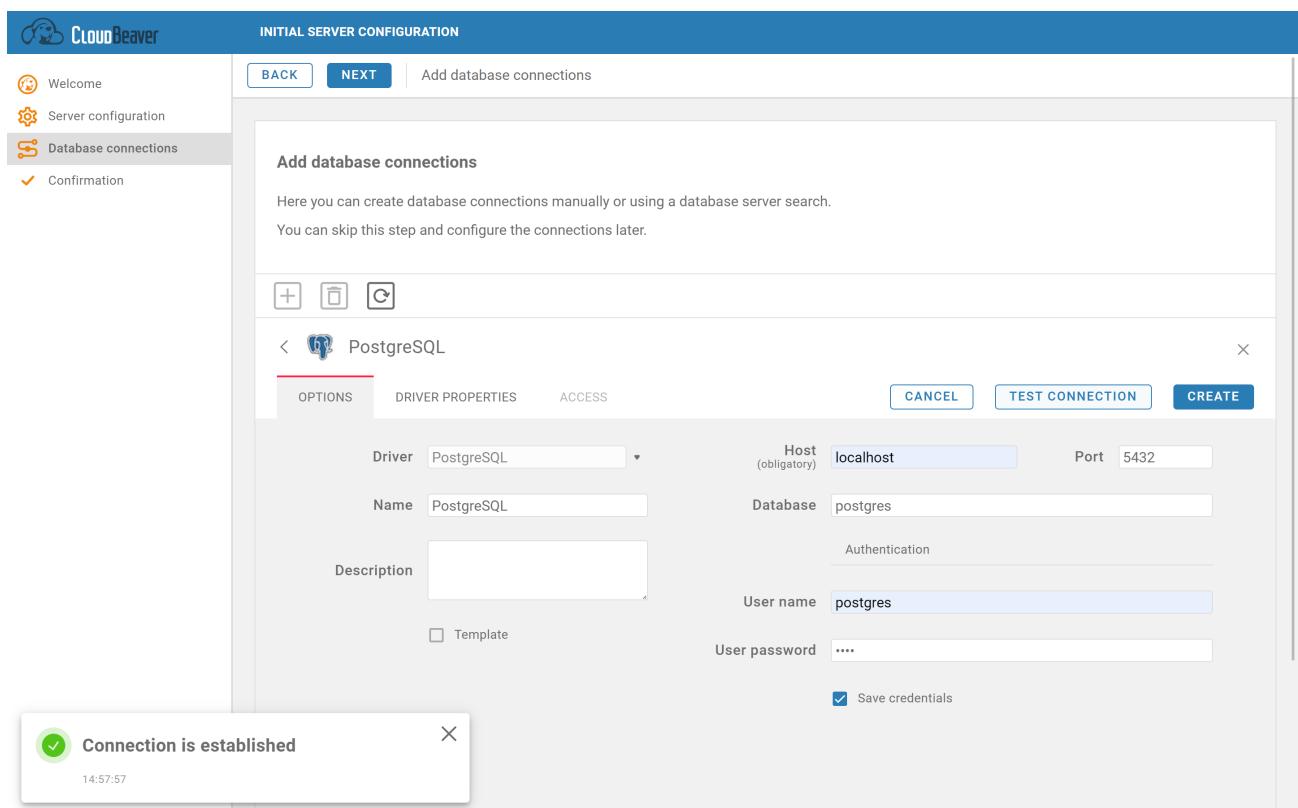
The screenshot shows the 'INITIAL SERVER CONFIGURATION' step of the CloudBeaver setup wizard. The left sidebar lists steps: Welcome, Server configuration (selected), Database connections, and Confirmation. The main area has 'BACK' and 'NEXT' buttons. A message says: 'You can configure the main server parameters here.' Below it, a note states: '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 section is divided into 'Server Info' and 'Configuration'. Under 'Server Info', fields include 'Server Name \*' (CloudBeaver), 'Session lifetime \*' (30), 'Administrator' (disabled), 'Admin user name \*' (cbadmin), and 'Admin password \*' (disabled). Under 'Configuration', three toggle switches are shown: 'Allow anonymous access' (on), 'Enable users authentication' (on), and 'Enable custom connections' (off). Descriptions for each switch are provided.

Setting	Description
Allow anonymous access	Allows to work with CloudBeaver without user authentication
Enable users authentication	Allows users to authenticate. Otherwise only anonymous access is enabled
Enable custom connections	Allows users to create custom connections. Otherwise, all new connections can be created from the administration part only

2. Database connections - in that step you can add some common connections for yours databases



The application will try to find databases on the host machine, so you can simply add existing databases or enter hosts to search in (hosts can be divided by spaces or comma):



The screenshot shows the 'INITIAL SERVER CONFIGURATION' screen in CloudBeaver. On the left sidebar, there are five tabs: 'Welcome', 'Server configuration', 'Database connections' (which is selected and highlighted in orange), and 'Confirmation'. At the top right, there are 'BACK' and 'NEXT' buttons, and a link to 'Add database connections'. The main area is titled 'Add database connections' and contains instructions: 'Here you can create database connections manually or using a database server search. You can skip this step and configure the connections later.' Below this is a table with three rows, each representing a database connection:

	NAME	ADDRESS	TEMPLATE
<input type="checkbox"/>	PostgreSQL	localhost:5432	<input type="checkbox"/>
<input type="checkbox"/>	MySQL (local)	172.17.0.1:3306	<input type="checkbox"/>
<input type="checkbox"/>	SQLite - Chinook (Sample)		<input type="checkbox"/>

## Entering administration page

After the initial server configuration, you can login as an administrator:

- If the authentication is enabled: *Settings (top right corner, the cog icon) > login*
- Another option is to enter `/#/admin` page

The screenshot shows a 'Authentication' dialog box centered over a dark background. The dialog has two tabs at the top: 'LOCAL' (which is selected) and 'AWS'. It contains two input fields: 'User name' and 'User password', both with placeholder text. At the bottom right is a blue 'LOGIN' button.

After successful authentication with the administrator account, you can enter *Administration* page:

- via settings *Settings (top right corner, the cog icon) > Administration*
- Directly go to `/#/admin` page

## Connection management

On the administration page, select the *Connection Management* tab. On the *Connection Management* tab you can create or delete connections, and configure connections (connection settings, users access).

### 1. Connection list

A screenshot of the CloudBeaver interface showing the 'Connection Management' tab. The left sidebar has 'Connection Management' selected. The main area shows a table with two columns: 'CONNECTION NAME' and 'ADDRESS'. Three connections are listed: 'PostgreSQL (Template)' at 'localhost:5432', 'PostgreSQL@localhost' at 'localhost:5432', and 'SQLite - Chinook (Sample)'. Each connection has a small icon and a dropdown arrow to its left.

### 2. Connection creation

Click on the Add icon on the tools bar to open the connection creation wizard:

A screenshot of the 'Add Database' wizard in CloudBeaver. The left sidebar has 'Connection Management' selected. The main area is titled 'Add Database' with tabs for 'CUSTOM' and 'SEARCH'. A search bar says 'Type driver name...'. Below it is a list of drivers with icons: MySQL, Oracle, PostgreSQL, SQL Server, SQLite, ClickHouse, Derby Server, Firebird, H2 Embedded, and Trino.

The connection can be created from the database driver or from databases searched in the entered hosts (hosts can be divided by spaces or comma).

Before adding a connection you can test the connection. For that, you should fill in the authentication parameters (if needed). If the *Template* is checked, then users can add that connection from *Connection > From Template* otherwise the connection will be presented in the navigation tree if the user has access to that connection (directly or by role). Connection access can be configured on the connection *Access* tab.

The connection authentication parameters can be filled in by the administrator and saved for all users, so they do

not need to enter the connection credentials.

### 3. Connection edit

The connection can be edited after creation. In the connections table click on the connection name or the angle icon.

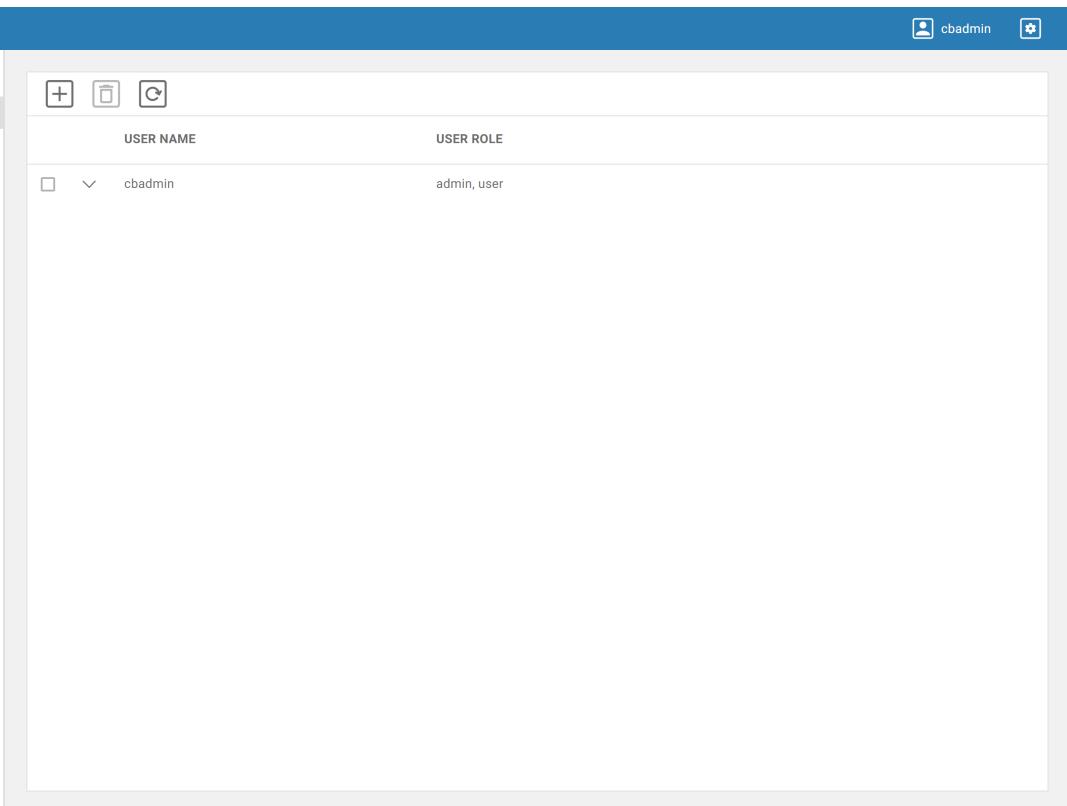
### 4. Connection deletion

The connection can be deleted. Select the connections in the table and click on the trash icon on the tools bar.

## User management

On the administration page, select the *Users* tab. On the *Users* tab you can create or delete, and create or edit users.

### 1. User list



USER NAME	USER ROLE
cbadmin	admin, user

### 2. User creation

Click on the Add icon on the tools bar to open the user creation form:

The screenshot shows the 'User Creation' dialog in CloudBeaver. The 'INFO' tab is selected. In the 'CREDENTIALS' section, there are fields for 'User name \*' (with placeholder 'cbadmin'), 'User password \*', and 'Repeat password \*'. In the 'USER ROLE' section, there are checkboxes for 'Admin' and 'User'. Below the dialog, a preview table shows a single row: 'USER NAME' (cbadmin) and 'USER ROLE' (admin, user). There is also a small info icon next to the preview table.

You can set the user credentials and roles. You can also configure the connections access.

Connection access can be provided via a user role, or directly.

### 3. User edit

The user can be edited after the creation. In the users table, click on the user name or the angle icon.

### 4. User deletion

The User can be deleted. Select the users in table and click on the trash icon on the tools bar.

## Server configuration

On the administration page select the *Server configuration* tab. On the *Server configuration* tab you can edit the base application settings.

The screenshot shows the 'Server configuration' dialog in CloudBeaver. The 'SERVER INFORMATION' tab is active. It contains fields for 'Server Name \*' (CloudBeaver CE Server), 'Server URL \*' (http://stage.web.dbeaver.net:8090), and 'Session lifetime' (30). The 'SECURITY' tab has two options: 'Save credentials' (selected) and 'Save users credentials'. The 'AUTHENTICATION SETTINGS' tab has one option: 'Allow anonymous access' (unchecked). On the right side, under 'CONFIGURATION', there are two sections: 'Enable custom connections' (selected) and 'Navigator simple view'.

# Server configuration

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 shows the 'Server configuration' page in CloudBeaver. On the left sidebar, 'Server configuration' is selected. The main area is divided into several sections:

- SERVER INFORMATION**:
  - Server Name \***: Cloudbeafer EE Web Server
  - Server URL \***: http://localhost:3100
  - Session lifetime \***: 30
- CONFIGURATION**:
  - Enable custom connections**: Allows users to create custom connections. Otherwise, all new connections can be created from the administration part only.
  - Navigator simple view**: By default, all user's new connections will contain only basic information in navigation tree.
- SERVICES**:
  - AWS**: AWS services
- AUTHENTICATION SETTINGS**:
  - Allow anonymous access**: Allows to work with CloudBeaver without user authentication.
  - Local**: Local name/password based authentication
  - AWS IAM**: Amazon Web Services authentication
  - OpenId**: Openid authentication provider. Edit configurations
  - SAML**: SAML authentication provider. Edit configurations
- SECURITY**:
  - Save credentials**: Allow to save credentials for pre-configured database.
  - Save users credentials**: Allow to save credentials for non-admin users.

## 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 Management

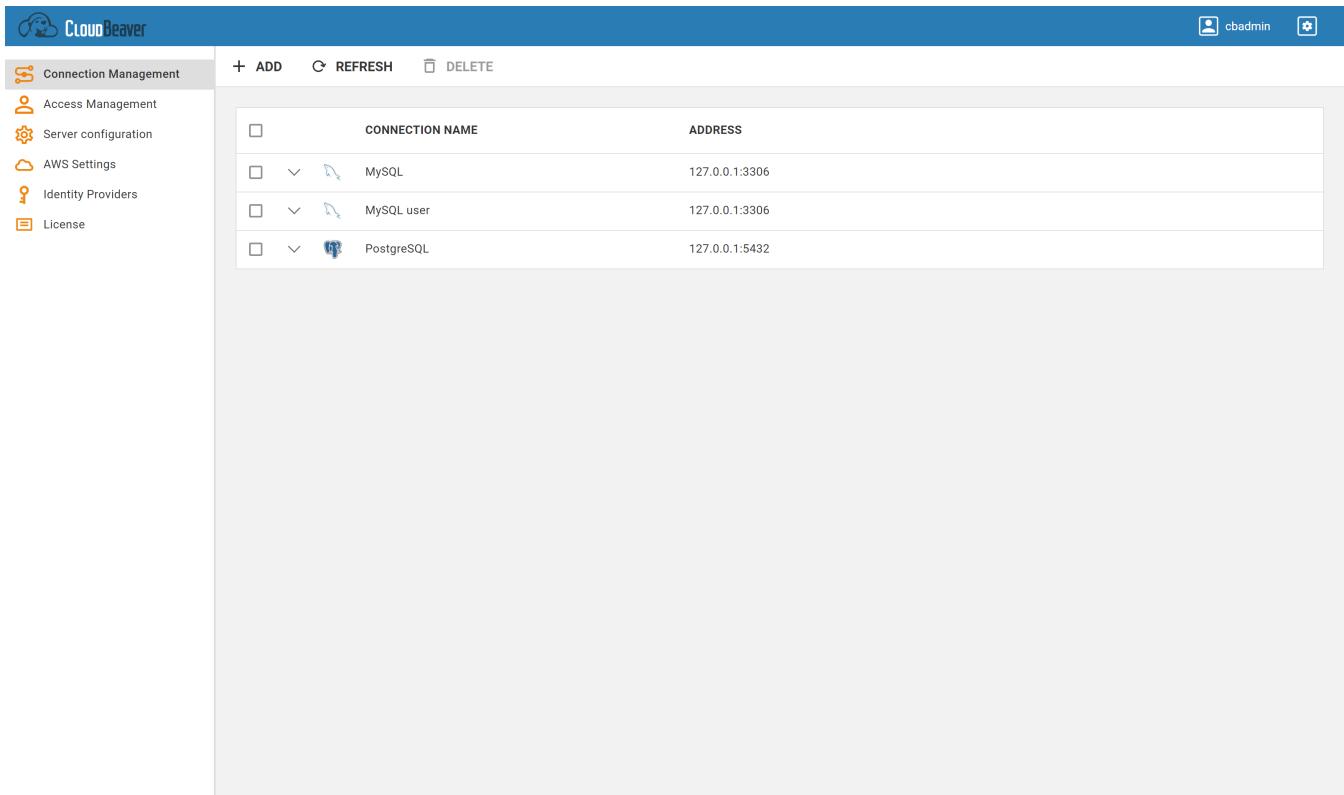
## Features

- [Description](#)
- [Creation](#)
- [Connection Form](#)

## Overview

### Description

You can add, edit, or remove shared database connections or database connection templates on the [Connection Management](#) page in administration.



The screenshot shows the CloudBeaver Connection Management interface. The left sidebar has a 'Connection Management' section selected. The main area displays a table with three rows:

	CONNECTION NAME	ADDRESS
<input type="checkbox"/>	MySQL	127.0.0.1:3306
<input type="checkbox"/>	MySQL user	127.0.0.1:3306
<input type="checkbox"/>	PostgreSQL	127.0.0.1:5432

### Creation

Click the [Add](#) button in the top toolbar to open the connection creation form.

The screenshot shows the CloudBeaver interface with the 'Connection Management' tab selected. In the center, there's a search interface titled 'Add Database'. The 'SEARCH' tab is active. A search bar contains the text 'localhost'. Below it, two database entries are listed: 'localhost (127.0.0.1:3306)' and 'localhost (127.0.0.1:5432)'. At the bottom of the search interface, there are columns for 'CONNECTION NAME' and 'ADDRESS', with a single entry: 'MySQL' and '127.0.0.1:3306'.

You can use the **Search** tab to find databases on the cloudbeaver host machine or provided host. You can choose database type if several databases can be hosted on the same port. You can write several hosts to search on: `localhost`, `yourhost.com` or `localhost, yourhost.com`. Click on the connection in the list to open the creation form (You can also select database type there).

The screenshot shows the CloudBeaver interface with the 'Connection Management' tab selected. In the center, there's a search interface titled 'Add Database'. The 'CUSTOM' tab is active. A search bar contains the placeholder 'Type driver name...'. Below it, a list of database drivers is shown in a table format:

ICON	DRIVER NAME	DESCRIPTION
	DB2 LUW	DB2 for Linux/Unix/Windows driver
	MariaDB	MariaDB JDBC driver
	MongoDB	Driver for MongoDB
	MySQL	Driver for MySQL 8 and later
	Oracle	Oracle JDBC driver
	PostgreSQL	PostgreSQL standard driver
	SQL Server	Microsoft JDBC Driver for SQL Server (MSSQL)
	SQLite	SQLite JDBC driver
	Apache Hive	Apache Hive JDBC
	Athena	Simba AWS Athena driver
	BigTable	Google BigTable Driver
	Cassandra	Driver for Apache Cassandra 2.x/3.x/4.x
	ClickHouse	Yandex ClickHouse

At the bottom of the search interface, there are columns for 'CONNECTION NAME' and 'ADDRESS', with a single entry: 'MySQL' and '127.0.0.1:3306'.

You can use the **Custom** tab to create a connection for the specified database or driver. You can search databases by name.

## Connection Form

You can set base connection parameters, driver settings, SSH tunnel, and access in the connection form. A connection template will be created if the **Template** checkbox is checked. To check the connection to the database, click on the **Test connection** button; if SSH is configured, it will be used to test the connection. Users will be asked to enter credentials if the connection requires authentication. An administrator can set authentication parameters and save them (**Save credentials** checkbox in **Authentication** section); in that case, any user that has access to a connection will be able to connect without entering credentials.

**CloudBeaver**

Connection Management    ADD    REFRESH    DELETE

< Oracle

OPTIONS    DRIVER PROPERTIES    SSH TUNNEL    ACCESS

Driver: Oracle    Connection name: Oracle@localhost

Host: localhost    Port: 1521

Database: ORCL

Template:

Description:

AUTHENTICATION

User name:  User password:   Save credentials

SETTINGS

Service type: SID    Role: NORMAL

CONNECTION NAME    ADDRESS

MySQL    127.0.0.1:3306

This screenshot shows the 'Connection Management' section of the CloudBeaver interface. A new Oracle connection is being configured. The 'OPTIONS' tab is selected, displaying fields for Driver (Oracle), Connection name (Oracle@localhost), Host (localhost), Port (1521), Database (ORCL), and a Template checkbox. The 'ACCESS' tab is also visible. Below the connection details, a table lists the connection name (MySQL) and its address (127.0.0.1:3306).

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

**CloudBeaver**

Connection Management    ADD    REFRESH    DELETE

< Oracle

OPTIONS    DRIVER PROPERTIES    SSH TUNNEL    ACCESS

Search for user or role name:

USER OR ROLE NAME    DESCRIPTION

Administrators see all connections.

cbadmin

Search for user or role name:

User    Standard user

cbadmin

CONNECTION NAME    ADDRESS

MySQL    127.0.0.1:3306

This screenshot shows the 'ACCESS' tab of the CloudBeaver interface for the Oracle connection. It lists users and roles with checkboxes for selection. On the left, a table shows 'USER OR ROLE NAME' and 'DESCRIPTION'. On the right, another table shows the same information with one row ('User') having a checked checkbox. Below the tables, a table lists the connection name (MySQL) and its address (127.0.0.1:3306).

# Users

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

## CloudBeaver CE

---

### Local user creation

1. Go to the Access management tab of the Administration Menu and press the Add button.
2. Create a username and password.
3. Grant a role to the user. It will define the user's permission (you can find more information about roles at [Role management](#) article).
4. Give connection access to the user in the Connection Access tab if it is necessary.
5. Press the Create button.

[https://github.com/dbeaver/cloudbeaver/wiki/images/administration/access\\_management/create\\_user\\_dialog.png](https://github.com/dbeaver/cloudbeaver/wiki/images/administration/access_management/create_user_dialog.png) The created user can be authorized to CloudBeaver using local authentication and has permission according to his 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 CloudBeaver EE instance with AWS IAM or Federated authentication for the first time, the appropriate user is created in the application with the User role by default. The administrator can change the user's role after that. The creation of new AWS and Federated users is not possible by the Administrator as it only works with real AWS and Federated users.

## CloudBeaver AWS

---

CloudBeaver Enterprise Edition for AWS allows you to configure only AWS and Federated users, because it does not have local access and local users cannot be created there.

# Roles

You can use roles to manage connection permissions in CloudBeaver. As a role is added to a user, the role connections become available for the user on the main page.

The Admin and User roles are available by default.

Role	Connections created in Easy Config	Connections created in Administration	Administration menu
Admin	Yes	Yes	Yes
User	Available by default. Can be deleted	Not available by default. Can be added	No

The Administrator can create **custom roles** on the Roles tab of the Administration menu.

Role creation

CUSTOM ROLES

OPTIONS USERS CONNECTIONS

CREATE

Search for connection name...

CONNECTION NAME	ADDRESS
<input type="checkbox"/> MySQL	localhost:3306

Search for connection name...

CONNECTION NAME	ADDRESS
<input type="checkbox"/> MySQL	localhost:3306
<input checked="" type="checkbox"/> Oracle@localhost	localhost:1521
<input type="checkbox"/> PostgreSQL	localhost:5432
<input type="checkbox"/> PostgreSQL (Template)	localhost:5432

Also, connection access can be configured in the Connection and User dialogs.

## Connection dialog

Connection Management

+ ADD REFRESH DELETE

OPTIONS DRIVER PROPERTIES SSH TUNNEL ACCESS

TEST CONNECTION SAVE

Search for user or role name

Search for user or role name

USER OR ROLE NAME	DESCRIPTION
Administrators see all connections.	
<input type="checkbox"/> User	Standard user

USER OR ROLE NAME	DESCRIPTION
<input type="checkbox"/> 1	test
<input type="checkbox"/> User	Standard user
<input type="checkbox"/> cbadmin	

# Authentication methods

CloudBeaver offers several authentication methods. The administrator can set them when configuring CloudBeaver for the first time. Or it can be done later in the Administration Menu.

## CloudBeaver CE

The screenshot shows the 'Server configuration' section of the CloudBeaver CE interface. It includes fields for 'Server Name' (CloudBeaver CE Server), 'Server URL' (https://), and 'Session lifetime' (30). The 'AUTHENTICATION SETTINGS' section is highlighted with a red box and contains two options: 'Allow anonymous access' (unchecked) and 'Local' (checked). Other sections include 'CONFIGURATION' (with 'Enable custom connections' and 'Navigator simple view' options) and 'SECURITY' (with 'Save credentials' and 'Save users credentials' options).

### Anonymous access

Users can work with CloudBeaver without authorization. Connections become available for anonymous access when the administrator:

1. creates connections in the Connection Management Menu and gives access to them for the User role (you can find more information for the roles at [Role management](#) article).
2. enables the Custom connections option in the Administration Menu. In this case connections can be configured on the main page by anonymous users and disappear after the session expirations.

### Local access

It is the local name/password based authentication. The administrator has to create users in the Administration and grant them a role which will define users' permissions (more information about users can be found at [Users](#) article).

## CloudBeaver EE

CloudBeaver Enterprise Edition also supports AWS IAM and SAML authentication methods.

The screenshot shows the 'Server configuration' section of the CloudBeaver EE interface. It includes fields for 'Server Name' (Cloudbeaver EE Web Server), 'Server URL' (https://), and 'Session lifetime' (30). The 'AUTHENTICATION SETTINGS' section is highlighted with a red box and contains three options: 'Allow anonymous access' (unchecked), 'Local' (checked), and 'AWS IAM' (checked). Other sections include 'CONFIGURATION' (with 'Enable custom connections' and 'Navigator simple view' options), 'SERVICES' (with 'AWS' and 'SAML' options), and 'SECURITY' (with 'Save credentials' and 'Save users credentials' options).

## AWS IAM access

Amazon Web Services authentication allows users to authorize to CloudBeaver EE with IAM credentials. Once an IAM user is authorized to CloudBeaver instance, the appropriate user is created in the application with the User role by default (you can find more information about AWS IAM authentication at [AWS IAM article](#)).

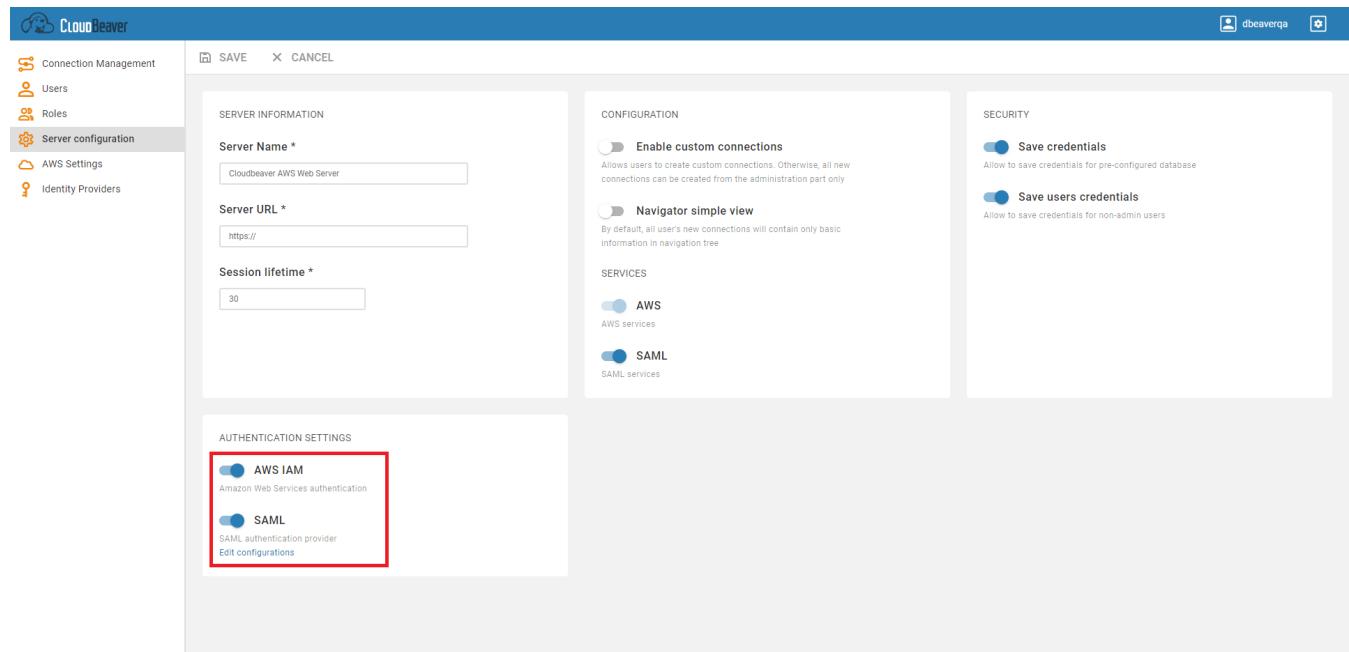
*Note: Local and AWS accounts, used during the first configuration of CloudBeaver EE instance, become associated with the administrator who configured it. It means that the administrator can login to the application with the local and the IAM credentials and a new user will not be created after using these IAM credentials.*

## SAML access

SSO (Single Sign-On) authentication can be used for access to CloudBeaver EE. Once an SSO user is authorized to CloudBeaver instance, the appropriate user is created in the application with the User role by default (you can find more information about SSO authentication at [Single Sign On article](#)).

# CloudBeaver AWS

CloudBeaver Enterprise Edition for AWS supports AWS IAM and SAML authentication methods, but local and anonymous authentication are not available in it.



# User 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.

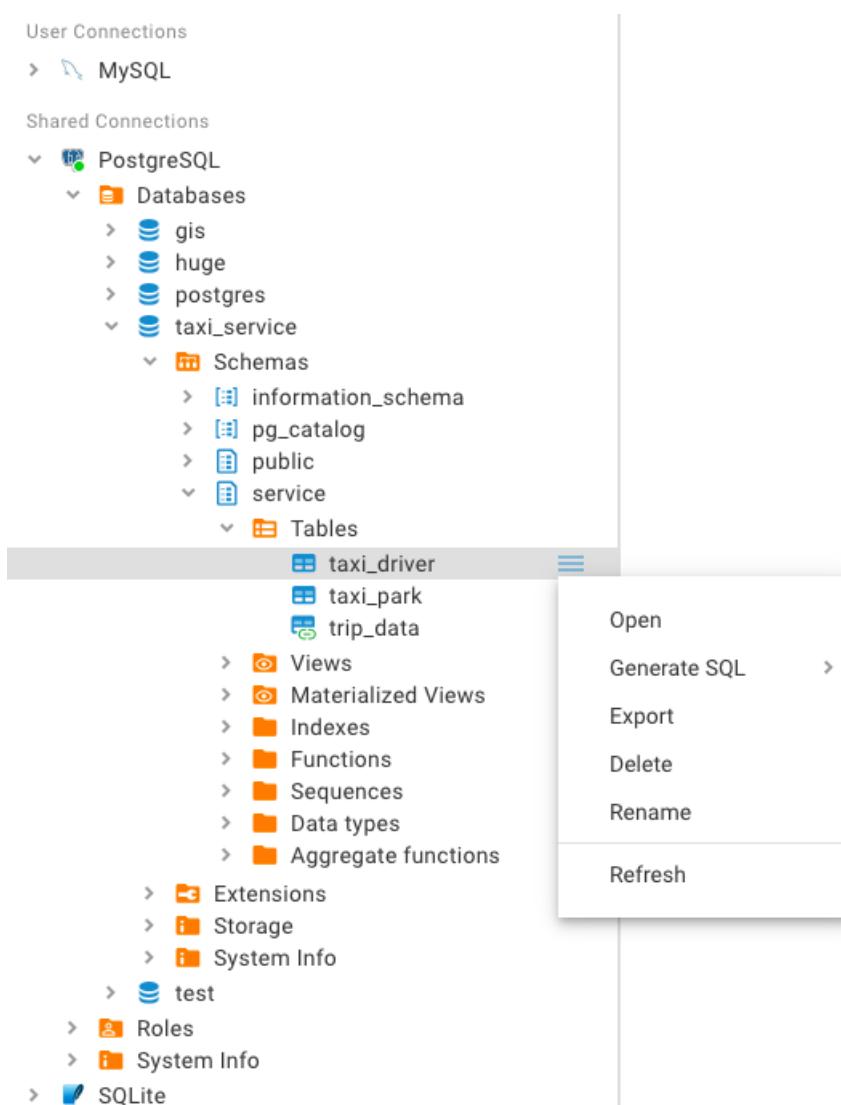
User configuration are stored in workspace sub-folders `user-projects / USER_NAME / .dbeaver`.

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](#)

# Database navigator

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.



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

<b>Menu item</b>	<b>Description</b>
Disconnect	Disconnects from the database
Delete	Deletes an object. <b>WARNING!</b> 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: - 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:

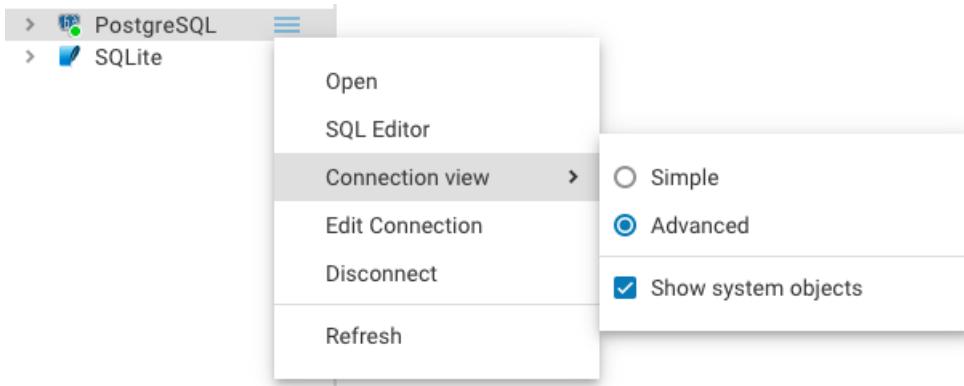
- open the connection;
- expand the connection in the Database Navigator;
- open the SQL Editor for the connection.

# 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.

# Data editor

## Features

---

- [Shortcuts](#)
- [Data Filters](#)
- [Data Ordering](#)
- [Value 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**.

The screenshot shows the CloudBeaver interface with a top toolbar. On the left is a search bar labeled "Enter a SQL expression to filter results". To its right are three icons: a magnifying glass, a green checkmark, and a red X. Below the toolbar is a table with columns: #, book\_ref, book\_date, and total\_amount. The table has 12 rows of data. A context menu is open over the first row, specifically over the "book\_ref" column cell containing "00034E". The menu items are: Sorting, Filters, Clipboard, Cell value, Custom, Reset filters / sorting, and Delete filter for book\_ref.

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 0002EO, then click **Filters -> Cell value** in the context menu and choose one of the expressions.

The screenshot shows a context menu open over a table cell containing "00034E". The menu is titled "Filters" and includes options like Sorting, Clipboard, Cell value, Custom, and Reset filters / sorting. The "Cell value" option is expanded, showing a list of SQL expressions related to the "book\_ref" column. The expressions include: book\_ref = '00034E', book\_ref <> '00034E', book\_ref > '00034E', book\_ref < '00034E', book\_ref LIKE '%00034E %', and book\_ref ILIKE '00034E'.

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.

This is a detailed view of the context menu for the "book\_ref" cell. The "Filters" section is expanded, showing the following options: Clipboard, Cell value, Custom, book\_ref IS NULL, book\_ref IS NOT NULL, Delete filter for book\_ref, Reset all filters, and 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.

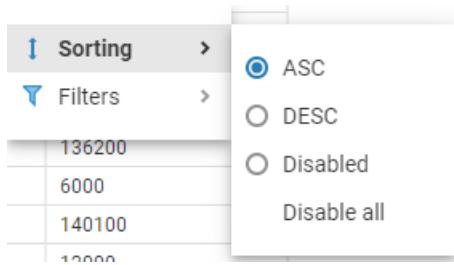
#	book_ref		book_date		total_amount	
---	----------	--	-----------	--	--------------	--

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 in value panel** on a cell context menu.

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

This screenshot shows the CloudBeaver Data Editor interface. The top navigation bar includes tabs for 'Properties', 'Data' (which is currently selected), and 'Diagram'. Below the navigation is a search bar with placeholder text 'Enter a SQL expression to filter results'. The main area is a table with columns: '#', 'film\_id', 'title', 'description', and 'release\_year'. A specific row for 'Bright Encounters' (film\_id 98) is selected. A context menu is open over this row, with the 'Show in value panel' option highlighted by a red box. To the right of the table, there is a vertical toolbar with a 'VALUE' button also highlighted by a red box. At the bottom of the screen are standard database management buttons: Refresh, Row Count (200), Sort, Filter, Save, Revert, Script, Export, and a status message indicating '200 row(s) fetched - 268ms'.

The **Value** viewer 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.

This screenshot shows the same CloudBeaver Data Editor interface as the previous one, but the Value panel is now open and visible on the right side of the screen. The 'Text' tab is selected, and the content area displays the string 'Ace Goldfinger'. The other tabs ('HTML', 'XML', 'JSON') are also visible but not selected. The rest of the interface, including the table, toolbar, and bottom buttons, remains the same as in the first screenshot.

# JSON and Document View

## Features

- 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.

The screenshot shows the CloudBeaver interface with the 'Data' tab selected. A search bar at the top contains the query 'year:1933, title:1'. Below the search bar, there are two sections of JSON data. The first section is for the movie 'King Kong' (1933), and the second is for a movie with an ID. At the bottom, there are navigation buttons for back, forward, and search, along with save, revert, and export options. The status bar indicates '200 row(s) fetched - 639ms'.

```
{
  "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/MVSBMTkxOTIxMDU2OV5BM15BanBnXkFtZTcwNjM5NjQyMg@@._V1_SX400_.jpg",
    "directors": [
      "Merian C. Cooper",
      "Ernest B. Schoedsack"
    ],
    "rating": 7,
    "rank": 3.551567890123456,
    "running_time_secs": 6.34
  }
}

year:1933, title:1234567890123456

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

### Document selection

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

Properties Data Diagram

Enter a SQL expression to filter results

year:1933,title:1

```

TABLE
JSON
  {
    year: 1933,
    title: "I",
    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/MV5BMkxOTIxMDU2OV5BM15BanBnXkFtZTcwNjHSNjQyMg@@._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".
      ]
    }
  }

```

200 200 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.

Properties Data Diagram

Enter a SQL expression to filter results

year:1933,title:1

```

TABLE
JSON
  ✓ 1  {
    2    "year": 1933,
    3    "title": "I",
    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/MV5BMkxOTIxMDU2OV5BM15BanBnXkFtZTcwNjHSNjQyMg@@._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: [
        "Fay Wray2".
      ]
    }
  }

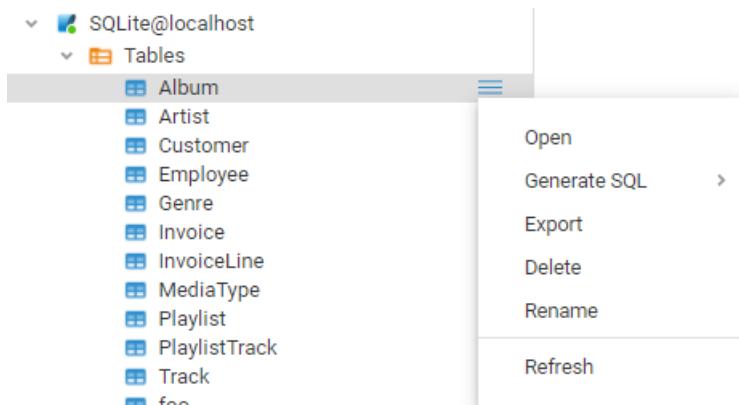
```

200 200 row(s) fetched - 639ms

# Data export

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:

3. CSV
4. DBUbit
5. JSON
6. Markdown
7. Source code
8. SQL
9. TXT
10. XML
11. XLSX
12. HTML

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

## Export configuration (CSV)

X

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.

# Entity Diagrams

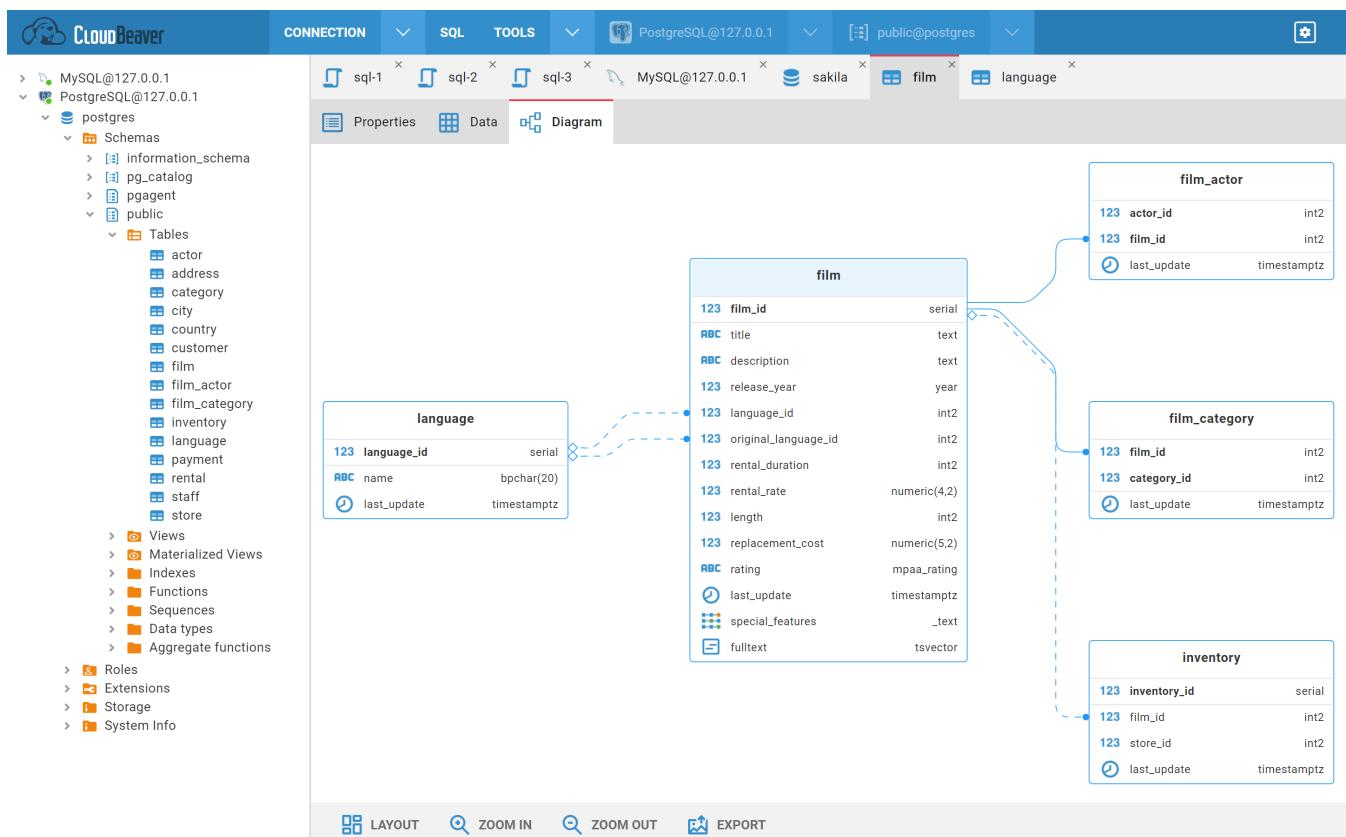
## Features

- Display entities with attributes
- Display relations
- Navigation to entity/attribute metadata editor
- Highlighting
- Auto layout
- Entity drag & drop
- Zoom
- Export to png/svg

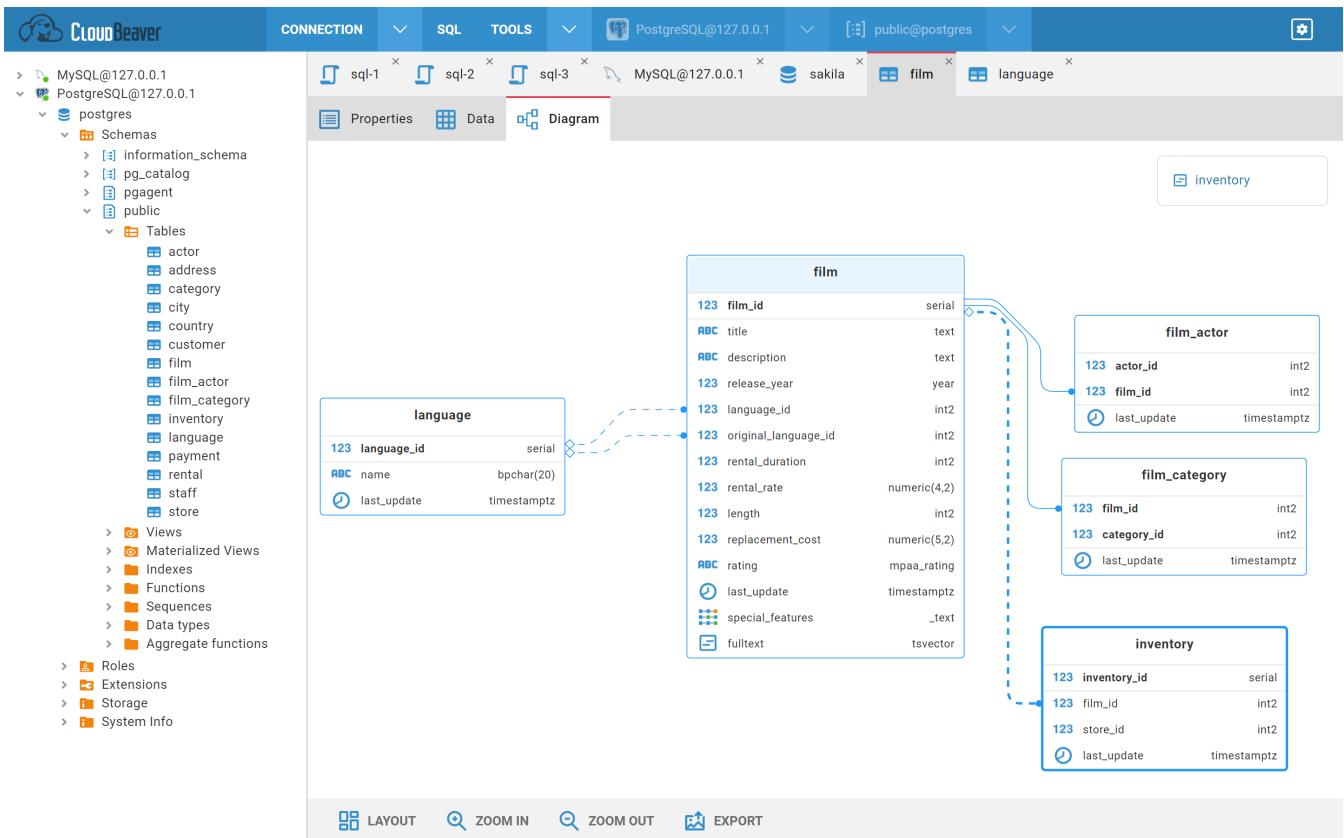
## Overview

### Display entities with attributes

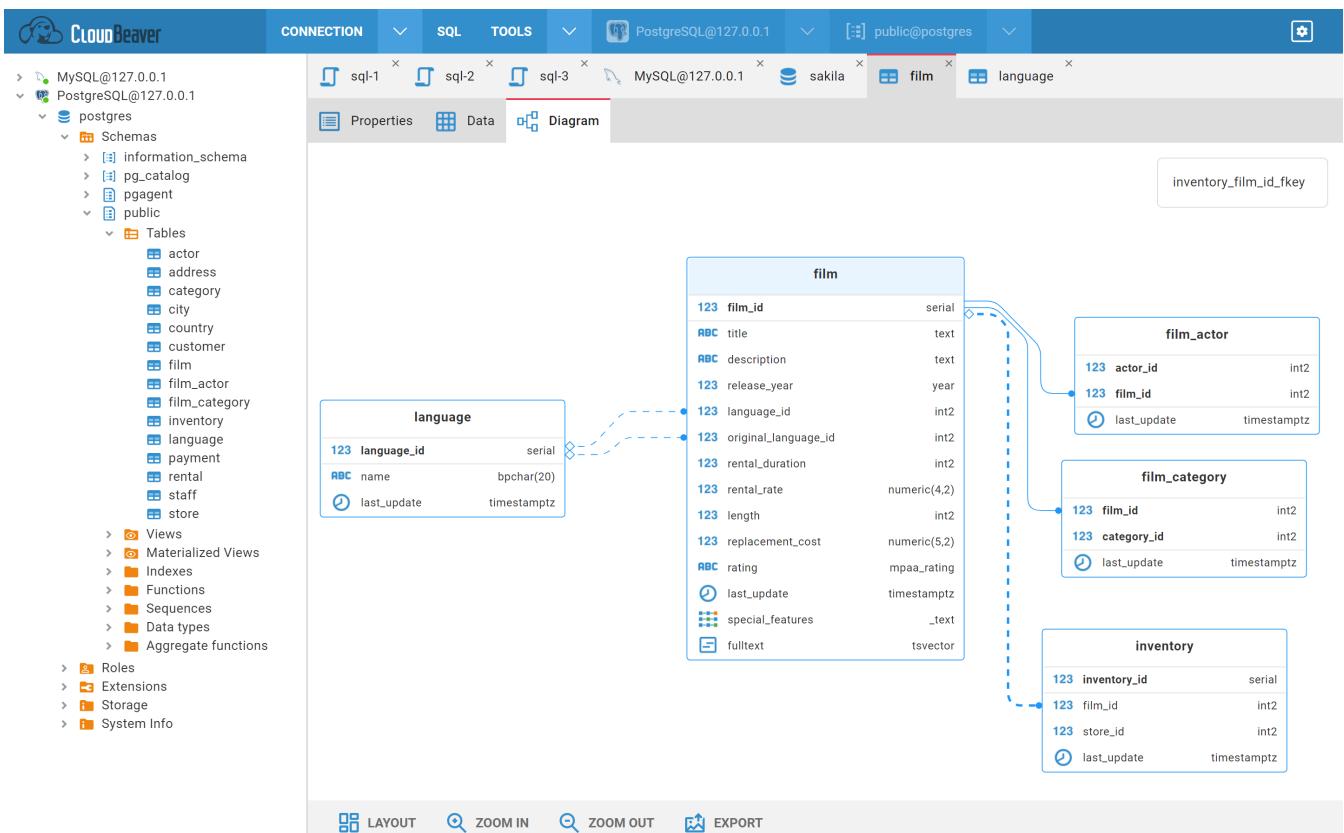
1. Navigate to your connection and open a table or schema
2. Select the "Diagram" tab (if the tab is not presented then the object does not support the diagram presentation)



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

The screenshot shows the CloudBeaver interface with a database schema diagram for the 'sakila' database. The 'film' table is highlighted. A modal dialog titled 'Export diagram' is open, showing options for 'File format' (SVG) and 'Transparent background'. The background shows the database schema with tables like 'film\_actor', 'film\_category', and 'inventory' connected to the 'film' table.

**Export diagram**

File format:   Transparent background

**CANCEL** **EXPORT**

**film**

**film\_actor**

- 123 actor\_id int2
- 123 film\_id int2
- last\_update timestamptz

**film\_category**

- 123 film\_id int2
- 123 category\_id int2
- last\_update timestamptz

**inventory**

- 123 inventory\_id serial
- 123 film\_id int2
- 123 store\_id int2
- last\_update timestamptz

**film**

**actor**

**address**

**category**

**city**

**country**

**customer**

**film**

**film\_category**

**inventory**

**language**

**payment**

**rental**

**staff**

**store**

**Views**

**Materialized Views**

**Indexes**

**Functions**

**Sequences**

**Data types**

**Aggregate functions**

**Roles**

**Extensions**

**Storage**

**System Info**

LAYOUT ZOOM IN ZOOM OUT EXPORT

# SQL Editor

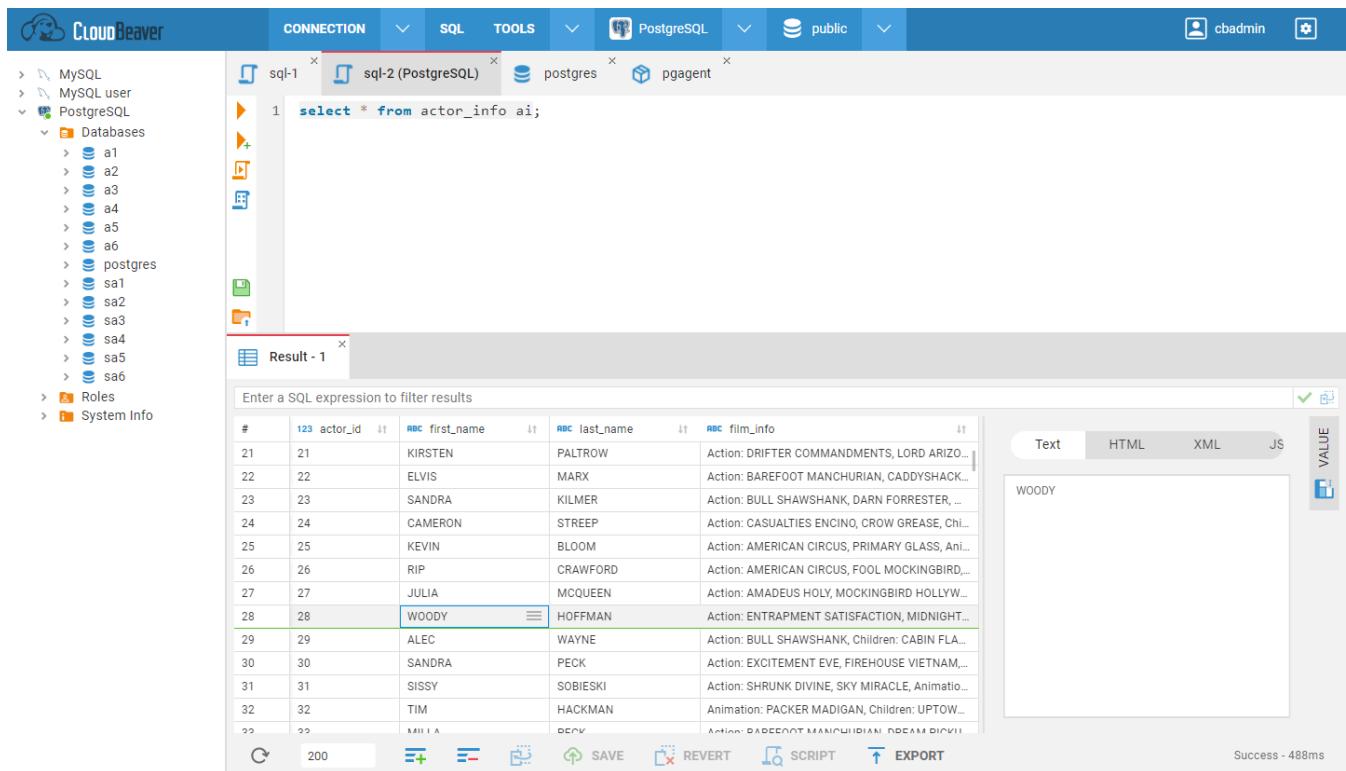
## Features

- Description
- Shortcuts
- Statement Execution
- Script execution
- Query Execution Plan

## Overview

### Description

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



The screenshot shows the CloudBeaver interface with a PostgreSQL connection. The left sidebar lists databases (a1-a6, postgres, sa1-sa6) and roles. The main area has tabs for sql-1, sql-2 (PostgreSQL), postgres, and pgagent. The sql-1 tab contains the query: `select * from actor_info ai;`. The result pane shows a table with columns: #, actor\_id, first\_name, last\_name, and film\_info. The row for actor\_id 28, first\_name WOODY, last\_name HOFFMAN is selected. To the right, a JSON viewer shows the selected row with fields: Text, HTML, XML, and JS. The status bar at the bottom right indicates "Success - 488ms".

### 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

### 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.

```

sql-1 x sql-2 (PostgreSQL) x postgres x pgagent x
1 select * from actor_info ai;
2 Select * from actor a;

Result - 2 x Result - 3 (1) x Result - 3 (2) x

Enter a SQL expression to filter results
# 123 actor_id first_name last_name film_info
1 1 PENELOPE GUINNESS Animation: ANACONDA CONFESSIONS, Children: ...
2 2 NICK WAHLBERG Action: BULL SHAWSHANK, Animation: FIGHT JA...
3 3 ED CHASE Action: CADDYSHACK JEDI, FORREST SONS, Clas...
4 4 JENNIFER DAVIS Action: BAREFOOT MANCHURIAN, Animation: AN...
5 5 JOHNNY LOLLOBRIGIDA Action: AMADEUS HOLY, GRAIL FRANKENSTEIN, ...
6 6 BETTE NICHOLSON Action: ANTITRUST TOMATOES, Animation: BIKIN...
7 7 GRACE MOSTEL Action: BERETS AGENT, EXCITEMENT EVE, Anima...
8 8 MATTHEW JOHANSSON Action: CAMPUS REMEMBER, DANCES NONE, Ani...
9 9 JOE SWANK Action: PRIMARY GLASS, WATERFRONT DELIVERA...
10 10 CHRISTIAN GABLE Action: LORD ARIZONA, WATERFRONT DELIVERA...
11 11 ZERO CAGE Action: DANCES NONE, HANDICAP BOONDOCK, ...
12 12 KARL BERRY Action: STAGECOACH ARMAGEDDON, Animation: ...
13 13 TIMA WOOD Action: ANTITRUST TOMATOES, CHILDREN'S BUCK

Success - 490ms

```

## 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.

```

sql-1 x sql-2 (PostgreSQL) x postgres x pgagent x jail x
1 INSERT INTO public.jail ("name")
2   VALUES ('Joker');
3 ▶ INSERT INTO public.jail ("name")
4   VALUES ('Penguin');
5 INSERT INTO public.jail ("name")
6   VALUES ('Court of Owls');

Statistics - 1 x

Queries: 1 / 3
Duration: 3 ms
Updated Rows: 1
Executing... CANCEL

INSERT INTO public.jail ("name")
VALUES ('Penguin')

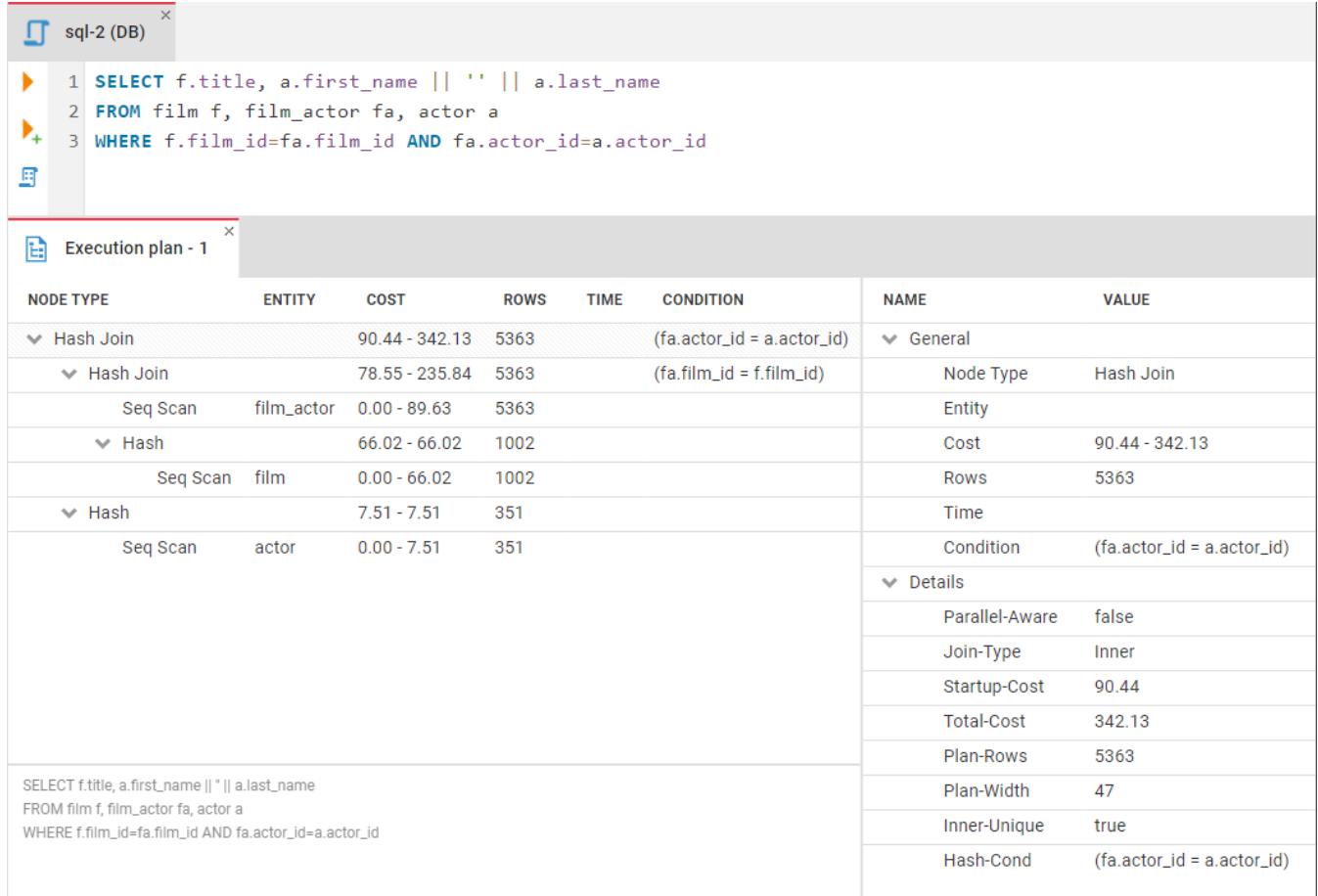
```

# Query 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.



The screenshot shows the CloudBeaver interface with a query editor and an execution plan viewer. The query editor contains the following SQL:

```
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
```

The execution plan viewer displays the following details:

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)

Below the table, the original SQL query is shown again:

```
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
```

On the far right, there is a detailed section with the following parameters:

Details	Value
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)

# Running from Docker

CloudBeaver container image is on DockerHub:<https://hub.docker.com/r/dbeaver/cloudbeaver>.

- `dbeaver/cloudbeaver:latest` - latest release build.
- `dbeaver/cloudbeaver:dev` - latest developer build.

## Installation

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

```
sudo docker pull dbeaver/cloudbeaver:latest
```

## Running

To run cloudbaver in the terminal:

```
sudo docker run --name cloudbeaver --rm -ti -p 8080:8978 -v /var/cloudbeaver/workspace:/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
```

## 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 '{ print $2 }' | cut -f2 -d: | head -n1)
docker run --name cloudbeaver --rm -ti -p 8080:8978 --add-host=host.docker.internal:${CB_LOCAL_HOST_ADDR} -v /var/cloudbeaver/workspace:/opt/cloudbeaver/workspace dbeaver/cloudbeaver:dev
```

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

# Build and deploy

## Introduction

CloudBeaver is a web server which provides a rich web interface.  
It consists of two parts:

- Server back-end. It is written in Java and reuses "platform" libraries of DBeaver.
- Front-end part. It is written in TypeScript and JavaScript.

This build process is relatively complicated (and also consists of two parts).

## Prerequisites

- [Java 11](#).
- [Apache Maven \(3.6+\)](#)
- [Node.js \(14.x\)](#)
- [Yarn](#)
- [NPM + Lerna](#)

### Install prerequisites on Ubuntu:

Add Node.js (version 14.x minimal version 14.17) and Yarn latest version repo:

```
curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -
echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.list
curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
```

Install packages:

```
sudo apt update
sudo apt install openjdk-11-jdk
sudo apt install maven
sudo apt install yarn
sudo apt install nodejs
sudo apt install npm
sudo npm install -g lerna
```

### Install prerequisites on Windows:

- Download and install [AdoptOpenJDK](#)
- Download and extract [Maven](#).
- Add path to Maven bin folder in system PATH
- Download and install [Node.js](#). NPM comes along with Node.js, NPX/Lerna will be installed in the build script.
- Download and install [Yarn](#)

If lerna is not installed along with Node.js you can do it manually:

```
npm install -g lerna
```

## Build and deploy

```
git clone https://github.com/dbeaver/cloudbeaver.git
cd cloudbeaver/deploy
./build.sh
```

The final artifacts can be found in [deploy/cloudbeaver](#).

## Running server

```
cd cloudbeaver  
./run-server.sh
```

By default, the server listens to port `8978` (you can change it in `conf/cloudbeaver.conf`). So just navigate to <http://localhost:8978>. You can configure [Nginx](#), Apache or any other web server in front of it.

*Note: you must be in the server root directory to run it.*

If you need to run it from some other directory, then you can set the environment variable `CLOUDBEAVER_HOME` to the server root directory.

# Demo Server

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>**

## 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!

# 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 folder `/etc/cloudbeaver/`. 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 JSONC format (JSON with comments and without redundant quotes). It can be parsed by most of the JSON parsers 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` but 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,  
  
        database: {  
            url: "jdbc:h2:${workspace}/.data/cb.h2.dat",  
            initialDataConfiguration: "conf/initial-data.conf",  
            pool: {  
                maxConnections: 100  
            }  
        },  
        app: {  
            anonymousAccessAllowed: true,  
            anonymousUserRole: "user",  
            supportsCustomConnections: false,  
  
            resourceQuotas: {  
                dataExportFileSizeLimit: 10000000,  
                sqlMaxRunningQueries: 3,  
                sqlResultSetRowsLimit: 100000,  
                sqlResultSetMemoryLimit: 2000000  
            },  
            plugins: {}  
        }  
    }  
}
```

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 to all interfaces.
serverURL	Server address (full URL). Used to generate links and for third-party services integration.

Name	Description
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 GraphQL console is enabled on the server.

## Database configuration

Configures CloudBeaver database where it keeps users, credentials and permissions.

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

## Application parameters:

In the section `app` :

Name	Description
anonymousAccessEnabled	Allows anonymous access. Anonymous users work with the role, 'User'.
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	When there is only one federation authentication configuration then redirect to it automatically

## 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)
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

# Automatic server configuration

On the first start CloudBeaver server will show administrator interface for server configuration.  
In some cases the server must be configured automatically (e.g. when it is run in 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	admin
CB_ADMIN_PASSWORD	Administrator user password	S0mePazzworD

These parameters can be specified in:

- OS environment variables
- configuration file `cloudbeaver.auto.conf` which must be placed in the same location as the `cloudbeaver.conf` file.

## Datasources configuration

---

You can find a detailed description [here](#)

# Configuring server datasources

## 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

## 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.

# CloudBeaver and Nginx

## 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;  
}
```

# Product configuration parameters

Some variables can be configured via product.conf (server), or via config.json5 (for webapp).

The list of configurable variables are:

Variable	Value	Description
core.user.defaultTheme	light	Default theme (light or dark)
core.user.defaultLanguage	en	Default language (language code)
core.app.logViewer.refreshTimeout	3000	Log viewer refresh interval in ms
core.app.logViewer.maxLogRecords	2000	Maximum rows in log viewer
core.app.logViewer.maxFailedRequests	3	Count failed log viewer request before disabling
core.app.navigationTree.childrenLimit	500	Maximum children in navigation tree
core_events.notificationsPool	5	Maximum notifications
plugin_erd_viewer.maxColumnsToDisplay	7500	Maximum columns in ERD
plugin_data_export.disabled	true	Disable data export functionality
core.app.sqlEditor.maxFileSize	100	Max size of sql script that can be uploaded (KB)
core.app.metadata.deleting	true	Allow deleting metadata objects
core.app.metadata.editing	true	Allow editing metadata objects
core.authentication.primaryAuthProvider	local	Primary auth provider

You can also specify these parameters in the config file `/workspace/.data/.product.runtime.conf`. These properties have a higher priority than `product.conf`.

For example, if you want to disable the data export functionality and increase refresh timeout for the [Log Viewer](#), you can do it this way.

1. Open or create `.product.runtime.conf` in folder `.data`
2. Paste the following code

```
"refreshTimeout": 7000
} }, "plugin_data_export": { "disabled": true } }
```

# Command line parameters

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/cloudbeaver.conf -nl en -registryMultiLang  
usage
```

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` }
```

# Overview

## 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/](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](#).

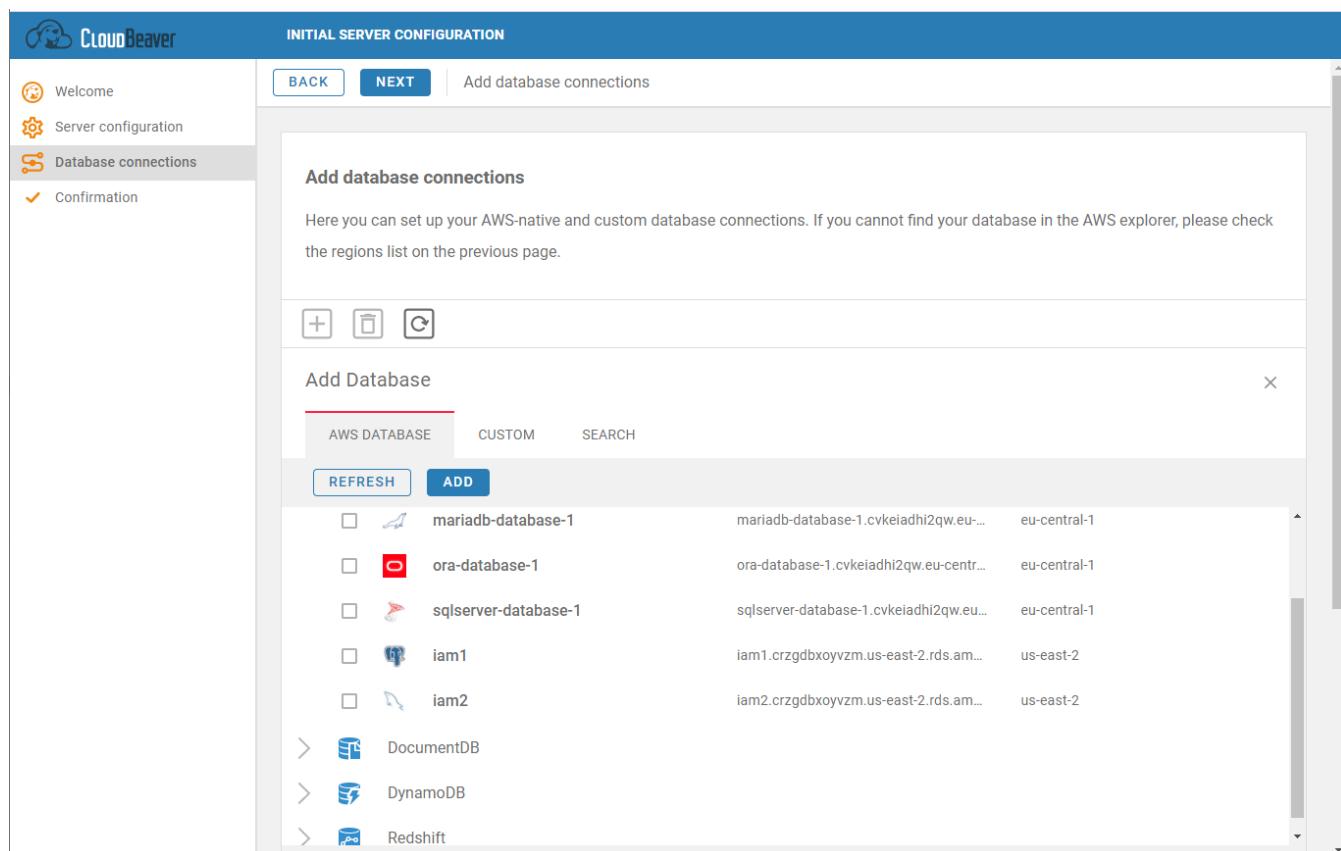
You can connect to your EC2 instance using SSH. You need to specify the SSH keypair during the EC2 instance launch. You can use the user name `ubuntu` to connect.

## Docker image

Not yet supported. We are working on the new docker image for the AWS version of CloudBeaver.

## Cloud explorer

You can use an embedded [Cloud explorer](#) in order to find and add existing AWS databases:



The screenshot shows the 'INITIAL SERVER CONFIGURATION' screen of CloudBeaver. On the left, a sidebar lists steps: Welcome, Server configuration (selected), Database connections, and Confirmation. The main area has a 'BACK' button, a 'NEXT' button, and a link to 'Add database connections'. A sub-section titled 'Add database connections' explains how to set up AWS-native and custom database connections. Below it is a 'Add Database' dialog with tabs for 'AWS DATABASE', 'CUSTOM', and 'SEARCH'. The 'AWS DATABASE' tab is active, displaying a list of databases with checkboxes and details like region. The list includes:

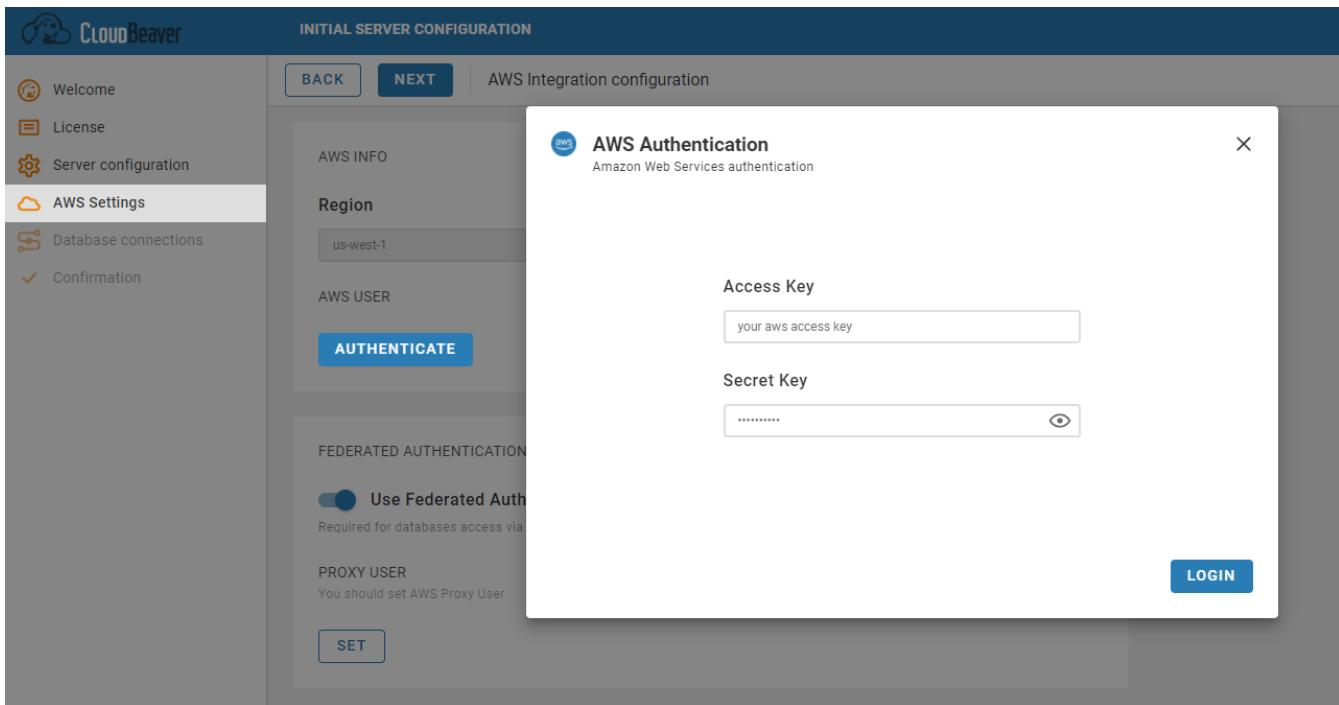
Database	Region
mariadb-database-1	eu-central-1
ora-database-1	eu-central-1
sqlserver-database-1	eu-central-1
iam1	us-east-2
iam2	us-east-2
DocumentDB	
DynamoDB	
Redshift	

# AWS Settings

## Authentication

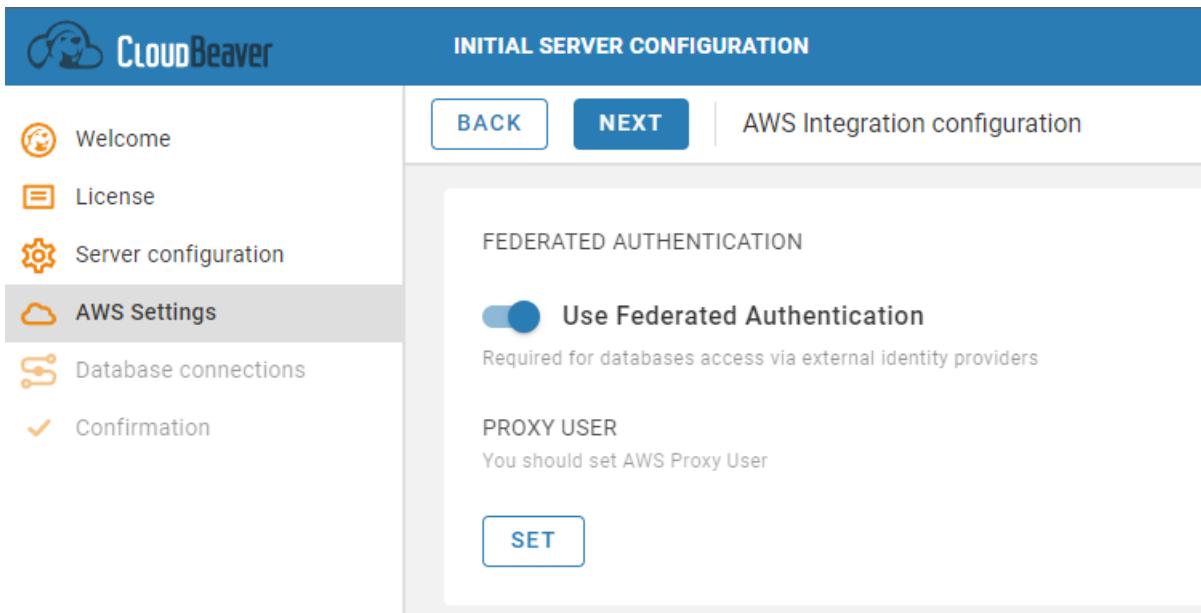
To get access to the AWS Services from the CloudBeaver you need to authenticate to your AWS Account. All CloudBeaver AWS settings will depend on this AWS account permissions. You will not be able to change entered AWS credentials after the end of configuration process. Also, these credentials will be 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](#).



## 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](#).



## 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 (Mumbai) (ap-south-1) Asia Pacific (Osaka) (ap-northeast-3) Asia Pacific (Seoul) (ap-northeast-2) Asia Pacific (Singapore) (ap-southeast-1) Asia Pacific (Sydney) (ap-southeast-2) Asia Pacific (Tokyo) (ap-northeast-1) China (Beijing) (cn-north-1) China (Ningxia) (cn-northwest-1) Middle East (Bahrain) (me-south-1)

## EUROPE

 Europe (Frankfurt) (eu-central-1) Europe (Ireland) (eu-west-1)

# Cloud Explorer

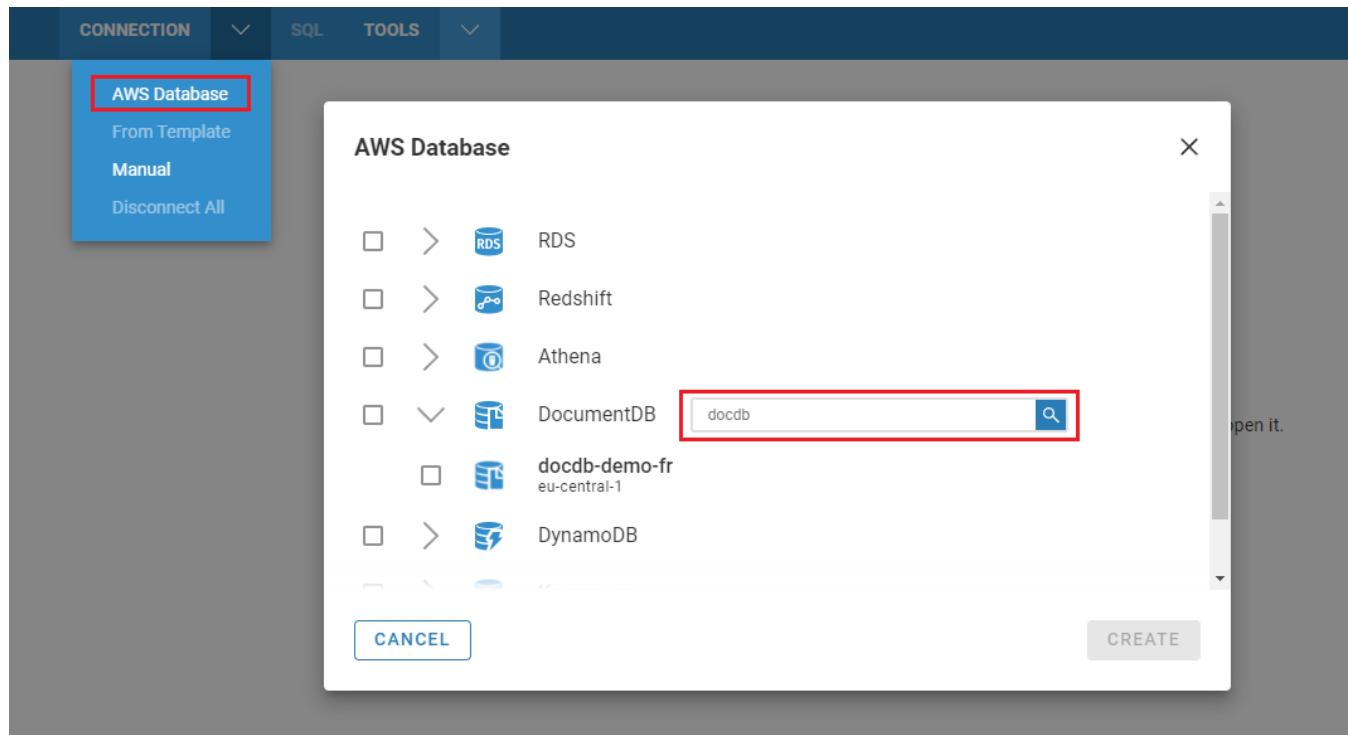
Once you configure the [AWS Settings](#) you can open the Cloud Explorer and start adding AWS database connections. The principle is similar to the usual connections that means you can add connection from the [Administration part](#) and from the Public part of the app.

The screenshot shows the CloudBeaver Cloud Explorer interface. The left sidebar has a 'Connection Management' tab selected, showing options like Users, Roles, Server configuration, AWS Settings, Identity Providers, and License. The main area is titled 'Add Database' with tabs for AWS DATABASE, CUSTOM, and SEARCH. Below is a 'REFRESH' button and an 'ADD' button. A hierarchical tree view lists databases grouped by service type: RDS, Redshift, Athena, DocumentDB, DynamoDB, and Keyspaces. Each item has a checkbox icon and a right-pointing arrow icon.

Service Type	Database Name
RDS	RDS
Redshift	Redshift
Athena	Athena
DocumentDB	DocumentDB
DynamoDB	DynamoDB
Keyspaces	Keyspaces

In the center of the Cloud Explorer you can see AWS databases in a hierarchical view. All databases are grouped by database/service type. When you expand one of the top elements, CloudBeaver will start to search AWS databases considering the regions you have selected in the [AWS Settings](#) section.

If you have a large number of databases in your cloud, you can search or filter them using filter text above the every top element You can also check any number of databases in the Cloud Explorer using the checkbox control on the left side of the Cloud Explorer tree.



# Authentication / IAM

## AWS Authentication

---

CloudBeaver Enterprise for AWS requires AWS IAM authentication to work with databases. You must enter a valid Access Key and Secret Key in order to login.

CloudBeaver Enterprise for AWS does not keep your access/secret keys on the server-side. They are not saved in a database or in configuration files.

Once your session expires, you will need to authenticate again. Authentication is always performed through remote AWS services.

## Server configuration

---

When you open CloudBeaver EE in AWS for the first time you must enter your access/secret keys.

The AWS user who configures CloudBeaver will become an administrator in this CloudBeaver EE instance (this user will have administrator permissions).

After the server configuration finishes the current AWS account (the account to which administrator belongs), it will be associated with this CloudBeaver EE instance. Only AWS users from this account can authenticate in this CloudBeaver EE instance.

You cannot create new users in CloudBeaver EE for AWS as it only works with real AWS users.

The Administrator may grant different roles (including Administrator role) to the other AWS users in this account after they authenticate in this CloudBeaver instance.

## IAM permissions

---

CloudBeaver EE 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 EE 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.

## Database authentication

---

As most AWS databases have their own authorization system (excluding DynamoDB) each database may require additional authentication parameters.

It is usually a username/password pair.

If you use IAM authentication for RDS/Aurora databases, then only the database username may be required and you can leave the password field empty.

# Authentication / AWS SSO

## 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:

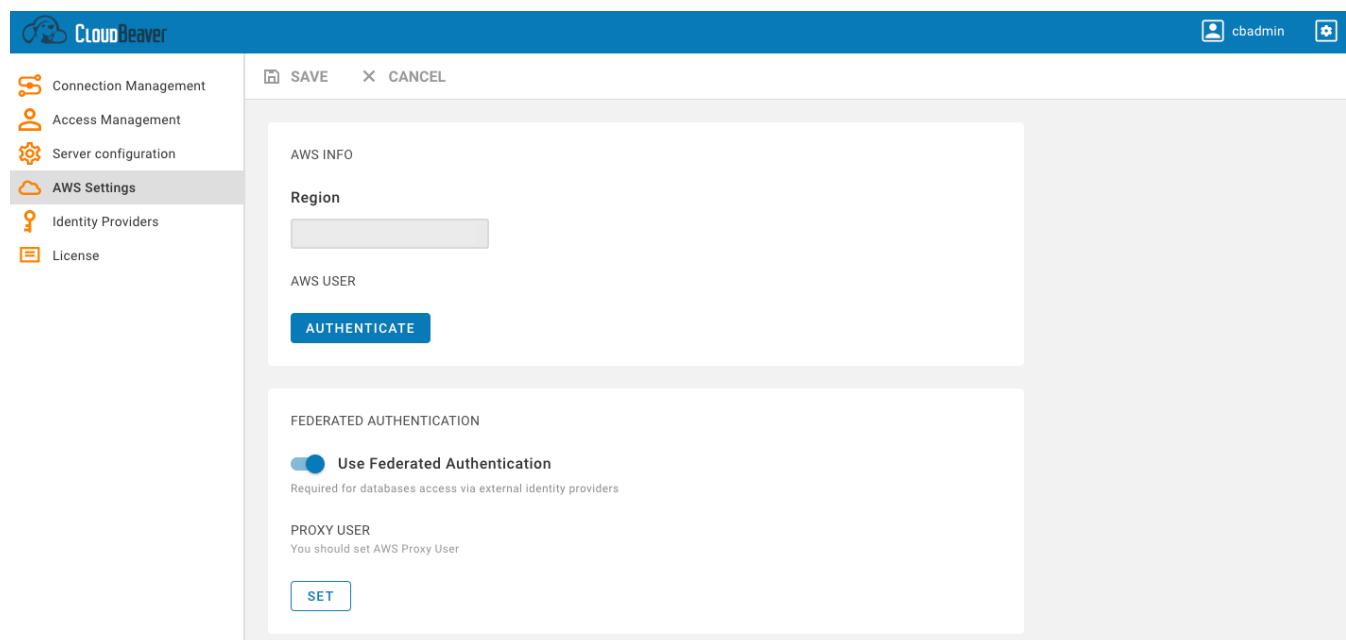
- log in to the application by users who have been given rights.
- get access to databases according to users' roles.

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: [SAML Authentication](#)

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.

### 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.

# Overview

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

- Microsoft SQL Server
- Sybase, SAP ASE
- Snowflake
- Vertica
- Netezza
- Hive

- NoSQL databases

- MongoDB
- Cassandra
- InfluxDB
- Couchbase

- AWS databases

- RDS/Aurora
- Athena
- Redshift
- DynamoDB
- DocumentDB
- Keyspaces

# SSO

## 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:

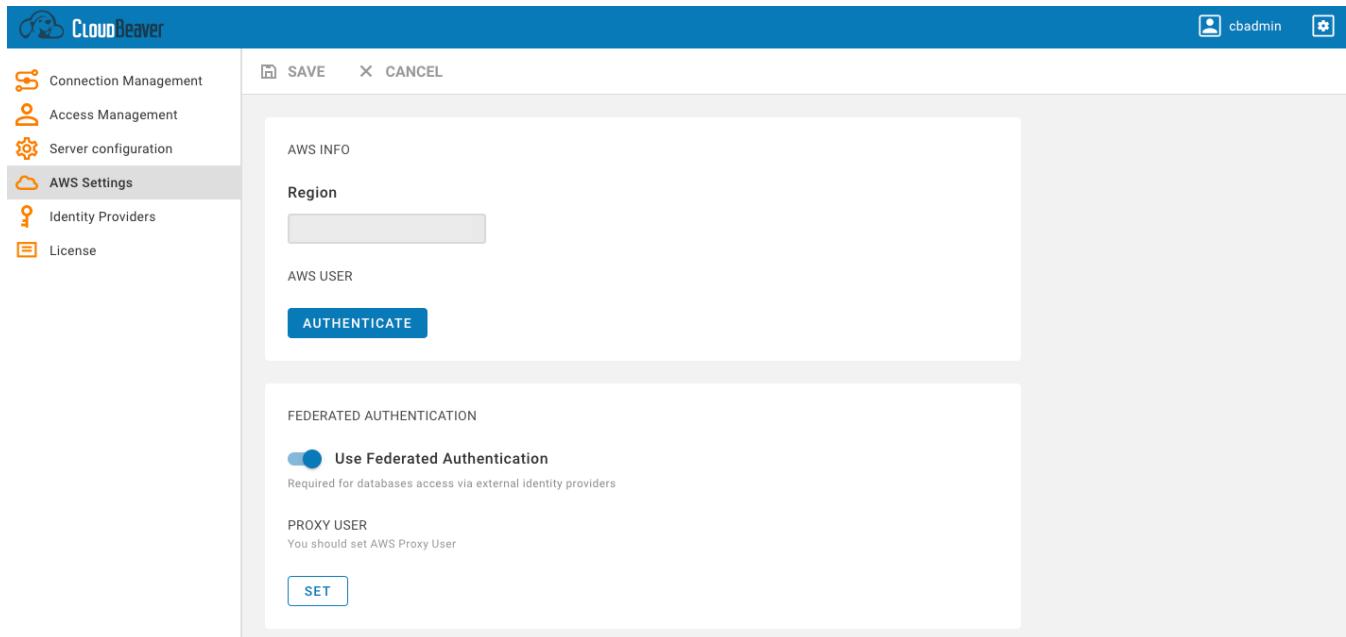
- log in to the application by users who have been given rights.
- get access to databases according to users' roles.

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: [SAML Authentication](#)

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.

### 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 authentication

## 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 administration interface. On the left sidebar, the 'Server configuration' option is selected. In the main panel, under 'Server URL \*' and 'Session lifetime \*', there are input fields. To the right, under 'Navigator simple view', a toggle switch is off. Under 'SERVICES', the 'AWS' toggle switch is on. In the 'AUTHENTICATION SETTINGS' section, several options are listed: 'Allow anonymous access' (off), 'Local' (on), 'AWS IAM' (off), 'OpenId' (on), and 'SAML' (on). Under 'SECURITY', two options are available: '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 'Configuration creation' dialog for 'Identity Providers'. The 'OPTIONS' tab is selected. In the 'SAML' section, the 'Provider \*' dropdown is set to 'SAML' and the 'ID \*' field is empty. The 'Configuration name \*' field is empty. The 'Description' and 'Icon URL' fields are also empty. A 'Disabled' checkbox is unchecked. To the right, there are four empty input fields for 'IDP Entity ID', 'IDP signon URL', 'IDP logout URL', and 'IDP x509 certificate'.

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.

The screenshot shows the CloudBeaver interface for managing identity providers. On the left, there's a sidebar with options like Connection Management, Access Management, Server configuration, AWS Settings, Identity Providers (which is selected and highlighted in blue), and License. The main area has tabs for ADD, REFRESH, and DELETE. Below these are sections for OPTIONS and SAML. In the SAML section, there are fields for IDP Entity ID, IDP signon URL, IDP logout URL, and IDP x509 certificate. To the right, under the 'LINKS' tab, there are 'Sign in' and 'Sign out' URLs. A prominent red double-headed arrow points between the 'Download metadata file' link in the 'LINKS' section and the 'Metadata' table below.

Attribute	Value	Meaning
Metadata	<a href="https://test.ee.cloudbeaver.io/api/saml/test/metadata">https://test.ee.cloudbeaver.io/api/saml/test/metadata</a>	The URL where the SAML metadata file is located.

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	<a href="https://HOST_NAME/api/saml/CONFIG_ID/acs">https://HOST_NAME/api/saml/CONFIG_ID/acs</a>
Application SAML audience	<a href="https://HOST_NAME/api/saml/CONFIG_ID/metadata">https://HOST_NAME/api/saml/CONFIG_ID/metadata</a>

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
Metadata	<a href="https://test.ee.cloudbeaver.io/api/saml/test/metadata">https://test.ee.cloudbeaver.io/api/saml/test/metadata</a>	The URL where the SAML metadata file is located.

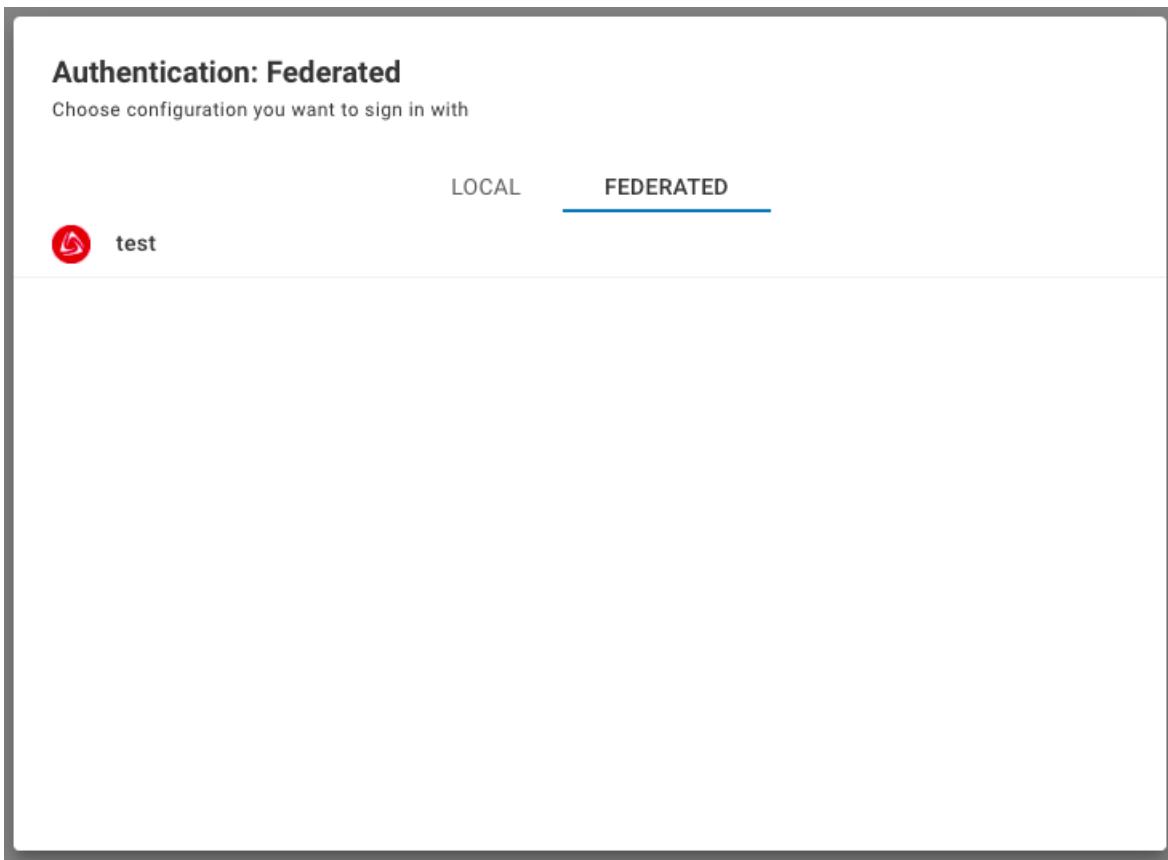
Attribute	Value	Meaning
Subject	#{user:email}	User unique identifier (nameId). It is usually an email address.
<a href="https://aws.amazon.com/SAML/Attributes/SessionDuration">https://aws.amazon.com/SAML/Attributes/SessionDuration</a>	1800	Session duration in seconds. 1800 (30 minutes) is the default value
<a href="https://aws.amazon.com/SAML/Attributes/Role">https://aws.amazon.com/SAML/Attributes/Role</a>	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](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 authentication

## OpenID configuration

### Enabling OpenID authentication

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

The screenshot shows the CloudBeaver administration interface. The left sidebar has tabs: Connection Management, Access Management, Server configuration (selected), AWS Settings, Identity Providers (selected), and License. The main area has tabs: SAVE and CANCEL. It shows fields for Server URL \* and Session lifetime \*. On the right, there are sections for Navigator simple view (disabled), SERVICES (AWS is selected), AUTHENTICATION SETTINGS (Allow anonymous access is off, Local is selected, AWS IAM and OpenId are off, SAML is off), and SECURITY (Save credentials and Save users credentials are selected). A note says 'By default, all user's new connections will contain only basic information in navigation tree'.

### Configuring external identity provider

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

The screenshot shows the CloudBeaver Identity Providers configuration creation dialog. The left sidebar has tabs: Connection Management, Access Management, Server configuration, Identity Providers (selected), License, and Version update. The main area has tabs: + ADD, REFRESH, DELETE. It shows a 'Configuration creation' dialog with an OPTIONS tab. Fields include Provider \* (OpenId selected), ID \*, Configuration name \*, Description, Icon URL, and a Disabled checkbox. To the right is an OPENID group with Client ID, Client Secret, IDP auth endpoint URL, and IDP token endpoint URL fields. Buttons at the bottom are CANCEL and CREATE.

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

Configuration in OpenID	Configuration in CloudBeaver
	Client ID
	Client Secret

Configuration in OpenID	Configuration in CloudBeaver
	IDP auth endpoint URL
	IDP token endpoint URL

### Testing OpenID 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 SSO.

The screenshot shows the 'Authentication' dialog in CloudBeaver. At the top, it says 'Authentication: Federated'. Below that, it says 'Choose configuration you want to sign in with'. There are two tabs: 'LOCAL' and 'FEDERATED', with 'FEDERATED' being underlined. Under the 'FEDERATED' tab, there is a single entry: 'test' with a small icon next to it. The rest of the screen is mostly blank white space.

# Deployment

The CloudBeaver EE container image is at DockerHub:<https://hub.docker.com/r/dbeaver/cloudbeaver-ee>.

- `dbeaver/cloudbeaver-ee:latest` - latest release build.
- `dbeaver/cloudbeaver:x.y.z` - particular product version.

It is a public repository so you can just pull it.

## Installation

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

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

## Running

To run cloudbeaver in terminal:

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

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

To run the server in 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 '{ print $2 }' | cut -f2 -d: | head -n1)
docker run --name cloudbeaver-ee -d --restart unless-stopped -ti -p 8080:8978 --add-host=host.docker.internal:${CB_LOCAL_HOST_ADDR} -v /var/cloudbeaver-ee/workspace:/opt/cloudbeaver/workspace dbeaver/cloudbeaver-ee:latest
```

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

## Docker parameters explanation

Parameters explanation:

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

Parameter	Explanation
dbeaver/cloudbeaver-ee:latest	Container ID