



CloudBeaver Documentation v.25.3

Table of contents

- **Getting started**
 - First steps
 - Application window overview
 - Basic operations
 - Demo server
- **Configure connection**
 - Create connection
 - Network configuration settings
 - SSH configuration
 - SSL configuration
 - Database authentication methods
 - Database Native authentication
 - MongoDB authentication
 - Microsoft Entra ID authentication
 - Oracle wallet authentication
 - Pgpss authentication
 - Salesforce authentication
 - Kerberos authentication
- **Databases support**
 - Supported databases
 - File drivers
 - Multi Source
 - CSV
 - JSON
 - Parquet
 - XLSX
 - XML
 - File based driver properties
 - Drivers management
- **Navigation**
 - Database Navigator
 - Toolbar
 - Settings menu
 - Folders

- Simple and Advanced view
- Properties Editor
- Data Editor
 - Data Editor
 - Data filters
 - Data ordering
 - Panels
 - Value Panel
 - Grouping Panel
 - Result details Panel
 - Managing charts
 - JSON and Document view
 - Spatial GIS data
- SQL Editor
 - SQL Editor
 - SQL formatting
 - SQL assist and auto complete
 - Query Execution Plan
 - Visual Query Builder
 - Client-side commands
 - Auto and Manual commit modes
 - Resource Manager
 - Server Output
- Entity relation diagrams
- Cloud services
 - Cloud Explorer
 - Cloud Storage
 - AWS settings
 - Azure permissions
- AI
 - AI Assistant
 - Data privacy
 - Features
 - AI Chat
 - Work with SQL
 - Explain query

- [Fix SQL errors](#)
 - [AI command](#)
 - [Work with objects](#)
 - [Describe object with AI](#)
- [Tutorials](#)
 - [Quick start with AI](#)
- [Settings](#)
 - [AI Assistant settings](#)
 - [Providers](#)
 - [OpenAI](#)
 - [GitHub Copilot](#)
 - [Azure OpenAI](#)
 - [Google Gemini](#)
 - [Ollama](#)
 - [Anthropic Claude](#)
 - [Amazon Bedrock](#)
- [Data transfer](#)
 - [Data export](#)
 - [Data import](#)
- [General user guide](#)
 - [Database Object Editor](#)
 - [Managing preferences](#)
 - [Managing preferences](#)
 - [Preferences](#)
 - [Settings](#)
 - [Accessibility guide](#)
 - [Keyboard shortcuts](#)
 - [Search](#)
- [Administration](#)
 - [Server configuration](#)
 - [Server configuration](#)
 - [Web server configuration](#)
 - [How to start](#)
 - [Server architecture](#)
 - [Server database](#)
 - [Domain manager](#)
 - [Product configuration parameters](#)

- Product configuration parameters
- Authentication settings
- Navigation tree settings
- Data spreadsheet settings
- Data export settings
- ERD settings
- Connection settings
- Query manager database configuration
- Security and access configuration
 - Server security and access configuration
 - Authentication
 - Authentication methods
 - Anonymous access
 - Local access authentication
 - Reverse proxy header authentication
 - LDAP authentication
 - Single sign-on (SSO)
 - SAML authentication
 - OpenID authentication
 - AWS OpenID authentication
 - AWS SAML database access via Okta
 - AWS IAM authentication
 - AWS OpenID database access via Okta
 - Snowflake OpenID database access via Okta
 - Okta OpenID authentication
 - Cognito OpenID authentication
 - JWT authentication
 - NTLM user authentication
 - Microsoft Entra ID authentication
 - Google authentication
 - Access management
 - Users
 - Users
 - Provisioning users
 - Password policy
 - User credentials storage
 - Teams

- [Admin password recovery](#)
 - [Initial data configuration](#)
 - [Global Permissions](#)
- [Proxy configuration](#)
- [WebSockets](#)
- [Multiple Server URLs](#)
- [Secret management](#)
- [Pre-configured database connections](#)
 - [Pre-configured database connections](#)
 - [Pre-configured permissions for connections](#)
- [Logs](#)
 - [Log viewer](#)
 - [Log customization](#)
- [Query manager](#)
 - [Query manager](#)
 - [Query history](#)
 - [Query manager database](#)
- [Workspace location](#)
- [Workspace migration guide](#)
- [Command line parameters](#)
- [Session manager](#)
- [Deployment](#)
 - [Deployment options](#)
 - [CloudBeaver Enterprise](#)
 - [Docker compose](#)
 - [AWS AMI](#)
 - [Google Cloud](#)
 - [Microsoft Azure](#)
 - [CloudBeaver AWS](#)
 - [AWS Marketplace](#)
 - [AWS AMI](#)
 - [Additional setup and management](#)
 - [Version update](#)
 - [Workspace backup](#)
 - [SSL certificate configuration](#)
 - [How to change Java security properties](#)
 - [How to connect CloudBeaver to a database on a separate machine in Azure](#)

- [Configuring HTTPS for Jetty server](#)
 - [Managing cloud deployed services](#)
- [API and Integration](#)
 - [API and Integration overview](#)
 - [GraphQL API](#)
 - [GraphQL API overview](#)
 - [Generate API token](#)
 - [GraphQL console](#)
 - [Run GraphQL requests from CLI](#)
 - [DBeaver Proxy driver](#)
 - [Proxy Driver](#)
 - [Connect Tableau to a database using DBeaver Proxy Driver](#)
- [Licenses](#)
 - [Import license](#)
 - [Update license](#)
 - [Reassign license](#)
 - [License types](#)
 - [License types](#)
 - [Multi server license](#)
- [About CloudBeaver](#)
 - [Editions](#)
 - [Enterprise edition](#)
 - [AWS edition](#)
 - [Release cycles](#)
 - [Customer technical support](#)
- [FAQ](#)

Getting started

First steps

This guide helps you quickly deploy CloudBeaver. You'll launch the server, open the app in your browser, and prepare it for database connections.

Deploy CloudBeaver

The easiest way to run CloudBeaver is with Docker Compose:

Tip

You can also deploy CloudBeaver on Kubernetes, or in the cloud. See [CloudBeaver deployment](#) for more options.

1. Clone the [deployment repo](#) and navigate to the repository:

```
git clone https://github.com/dbeaver/cloudbeaver-deploy
cd ~/cloudbeaver-deploy
```

2. Copy the example file to create your `.env` file:

```
cp .env.example .env
```

Then open the file and make sure to change the default database password:

```
nano .env
```

Tip

This file controls startup settings like ports, database config, and proxy type. See [environment file variables](#) and [server configuration](#).

3. Start the server:

```
docker-compose up -d
```

4. Open `http://<server-ip>:8978` in your browser to launch the easy config wizard.

- a. Follow the steps in the [easy config wizard](#).
- b. If you're using a paid version, you'll need to provide your license key. [Learn more](#)

Need custom networking, HTTPS, or a proxy?

See [Proxy configuration](#) and [Domain Manager](#).

5. Once setup is complete, continue with **Basic operations**.

Using the Community Edition?

See the [CloudBeaver Community deployment instructions](#) in the GitHub wiki.

Create a new connection

1. Click the **+** button in the top menu and select **New Connection**.
2. Choose a driver.
3. Fill in connection details.

Info

See [configuring connection settings](#) for help.

4. Click **Test** to check the connection (optional).
5. Click **Create**.

Resources for database operations

- **Interface overview**
Overview of CloudBeaver's web interface.
- **Database Navigator**
Navigation through database schemas and objects.
- **Database objects**
Managing database objects.
- **Data Editor**
Data management and editing.
- **Run queries**
Running SQL queries and scripts.

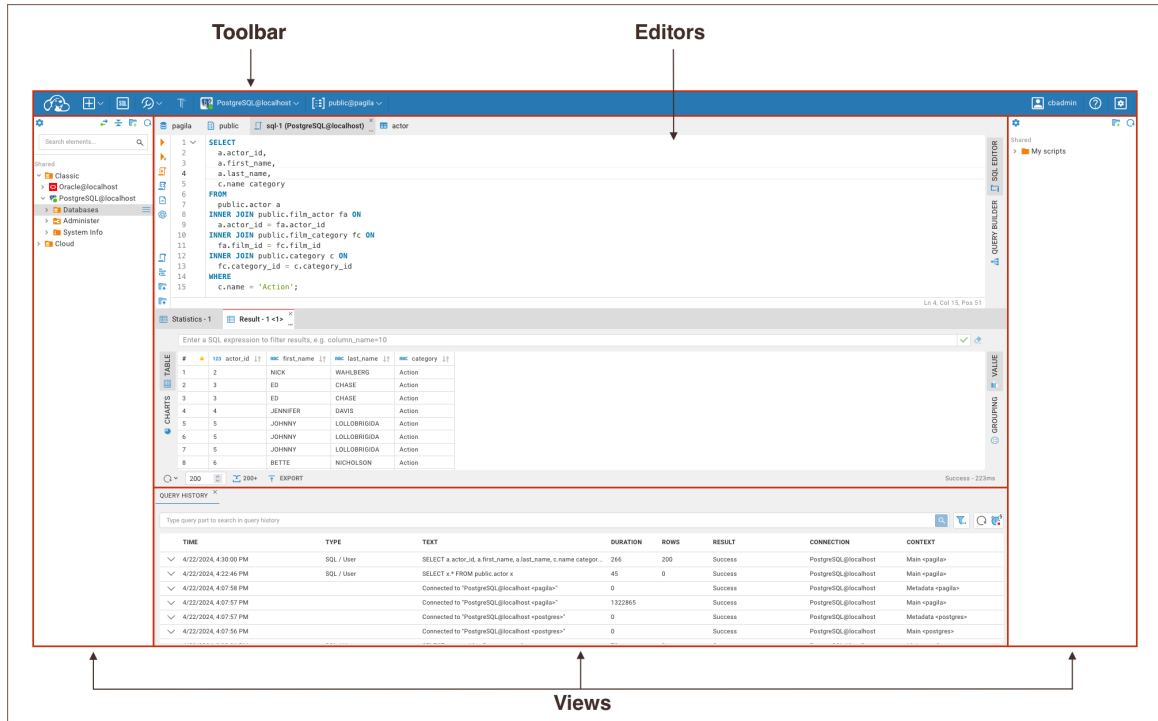
- **Database diagrams**
Database structure visualization.
- **Export data**
Export tables, query results in different formats.
- **Import data**
Import data into databases from files in different formats.
- **Use AI**
AI features for database tasks.

Additional resources

- **DBeaver Blog**
News and updates from the CloudBeaver team.
- **Twitter updates**
News and updates on Twitter.
- **Edition comparison**
Comparison of DBeaver editions.

Application window overview

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



Tip

For a practical demonstration of CloudBeaver's capabilities, please refer to our demo server. For more information, see [Demo Server](#).

Toolbar

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

- The **Connection** menu contains buttons for **creating new connections**, finding local databases, and opening **Cloud Explorer**.
- The **SQL Editor** button is for opening the **SQL Editor**.
- The **Tools** menu provides options to open the **Cloud Storage** view, **AI chat**, Scripts viewer, Query history, and Log viewer and Datasets viewer.
- The menu for switching between **Auto and Manual commit mode**.

- The **Database** menu allows for the management of database drivers and connections.
- The **Profile** menu contains user info, **preferences**, and capability for password management.
- The **Shortcuts** button contains shortcuts for CloudBeaver usage.
- The **Settings** menu provides options to open an **Administration menu**, view Product information, open **Preferences**, and Logout.

Note

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

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

Views

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

- **Database Navigator**: Provides a tree structure view of all the databases you have access to, allowing you to navigate through schemas, tables, and other database objects.
- **Query History**: Records the queries you have executed and allows you to revisit and analyze them.
- **Scripts viewer**: Manages and organizes your SQL scripts.
- **Cloud Storage**: Enables you to interact with files stored in cloud services.
- **Log viewer**: Displays the CloudBeaver logs. Except for the Database Navigator, to access these views, click on the **Tools** button and choose the necessary **View** from the list.

Note

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

Editors

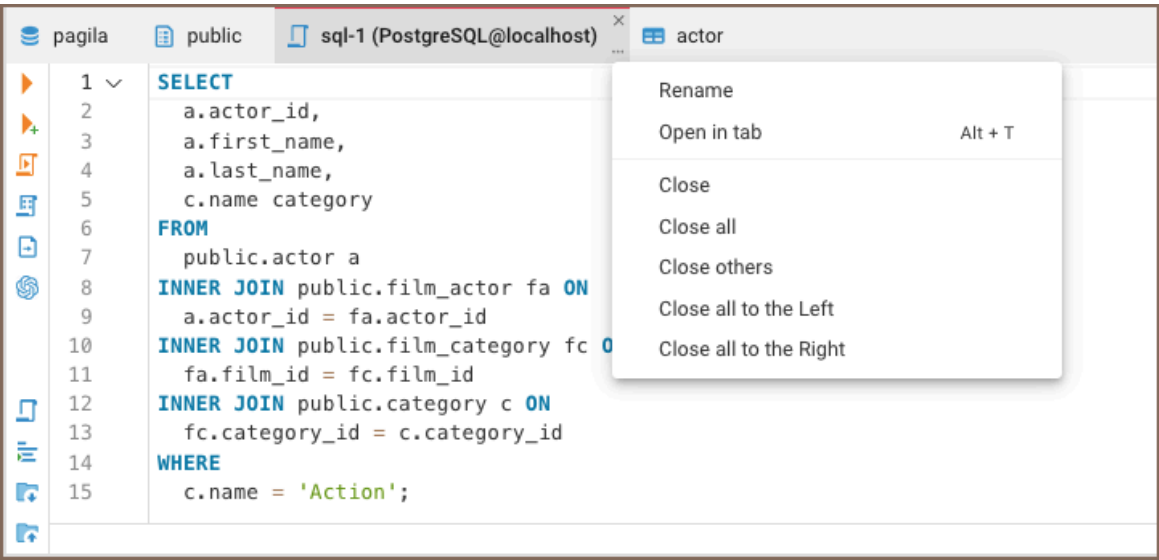
The **Editors** within the workspace are interactive windows that allow you to directly work with data and SQL queries.

- **Data Editor**: An interface for modifying and browsing table data.
- **SQL Editor**: An interface for writing, editing, and running SQL queries.

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

Managing Views and Editors

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



Button	Description
Rename	Rename the SQL Editor's tab.
Open in Tab	Opens the current editor in a separate tab.
Close	Close the current editor window or tab.
Close all	Close all open editor windows or tabs.
Close others	Close all editor windows or tabs except the currently active one.
Close all to the Left	Close all editor windows or tabs to the left of the current tab.

Button	Description
Close all to the Right	Close all editor windows or tabs to the right of the current tab.

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

CLOUD STORAGE

QUERY HISTORY

Type query part to search in query history

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

To do this:

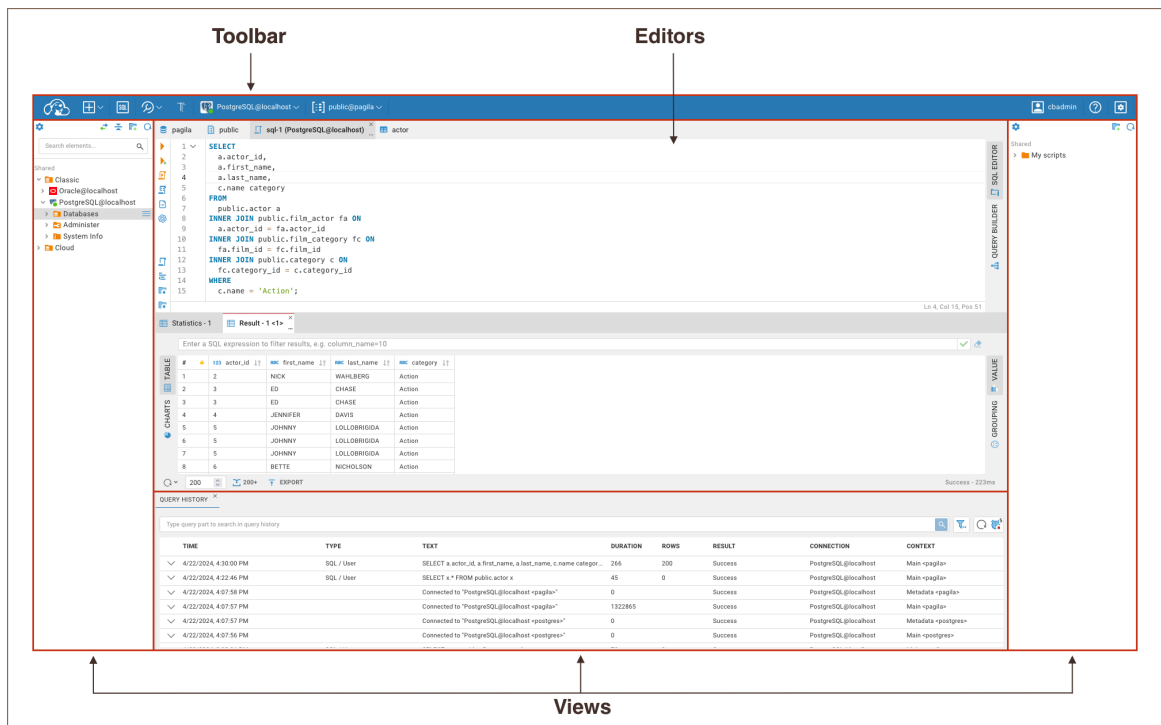
- **Resize:** Click and drag the borderline separating the **View** from the **Editor**. This allows you to customize the size of the **View** pane to your preference.
- **Maximize/Restore:** Use the maximize and restore buttons to toggle the View pane between the full workspace and its original size.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

Basic operations

This guide outlines the essential steps to quickly get started with CloudBeaver.

First access

When you open CloudBeaver in your browser for the first time, you'll land directly on the login page. After logging in, you can start working with your databases right away.

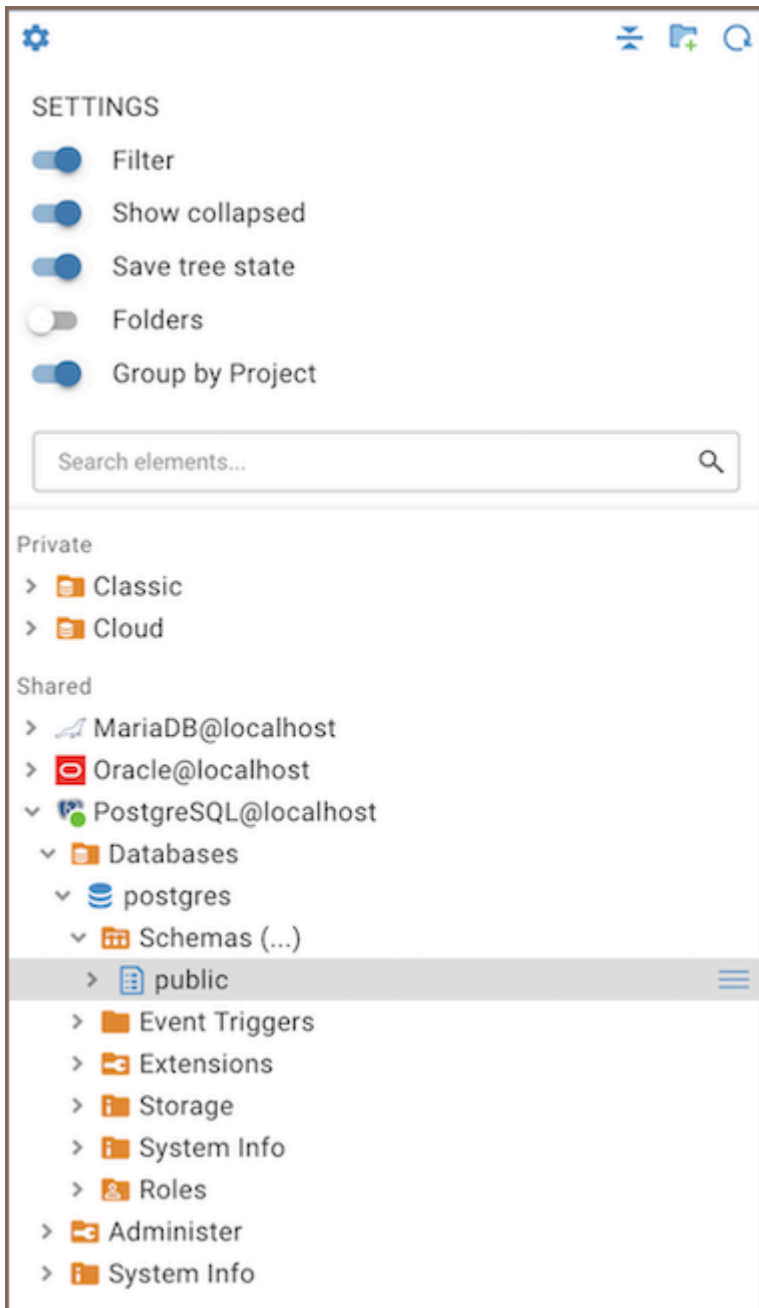


Tip

For a detailed understanding of CloudBeaver interface, refer to the [Application Window Overview](#) article.

Database navigator

After logging in, you'll see the **Database Navigator**. It's your main workspace for accessing and managing databases.

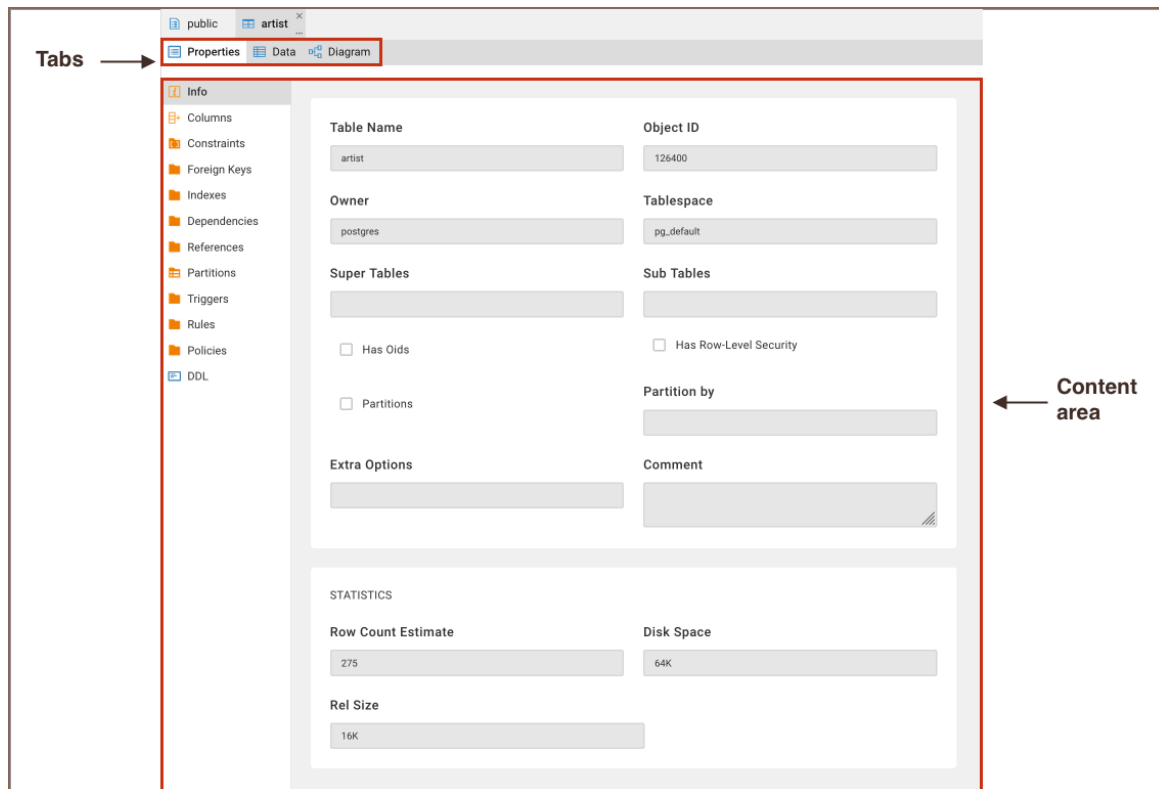


- To start, **create a connection** to a database.
- To explore the database objects, click on the arrow next to a database name to expand and see tables, views, and other metadata objects.

Object editor

From the Database Navigator, you can open the **Object Editor** to explore and manage database objects.

Double-click a table or other object to open it. The Object Editor provides detailed tabs for working with your data.



The Object Editor has three tabs to organize your work:

Tab	Description	Reference article
Properties	View and edit the properties of the database object.	Properties Editor
Data	Open the Data Editor to manage table and view data.	Data Editor
Diagram ★	Visualize table relationships with	Diagrams

Tab	Description	Reference article
	th Entity-Relationship Diagram S.	

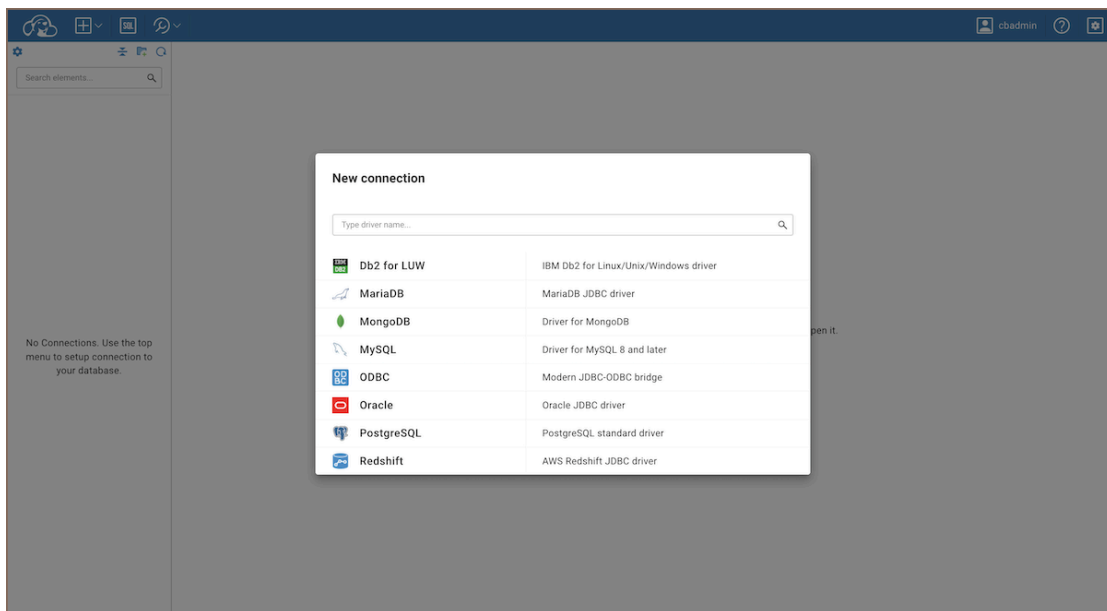
Basic operations

Now let's explore how to manage your data in CloudBeaver.

How to create the first connection

After logging in, create your first database connection:

1. Click **New Connection** on the welcome page.



2. Select your database type from the list.
3. Enter connection details:
 - Host.
 - Port.
 - Database name.
 - Database credentials.
4. Test the connection, then click **Finish**.

Tip

For a full list of supported databases, see [Supported Databases](#).

Object creation

Create new objects like tables, views, or schemas using SQL scripts.

1. In the **Database Navigator**, locate your database.
2. Click **Open SQL Editor** button in the top toolbar.

The screenshot displays the SQL Editor interface. At the top, the **Toolbar** contains icons for various database actions, and the **Script panel** shows a SQL query: `select * from countries;`. Below the script panel, the **Result panel** displays the query results in a table format. The table has columns for country details and a region ID. The results are filtered to show 15 rows. The interface also includes a **Query Builder** sidebar on the right and a **Result** tab at the bottom.

#	country_id	name	area	national_day	country_code2	country_code3	region_id
1	1	Aruba	193	[NULL]	AW	ABW	1
2	2	Afghanistan	652090	1919-08-19	AF	AFG	2
3	3	Angola	1246700	1975-11-11	AO	AGO	3
4	4	Anguilla	96	1967-05-30	AI	AIA	1
5	5	Albania	28748	1912-11-28	AL	ALB	4
6	6	Andorra	468	[NULL]	AD	AND	4
7	7	Netherlands Antilles	800	[NULL]	AN	ANT	1
8	8	United Arab Emirates	83600	1971-12-02	AE	ARE	5
9	9	Argentina	2780400	1816-07-09	AR	ARG	6
10	10	Armenia	29800	1991-09-21	AM	ARM	5
11	11	American Samoa	199	[NULL]	AS	ASM	7
12	12	Antarctica	13120000	[NULL]	AQ	ATA	8
13	13	French Southern territories	7780	[NULL]	TF	ATF	8
14	14	Antigua and Barbuda	442	1981-11-01	AG	ATG	1
15	15	Australia	7741220	[NULL]	AU	AUS	9

3. Write your SQL script:

Example

```
CREATE TABLE example_table (  
  id INT PRIMARY KEY,  
  name VARCHAR(100)  
);
```

4. Run the script:

- Click the **Execute SQL Statement** button  or press `Ctrl+Enter` to run it and create the object.

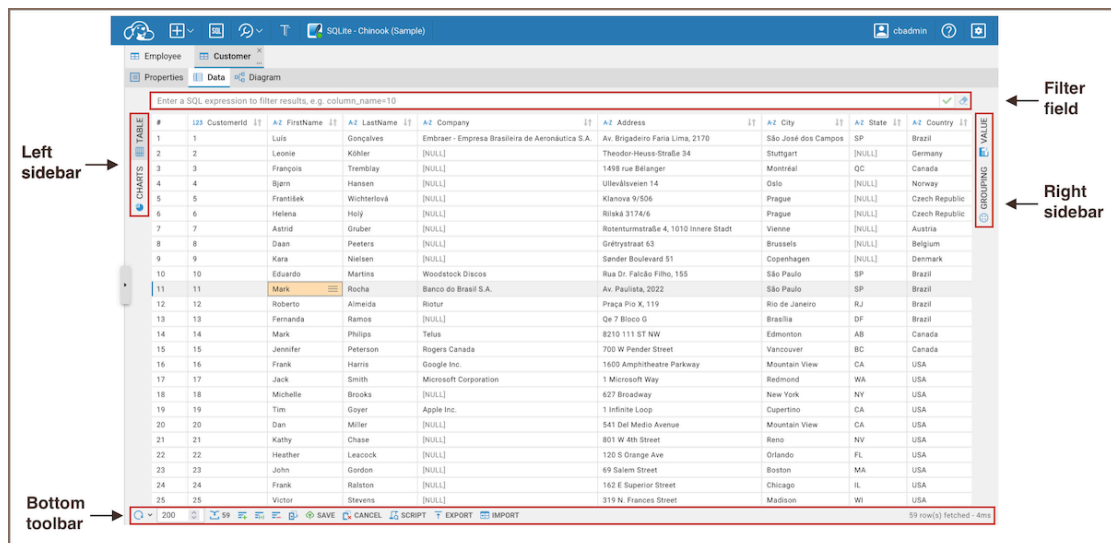
Tip

See more information on [SQL Editor](#).

Work with data

1. View and edit data:

- Double-click a table in the **Database Navigator**, or right-click it and select **Open**.
- Edit data directly in the grid and click **Save**.



The screenshot shows the SQL Editor interface with the 'Customer' table open. The data grid displays 25 rows of customer information. The left sidebar contains icons for 'TABLE', 'CHARTS', and 'VALUE'. The right sidebar contains a 'Filter field' and a 'Right sidebar' section. The bottom toolbar includes buttons for 'SAVE', 'CANCEL', 'SCRIPT', 'EXPORT', and 'IMPORT'. The status bar at the bottom indicates '59 row(s) fetched - 4ms'.

2. Insert new data:

- Click the **Add** button  to add new records.
- Fill out the row and click **Save**.

3. Delete data:

- Select rows and click the **Delete selected** button .
- Confirm and save changes.

Tip

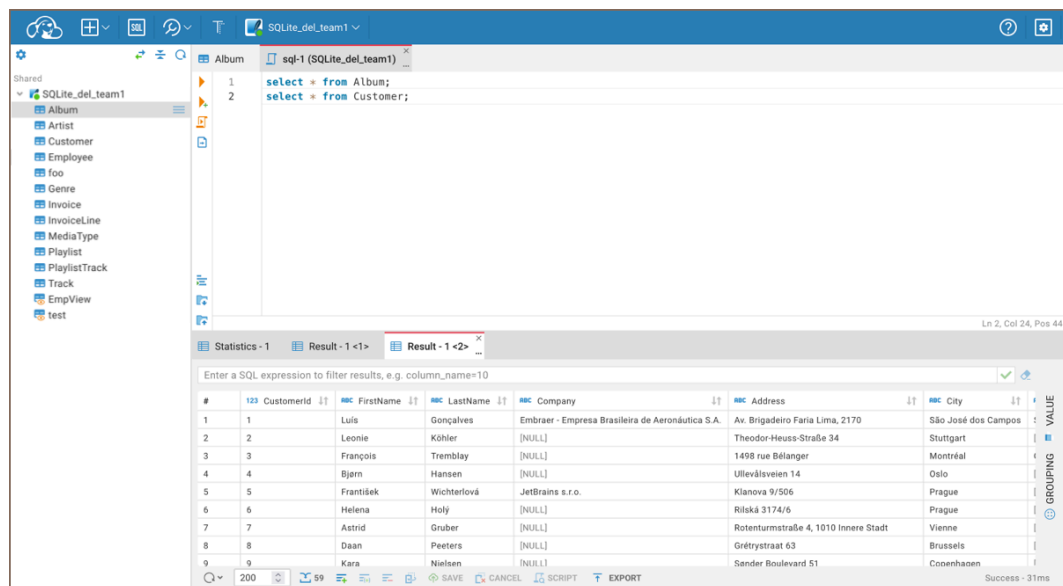
See more information on [Data Editor](#).

Execute SQL queries

1. In the **Database Navigator**, locate your database.
2. Click **Open SQL Editor** button in the top toolbar.
3. **Write your query:**
 - a. In the **SQL Editor**, type your SQL query.

Example

```
SELECT * FROM Album; .
```



Tip

Use **AI SQL Assistance** to improve your SQL scripts.

- b. Press **Ctrl+Enter** or right-click the query and click **Execute SQL Statement** button.
4. **View results:**
 - Results appear in the bottom panel of the **SQL Editor**.

Tip

Want to work faster? Check out the list of available keyboard shortcuts in [Shortcuts](#).

Demo server

CloudBeaver's demo server offers a comprehensive preview of the functionalities available in CloudBeaver Enterprise. This initial experience allows you to explore the software using pre-configured databases.

Demo server

Access the demo server at <https://demo.cloudbeaver.io>. The server comes pre-installed with the following databases:

- MariaDB
- MongoDB
- PostgreSQL
- SQLite

Custom Connections

The CloudBeaver Enterprise demo server doesn't support creating custom connections. To connect CloudBeaver to your databases, install CloudBeaver in your infrastructure. Deploy your own server instance to enable custom connection features using one of these methods:

- [Deploy from a Docker image](#)
- [Use AWS AMI deployment](#)
- [Deploy in Google Cloud](#)
- [Deploy on Microsoft Azure](#)

Note

An Enterprise license is required to deploy a custom server instance. For more details, see [CloudBeaver licensing](#).

Security

The demo server is not focused on security measures and contains no sensitive data. It operates in a completely isolated environment. If you identify any security vulnerabilities, please report them through our [GitHub issues page](#).

Configure connection

Create connection

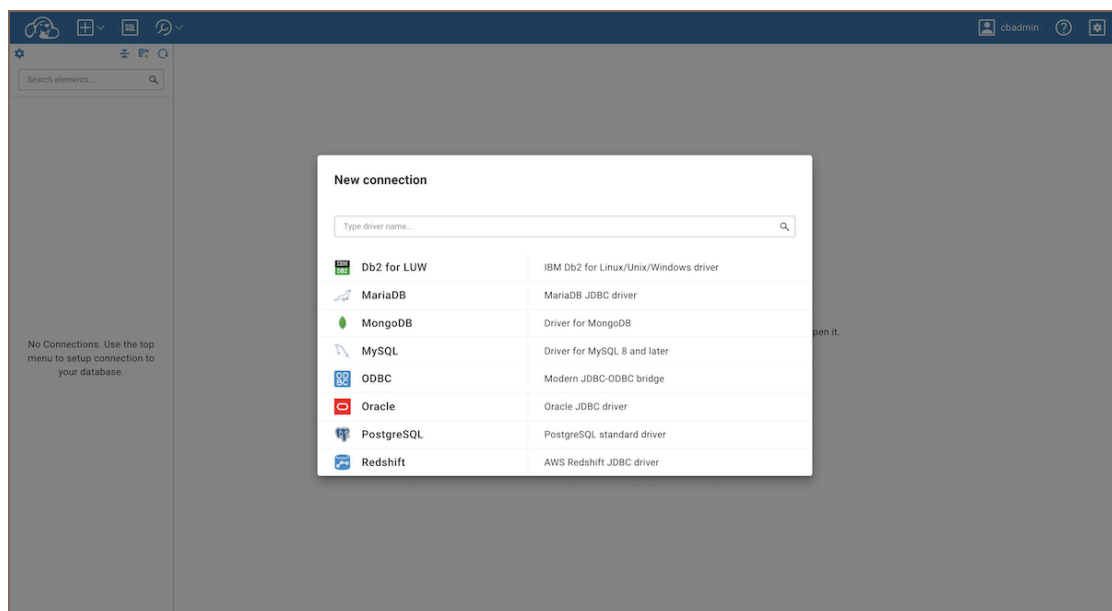
Creating a new database connection

CloudBeaver provides multiple methods to set up a connection. To create a connection, do one of the following:

Use the new connection wizard

To establish a new connection, follow these steps:

1. Initiate new connection:
 - Click the **+** button in the top menu.
 - Select **New Connection** from the dropdown options.



2. Select the appropriate driver from the available options.

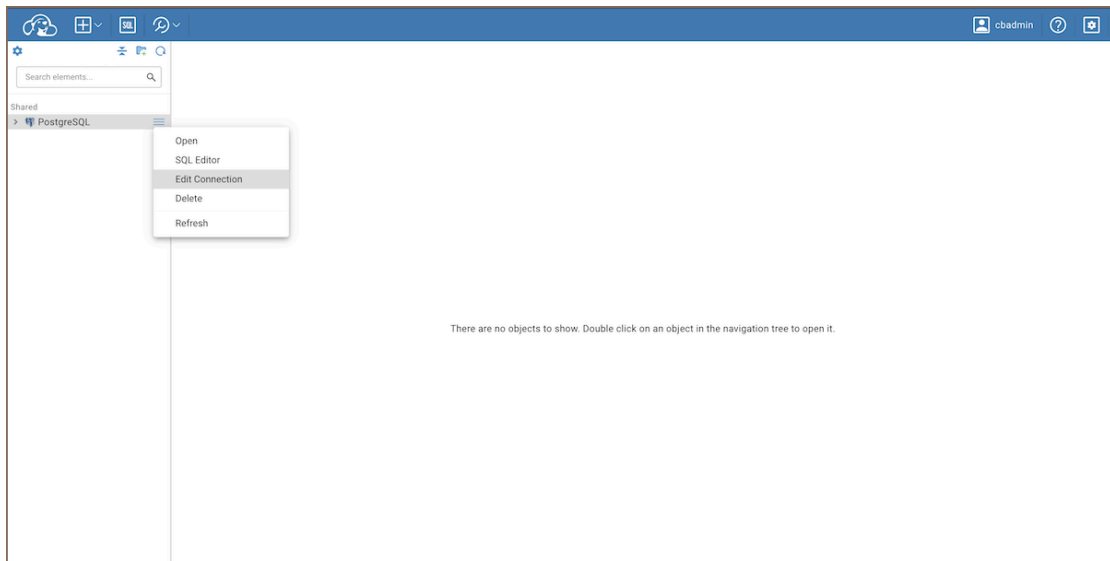
Note

If no driver is available for your database, you may need to [create a custom driver](#). By default, some drivers require [downloading](#). If a driver is missing, you'll see a warning message.

3. Enter all necessary details in the form fields provided. For detailed steps on filling out these fields, refer to the [section below](#).

The screenshot shows the 'Driver Properties' dialog box in DBeaver, specifically the 'Configuration' tab. The 'Driver' is set to 'PostgreSQL'. Under 'Configuration', 'Manual' is selected. The 'Host' is 'localhost', 'Port' is '5432', and 'Database' is 'postgres'. The 'Connection name' is 'PostgreSQL@localhost', 'Project' is 'Private', and there is a 'Description' field. On the right, the 'AUTHENTICATION' section shows 'Database Native' selected, with 'User name' and 'User password' fields. A checkbox for 'Save credentials for the current user' is present. Below this, the 'MISC' section shows 'User role' and expandable sections for 'DATABASE LIST', 'SQL', and 'EXPERT SETTINGS'. At the top right are 'CANCEL', 'TEST', and 'CREATE' buttons. The left sidebar shows 'No Connections. Use the top menu to setup connection to your database.'

4. Test the connection (optional):
 - Click the **Test** button to ensure the connection parameters are correct.
5. After confirming all details, click the **Create** button.
6. Access and modify connection:
 - The new connection will appear in the connection navigator menu.
 - To edit the connection details, click the context button next to the connection name and choose **Edit Connection**.



Use the pre-configured connections

Access these connections through `${WORKSPACE}/workspace/GlobalConfiguration/.dbeaver/data-sources.json`. They are always visible in the **Database Navigator** and are not editable, as they are managed by administrators.

Info

For details, see [Pre-configured database connections](#).

Use the Cloud Explorer

Cloud connections are set up through your cloud service provider and automatically populated in your navigator tree, based on configurations managed by the server administrator. For a comprehensive guide on configuring and managing cloud connections, please refer to the **Cloud Explorer** article.

Configuring connection settings

In the **New connection** wizard, on the Main tab, you need to set all primary connection settings.

The screenshot shows the 'DRIVER PROPERTIES' dialog in DBeaver. The 'Configuration' tab is selected, and the 'Manual' radio button is chosen. The fields are filled with the following values: Driver: PostgreSQL, Host: localhost, Port: 5432, Database: postgres, Connection name: PostgreSQL@localhost, Project: Private, and Description: (empty). The 'Authentication' section has 'Database Native' selected. The 'Misc' section has 'User role' set to (empty), and expandable sections for 'DATABASE LIST', 'SQL', and 'EXPERT SETTINGS'. The 'Save credentials for the current user' checkbox is unchecked. The 'CREATE' button is highlighted in blue.

Here's what you typically need to configure:

Setting	Description
Host	The address of your database server.
Port	The port number used to connect.
Database	The name of the database you want to connect to.
Connection name	A name you choose for this connection. It shows up in the Database Navigator.
Project	Links the connection to a specific Project. You can choose between "Shared" and "Private".
Description	A short description of the connection.
Authentication	Choose the type of authentication the database requires.

Credential saving options

This table describes how credential options vary based on the project type:

Project	Option	Description
Private	Save credentials for the current user.	These credentials will be used to make automatic connection.
Shared	Save credentials for all users with access.	These credentials will be used to make automatic connection for all users having access. For more details, see Users and Teams .

Note

Save credentials checkbox ignored when using [Secret Providers](#).

Expert settings

Setting	Description
Keep alive (in seconds)	Sets the interval in seconds for sending signals to keep the connection active. Set to 0 to disable.
Auto commit	Automatically commits changes during a transaction if enabled. For more information, see Auto and Manual commit

Note

The number and type of connection properties are very dependent on the driver.

Additional settings

When setting up a new connection, you may need to configure additional options:

Additional tab	Description
Driver Properties	Each driver has its own set of additional properties.
SSH Tunnel	Configure an SSH tunnel if your database requires a secure connection.
SSL	Enable and configure SSL if your connection requires encryption.

Tip

For secure database connections, store your SSH credentials in `data-sources.json` as follows: - **SSH Key**: Assign your SSH key to `connections.configuration.handlers.ssh_tunnel.properties.keyValue`. - **SSH Username**: Store your SSH username under `connections.configuration.handlers.ssh_tunnel.user`.

Secure Storage with Secret Providers

CloudBeaver supports various cloud-based secret providers to retrieve database credentials securely. For detailed setup instructions, see [Secret Providers](#).

Manage access permissions

In the Community, Enterprise and AWS Editions, you can manage access permissions for a database:

1. Initiate the creation of a new database connection as an administrator.
2. Find the **Project** dropdown menu and select **Shared** project.
3. Complete the necessary fields based on the selected type of credentials.
4. Go to **Access** tab, click **Edit**, check the box next to their names and click the **Add** button.

MAINDRIVER PROPERTIESSSH TUNNELSSLACCESS

CANCELTESTCREATE

Search for user or team name

DELETEEDIT

☐	USER OR TEAM NAME	DESCRIPTION
Administrators see all connections.		
☐	 Developer	
☐	 Denis	
☐	 Eugene	
☐	 Ksenia	

Search for user or team name

ADD

☐	USER OR TEAM NAME	DESCRIPTION
☐	 Developer	
☐	 Denis	
☐	 Eugene	
☐	 Ksenia	

To manage access permissions for a database, ensure the **Enable private connections** setting is configured correctly in the **Server configuration** settings:

- If **Enable private connections** is turned on, switch the project mode to **Shared** from the dropdown menu of the Project settings to access the **Access** tab.
- If **Enable private connections** is turned off, the **Access** tab is available by default.

Note

For Private projects, credentials saved during connection creating or database login are stored permanently for the current user. For Shared projects, credentials saved while creating a connection are available to all users with access, but if saved during login, they are retained only for the current session.

Network configuration settings

SSH configuration

CloudBeaver supports the following SSH authentication methods:

- Username/password
- Public key authentication

Configuring SSH tunnels

1. Start by [creating a connection](#).
2. Set the hostname to `localhost` and the port to match the remote database in **Main** tab.

Tip

This step ensures CloudBeaver connects to the database on the remote machine through the SSH tunnel, treating the remote database as if it were running locally.

For example, if the database server listens on port 3306, the SSH tunnel forwards this port to your server machine. In CloudBeaver, set the hostname to `localhost` and the port to 3306 to use the local end of the tunnel. Ensure that the local port specified in the SSH tunnel configuration matches the port set in CloudBeaver's connection settings. Verify that no other local services use the same port to avoid conflicts.

3. Go to the **SSH Tunnel** tab in the connection settings.
4. Enable the checkbox **Use SSH Tunnel** and choose the authentication method from **Authentication method** dropdown.
5. Depending on the chosen authentication method, configure the required settings:

Field	Description
Host	Enter the hostname or IP address of the SSH server.
Port	Specify the SSH server port (default is 22).
User	Provide the username to authenticate with.
Password	Enter the password for the specified user.

Field	Description
Private key	Paste the private key or use the File button to upload it.
Passphrase	(Optional) Enter the passphrase if the private key is encrypted.

6. Click **Test tunnel** to verify the parameters.

Tip

To avoid entering credentials every time, enable the **Save credentials for the current user** option. Depending on the connection type, the checkbox name may vary. For more details, see [Credential saving options](#).

Advanced settings

Setting	Description
Alive interval (ms)	Sets the interval in milliseconds for sending a keep-alive packet to keep the connection active. A value of 0 disables keep-alive packets.
Connect timeout (ms)	Sets the maximum time in milliseconds to establish a tunnel connection before timing out.

SSL configuration

CloudBeaver supports the ability to establish connections using Secure Socket Layer (SSL), a protocol for creating authenticated and encrypted links between networked computers.

Note

SSL parameters are optional and should only be specified if required by your server configuration.

Setting up SSL configuration

1. Start by [creating a connection](#).
2. Go to the **SSL** tab in the connection settings.
3. Enable the checkbox **Enable SSL**
4. Configure the required settings:

Parameter	Description
CA Certificate	Path to the Certificate Authority (CA) certificate.
Client Certificate	Path to the client's public key certificate.
Client Private Key	Path to the client's private key.
SSL Mode	Desired security level for SSL.
SSL Factory	Provide SSLSocketFactory factory class name.
Cipher Suites (optional)	Specify a comma-separated cipher suites list to use.
Skip host name validation	Whether to bypass hostname validation.
Use self signed certificate (non-secure)	Acceptance of self-signed certificates.
Force TLS 1.2	Enforce using TLS version 1.2.

Parameter	Description
Verify server certificate	Ensures the server certificate is valid.
Allow public key retrieval	Allows the retrieval of the server's public key.

Note

The availability of these parameters can vary depending on the database.

5. If all other necessary connection settings are configured, complete the setup by clicking **Finish**, or test the configuration by clicking **Test Connection** to ensure that SSL is functioning correctly.

Tip

To avoid entering credentials every time, enable the **Save credentials for the current user** option. Depending on the connection type, the checkbox name may vary. For more details, see [Credential saving options](#).

Setting up SSL configuration via Driver properties

For some databases, an alternative method to configure SSL settings is available through the **Driver properties** tab. This approach allows access to additional or driver-specific SSL options that may not be present in the main SSL tab:

1. Select the **Driver properties** tab within the **Connection settings**.

The screenshot shows the 'DRIVER PROPERTIES' tab in the CloudBeaver interface. At the top, there are tabs for 'MAIN', 'DRIVER PROPERTIES', 'SSH TUNNEL', and 'SSL'. On the right side of the tab bar are three buttons: 'CANCEL', 'TEST', and 'CREATE'. Below the tab bar is a table with two columns: 'NAME' and 'VALUE'. The 'NAME' column has a search bar with 'ssl' entered and a close button. The table lists various SSL-related properties. Some properties have dropdown arrows next to their values. Below the table is a '+ Add property' link.

NAME	VALUE
+ Add property	
gsslib	auto
ssl	
sslcert	
sslfactory	org.postgresql.ssl.LibPQFactory
sslfactoryarg	
sslhostnameverifier	
sslkey	
sslmode	
sslpassword	
sslpasswordcallback	
sslResponseTimeout	5000
sslrootcert	

2. Look for properties prefixed with `ssl`, such as `sslResponseTimeout`, `sslfactoryarg`, `sslhostnameverifier`, etc. Configure the necessary properties according to your database and server requirements.
3. If all other necessary connection settings are configured, complete the setup by clicking **Finish**, or test the configuration by clicking **Test Connection** to ensure that SSL is functioning correctly.

Import SSL certificates on startup

CloudBeaver can import certificates from a predefined folder at startup. This method is useful for databases protected by self-signed certificates. It ensures secure connections without manual setup, especially when a firewall or internal security policies require specific certificates.

For Docker Compose

To add a certificate, place the file in the `{workspace}/.data/custom-certs` folder on the server. CloudBeaver will detect and import it on startup.

Info

For more details on `workspace` location, see [Workspace location](#)

To manually copy a certificate, use the following command from the Compose project directory:

```
docker compose cp /path/to/cacert/root.chain.pem cloudbeaver:/opt/cloudbeaver/
workspace/.data/custom/
```

After copying the certificate, restart the cluster (if it is already running) to apply the changes:

```
docker compose restart
```

For Kubernetes

Since if you mount the folder with certificates inside the workspace before the first startup - there may be problems with access rights inside the system `.data` folder, it is recommended to use `/opt/cloudbeaver/conf/custom-certs` folder. So you need to create a volumen and mount it with the `/opt/cloudbeaver/conf/custom-certs`.

All supported certificate folders

1. `{workspace}/.data/custom-certs`
2. `{workspace}/.data/custom`
3. `/opt/cloudbeaver/conf/custom-certs`
4. `/opt/cloudbeaver/conf/custom`

Read more about [CloudBeaver deployment](#).

Important

- Only certificates are imported, not private keys.
- The `custom-certs` directory must not contain any extra files or subfolders.

Database authentication methods

Database Native authentication

CloudBeaver includes support for Database Native authentication, enabling secure connections to your databases.

The screenshot shows the CloudBeaver configuration window for a PostgreSQL database. The interface is divided into several sections: Driver, Configuration, Host, Port, Database, Connection name, Project, and Description. The Authentication section is highlighted with a red box. It contains a dropdown menu for 'Authentication' set to 'Database Native', fields for 'User name' (postgres) and 'User password' (masked), and a checkbox for 'Save credentials for the current user'. Below the Authentication section, there are expandable sections for 'MISC', 'User role', 'DATABASE LIST', 'SQL', and 'EXPERT SETTINGS'.

Settings

Here are the steps to configure Database Native authentication:

1. From the **Authentication** dropdown menu, select the **Database Native** method.
2. Enter the username and password that are associated with your database in the **Username** and **Password** fields respectively.

Tip

If you check the **Save credentials for the current user** option, you will not be asked to input your credentials every time you connect. For more details on connection settings, see [Create Connection](#).

3. Once you have input your credentials, click the **Test** button to verify your settings. If everything is configured correctly, CloudBeaver should now be able to connect to your database using the Database Native authentication method.

MongoDB authentication

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

CloudBeaver is fully compatible with the MongoDB authentication method, offering you a secure way to access your MongoDB databases.

The screenshot displays the CloudBeaver configuration interface for MongoDB. The interface is divided into two main sections: 'Driver' and 'Configuration' on the left, and 'AUTHENTICATION' on the right. The 'Driver' section includes fields for 'Host' (localhost), 'Port' (27017), and 'Database' (local). The 'Configuration' section includes 'Connection name' (MongoDB@localhost), 'Project' (Private), and 'Description'. The 'AUTHENTICATION' section is highlighted with a red border and contains fields for 'User name' (mongo_user), 'User password' (masked), and a checkbox for 'Save credentials for the current user'. Below this, the 'SETTINGS' section includes a dropdown for 'Mechanism' (SCRAM-SHA-1), a text field for 'Source' (admin), and a text field for 'Replica Set'. At the bottom, there is a link to 'EXPERT SETTINGS'. The top of the interface has tabs for 'MAIN', 'DRIVER PROPERTIES', 'SSH TUNNEL', and 'SSL', along with 'CANCEL', 'TEST', and 'CREATE' buttons.

Settings

Here are the steps to configure MongoDB authentication:

1. Start by **creating a connection** for MongoDB.
2. Enter the username and password in the **Username** and **Password** fields respectively.

Tip

If you check the **Save credentials for the current user** option, you will not be asked to input your credentials every time you connect. For more details on connection settings, see [Create Connection](#).

3. From the **Mechanism** dropdown menu, select an option. The following table outlines the available authentication mechanisms:

Mechanism	Description
SCRAM-SHA-1	A password-based authentication mechanism using SHA-1 hashing algorithm. Suitable for securing data transmissions.
SCRAM-SHA-256	An upgraded version of SCRAM-SHA-1, using the stronger SHA-256 hashing algorithm. Offers enhanced security.
MONGODB-CR	A protocol specific to MongoDB, utilizing a challenge-response mechanism for authentication.
PLAIN	A simple, unencrypted password-based authentication mechanism. Not recommended for production environments due to security vulnerabilities.
GSSAPI	Based on the Kerberos protocol, this mechanism offers secure password authentication.
NONE	No authentication is required. This method should only be used in secure, isolated environments.

Info

You can read more about these mechanisms on the [MongoDB documentation](#) page.

4. Fill in the additional fields as necessary.

Field	Description
Source	The authentication database to use. Typically, this is <code>admin</code> for administrative users.
Replica Set	Required when your MongoDB setup includes a replica set.

5. Once you have input your credentials, click the **Test** button to verify your settings. If everything is configured correctly, CloudBeaver should now be able to connect to your database using the MongoDB authentication method.

Microsoft Entra ID authentication

Note

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

CloudBeaver includes support for Microsoft Entra ID authentication, enabling secure connections to your databases.

The screenshot shows the CloudBeaver configuration interface for a PostgreSQL driver. The interface is divided into several sections:
 - **Driver:** A dropdown menu set to 'PostgreSQL'.
 - **Configuration:** Two radio buttons, 'Manual' (selected) and 'URL'.
 - **Host *:** A text input field containing 'localhost'.
 - **Port:** A text input field containing '5432'.
 - **Database:** A text input field containing 'postgres'.
 - **Connection name *:** A text input field containing 'PostgreSQL@localhost'.
 - **Project:** A dropdown menu set to 'Private'.
 - **Description:** A large text area.
 - **Authentication:** A section highlighted with a red border, containing a dropdown menu set to 'Microsoft Entra ID for PostgreSQL', a text input field for 'Azure Group Name', and a checkbox labeled 'Save credentials for the current user'.
 - **MISC:** A section containing a 'User role' text input field, a 'DATABASE LIST' section with a right-pointing arrow, and an 'SQL' section with a right-pointing arrow.
 - **EXPERT SETTINGS:** A section with a right-pointing arrow.
 At the top right, there are three buttons: 'CANCEL', 'TEST', and 'CREATE'.

Info

To use Microsoft Entra ID authentication for database connections, sign in to CloudBeaver through the **Microsoft Entra ID** provider. Azure authentication works only within an active Entra ID session. For setup details, see [Set up Microsoft Entra ID provider](#).

Settings

Here are the steps to configure Microsoft Entra ID authentication:

1. From the **Authentication** dropdown menu, select the **Microsoft Entra ID** method.
2. Click the **Authenticate** button that appears. You'll be prompted to log in through Azure.
 - Alternatively, log in using the **Entra ID** identity provider directly.
3. If necessary, provide the **Azure Group Name** for group-based access.
4. Once you have input database connection details, click the **Test** button to verify your settings. If everything is configured correctly, CloudBeaver should now be able to connect to your database using the Microsoft Entra ID authentication method.

Tip

If you check the **Save credentials for the current user** option, you will not be asked to input your credentials every time you connect. For more details on connection settings, see [Create Connection](#).

Oracle wallet authentication

Note

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

CloudBeaver includes support for Oracle Wallet authentication, enabling secure connections to your databases.

The screenshot shows the 'DRIVER PROPERTIES' dialog in CloudBeaver. The 'Driver' is set to 'Oracle'. The 'JDBC URL' is 'jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps...'. The 'Connection name' is 'Oracle@jdbc:oracle:thin:@(D'. The 'Project' is 'Private'. The 'Description' field is empty. The 'AUTHENTICATION' section is highlighted with a red border. It shows 'Oracle Wallet' selected in the authentication dropdown. The 'User name' is 'system' and the 'User password' is masked with '.....'. Below this is the 'Oracle wallet archive' section with an 'UPLOAD FILE' button and a checkbox for 'Save credentials for the current user'. The 'EXPERT SETTINGS' section is collapsed.

Settings

Here are the steps to configure Oracle Wallet authentication:

1. From the **Authentication** dropdown menu, select the **Oracle Wallet** method.
2. Create an Oracle wallet if you do not already have one. For more information, see the official Oracle documentation: [Oracle Wallet Configuration Guide](#).

Tip

For CloudBeaver, the Oracle Wallet files must be provided as a zip archive.

3. Enter the username and password that are associated with your database in the **User name** and **User password** fields respectively.

Tip

If you check the **Save credentials for the current user** option, you will not be asked to input your credentials every time you connect. For more details on connection settings, see [Create Connection](#).

4. Once you have input your credentials, click the **Test** button to verify your settings. If everything is configured correctly, CloudBeaver should now be able to connect to your database using the Oracle Wallet authentication method.

Pgpass authentication

CloudBeaver is fully compatible with the PgPass authentication method, offering a secure way to access your PostgreSQL databases.

Note

Only an admin can create a PgPass connection.

Settings

PostgreSQL's PgPass method allows authentication using credentials stored in a `.pgpass` file. Follow the steps below to configure it in CloudBeaver.

Server configuration

1. Create the `.pgpass` file. Each line in the file follows the format:

```
hostname:port:database:username:password
```

Save this line in the `.pgpass` file and ensure the file permissions are set to be readable only by the user.

Info

For more information on `.pgpass`, see [PostgreSQL documentation](#).

2. Open a terminal and run the following command to list Docker containers:

```
docker ps
```

3. Copy the ID of the CloudBeaver container from the list.
4. Transfer the `.pgpass` file to the container:

- **For versions before 24.3.4:**

```
docker cp .pgpass "<container-id>:/root/.pgpass"  
docker exec "<container-id>" chmod 600 /root/.pgpass
```

- **Starting from version 24.3.4:**

```
docker cp .pgpass "<container-id>":/opt/cloudbeaver/.pgpass
docker exec "<container-id>" chmod 600 /opt/cloudbeaver/.pgpass
```

Note

Ensure the `.pgpass` file is placed in the correct directory based on your CloudBeaver version.

Create connection

After transferring the `.pgpass` file to the container, follow these steps to finalize the configuration:

1. Open CloudBeaver and start **creating a connection**.
2. In the **Authentication** dropdown menu, select **PostgreSQL PgPass**.
3. Enter the username associated with the PostgreSQL database in the **Username** field.
4. Test the connection by clicking the **Test** button. If the `.pgpass` file is properly configured and the credentials are correct, CloudBeaver will connect successfully.

Salesforce authentication

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

CloudBeaver is fully compatible with the Salesforce password and OAuth authentication methods, offering you a secure ways to access your Salesforce databases.

Password authentication

Here are the steps to configure Salesforce authentication:

1. From the **Authentication** dropdown menu, select the **Salesforce password authentication** method.
2. Enter the username, password and security token that are associated with your Salesforce database in the **Username**, **Password** and **Security token** fields respectively.

Note

Having a security token is necessary for securely connecting to Salesforce from CloudBeaver. This token can be generated in your Salesforce account settings.

3. Click the **Test** button to verify your settings. If everything is configured correctly, CloudBeaver should now be able to connect to your database using the Salesforce authentication method.

Tip

If you check the **Save credentials for the current user** option, you will not be asked to input your credentials every time you connect.

MAIN DRIVER PROPERTIES SSH TUNNEL CANCEL TEST CREATE

Driver
Salesforce

Host *
login.salesforce.com

Port
443

Database
db

Connection name *
Salesforce@login.salesforce.com

Project
Private

Description

AUTHENTICATION
Salesforce password authentication

User name
salesforce_user

User password

Security token

☒ Save credentials for the current user

EXPERT SETTINGS

Keep alive (in seconds)
0

☒ Auto commit

OAuth authentication

Follow these steps to configure Salesforce OAuth authentication:

1. From the **Authentication** dropdown menu, select **Salesforce OAuth authentication**.
2. Enter the **Client ID** and **Client Secret** associated with your Salesforce application in their respective fields.

Tip

The **Client Secret** field is optional, but entering it will allow a more secure connection to your Salesforce account.

3. Click the **Test Connection** button to verify your settings. If configured correctly, CloudBeaver will establish a connection using the Salesforce OAuth authentication method.

MAIN

DRIVER PROPERTIES

SSH TUNNEL

CANCEL

TEST

CREATE

Driver

Salesforce

Host *

login.salesforce.com

Port

443

Database

db

Connection name *

Salesforce@login.salesforce.com

Project

Private

Description

AUTHENTICATION

Salesforce OAuth authentication

clientSecret

clientID

Save credentials for the current user

EXPERT SETTINGS

Keep alive (in seconds)

0

Auto commit

Kerberos authentication

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

CloudBeaver includes support for Kerberos authentication, enabling secure connections to your databases.

The screenshot shows the 'DRIVER PROPERTIES' tab in the CloudBeaver configuration window. The 'Driver' is set to 'MySQL'. The 'Configuration' section has 'Manual' selected. The 'Host' is 'localhost', 'Port' is '3306', and 'Database' is empty. The 'Connection name' is 'MySQL@localhost', 'Project' is 'Private', and 'Description' is empty. The 'AUTHENTICATION' section is highlighted with a red border and contains the following fields: 'MySQL Kerberos' (selected in the dropdown), 'User name', 'User password', 'Kerberos user name', 'KDC server', and 'Realm name'. There is a checkbox for 'Save credentials for the current user'. Below this, the 'ADVANCED' section shows 'Server Time Zone'. At the bottom, there is a link for 'EXPERT SETTINGS'. The top right of the window has 'CANCEL', 'TEST', and 'CREATE' buttons.

Supported databases

- AlloyDB for PostgreSQL
- CockroachDB
- Db2 for LUW

Note

Kerberos authentication is only supported with **version 11.x of the Db2 driver**.

- Fujitsu Enterprise Postgres
- Greenplum
- SAP HANA
- MariaDB
- Oracle Database
- PostgreSQL
- MySQL
- Amazon Redshift
- Teradata
- TimescaleDB
- Trino (formerly PrestoSQL)
- Yellowbrick Data Warehouse
- YugabyteDB

Settings

Here are the steps to configure Kerberos authentication:

1. From the **Authentication** dropdown menu, select the **Kerberos** method.
2. Fill in the following fields:

Setting	Description
User name	Specifies the name of the user or role within the database.
User password	The password associated with your Kerberos user.
Kerberos user name	A unique identity in the Kerberos system to which Kerberos can assign tickets, enabling access to services that are Kerberos-aware.
Realm name	The domain over which a Kerberos authentication server can authenticate users, hosts, or services. It's often the uppercase version of the D

Setting	Description
	NS domain name it oversees.
KDC server	The hostname of your Kerberos Key Distribution Center (KDC), which is a service that provides session tickets and keys within an Active Directory domain.

Tip

If you check the **Save credentials for the current user** option, you will not be asked to input your credentials every time you connect.

- Once you have input your credentials, click the **Test** button to verify your settings. If everything is configured correctly, CloudBeaver should now be able to connect to your database using the Kerberos authentication method.

Databases support

Supported databases

CloudBeaver supports a wide range of databases, including relational, NoSQL, and embedded, providing easy integration with different systems and business needs.

Note

Not all databases in the list are available for connection by default. If you don't see the database you need, contact your server administrator. For instructions on adding new database drivers, refer to [the instruction](#).

Relational databases

- Apache Kyuubi
- ClickHouse
- Db2 iSeries/AS 400 for IBM i
- Db2 LUW
- DuckDB
- Firebird
- H2 Embedded
- MariaDB
- MySQL
- Oracle
- PostgreSQL
- SQL Server
- SQLite
- Trino
- Altibase
- Apache Arrow
- Apache Calcite Avatica
- Apache Kylin
- Aurora DSQL

- Databricks OSS
- Azure SQL Server
- Babelfish via TDS (beta)
- CUBRID
- Cache
- ClickHouse (Legacy)
- Cloudberry
- CloudSQL - MySQL
- CloudSQL - PostgreSQL
- CloudSQL - SQL Server
- CockroachDB
- CrateDB
- CrateDB (Legacy)
- Dameng
- Db2 for z/OS
- Denodo 8
- Derby Embedded
- Derby Server
- Dremio
- EDB
- Exasol
- Fujitsu Enterprise Postgres
- GaussDB
- Google AlloyDB
- Google Cloud Spanner
- Greenplum
- Greengage
- H2 Embedded V.2
- H2 Server
- H2GIS Embedded
- H2GIS Server

- HANA
- HSQL Embedded
- HSQL Server
- Informix
- Ingres
- InterSystems IRIS
- JDBCX
- Jennifer
- MS Access (UCanAccess)
- Materialize
- MaxDB
- Mimer SQL
- MonetDB
- MySQL 5 (Legacy)
- NDB Cluster
- NetSuite
- Netezza
- NuoDB
- ODBC
- OceanBase
- Ocient
- OmniSci (formerly MapD)
- OpenEdge
- OpenSearch
- Pervasive SQL
- PrestoDB
- PrestoSQL
- Raima
- Redshift
- Redshift Serverless
- RisingWave

- SAP ASE jConnect
- SQL Server (Old driver, jTDS)
- SQLite Crypt
- SQream DB
- Salesforce
- Salesforce Data Cloud
- SingleStore
- Snowflake
- StarRocks
- Sybase jConnect
- Sybase jTDS
- Teradata
- TiDB
- Vertica
- Virtuoso
- Yellowbrick
- YugabyteDB

NoSQL databases

- Apache Drill
- Apache Hive
- Apache Ignite
- Apache Phoenix
- Apache Spark
- Athena
- Azure Cosmos DB for Cassandra
- Azure Cosmos DB for MongoDB
- Azure Cosmos DB for NoSQL
- Cassandra
- Cloudera Impala

- CouchDB
- Couchbase
- Couchbase 5+
- CSV
- CUBRID
- DBF
- DocumentDB
- DynamoDB
- Elasticsearch
- FerretDB
- Firestore
- Gemfire XD
- Google BigQuery
- Google Cloud Bigtable
- InfluxDB
- InfluxDB 2
- InfluxDB 3
- Kafka (ksqlDB)
- Keyspaces
- Machbase
- MongoDB
- Neo4j
- Neptune
- Open Distro Elasticsearch
- OrientDB
- Parquet
- Raima
- Redis
- ScyllaDB
- SnappyData
- Solr

- TDengine
- TDengine Cloud
- TimescaleDB
- Timestream
- XLSX
- Yugabyte CQL

File drivers

Multi Source

Note

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

This guide explains how to set up and use the **Multi Source** driver in CloudBeaver. This driver allows you to work with multiple files of different formats within a single connection, including:

- [CSV](#)
- [JSON](#)
- [Parquet](#)
- [XML](#)
- [XLSX](#)

You can query data from different sources simultaneously, apply filters and sorting, and even join data across files.

Important

When using the **Multi Source** driver, all connected files are read-only. To make changes, update the original files outside CloudBeaver.

Multi Source driver connection

This section describes two ways to set up a connection: opening a file from Cloud Storage or creating a connection using the wizard.

Note

The connection from Cloud Storage is temporary and will be removed when the session ends.

Open a file from Cloud Storage

You can open files directly from your [Cloud Storage](#). Open **Cloud Storage**, find the file you need, and double-click it to create a connection. The connection will appear in the **File databases** folder in [Database Navigator](#).

Info

You'll need to be logged in using the **Identity Provider** associated with your Cloud Storage.

Create a connection

To create a connection in CloudBeaver, **use the new connection wizard**, select the **Files - Multi Source** driver, and fill the following fields:

MAIN DRIVER PROPERTIES

Driver

Files - Multi Source

Configuration

☒ Manual ☐ URL

Database

s3://AWS/files/path/example/

Connection name *

Files - Multi Source@localhost

Project

Private

Description

EXPERT SETTINGS

Keep alive (in seconds)

0

☒ Auto commit

☐ Read-only connection

Field	Description	Options
Configuration	Choose how to specify the connection details.	- Manual : Enter the database path manually. - URL : Provide a connection URL.
Database	Enter the file/folder path.	Use the appropriate format for your cloud provider. For instance: <code>s3://AWS/files/path/example/</code>
Connection name	Enter a custom name for your connection.	Defaults to <code>Files - Multi Source@localhost</code> .

Field	Description	Options
Project	Select the project where this connection belongs.	Default: Private .
Description	(Optional) Add details about this connection.	
Keep alive (in seconds)	Set how long the connection stays active.	Default: 0 (no timeout).
Auto commit	Enable automatic transaction commits. See more details on Auto and Manual commit modes .	Enabled by default.
Read-only connection	Restrict the connection to read-only mode.	Optional checkbox.

Tip

When using the **folder** path in the **Database** field, CloudBeaver scans the directory up to two levels deep for supported files. If the folder contains multiple files, CloudBeaver organizes them into schemas based on their directory structure. For more information, see **folder structure**.

Features and capabilities

Advanced SQL query capabilities

The **Multi Source** driver supports the full range of SQL queries:

- **Simple queries** (e.g., `SELECT * FROM table`): Data is read directly from the files.
- **Complex queries** (e.g., using `WHERE`, `JOIN`, `ORDER BY`, `GROUP BY`): When a complex query is executed for the first time, the driver imports the data from all selected files into an **internal database** to enable advanced SQL functions. Subsequent queries run faster because the data is already imported into an internal database.

Note

If you want to join data from different files, they must be opened in the same connection. To do this, use a folder path instead of a single file when creating the connection.

Folder structure

When working with multiple files and folders, CloudBeaver organizes them as follows:

Location	Schema in CloudBeaver
Root files	Default schema
Subfolder files	Schema named after the subfolder
Files in deeper folders	Ignored

If your folder structure looks like this:

```
Data/  
├─ employees.csv  
├─ sales.json  
└─ Reports/  
    ├─ monthly.xml  
    ├─ yearly.parquet  
    └─ summary.xlsx
```

CloudBeaver will create:

- Default schema: employees, sales
- Reports schema: monthly, yearly, summary

Tip

To focus on specific files, consider selecting individual files or folders when configuring the connection.

Internal database

When you execute a complex query (such as `WHERE`, `JOIN`, `GROUP BY`, or `ORDER BY`.) on multiple files for the first time, the **Multi Source** driver processes the data by importing it into a temporary internal SQLite database.

This internal database stores data temporarily on the server during your session and is cleared when session ends.

Additional features

CloudBeaver provides additional features compatible with **Multi Source** driver, but not exclusive to it:

Category	Feature
Data Export	Data Export
Data Visualization	Visual Query Builder
	Charts

Info

For more details on driver properties, see [File-based driver properties](#).

CSV

Note

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

This guide provides instructions on how to set up and use CSV files with CloudBeaver. The **CSV Pro** driver allows you to work with CSV data as if it were in a database. You can retrieve data and apply filters, sorting, and other operations, even combining data from multiple CSV files.

Important

When using the **CSV Pro** driver, all connected CSV files are read-only. To make changes, you need to update the original files outside CloudBeaver.

CSV Files driver connection

This section describes two ways to set up a connection: opening a file from Cloud Storage or creating a connection using the wizard.

Note

The connection from Cloud Storage is temporary and will be removed when the session ends.

Open a file from Cloud Storage

You can open files directly from your [Cloud Storage](#). Open **Cloud Storage**, find the file you need, and double-click it to create a connection. The connection will appear in the **File databases** folder in [Database Navigator](#).

Info

You'll need to be logged in using the [Identity Provider](#) associated with your Cloud Storage.

Create a connection

To create a connection in CloudBeaver, [use the new connection wizard](#), select the **CSV** driver, and fill the following fields:

MAIN DRIVER PROPERTIES

Driver

CSV Pro

Manual

URL

Database

s3://AWS/files/path/example/

Connection name *

CSV Pro@localhost

Project

Private

Description

EXPERT SETTINGS

Keep alive (in seconds)

0

☒ Auto commit

☐ Read-only connection

Field	Description	Options
Configuration	Choose how to specify the connection details.	- Manual: Enter the database path manually. - URL: Provide a connection URL.
Database	Enter the file/folder path.	Use the appropriate format for your cloud provider. For instance: <code>s3://AWS/files/path/example/</code>
Connection name	Enter a custom name for your connection.	Defaults to <code>CSV Pro@localhost</code> .
Project	Select the project where this connection belongs.	Default: Private .
Description	(Optional) Add details about this connection.	
Keep alive (in seconds)	Set how long the connection stays active.	Default: <code>0</code> (no timeout).
Auto commit	Enable automatic transaction c	Enabled by default.

Field	Description	Options
	ommits. See more details on Auto and Manual commit modes .	
Read-only connection	Restrict the connection to read-only mode.	Optional checkbox.

Tip

When using the **folder** path in the **Database** field, CloudBeaver scans the directory up to two levels deep for supported files. If the folder contains multiple files, CloudBeaver organizes them into schemas based on their directory structure. For more information, see [folder structure](#).

Features and capabilities

Advanced SQL query capabilities

The **CSV Pro** driver supports the full range of SQL queries:

- **Simple queries** (e.g., `SELECT * FROM table`): Data is read directly from the CSV file.
- **Complex queries** (e.g., using `WHERE`, `JOIN`, `ORDER BY`, `GROUP BY`): When a complex query is executed for the first time, the driver imports the entire CSV file into an **internal database** to enable advanced SQL functions. Subsequent queries run faster because the data is already imported into internal database.

Note

If you want to join data from different files, they must be opened in the same connection. To do this, use a folder path instead of a single file when creating the connection.

Folder structure

When working with a folder containing multiple CSV files, CloudBeaver organizes them as follows:

Folder structure	Schema in CloudBeaver
Root files	<code>Default</code> schema
Subfolder files	Schema named after the subfolder

Folder structure	Schema in CloudBeaver
Files in deeper folders	Ignored

If your folder looks like this:

```
Data/
├─ employees.csv
├─ sales.csv
└─ Reports/
   └─ monthly.csv
   └─ yearly.csv
```

CloudBeaver will create:

- Default schema: `employees`, `sales`
- Reports schema: `monthly`, `yearly`

Tip

To focus on specific files, consider selecting individual files or folders when configuring the connection.

Internal database

When you execute a complex query (such as `WHERE`, `JOIN`, `GROUP BY`, or `ORDER BY`), on an **CSV** file for the first time, the **CSV Pro** driver processes the data by importing it into a temporary internal SQLite database.

This internal database stores data temporarily on the server during your session and is cleared when session ends.

Additional features

CloudBeaver provides additional features compatible with **CSV Pro** driver, but not exclusive to it:

Category	Feature
Data Export	Data Export
Data Visualization	Visual Query Builder

Category	Feature
	Charts

Info

For more details on driver properties, see [File-based driver properties](#).

JSON

Note

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

This guide provides instructions on how to set up and use JSON files with CloudBeaver. The **JSON** driver supports standard JSON files and allows you to work with JSON data as if it were in a database. You can retrieve data and apply filters, sorting, and other operations, even combining data from multiple files.

Important

When using the **JSON** driver, all connected JSON files are read-only. To make changes, you need to update the original files outside CloudBeaver.

JSON Files driver connection

This section describes two ways to set up a connection: opening a file from Cloud Storage or creating a connection using the wizard.

Note

The connection from Cloud Storage is temporary and will be removed when the session ends.

Open a file from Cloud Storage

You can open files directly from your [Cloud Storage](#). Open **Cloud Storage**, find the file you need, and double-click it to create a connection. The connection will appear in the **File databases** folder in [Database Navigator](#).

Info

You'll need to be logged in using the [Identity Provider](#) associated with your Cloud Storage.

Create a connection

To create a connection in CloudBeaver, [use the new connection wizard](#), select the **JSON** driver, and fill the following fields:

MAIN DRIVER PROPERTIES

Driver

JSON

Configuration

☒ Manual
☐ URL

Database

Connection name *

JSON@localhost

Project

Private

Description

EXPERT SETTINGS

Keep alive (in seconds)

0

☒ Auto commit
☐ Read-only connection

Field	Description	Options
Configuration	Choose how to specify the connection details.	- Manual: Enter the database path manually. - URL: Provide a connection URL.
Database	Enter the file/folder path.	Use the appropriate format for your cloud provider. For instance: <code>s3://AWS/files/path/example/</code>
Connection name	Enter a custom name for your connection.	Defaults to <code>JSON@localhost</code> .
Project	Select the project where this connection belongs.	Default: Private .
Description	(Optional) Add details about this connection.	
Keep alive (in seconds)	Set how long the connection stays active.	Default: <code>0</code> (no timeout).
Auto commit	Enable automatic transaction c	Enabled by default.

Field	Description	Options
	ommits. See more details on Auto and Manual commit modes .	
Read-only connection	Restrict the connection to read-only mode.	Optional checkbox.

Tip

When using the **folder** path in the **Database** field, CloudBeaver scans the directory up to two levels deep for supported files. If the folder contains multiple files, CloudBeaver organizes them into schemas based on their directory structure. For more information, see [folder structure](#).

Features and capabilities

Advanced SQL query capabilities

The **JSON** driver supports the full range of SQL queries:

- **Simple queries** (e.g., `SELECT * FROM table`): Data is read directly from the JSON file.
- **Complex queries** (e.g., using `WHERE`, `JOIN`, `ORDER BY`, `GROUP BY`): When a complex query is executed for the first time, the driver imports the entire JSON file into an **internal database** to enable advanced SQL functions. Subsequent queries run faster because the data is already imported into internal database.

Note

If you want to join data from different files, they must be opened in the same connection. To do this, use a folder path instead of a single file when creating the connection.

Folder structure

When working with a folder containing multiple JSON files, CloudBeaver organizes them as follows:

Folder structure	Schema in CloudBeaver
Root files	<code>Default</code> schema
Subfolder files	Schema named after the subfolder

Folder structure	Schema in CloudBeaver
Files in deeper folders	Ignored

If your folder looks like this:

```
Data/
├─ employees.json
├─ sales.json
└─ Reports/
    ├─ monthly.json
    └─ yearly.json
```

CloudBeaver will create:

- Default schema: `employees`, `sales`
- Reports schema: `monthly`, `yearly`

Tip

To focus on specific files, consider selecting individual files or folders when configuring the connection.

Internal database

When you execute a complex query (such as `WHERE`, `JOIN`, `GROUP BY`, or `ORDER BY`.), on an **JSON** file for the first time, the **JSON** driver processes the data by importing it into a temporary internal SQLite database.

This internal database stores data temporarily on the server during your session and is cleared when session ends.

Additional features

CloudBeaver provides additional features compatible with **JSON** driver, but not exclusive to it:

Category	Feature
Data Export	Data Export
Data Visualization	Visual Query Builder

Category	Feature
	Charts

Info

For more details on driver properties, see [File-based driver properties](#).

Parquet

Note

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

This guide provides instructions on how to set up and use Parquet files with CloudBeaver. The **Parquet** driver allows you to work with Parquet data as if it were in a database. You can retrieve data and apply filters, sorting, and other operations, even combining data from multiple files.

Important

When using the **Parquet** driver, all connected Parquet files are read-only. To make changes, you need to update the original files outside CloudBeaver.

Parquet Files driver connection

This section describes two ways to set up a connection: opening a file from Cloud Storage or creating a connection using the wizard.

Note

The connection from Cloud Storage is temporary and will be removed when the session ends.

Open a file from Cloud Storage

You can open files directly from your [Cloud Storage](#). Open **Cloud Storage**, find the file you need, and double-click it to create a connection. The connection will appear in the **File databases** folder in [Database Navigator](#).

Info

You'll need to be logged in using the [Identity Provider](#) associated with your Cloud Storage.

Create a connection

To create a connection in CloudBeaver, [use the new connection wizard](#), select the **Parquet** driver, and fill the following fields:

MAIN DRIVER PROPERTIES

Driver

Parquet

▼

Configuration

☒ Manual

☐ URL

Database

Connection name *

Parquet@localhost

Project

Private

▼

Description

EXPERT SETTINGS

Keep alive (in seconds)

0

⌵

☒ Auto commit

☐ Read-only connection

Field	Description	Options
Configuration	Choose how to specify the connection details.	- Manual: Enter the database path manually. - URL: Provide a connection URL.
Database	Enter the file/folder path.	Use the appropriate format for your cloud provider. For instance: <code>s3://AWS/files/path/example/</code>
Connection name	Enter a custom name for your connection.	Defaults to <code>Parquet@localhost</code> .
Project	Select the project where this connection belongs.	Default: Private .
Description	(Optional) Add details about this connection.	
Keep alive (in seconds)	Set how long the connection stays active.	Default: <code>0</code> (no timeout).
Auto commit	Enable automatic transaction c	Enabled by default.

Field	Description	Options
	ommits. See more details on Auto and Manual commit modes .	
Read-only connection	Restrict the connection to read-only mode.	Optional checkbox.

Tip

When using the **folder** path in the **Database** field, CloudBeaver scans the directory up to two levels deep for supported files. If the folder contains multiple files, CloudBeaver organizes them into schemas based on their directory structure. For more information, see [folder structure](#).

Features and capabilities

Advanced SQL query capabilities

The **Parquet** driver supports the full range of SQL queries:

- **Simple queries** (e.g., `SELECT * FROM table`): Data is read directly from the Parquet file.
- **Complex queries** (e.g., using `WHERE`, `JOIN`, `ORDER BY`, `GROUP BY`): When a complex query is executed for the first time, the driver imports the entire Parquet file into an **internal database** to enable advanced SQL functions. Subsequent queries run faster because the data is already imported into internal database.

Note

If you want to join data from different files, they must be opened in the same connection. To do this, use a folder path instead of a single file when creating the connection.

Folder structure

When working with a folder containing multiple Parquet files, CloudBeaver organizes them as follows:

Folder structure	Schema in CloudBeaver
Root files	<code>Default</code> schema

Folder structure	Schema in CloudBeaver
Subfolder files	Schema named after the subfolder
Files in deeper folders	Ignored

If your folder looks like this:

```
Data/
├─ employees.parquet
├─ sales.parquet
└─ Reports/
   └─ monthly.parquet
   └─ yearly.parquet
```

CloudBeaver will create:

- Default schema: `employees`, `sales`
- Reports schema: `monthly`, `yearly`

Tip

To focus on specific files, consider selecting individual files or folders when configuring the connection.

Internal database

When you execute a complex query (such as `WHERE`, `JOIN`, `GROUP BY`, or `ORDER BY`.), on an **Parquet** file for the first time, the **Parquet** driver processes the data by importing it into a temporary internal SQLite database.

This internal database stores data temporarily on the server during your session and is cleared when session ends.

Additional features

CloudBeaver provides additional features compatible with **Parquet** driver, but not exclusive to it:

Category	Feature
Data Export	Data Export

Category	Feature
Data Visualization	Visual Query Builder
	Charts

Info

For more details on driver properties, see [File-based driver properties](#).

Supported compression formats

CloudBeaver supports the following compression formats for Parquet files:

Format	Supported out of the box	Notes
Zstandard (<code>zstd</code>)	Yes	Works without additional setup
Snappy	No	Add the <code>snappy-java</code> JAR to the driver library
<code>LZ4_RAW</code>	No	Add the <code>aircompressor</code> JAR to the driver library

Info

For details on adding a `JAR` files to the driver library, see [Driver Manager](#).

XLSX

Note

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

This guide provides instructions on how to set up and use XLSX files with CloudBeaver. The **XLSX** driver allows you to work with XLSX data as if it were in a database. You can retrieve data and apply filters, sorting, and other operations, even combining data from multiple files.

Important

When using the **XLSX** driver, all connected XLSX files are read-only. To make changes, you need to update the original files outside CloudBeaver.

XLSX Files driver connection

This section describes two ways to set up a connection: opening a file from Cloud Storage or creating a connection using the wizard.

Note

The connection from Cloud Storage is temporary and will be removed when the session ends.

Open a file from Cloud Storage

You can open files directly from your [Cloud Storage](#). Open **Cloud Storage**, find the file you need, and double-click it to create a connection. The connection will appear in the **File databases** folder in [Database Navigator](#).

Info

You'll need to be logged in using the [Identity Provider](#) associated with your Cloud Storage.

Create a connection

To create a connection in CloudBeaver, [use the new connection wizard](#), select the **XLSX** driver, and fill the following fields:

MAIN DRIVER PROPERTIES

Driver

XLSX

Configuration

Manual

URL

Database

Connection name *

XLSX@localhost

Project

Private

Description

EXPERT SETTINGS

Keep alive (in seconds)

0

Auto commit

Read-only connection

Field	Description	Options
Configuration	Choose how to specify the connection details.	- Manual: Enter the database path manually. - URL: Provide a connection URL.
Database	Enter the file/folder path.	Use the appropriate format for your cloud provider. For instance: <code>s3://AWS/files/path/example/</code>
Connection name	Enter a custom name for your connection.	Defaults to <code>XLSX@localhost</code> .
Project	Select the project where this connection belongs.	Default: Private .
Description	(Optional) Add details about this connection.	
Keep alive (in seconds)	Set how long the connection stays active.	Default: <code>0</code> (no timeout).
Auto commit	Enable automatic transaction commit	Enabled by default.

Field	Description	Options
	ommits. See more details on Auto and Manual commit modes .	
Read-only connection	Restrict the connection to read-only mode.	Optional checkbox.

Tip

When using the **folder** path in the **Database** field, CloudBeaver scans the directory up to two levels deep for supported files. If the folder contains multiple files, CloudBeaver organizes them into schemas based on their directory structure. For more information, see [folder structure](#).

Features and capabilities

Advanced SQL query capabilities

The **XLSX** driver supports the full range of SQL queries:

- **Simple queries** (e.g., `SELECT * FROM table`): Data is read directly from the XLSX file.
- **Complex queries** (e.g., using `WHERE`, `JOIN`, `ORDER BY`, `GROUP BY`): When a complex query is executed for the first time, the driver imports the entire XLSX file into an **internal database** to enable advanced SQL functions. Subsequent queries run faster because the data is already imported into internal database.

Note

If you want to join data from different files, they must be opened in the same connection. To do this, use a folder path instead of a single file when creating the connection.

Folder structure

When working with a folder containing multiple XLSX files, CloudBeaver organizes them as follows:

Folder structure	Schema in CloudBeaver
Root files	<code>Default</code> schema
Subfolder files	Schema named after the subfolder

Folder structure	Schema in CloudBeaver
Files in deeper folders	Ignored

Note

Each file becomes a schema, and each sheet in a file becomes a table in that schema.

If your folder looks like this:

```
Data/
├── employees.xlsx
│   ├── Sheet1
│   └── Sheet2
├── sales.xlsx
│   ├── SalesData
│   └── Summary
└── Reports/
    ├── monthly.xlsx
    │   ├── January
    │   └── February
    └── yearly.xlsx
        ├── 2023
        └── 2024
```

CloudBeaver will create:

- **Schema:** Default **Tables:**
- employees.Sheet1
- employees.Sheet2
- sales.SalesData
- sales.Summary
- **Schema:** Reports **Tables:**
- monthly.January
- monthly.February
- yearly.2023
- yearly.2024

Tip

To focus on specific files, consider selecting individual files or folders when configuring the connection.

Internal database

When you execute a complex query (such as `WHERE` , `JOIN` , `GROUP BY` , or `ORDER BY` .), on an **XLSX** file for the first time, the **XLSX** driver processes the data by importing it into a temporary internal SQLite database.

This internal database stores data temporarily on the server during your session and is cleared when session ends.

Additional features

CloudBeaver provides additional features compatible with **XLSX** driver, but not exclusive to it:

Category	Feature
Data Export	Data Export
Data Visualization	Visual Query Builder
	Charts

Info

For more details on driver properties, see [File-based driver properties](#).

XML

Note

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

This guide provides instructions on how to set up and use XML files with CloudBeaver. The **XML** driver supports standard XML files, including both attribute-based and nested element-based structures. You can retrieve data and apply filters, sorting, and other operations, even combining data from multiple files.

Important

When using the **XML** driver, all connected **XML** files are read-only. To make changes, you need to update the original files outside CloudBeaver.

XML Files driver connection

This section describes two ways to set up a connection: opening a file from Cloud Storage or creating a connection using the wizard.

Note

The connection from Cloud Storage is temporary and will be removed when the session ends.

Open a file from Cloud Storage

You can open files directly from your [Cloud Storage](#). Open **Cloud Storage**, find the file you need, and double-click it to create a connection. The connection will appear in the **File databases** folder in [Database Navigator](#).

Info

You'll need to be logged in using the [Identity Provider](#) associated with your Cloud Storage.

Create a connection

To create a connection in CloudBeaver, [use the new connection wizard](#), select the **XML** driver, and fill the following fields:

MAIN DRIVER PROPERTIES

Driver

XML

Configuration

☒ Manual
☐ URL

Database

Connection name *

XML@localhost

Project

Private

Description

EXPERT SETTINGS

Keep alive (in seconds)

0

☒ Auto commit
☐ Read-only connection

Field	Description	Options
Configuration	Choose how to specify the connection details.	- Manual: Enter the database path manually. - URL: Provide a connection URL.
Database	Enter the file/folder path.	Use the appropriate format for your cloud provider. For instance: <code>s3://AWS/files/path/example/</code>
Connection name	Enter a custom name for your connection.	Defaults to <code>XML@localhost</code> .
Project	Select the project where this connection belongs.	Default: Private .
Description	(Optional) Add details about this connection.	
Keep alive (in seconds)	Set how long the connection stays active.	Default: <code>0</code> (no timeout).
Auto commit	Enable automatic transaction c	Enabled by default.

Field	Description	Options
	ommits. See more details on Auto and Manual commit modes .	
Read-only connection	Restrict the connection to read-only mode.	Optional checkbox.

Tip

When using the **folder** path in the **Database** field, CloudBeaver scans the directory up to two levels deep for supported files. If the folder contains multiple files, CloudBeaver organizes them into schemas based on their directory structure. For more information, see [folder structure](#).

Features and capabilities

Advanced SQL query capabilities

The **XML** driver supports the full range of SQL queries:

- **Simple queries** (e.g., `SELECT * FROM table`): Data is read directly from the XML file.
- **Complex queries** (e.g., using `WHERE`, `JOIN`, `ORDER BY`, `GROUP BY`): When a complex query is executed for the first time, the driver imports the entire XML file into an **internal database** to enable advanced SQL functions. Subsequent queries run faster because the data is already imported into internal database.

Note

If you want to join data from different files, they must be opened in the same connection. To do this, use a folder path instead of a single file when creating the connection.

Folder structure

When working with a folder containing multiple XML files, CloudBeaver organizes them as follows:

Folder structure	Schema in CloudBeaver
Root files	<code>Default</code> schema
Subfolder files	Schema named after the subfolder

Folder structure	Schema in CloudBeaver
Files in deeper folders	Ignored

If your folder looks like this:

```
Data/
├─ employees.xml
├─ sales.xml
└─ Reports/
   └─ monthly.xml
   └─ yearly.xml
```

CloudBeaver will create:

- Default schema: `employees`, `sales`
- Reports schema: `monthly`, `yearly`

Tip

To focus on specific files, consider selecting individual files or folders when configuring the connection.

Internal database

When you execute a complex query (such as `WHERE`, `JOIN`, `GROUP BY`, or `ORDER BY`), on an **XML** file for the first time, the **XML** driver processes the data by importing it into a temporary internal SQLite database.

This internal database stores data temporarily on the server during your session and is cleared when session ends.

Additional features

CloudBeaver provides additional features compatible with **XML** driver, but not exclusive to it:

Category	Feature
Data Export	Data Export
Data Visualization	Visual Query Builder

Category	Feature
	Charts

Info

For more details on driver properties, see [File-based driver properties](#).

File based driver properties

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

File drivers allow you to interact with various file formats, like **CSV**, **Excel**, **JSON**, **XML**, and **Parquet**, as if they were databases. Each driver comes with customizable properties to control how data is read and processed.

General properties

Property	Description	Default value
<code>defaultSchema</code>	Default schema name.	<code>default</code>
<code>scanSubfolders</code>	Scan subfolders for data files.	<code>true</code>
<code>subfolderNameSeparator</code>	Defines the separator used in subfolder names within schema names.	<code>--</code>
<code>internalDbBatchSize</code>	Internal database batch size.	<code>1000</code>
<code>internalDbTransactionSize</code>	A number of batches in a single transaction.	<code>10</code>
<code>useInternalDb</code>	Use internal database for complex queries.	<code>true</code>

CSV driver properties

Property	Description	Default value
<code>separator</code>	The delimiter to use for separating entries.	<code>,</code>

Property	Description	Default value
<code>escapeChar</code>	The character to use for escaping a separator or quote.	<code>\</code>
<code>quoteChar</code>	The character to use for quoted elements.	<code>"</code>
<code>strictQuotes</code>	Sets if characters outside the quotes are ignored.	<code>false</code>
<code>ignoreLeadingWhitespace</code>	If true, parser should ignore white space before a quote in a field.	<code>true</code>
<code>ignoreQuotations</code>	If true, treat quotations like any other character.	<code>false</code>
<code>nullFieldIndicator</code>	Which field content will be returned as null.	<code>NEITHER</code>
<code>trimWhitespaces</code>	If true, parser should trim whitespaces from the beginning and end of the field.	<code>true</code>
<code>header</code>	If true, the first row is treated as a header.	<code>true</code>
<code>sampleRows</code>	Number of rows to extract metadata from.	<code>5</code>
<code>wildcard</code>	Wildcard for file names.	<code>**.{csv, tcv, txt}</code>

XLSX driver properties

Property	Description	Default value
<code>header</code>	Indicates if the file has a header.	<code>true</code>

Property	Description	Default value
<code>sampleRows</code>	Number of rows to extract metadata from.	<code>5</code>
<code>wildcard</code>	Wildcard for file names.	<code>**.{xlsx}</code>
<code>schemaNameMode</code>	Defines how the schema name is formed:	<code>RELATIVE_DIR_PATH</code>
	- <code>RELATIVE_DIR_PATH</code> : Uses the concatenated relative path of the directory, and the table name is <code>\$file_name + _ + \$sheet_name</code> .	
	- <code>RELATIVE_FILE_PATH</code> : Uses the relative file path to generate the schema name. Each sheet in the file becomes a table.	

JSON driver properties

Property	Description	Default value
<code>sampleRows</code>	Number of rows to extract metadata from.	<code>5</code>
<code>wildcard</code>	Wildcard for file names.	<code>**.{json}</code>

XML driver properties

Property	Description	Default value
<code>sampleRows</code>	Number of rows to extract metadata from.	<code>5</code>

Property	Description	Default value
<code>wildcard</code>	Wildcard for file names.	<code>**.{xml}</code>

Parquet driver properties

Property	Description	Default value
<code>tmpFolder</code>	Temporary folder for storing downloaded Parquet files from Cloud Storages .	
<code>wildcard</code>	Wildcard for file names.	<code>**.{parquet}</code>

Drivers management

Note

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

CloudBeaver includes a selection of built-in drivers, but not all are preloaded with `.jar` files. This keeps the installation lightweight while allowing you to manage drivers as needed.

You can:

- [Download missing driver files](#) from official Maven and Git repositories if a listed driver isn't fully installed.
- [Add a custom driver](#) if the one you need isn't included.

Important

Only an administrator can perform these actions.

Driver Manager

To begin adding a new driver, navigate to **Administration -> Driver Management**.

Create a custom driver

1. Click **Add** to start adding a new driver.
2. In the form, select a driver from the **Driver Type** dropdown menu.
3. Fill in the necessary fields. For more details, see [Driver parameters](#).
4. Click **Create** to save the driver.

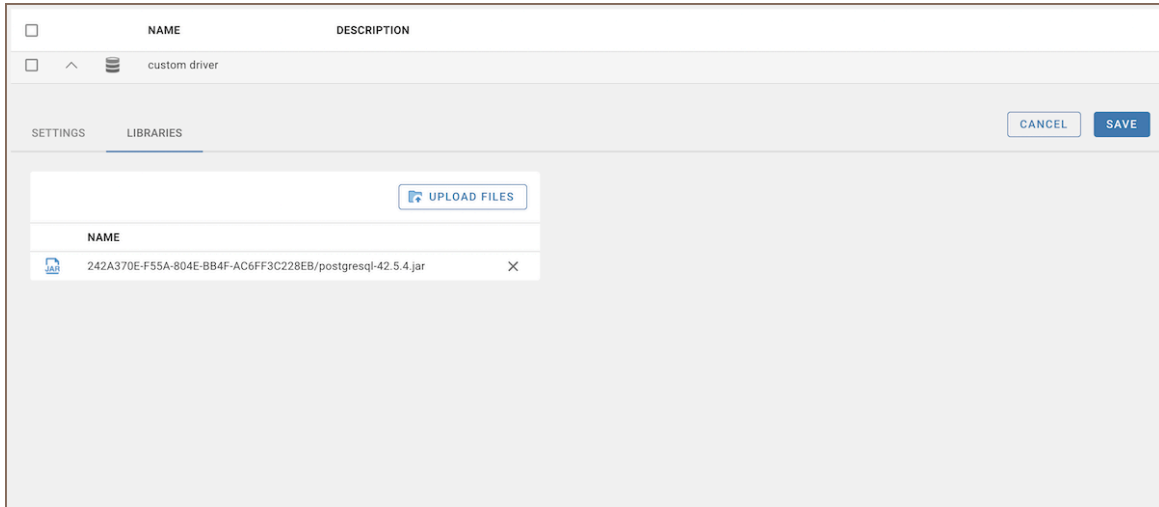
Upload binary files

After creating the driver, upload binary files:

1. Find your driver in the **Driver Management** list and select it.
2. Open the **Libraries** tab.
3. Click **Add File** to upload the required `.jar` files.
4. Confirm that the files appear in the table.
5. Click **Save**.

Info

Once the binary files are uploaded, you can [create a connection](#).



Generic driver

The **Generic** driver allows you to connect to databases that do not have a predefined JDBC driver. It works with any database that supports JDBC, giving you control to manually load the required JDBC library.

This driver is useful for connecting to custom or uncommon databases. It ensures flexibility in setting connection parameters, but you must ensure that the JDBC driver you provide is fully compatible with the database to avoid issues with connections or queries.

URL template

JDBC drivers use URLs to connect to databases, similar to web URLs. These URLs typically follow the format `jdbc:vendor:host:port/database`.

Example

PostgreSQL database might use `jdbc:postgresql:localhost:5432/postgres`.

Download binary files

If the required driver exists in the **Driver Management** list but its libraries haven't been installed, you can download them.

1. Go to **Settings -> Administration -> Server Configuration**.
2. Enable **Allow drivers downloading**.
3. Open the **Libraries** tab of the required driver in **Administration -> Driver Management**.
4. Click **Download** to fetch `.jar` files from an official Maven or Git repository.
5. Confirm that the files appear in the table.
6. Click **Save**.
7. **Create a connection**.

Edit an existing driver

1. Open the required driver in **Administration -> Driver Management**.
2. Update the necessary fields. For more details, see **Driver parameters**.

Warning

Changing the driver class will delete the existing driver.

3. Go to the **Libraries** tab to manage driver files:
 - Remove old files.
 - Upload new ones.
4. Click **Save**.

The screenshot shows the 'Oracle JDBC driver' configuration page in the Administration console. At the top, there is a table with columns 'NAME' and 'DESCRIPTION'. Below the table, there are tabs for 'SETTINGS' and 'LIBRARIES'. The 'SETTINGS' tab is active, showing various configuration fields. On the right side of the settings panel, there are 'CANCEL' and 'SAVE' buttons.

NAME	DESCRIPTION
Oracle	Oracle JDBC driver

SETTINGS | **LIBRARIES** | **CANCEL** | **SAVE**

Driver Name *: Oracle
Driver Type *: Oracle (dropdown)
Class Name *: oracle.jdbc.OracleDriver
URL Template: jdbc:oracle:thin:@(host):(port)/(database)
Description: Oracle JDBC driver

DEFAULTS
Default Port: 1521
Default database: ORCL

Restore default driver settings

To reset a driver to its original settings:

1. Open the required driver in **Administration -> Driver Management**.
2. Click **Reset to default**.

Note

This option is only available for built-in drivers. Custom drivers cannot be restored to default.

Delete an existing driver

Note

This action removes only the driver reference from the configuration file. To delete the driver file itself, go to your [workspace](#) and remove it manually.

1. Open the **Driver Management** tab.
2. Locate the driver you want to delete in the list.
3. Check the checkbox next to the driver's name.
4. Click the **Delete** button to remove the driver.

Tip

Instead of deleting a driver, you can disable it in the **Server Configuration** tab. For more details, see [Server configuration administration](#).

Filtering drivers

You can filter drivers using three options:

- Search
- Custom/Default drivers
- Driver State (`Enabled` / `Disabled` / `All`)

To access filter options, click the filter button:

+ ADD
REFRESH
DELETE

Search for the driver name...

Enabled
Custom drivers

Driver parameters

Parameter	Description
Driver Name	The name you assign to the driver. This can be any descriptive name that helps you identify the driver among others.
Driver Type	Specifies the type of the driver used for connecting to the database. Users typically select the appropriate driver type based on the database being connected to. For unsupported databases, the Generic type is often used. For more information see section below.
Class Name	The fully qualified name of the Java class for the JDBC driver. This is the main class that Java uses to interact with the database. It can often be found in the driver's documentation or by inspecting the <code>.jar</code> file.
URL Template	A template for the database connection URL. If left empty, you must specify the JDBC URL manually for each connection. It's recommended to provide a template to simplify connection setup. For more information see section below.
Default Port	The port number typically used by the database. This is optional and can be left blank if the port is non-standard or varies between installations.
Default Database	The default database name to use for connections. This can simplify the setup process by pre-filling the database field when creating connections. Leave blank if no default database is applicable.

Parameter	Description
Description	A brief explanation or notes about the driver. This is useful for adding contextual information, such as the database type, special requirements, or any custom configurations.

Adding a new driver through server configuration

This section provides detailed instructions on how to add a new database driver through server configuration.

Tip

To add a custom driver to the CloudBeaver Docker image, you must build from the source. Modify the source code to include your driver, rebuild the project, and then create a new Docker image using the Docker packager. Changes cannot be made directly in the Docker container.

Driver dependencies

Note

These operations require server access and should be performed by users with appropriate permissions.

Driver dependencies are managed within the server configuration. To add a new driver:

1. Navigate to the `server/drivers` directory.
2. Create a new folder for the driver you wish to add.
3. Copy the POM (Project Object Model) file from an existing driver configuration into the new folder.
4. Modify the POM file to reflect the new driver's module name and dependencies.

PostgreSQL driver configuration example:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <artifactId>drivers.postgresql</artifactId>
  <version>1.0.0</version>
```

```

<parent>
  <groupId>io.cloudbeaver</groupId>
  <artifactId>drivers</artifactId>
  <version>1.0.0</version>
  <relativePath>../</relativePath>
</parent>

<properties>
  <deps.output.dir>postgresql</deps.output.dir>
</properties>

<dependencies>
  <dependency>
    <groupId>org.postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <version>42.2.20</version>
  </dependency>
  <dependency>
    <groupId>net.postgis</groupId>
    <artifactId>postgis-jdbc</artifactId>
    <version>2.5.0</version>
  </dependency>
</dependencies>

</project>

```

Registering the new driver in the server

To make the new driver available within the CloudBeaver platform, perform the following steps:

1. Open the file `server/bundles/io.cloudbeaver.resources.drivers.base/plugin.xml`.
2. Copy an existing driver configuration and adjust it for the new driver.

Configuration lines

- **Resource Definition:**

```
<resource name="drivers/driver-name"/>
```

This line specifies that the folder `drivers/driver-name` contains the necessary JAR files for the driver.

- **Bundle ID:**


```
<bundle id="drivers.driver-name" />
```

This line is required for enabling the driver configuration within the DBeaver platform.

- **Driver ID:**

```
<driver id="generic:driver-name" />
```

This line enables the DBeaver driver in CloudBeaver. The driver ID should match the one specified in the database-specific `plugin.xml` files on the CloudBeaver platform.

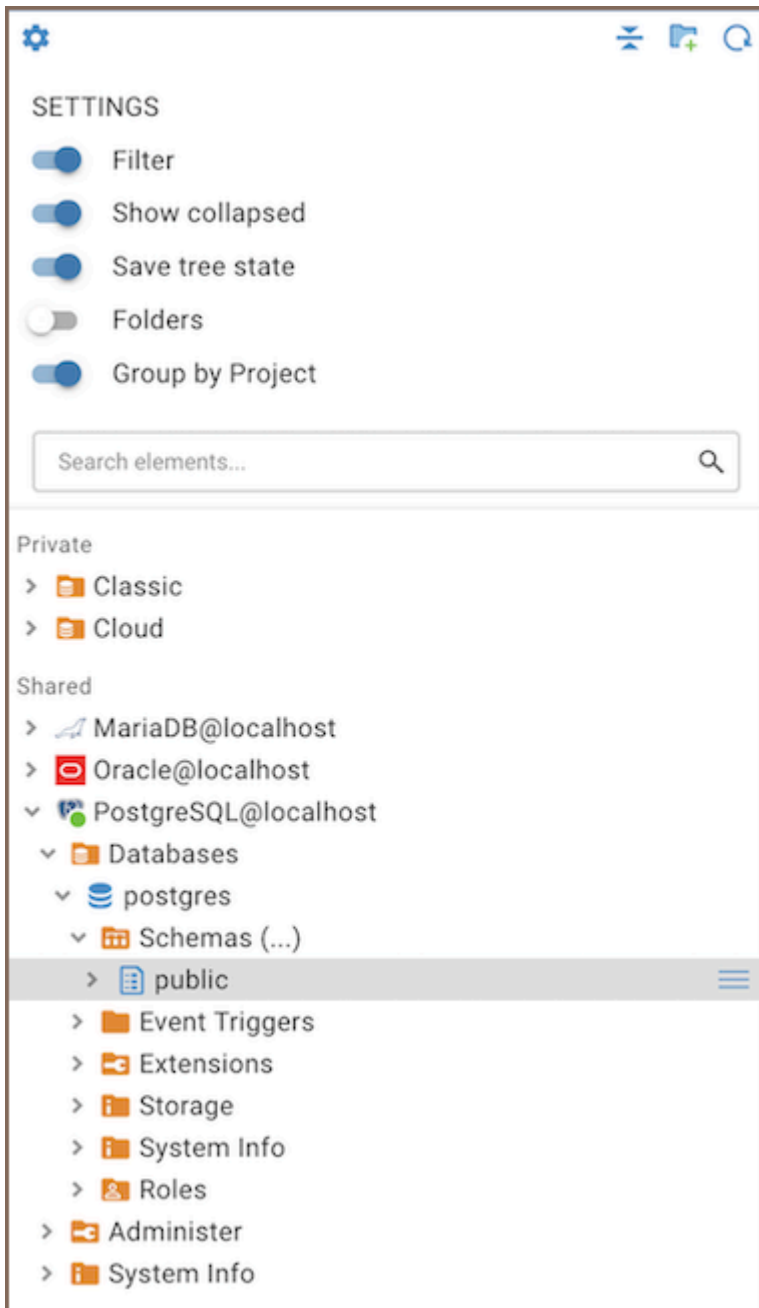
Navigation

Database Navigator

The Database Navigator is a primary interface component for managing database structures and contents. It is positioned on the left-hand side of the application window. Key elements of the Database Navigator include the Toolbar, the Settings menu, and a hierarchical Object tree.

Info

For details about the options available in the Toolbar and the Settings menu, see [the Toolbar](#) and [the Settings menu](#) guides.



Object tree

The Database Navigator displays a hierarchical tree that includes the following types of objects:

- **Folders**

- **Database connections**
 - **Database objects** (varying by database type, such as Tables, Views, Indexes, etc.)

Tip

You can store Database connections in folders. For more information, see [Database Navigator folders](#).

Context menu

To access 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. You can also access the context menu by right-clicking on the object in the Database Navigator.

The table below summarizes the available context menu items for each object type within the tree. The availability of these items depends on the specific database and object types.

Menu item	Description
Open	Opens an object in a separate Document viewer.
Filter	Allows inclusion or exclusion of database objects based on specified criteria. You can apply masks (%, _ and *) to define these criteria.
SQL Editor	Opens a new SQL Editor for the connection. For more information, see SQL Editor .
Connection view	Changes the view of the Database Navigator. For more information, see Simple and advanced view .
Edit Connection	Opens the Connection Configuration window that allows the configuration of connection settings. For more information, see Connection Configuration .
Disconnect	Disconnects from the database.
Disconnect all	Disconnects all database connections.
Delete	Deletes an object. The Delete action is irreversible and will remove the object from both the naviga

Menu item	Description
	tor tree and the database or file system permanently.
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: <code>SELECT</code> , <code>INSERT</code> , <code>UPDATE</code> , <code>DELETE</code> , <code>MERGE</code> , <code>DDL</code> . Clicking one of the items (e.g., <code>INSERT</code>) generates a relevant query in a separate window.
Export	Opens the Export window for table data export. For more information, see Data Export .
Describe object with AI	Describe a selected table, view, or other object using AI. For more information, see Describe database object .

Shortcuts

Shortcut	Description
<code>Ctrl + F</code>	Enable filtering
<code>Ctrl + Shift + ,</code>	Link with editor
<code>Ctrl + Shift + /</code>	Collapse all

Configuring Database Navigator in the configuration file

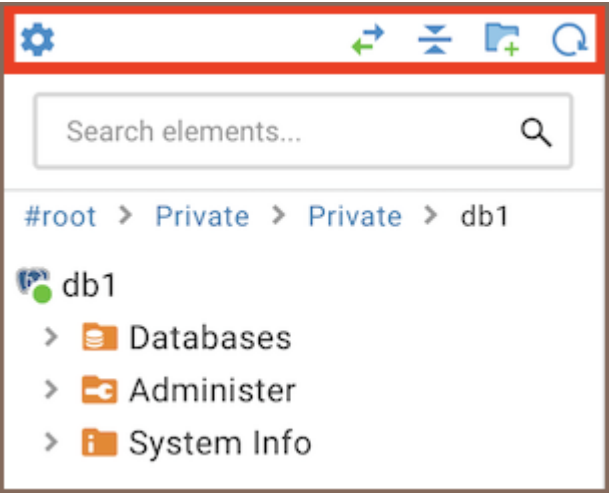
Administrators can customize the Database Navigator settings in the `app.defaultNavigatorSettings` section of the configuration file. These settings control the visibility and organization of database objects within the navigator.

Info

For more information on how to adjust Database Navigator settings through the configuration file, see the [Server configuration](#).

Toolbar

The **Database Navigator** contains various tools on the top toolbar.



Tool descriptions

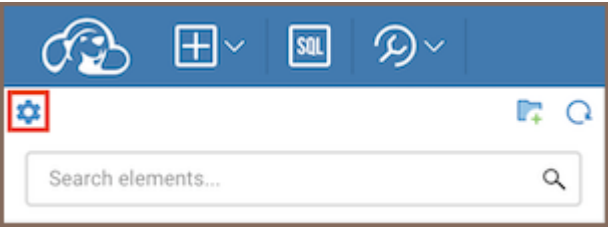
The table below contains a detailed description of each tool.

Tool	Icon	Description
Settings		Opens the settings menu with additional tools. For more information, see Settings menu .
Link with editor		Synchronises the active SQL Editor, Data Editor or Metadata Editor with the element in the Database Navigator. It becomes visible when an Editor for a connection is opened.
Collapse all		Collapses all expanded objects in the Database Navigator. It becomes visible when at least one object is expanded.

Tool	Icon	Description
New folder		Creates a folder in the Navigator or tree to group connections. For more information, see Folders in Database Navigator .
Refresh		Refreshes all objects in the Database Navigator.

Settings menu

The Database Navigator's Settings menu lets you customize and improve your navigation. To open it, click **Settings** button in the upper-left corner of the **Database Navigator**.



Available tools

The Settings menu includes the following options:



Option	Description
Filter	Search for objects in the Database Navigator. Enable the filter to show a search field, then type an object name to find matching items. Searches only visible objects.
Show collapsed	Expands the Filter function to include collapsed folders, as long as they've been previously expanded.
Save tree state	Keeps the Database Navigator's state after refreshing the browser. Enabled by default.

Option	Description
Folders	Displays only one level of folders. Shows the full path as breadcrumbs. When you double-click an object, the path appears in the Navigator tree.
Group by Project	Hides project names from the Database Navigator.
Show objects description	Shows schema and table comments in the Database Navigator, adding context for database objects.

Folders

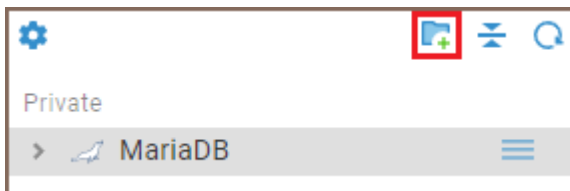
Users can conveniently group and store connections using folders in the Database Navigator. Folders can be created by any logged-in user in the Private project, and by administrators in the Shared project.

Managing connections and folders

Folder creation

To create a folder in the Database Navigator, follow these steps:

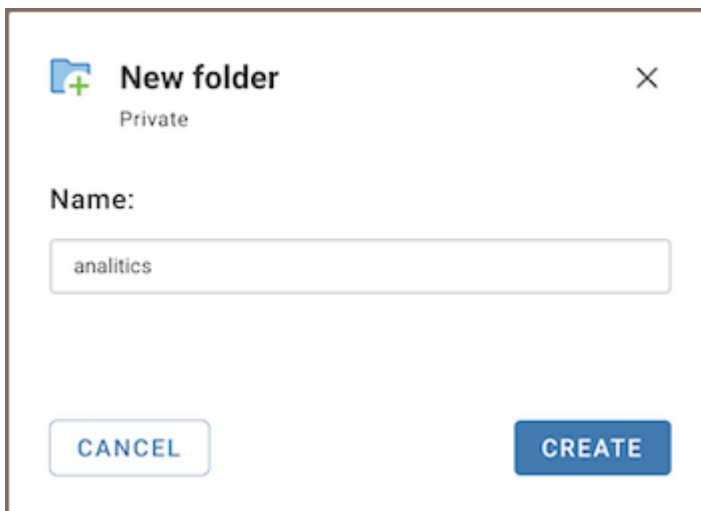
1. Press the **New folder** button on the top Navigator toolbar.



2. Enter a folder name in the popup dialog box and press the **Create** button.

Note

Folder names may not contain the following symbols `?, U, [, ^, /, :, ', ", \, <, >, |, ?, ], +, *` and cannot start with a `.`. Each folder name must be unique within the project. If a user attempts to use unsupported characters or a non-unique name, an error message will prompt a name change.



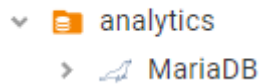
Tip

If the option to enable private connections is activated in the [settings](#), you can select whether folders are shared or private during creation.

Adding connections to folders

To add a connection to a folder, drag and drop the connection into the folder. The connection will then appear in the Database Navigator within the folder.

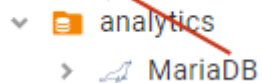
Private



Moving connections

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

Private

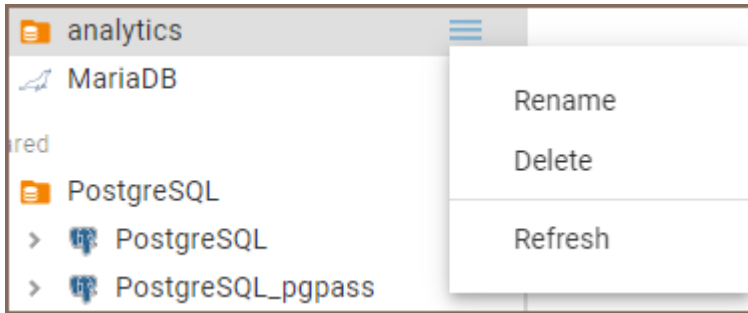


Creating and managing subfolders

To create a subfolder, drag and drop one folder into another, or click on a folder and create a new one. To move a subfolder, drag and drop it above the list of Navigator objects or into another folder.

Renaming a folder

To rename a folder, select **Rename** from the folder's context menu and change the name in the popup dialog box.



Deleting a folder

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

Note

Deleting a folder will only remove the folder and any subfolders. All connections will remain in the Database Navigator.

Refreshing folder content

To update a folder's content, select **Refresh** from the folder context menu.

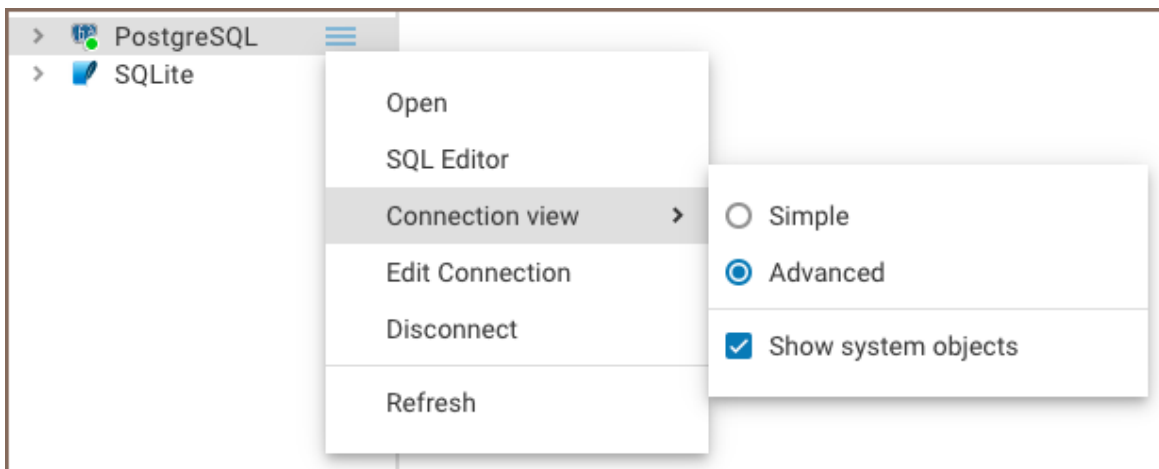
Simple and Advanced view

Connection views

Simple and Advanced Connection view defines what the **Database navigator** and the Metadata Editor structures will look like:

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

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



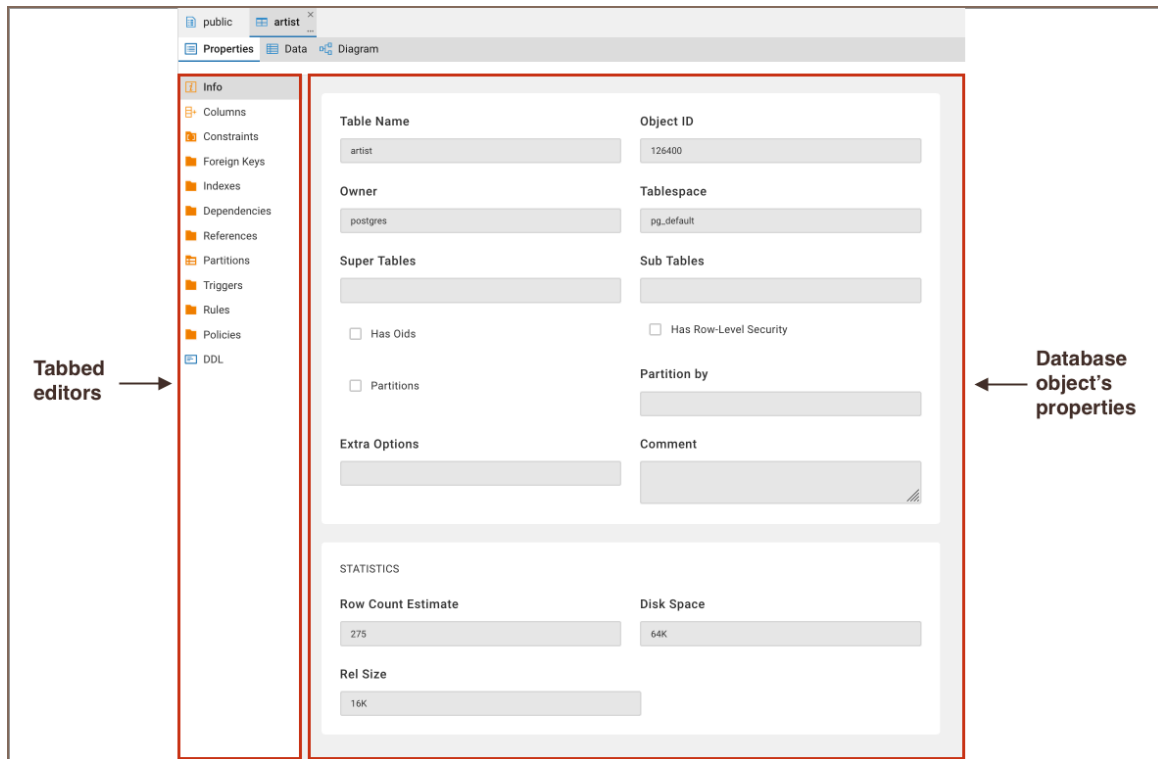
- **Show system objects** mode makes system schemas and tables available in the Database navigator and the Metadata Editor. It can be enabled in the connection context menu in the Database Navigator.

Tip

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

Properties Editor

When you open the **Properties** tab of the **Database Object Editor**, you'll see the following interface elements:



- **Database object's properties:** displays the properties of the current database object.
- **Tabbed editors:** These vary depending on your database. Depending on the database you're working with, you'll see different tabs reflecting that particular database's unique characteristics and requirements.
- The **Toolbar:** provides tools for managing objects.

Toolbar controls

To interact with the **Properties Editor** page, you can use Toolbar controls at the bottom of the editor provides the following tools:

Button	Description
	Deletes the item currently in focus.
	Saves the DDL script locally.
	Opens the DDL script in the SQL Editor .

Context menu

The **Properties** of the sub-entities have context menus that offer the same commands as the **Database Navigator**. To access these menus, right-click on the item.

DDL tab

The **DDL** tab displays the Data Definition Language (DDL) script for the selected database object. It lets you view, and edit the SQL code used to create or modify the object.

Key features:

- View the exact structure of tables, indexes, views, and other objects.
- Copy or save the script for version control or manual execution.
- Open the script in the **SQL Editor** for modification.

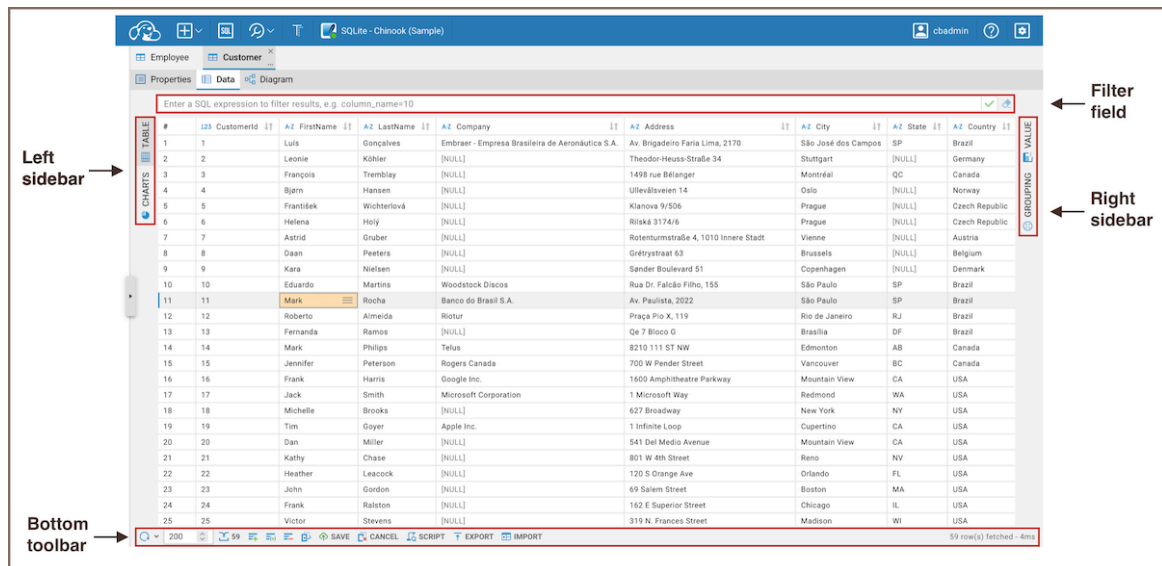
Data Editor

Data Editor

The Data editor appears:

- as the **Data** tab of the **Database Object Editor**, which is only available for tables and views;
- as the **Results** tab when you run a custom SQL query in **SQL Editor**.

The Data editor allows to view and edit data of a database table or view. The central part of the Data editor is the data grid. The editor also provides **filter field**, **bottom toolbar**, **left sidebar**, **right sidebar** and **cell context menu**:



Info

See all Data Editor **shortcuts**

Filter field

The top toolbar contains the following buttons:

Button	Name	Description
	Apply	Applies filters/sorting from filter field.
	Reset all filters/sorting	Removes all filters and orderings applied to the data in the filter field.

Left sidebar

The left sidebar contains the following tabs:

Button	Name	Description
 TABLE	Table	Switches to the grid view of data.
 CHARTS	Chart 	Switches to the Chart view .
 JSON	JSON 	Switches to the JSON view .

Right sidebar

The right sidebar contains the following tabs:

Button	Name	Description
 VALUE	Value	Opens Value panel .
 GROUPING	Grouping	Opens Grouping panel window tools.
 RESULT DETAILS	Result Details	Opens the Result Details panel . The Result Details panel is compatible with specific data bases such as BigQuery.

Button	Name	Description
 SUB COLLECTIONS	Sub Collections ★	Enables a hierarchical structure for data organization with specific databases such as Firestore.

Tip

For more details on working with GIS data in right sidebar, see [Spatial data](#)

Bottom toolbar

The bottom toolbar provides the following buttons:

Button	Name	Description
	Refresh	Refreshes the whole results set, including all items that are not visible on the screen, while its dropdown option allows to customize the refresh frequency over a specific period.
	Result-set fetch size	Displays the initial fetch number of rows in the result set on loading or refreshing.
	Calculate total row count	Calculates the total number of rows in the table.
	Add new row	Adds a new empty row below the current row.
	Duplicate current row	Copies the current rows and pastes the copy below the current row.
	Delete current row	Colors the rows in focus in red to mark them for deletion.

Button	Name	Description
	Cancel selected	Discards selected changes to the data.
 SAVE	Save	Saves all unsaved changes to the data such as adding, duplicating, deleting rows, inline editing of values.
 CANCEL	Cancel	Discards all unsaved changes to the data.
 SCRIPT	Script	Opens a window with the Preview Changes script.
 EXPORT	Export	Opens the Export wizard.
 IMPORT	Import	Opens the Import wizard.

Note

Some of these buttons may be disabled if you are using a read-only connection, connecting to a read-only database, viewing the result of a complex query (such as a join on multiple tables), or working with a table that doesn't have a primary or unique key.

Cell context menu

Every cell in the table has a context menu. Click the cell's context menu icon to open it. The menu includes the following options:

Menu item	Description
Sorting	Sorts table data by the selected column. See the Data Ordering article.
Filters	Filters data based on specified conditions. See the Data Filters article.

Menu item	Description
Edit	Opens additional editing options.
Show in value panel	Displays the selected cell's content in the value panel.
Download	Available for XML and JSON cell data types. Downloads the cell content as a file.
Download and Upload	Available for BLOB cell data types. Allows downloading and uploading cell content.

Column description

By default, CloudBeaver shows column descriptions (comments or metadata) in the data grid. If defined in the database, the description appears below each column name in the table header.

Tip

To hide descriptions, turn off **Show columns description** in [Preferences](#).

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.

Using filters

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`.

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

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

`0002E0` , then click **Filters** -> **Cell value** in the context menu and choose one of the expressions.

00034E	2017-08-04T16:52:00Z	73300
000352	2017-08-04T16:52:00Z	109500
000374	2017-08-04T16:52:00Z	136200
00044D	2017-08-04T16:52:00Z	
00044E	2017-07-17T05:39:00Z	
000480	2017-07-28T09:00:00Z	
0004E1	2017-07-31T17:34:00Z	
000511	2017-07-01T03:40:00Z	
00053F	2017-08-06T03:15:00Z	
00054E	2017-07-15T10:54:00Z	
0005E7	2017-07-03T09:25:00Z	
0005F4	2017-08-09T02:14:00Z	95400
0005FF	2017-07-18T03:27:00Z	56300

Sorting

Filters

Clipboard

Cell value

Custom

book_ref IS NULL

book_ref IS NOT NULL

Reset filters / sorting

book_ref = '00034E'

book_ref <> '00034E'

book_ref > '00034E'

book_ref < '00034E'

book_ref LIKE '%00034E %'

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.

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

Data ordering

You can order data in a column by clicking the header or using the context menu.

Order data using the column header

Click the ordering icon () in a column header.

123 actor_id 	A-Z first_name	A-Z last_name	 last_update
--	----------------	---------------	---

Tip

The icon appears when you hover over the header and becomes visible permanently after sorting is applied.

Each click changes the sort state:

State	Icon
Ascending	
Descending	
None	

Tip

To order by multiple columns, hold **Ctrl** (Windows) or **Cmd** (Mac) and click columns in the order you want to apply sorting.

Reset data ordering

To reset data ordering:

- click the ordering icon () until all sorting is removed
- open the cell context menu (three-bar icon), then select **Sorting -> Disabled**
- if multiple sortings are applied, select **Sorting -> Disable all** in the cell context menu.

Tip

To clear all filters and sorting at once, open the cell context menu and select **Reset filters/sorting**

Panels

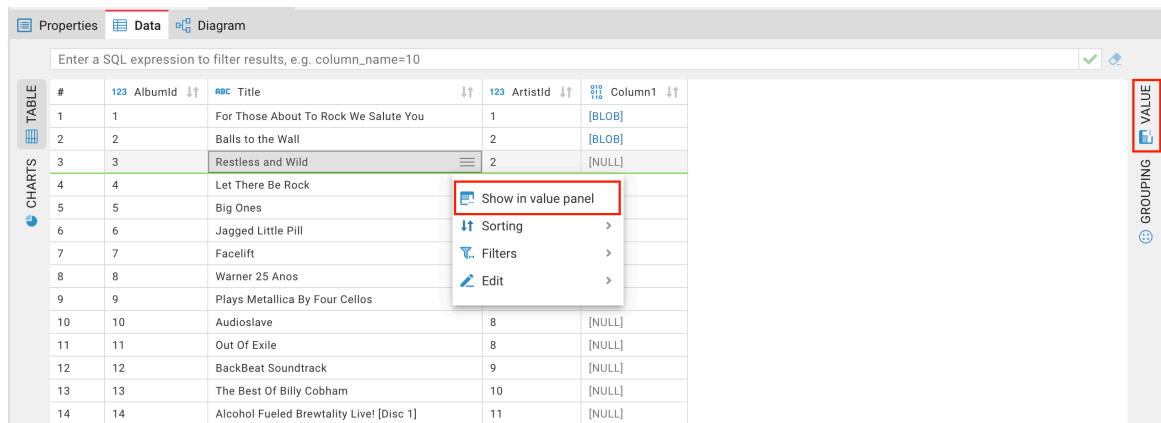
Value Panel

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.

Use value panel

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

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



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

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

Diagram

to filter results, e.g. column_name=10

ABC Title	123 ArtistId	Column1
For Those About To Rock We Salute You	1	[BLOB]
Balls to the Wall	2	[BLOB]
Restless and Wild	2	[NULL]
Let There Be Rock	1	[BLOB]
Big Ones	3	[NULL]
Jagged Little Pill	4	[NULL]
Facelift	5	[NULL]
Warner 25 Anos	6	[NULL]
Plays Metallica By Four Cellos	7	[NULL]
Audioslave	8	[NULL]
Out Of Exile	8	[NULL]

TEXT HTML XML JSON

Restless and Wild

VALUE

GROUPING

There are 3 available tabs for BLOB type of data:

- Text
- HEX
- Base64

Column1

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

IMAGE TEXT

TEXT HEX BASE64

iVB0Rw0KGgoAAAANSUhEUgAAAG4AAABuCAIAA

VALUE

GROUPING

Grouping Panel

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

Grouping panel

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

The screenshot shows the CloudBeaver interface with the 'actor' table selected. The 'Data' tab is active, displaying a table with 28 rows and 6 columns: #, actor_id, first_name, last_name, last_update, and age. A red arrow points from the 'last_update' column header in the table to the Grouping Panel on the right. The Grouping Panel is currently empty, displaying a message: 'There is no data to show. Drag and drop a column from the result viewer to group values.'

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

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

actor

Properties Data Diagram

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

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

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

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

400 200 EXPORT

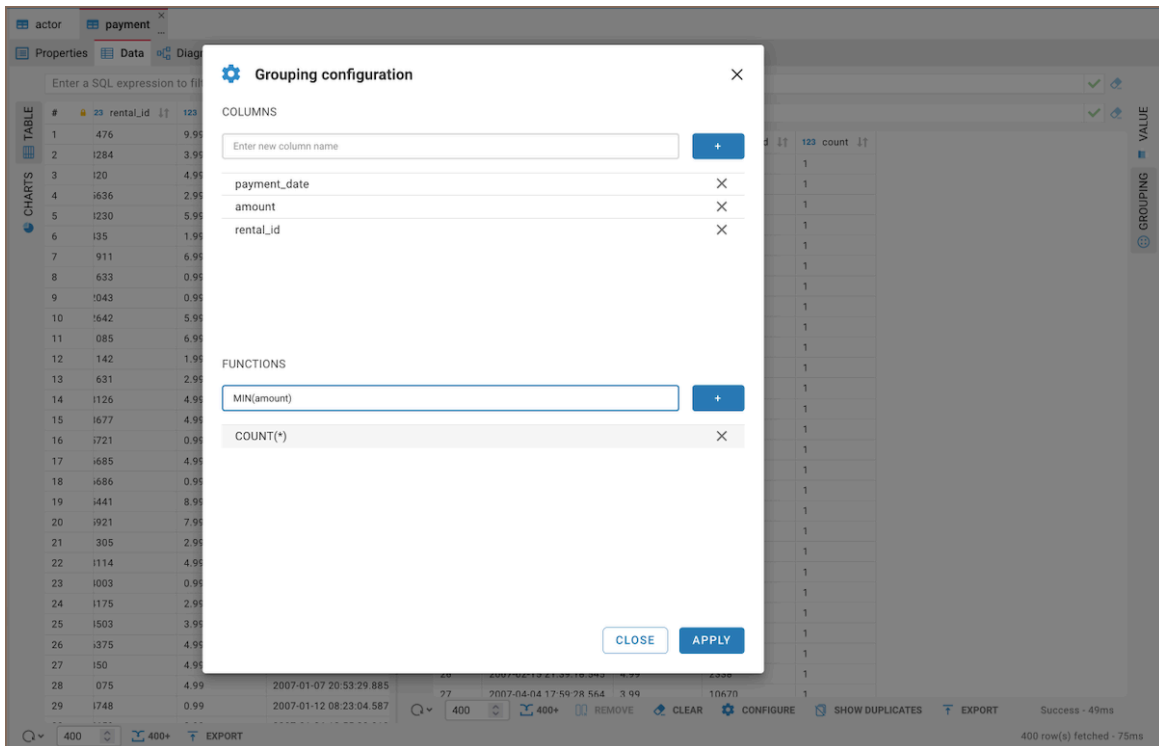
400 121 REMOVE CLEAR CONFIGURE SHOW DUPLICATES

200 row(s) fetched - 46ms

Function management

Adding a function

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



To add a function, follow these steps:

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

Note

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

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

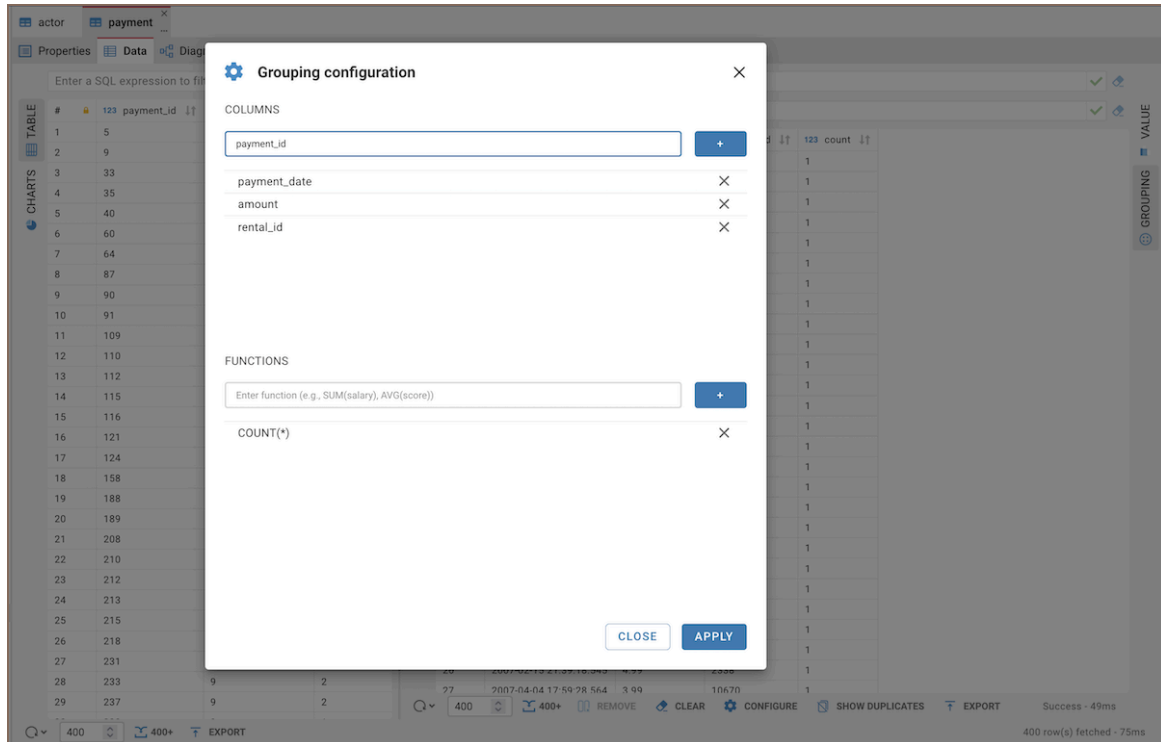
Removing a function

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

Column management

Adding a column

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



To add a column:

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

Tip

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

The screenshot shows the 'Grouping configuration' window on the left and a data table on the right. The configuration window has a 'COLUMNS' section with a text input containing 'TOTAL_PRICE / 100 * 20' and a 'FUNCTIONS' section with a text input containing 'count(*)'. The data table on the right has columns: '#', 'TOTAL_PRICE', 'TOTAL_PRICE / 100 * 20', and 'count_'. A red box highlights the first three columns, and a red arrow points from the 'TOTAL_PRICE' column to the 'TOTAL_PRICE / 100 * 20' column.

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

Removing a column

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

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

Filtering

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

Default sorting

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

Option	Icon	Description
Unsorted	↕	Leaves the data in its original order.
Ascending	↑	Sorts the grouped data in ascending order.
Descending	↓	Sorts the grouped data in descending order.

Show duplicates only

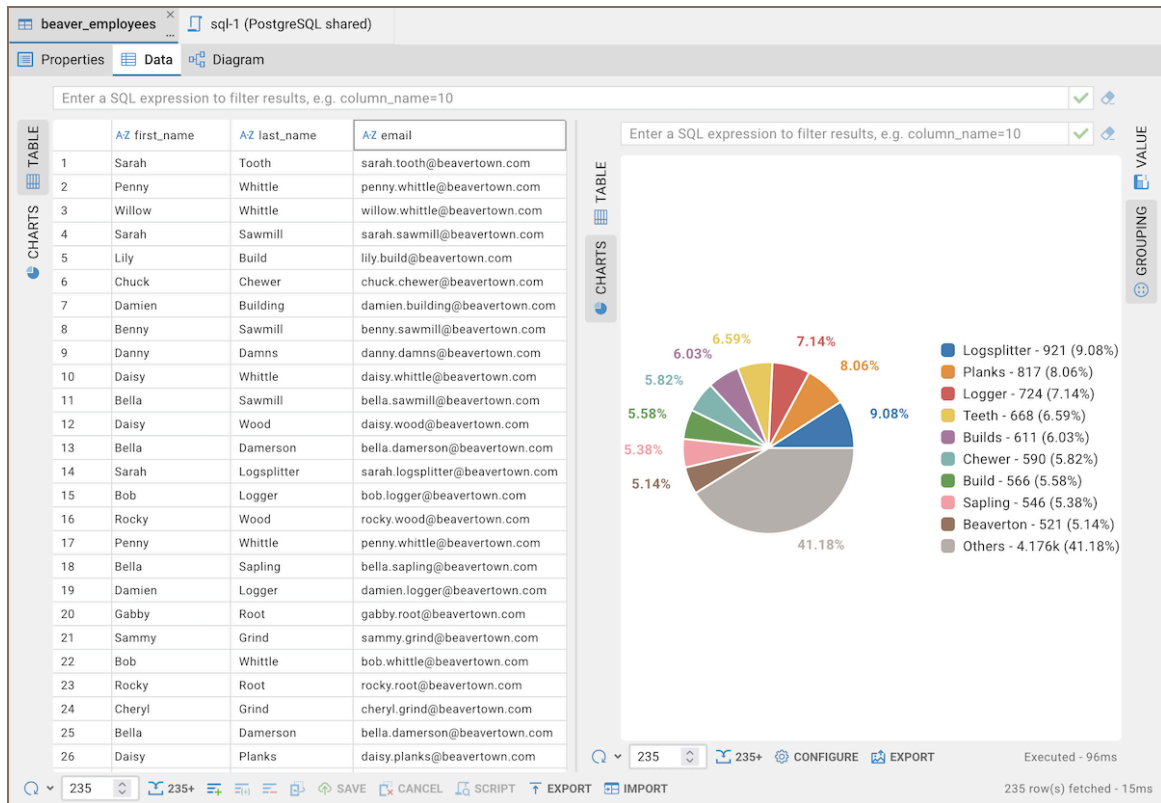
Click the **Show Duplicates** button  to filter the results and display only those rows where the `COUNT` is greater than `1`.

Charts

Note

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

You can create and manage various charts directly within the **Grouping Panel**. For details, see [Charts](#).



Export grouped data

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

Result details Panel

The Result details panel provides you with information about the execution of a `SELECT` query. To open the panel, click the **Result Details** button on the right-hand side of the **Data** tab.

Note

This panel is specific to BigQuery and provides details that help in monitoring query costs.

gcp_billing_export_v1_01336...

PropertiesDataDiagram

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

TABLE

#	ABC billing_account_id	ABC service_id	ABC service.description	ABC s
1	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
2	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
3	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
4	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
5	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
6	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
7	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
8	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
9	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
10	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
11	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
12	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
13	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
14	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
15	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
16	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
17	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
18	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
19	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
20	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
21	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
22	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
23	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
24	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F
25	013360-01AFAD-DB7DE8	5490-F7B7-8DF6	5490-F7B7-8DF6	143F

CHARTS

Name	Value
Job	job_ZHE8YIZDGz7e6Ghz-NXsuNZO8TUJ
Location	EU
Project	dbeavergcp
DML affected rows	none
Fetches from cache	false
Total rows	200
Total bytes processed	283,502,318
Transaction id	none
Total bytes billed	284,164,096
Total modified partitions	0
Total slot ms	40
Query	SELECT * FROM dbeavergcp.daily_cost_detail.gcp_billing_export_v1

200200+ADD DATASETEXPORT

200 row(s) fetched - 1250ms

Metrics

Result details panel includes the following metrics:

Name	Description
Job	The unique identifier of the job.
Location	The regional location of the data you have queried

Name	Description
	d.
Project	The project within which the query was executed.
DML affected rows	The number of rows affected by a Data Manipulation Language (DML) statement.
Fetches from cache	Indicates if the results were retrieved from the cache. <code>true</code> means no additional costs are incurred for data processing.
Total rows	The number of rows returned by the query.
Total bytes processed	The amount of data processed by the query. This is important for cost calculations as costs are incurred based on the amount of data processed. If the Total bytes processed is zero, it indicates that the query results were served from the cache, which does not incur costs.
Transaction id	The identifier for the transaction within which the query was executed, if applicable.
Total bytes billed	The total bytes billed for the query, which might be different from the bytes processed if discounts or caps are applied.
Total modified partitions	The total number of partitions that were modified as a result of the query execution.
Total slot ms	The total number of slot milliseconds consumed by the query.
Query	The SQL query statement that was executed.

Managing charts

Note

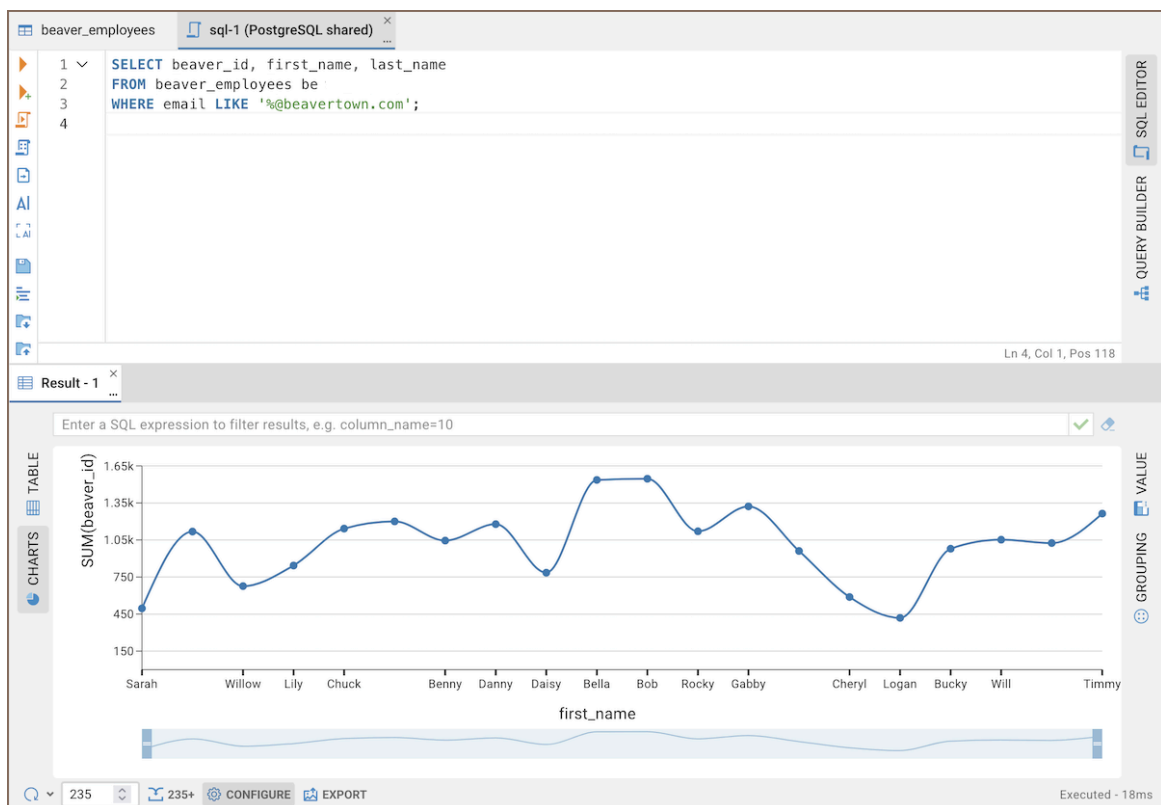
This feature is available in [Enterprise](#), [AWS](#), and [Team](#) editions only.

The default grid view of query results may look unremarkable. Use CloudBeaver's Charts feature to quickly explore patterns and trends without exporting data. You can visualize data in **Bar**, **Line**, **Pie**, or **Area** formats.

Creating Charts in the SQL Editor

To create a chart in the **SQL Editor**:

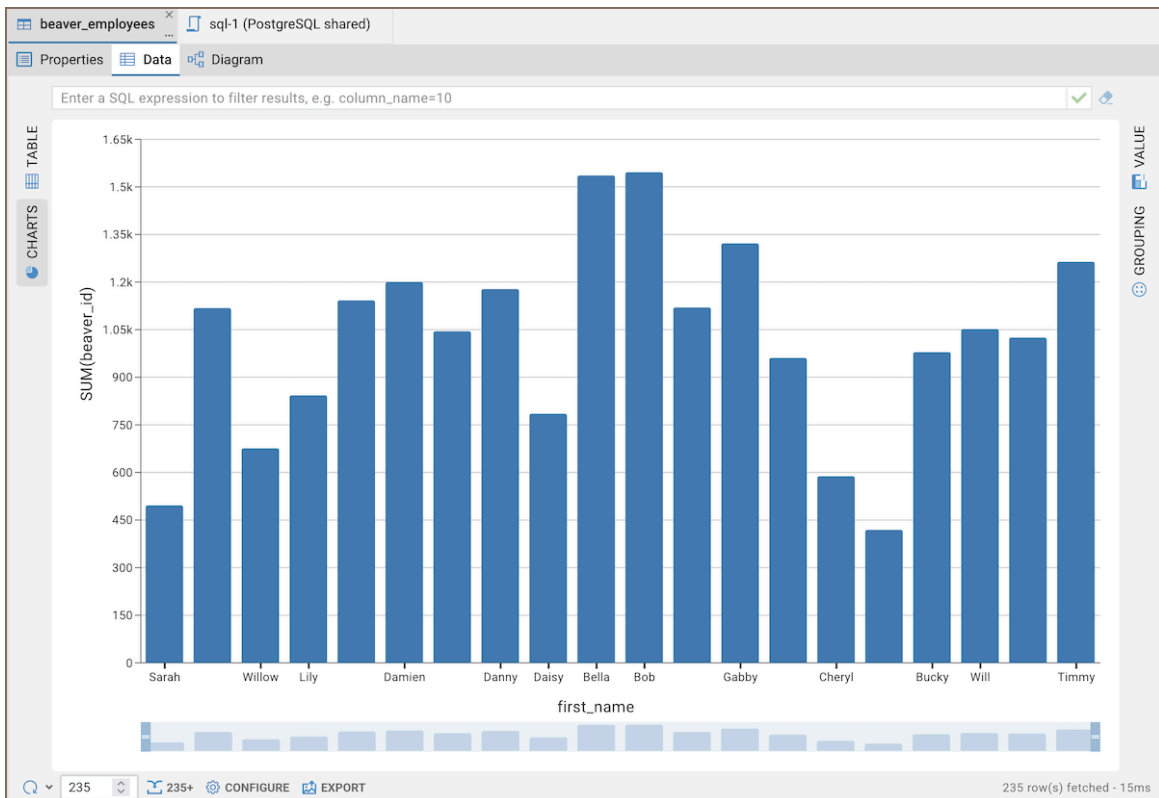
1. Run your query.
2. Click the **Charts** button ( **CHARTS**) on the left toolbar in the result area.
3. Your data will appear as a chart.



Creating Charts in the Data Editor

To visualize data from Data Editor:

1. (Optional) Sort and filter your data using the **filters**.
2. Click the **Charts** button ( **CHARTS**) in the **Data Editor**'s left toolbar.
3. Your data will appear as a chart.



Note

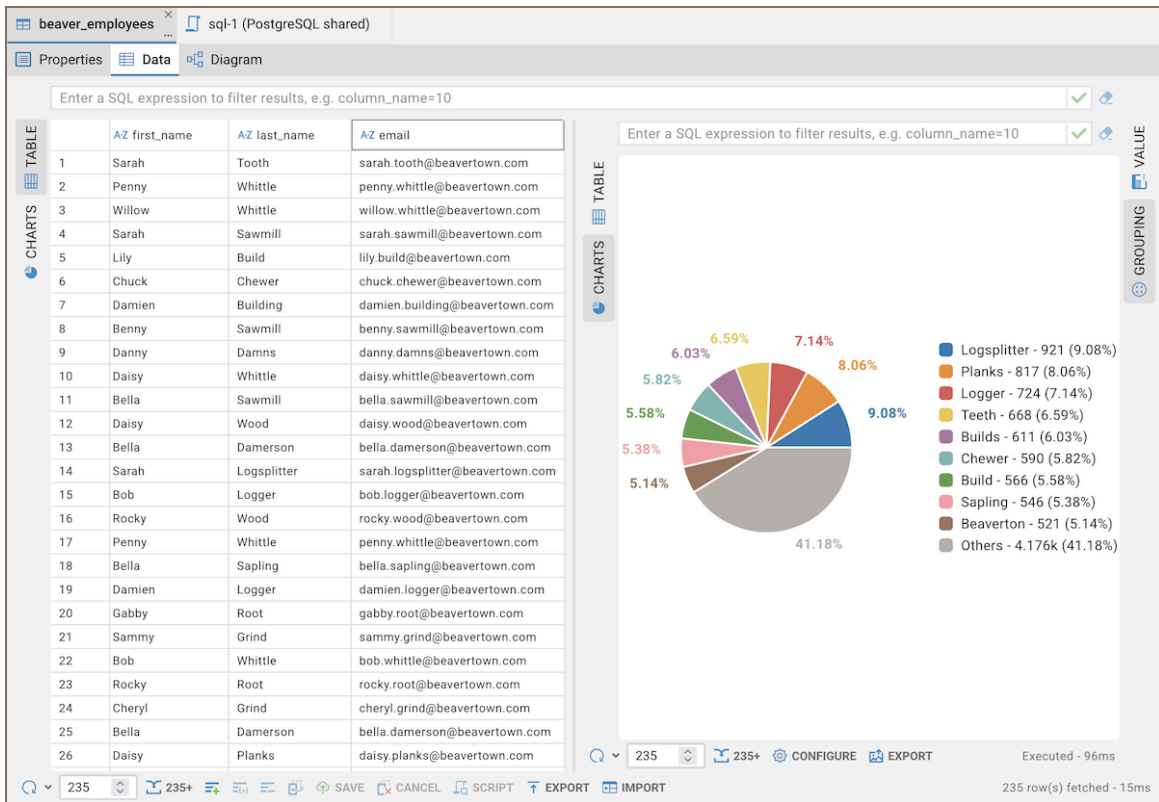
By default, the data for Y-axis is taken from the first column of the table containing numeric values.

Creating Charts in the Grouping Panel

To create charts from grouped data:

1. Group your data in the **Grouping Panel**.

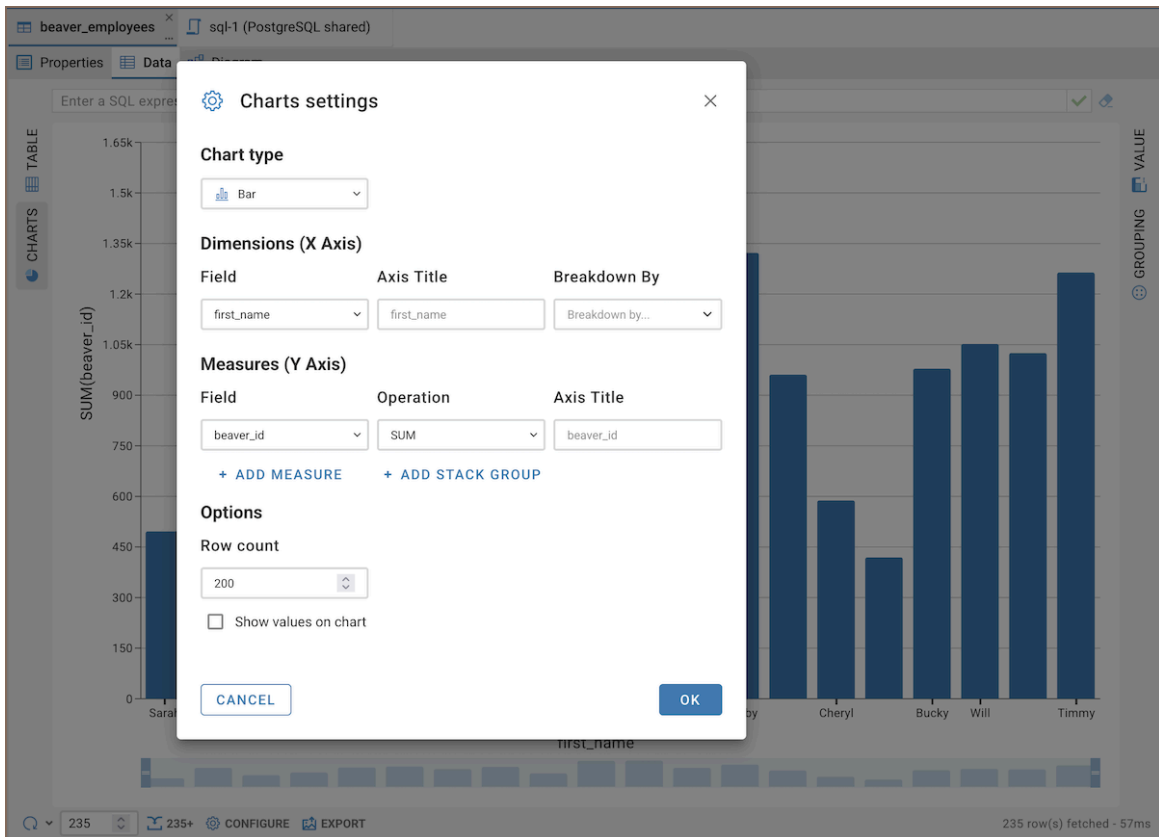
- Click the **Charts** button ( **CHARTS**).
- Your data will appear as a chart.



Editing chart settings

To edit your chart:

- Click the **Configure** button in the bottom menu.
- Adjust settings in the dialog box that appears.



Setting Axis X

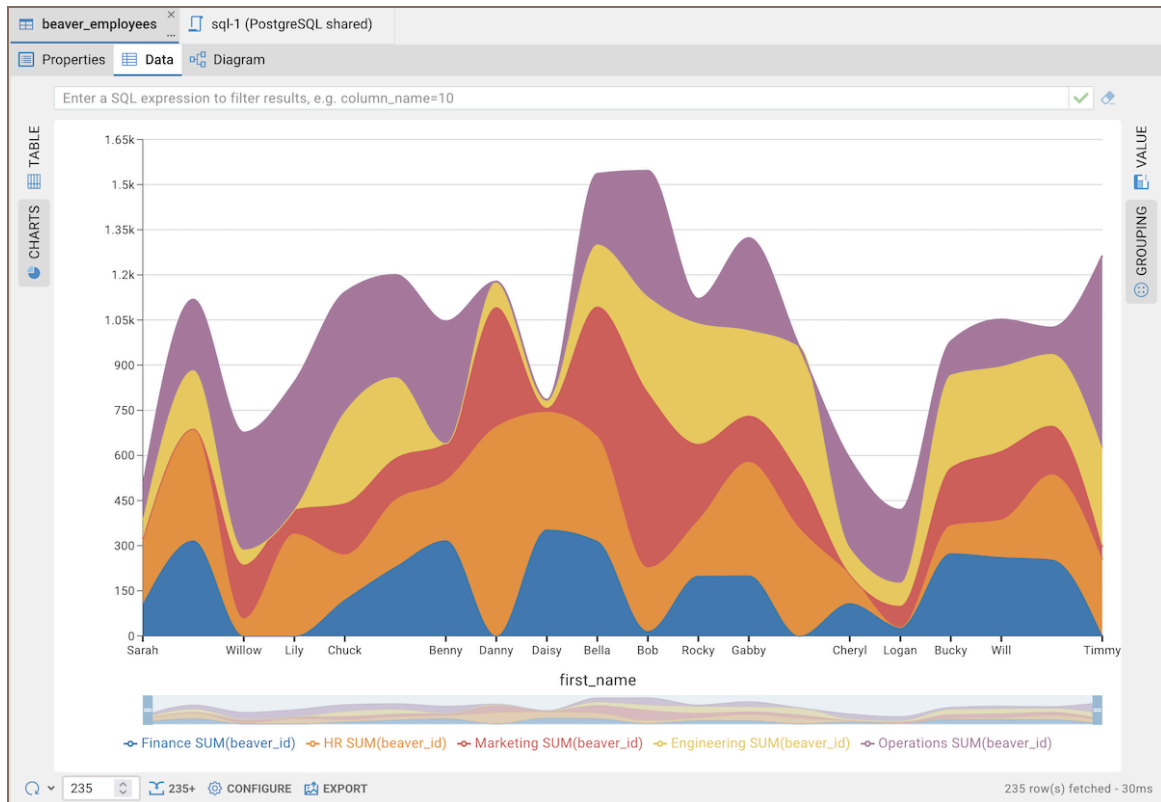
- from the **Field** dropdown, select a column for the X-axis
- name the X-axis in the **Axis title** field
- optionally, use **Breakdown by** to group results by another column

Setting Axis Y

- choose a column for the Y-axis from the **Field** dropdown
- select an **Operation** to define how values are aggregated (e.g., `SUM`, `AVG`, `COUNT`, `MIN`, `MAX`, `FIRST`, `LAST`)
- label the Y-axis in the **Axis title** field
- click **+ Add measure** to include additional measures

Using Breakdown by

The **Breakdown by** option lets you split the chart into multiple series based on the selected column. For example, if you choose `last_name`, the chart will display a separate line or area for each unique last name.



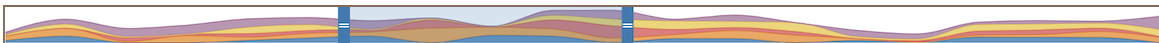
When you hover over the chart, a tooltip appears showing values for each series in that point.

Note

The **Breakdown by** option isn't available for **Pie** charts.

Adjust visible data range

Use the slider below the chart to control which part of the data is displayed. Drag the handles on both sides to zoom in or out on a specific data range.



Tip

This is useful when working with large datasets – you can focus on a smaller section without reloading or filtering data.

Setting other options

You can also set the following chart options:

Setting	Description
Chart type	Select visualization type: Bar , Pie , Line , or Area
Field	Choose a column for the X or Y axis
Axis title	Enter a custom label for the axis
Breakdown by	Group data by an additional column (creates separate series for each value; not available for Pie charts)
Operation	Select a function for data aggregation on the Y axis
	AVG - Calculates the average of the values
	SUM - Sums up all the values
	FIRST - Displays the first value
	LAST - Shows the last value
	COUNT - Counts all the values
	MIN - Finds the minimum value
	MAX - Identifies the maximum value
Add measure	Add another Y -axis field to display multiple measures on one chart

Setting	Description
Add stack group	Combine selected measures into a single stacked group. Each group appears as one stacked bar or a rea segment.
Row count	Limit the number of rows to include in the chart
Show values on chart	Display numeric values directly on the chart bars or points

Export chart image

You can export your chart as an image file.

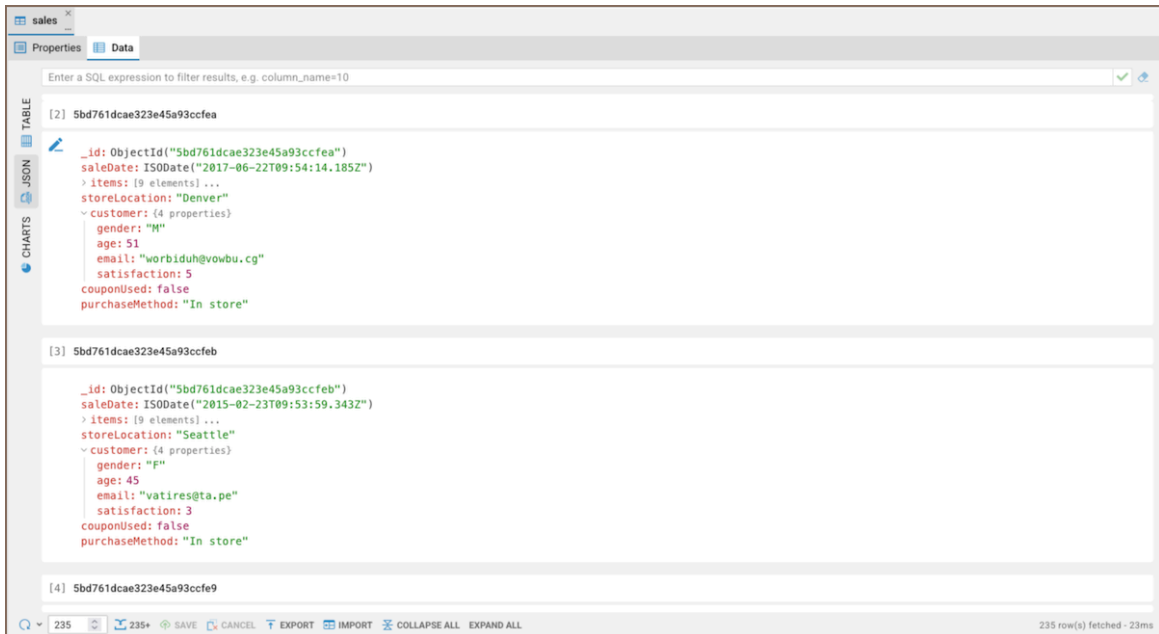
1. In the **Charts** window, click **Export** button ( **EXPORT**)
2. In the **Export image** dialog, enter a name for your file
3. Choose a **File format** (**SVG** or **PNG**)
4. (Optional) Enable **Transparent background** if you don't want a white background
5. Click **Export** to download the image

JSON and Document view

Note

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

CloudBeaver lets you view and edit `JSON` documents in databases such as **MongoDB**, **DynamoDB**, and other **NoSQL** databases.



Viewing documents

To work with `JSON` data, switch to the **JSON** tab available on the left side of the [Data Editor](#).

Info

You can collapse or expand `JSON` entries to focus on specific parts of the document. Click the arrow button () next to a key to fold or unfold that section.

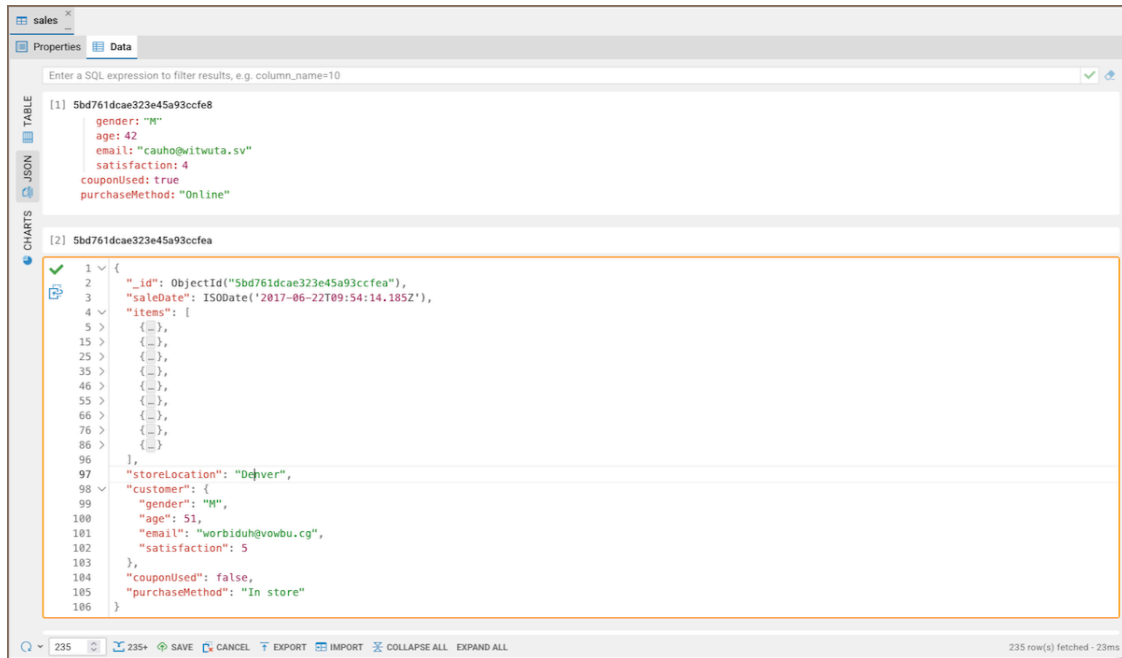
Editing documents

To edit a document:

1. Click the **Pencil** button () in the left panel.
2. The document switches to **editing mode**.
3. Click **Apply** () or **Cancel** () to save or discard your changes.

Info

If a document contains unsaved changes, it's highlighted with an orange border.



The screenshot shows a document editor interface. On the left, there is a sidebar with icons for 'TABLE', 'JSON', and 'CHARTS'. The main area displays a document with two sections. The first section, labeled '[1] 5bd761dcae323e45a93ccfe8', contains a JSON object with fields: 'gender: "M"', 'age: 42', 'email: "cauho@witwuta.sv"', 'satisfaction: 4', 'couponUsed: true', and 'purchaseMethod: "Online"'. The second section, labeled '[2] 5bd761dcae323e45a93ccfea', contains a JSON object with fields: '_id: ObjectId("5bd761dcae323e45a93ccfea")', 'saleDate: ISODate("2017-06-22T09:54:14.185Z")', 'items: [{...}]', 'storeLocation: "Dohver"', 'customer: { "gender: "M", "age: 51, "email: "worbiduh@vowbu.cg", "satisfaction: 5', 'couponUsed: false, and 'purchaseMethod: "In store"'. The second section is highlighted with an orange border, indicating unsaved changes. At the bottom, there is a status bar showing '235' and '235 row(s) fetched - 23ms'.

Spatial GIS data

CloudBeaver supports the visualization and management of spatial data. Spatial data, which includes geographic locations and geometric shapes.

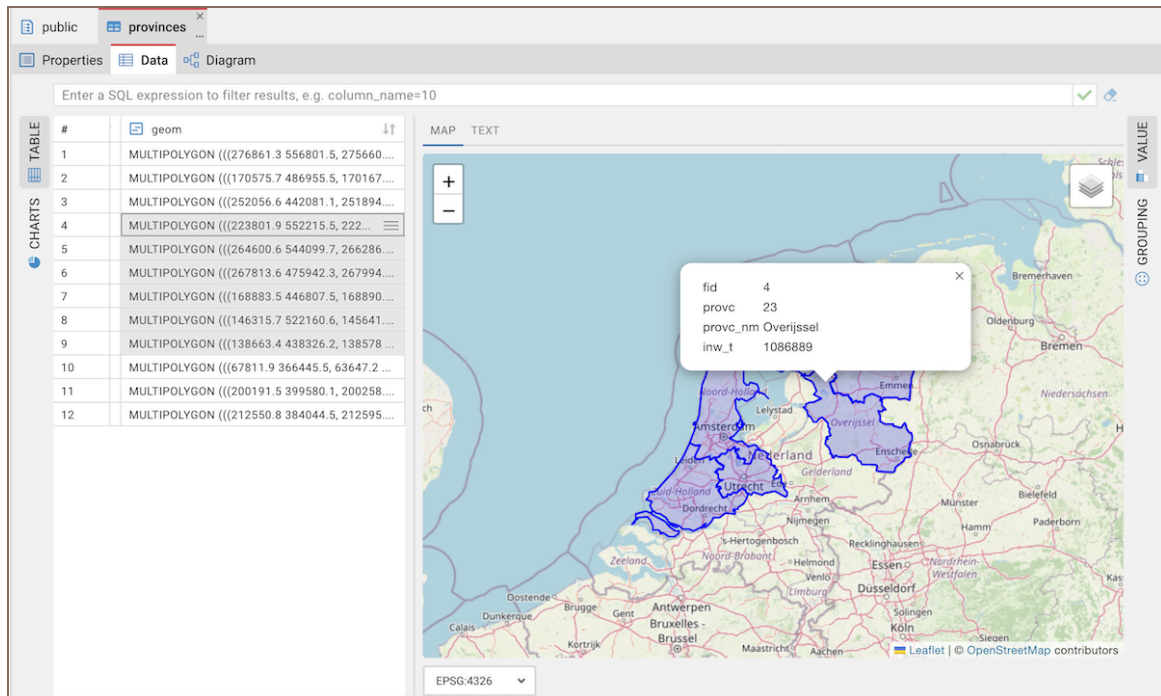
Spatial data, often represented as either geometric or geographical values, can be visualized on a map or graph. A geometric object is typically composed of a sequence of points that define its shape. For a more comprehensive understanding of spatial data, you can refer to this [detailed explanation](#).

CloudBeaver's support of spatial data covers the following databases:

- PostgreSQL (PostGIS)
- Greenplum ★
- CockroachDB ★
- MySQL
- MariaDB
- SQLite (GeoPackage)
- H2GIS ★
- SAP HANA ★
- DuckDB
- Redshift ★
- Exasol ★
- AltiBase ★
- BigQuery ★
- Oracle ★
- SQL Server ★
- Snowflake ★
- AlloyDB ★

Spatial data viewer

To access the spatial data viewer, select one or more rows containing GIS data from your table. After selecting the rows, open the Value Panel to display the selected spatial data on a map. For more information, see [Value Panel](#).



Abilities of the Spatial data viewer

- **Zooming:** Allows you to zoom in and out to tailor the map view, using either the mouse wheel or the on-screen icons.
- **Map display options:** You can choose between street or topography map view.
- **Data interaction:** When you click on a spatial object on the map, CloudBeaver displays associated information from every other column in the corresponding row.
- **Coordinate Reference System (CRS) selection:** A dropdown menu in the viewer allows you to select the appropriate CRS for your data. For more information about CRS, see [Spatial reference system](#).

SQL Editor

SQL Editor

With the SQL Editor in CloudBeaver, you can write and execute multiple SQL scripts within a single database connection, save them as files, and reuse them later.

Getting started

To start working with the SQL Editor, you have two options:

1. **Create a new SQL script:**

- click **Open SQL Editor** button in the top toolbar.

Info

For more information about toolbar, see [Toolbar in CloudBeaver](#).

2. **Open recent SQL script:**

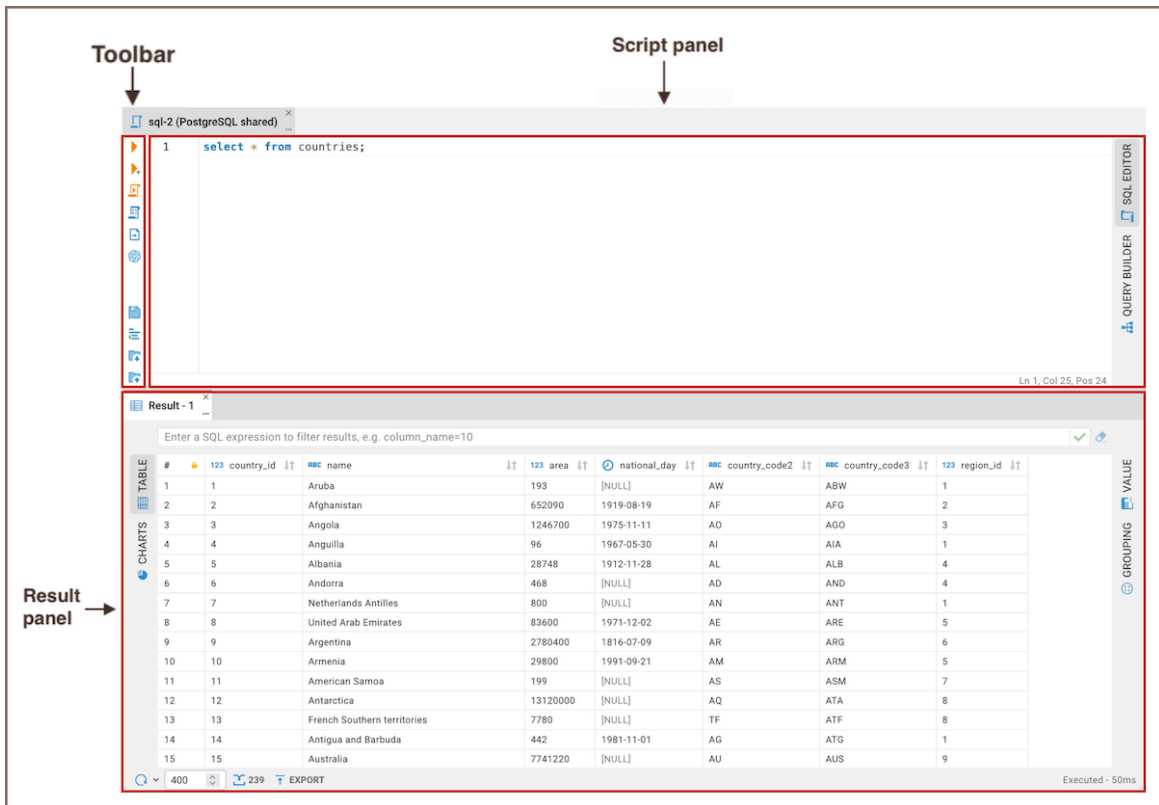
- Select **Tools -> Show scripts** from the top toolbar.
- A **Resource manager** window appears. Click any script to open it in a new tab. For more information see [Resource manager](#).

Note

A connection is created when you open the first SQL editor tab. Additional tabs for the same database reuse that connection.

SQL Editor overview

The SQL Editor contains the Script panel, the Result panel, and the Toolbar.



Script panel

The Script Panel is the primary area where you can write, edit, and manage your SQL scripts. It provides basic text editing features with the added benefits of specialized functionalities tailored for SQL development:

Results Panel

The results panel displays tabs with results in various formats. The tabs resulting from script execution represent instances of the **Data Editor**. You can create, edit and execute SQL scripts in the script panel and then see the results in the result tabs.

Toolbar

The Toolbar in the SQL Editor is equipped with various buttons:

Button name	Icon	Description	Additional resource
Execute SQL statement		Executes the SQL statement.	
Execute SQL statement in new tab		Executes the SQL statement in a new tab.	
Execute SQL script		Executes all SQL statements in the script panel.	
Explain execution plan		Provides a detailed explanation of the SQL execution plan.	Query Execution Plan
Show server output		Displays the server-side output, specific to the database being used.	Server Output
AI prompt 		Generates SQL code based on natural language prompts.	AI Smart Assistance
Save as script		Saves the current SQL script to a file.	Saving Scripts
Format SQL script		Formats the SQL script.	Formatting Guide
Download SQL script		Downloads the SQL script to the local machine.	
Upload SQL script		Uploads a SQL script from your local machine to the SQL Editor.	

Active database

You can change the connection linked to the current SQL editor or switch the active database/schema from the **main toolbar**, without losing your SQL query.

- To change the connection, click **Active datasource**, then select the required connection from the connections tree.



- To change the active database or schema, click **Active catalog/schema**, then select the required schema from the dropdown list.



Tip

To quickly link the SQL editor to the connection currently in focus in the **Database Navigator** (any object in the connection-table, folder, etc.), click **Link with editor**  or press `CTRL + SHIFT + ,` (`CMD + SHIFT + ,` for macOS).

Query execution

Statement execution

Place the cursor on the line with the statement or select part of the script to execute the statement.

Click on the **Execute SQL statement** button () in the left toolbar or use the **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.

The screenshot shows a database management tool interface. The top panel displays a script with two SQL statements: `select * from Album;` and `select * from Customer;`. The bottom panel shows the results of the second statement in a table format. The table has columns: #, CustomerId, FirstName, LastName, Company, Address, City, and VALUE. The results are as follows:

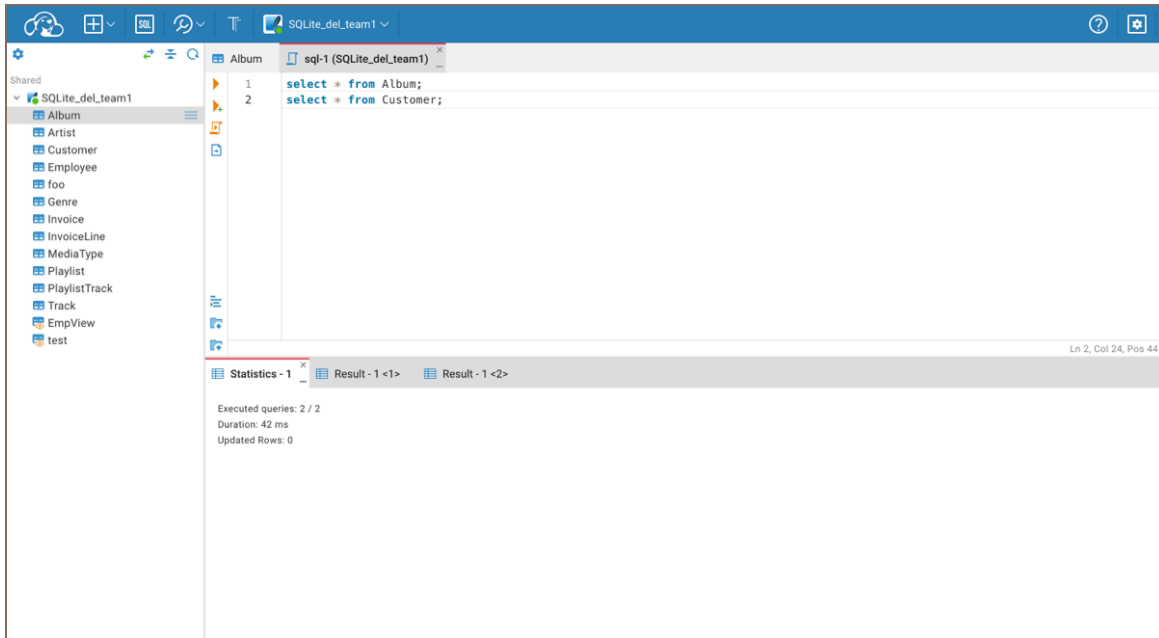
#	CustomerId	FirstName	LastName	Company	Address	City	VALUE
1	1	Luis	Gonçalves	Embraer - Empresa Brasileira de Aeronáutica S.A.	Av. Brigadeiro Faria Lima, 2170	São José dos Campos	
2	2	Leonie	Köhler	[NULL]	Theodor-Heuss-Straße 34	Stuttgart	
3	3	François	Tremblay	[NULL]	1498 rue Bélanger	Montréal	
4	4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

Tip

To view results in a separate tab, click **Execute SQL statement in a new tab** () instead. This opens the result in a new tab below the editor.

Script execution

Click on the **Execute SQL script** button () in the left toolbar or use the **shortcut** to execute the script. The summary result will be shown in the **Statistics** tab, and results will be shown in separate **Result** tabs.



Execution plan

Click **Explain execution plan** button () or use the **shortcut** to view the query plan if supported by the driver.

Info

See [Query execution plan](#) for details.

SQL Editor features

SQL assist and auto-complete

SQL Assist suggests SQL keywords, object names, and code snippets as you type.

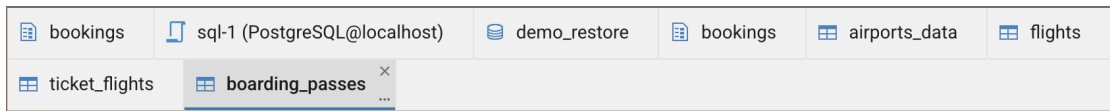
Info

Read more in [SQL Assist and auto-complete](#).

Display multiple tab rows

When many tabs are open, show them in multiple rows instead of scrolling:

1. Go to **Profile -> Preferences -> Interface**
2. Turn on the **Multiple rows for tabs** checkbox



Tip

Use this to avoid horizontal scrolling when many tabs are open.

Resource manager

To manage SQL scripts on the server file system, use the [Resource Manager](#).

Auto and Manual commit

By default, all database connections in the SQL Editor operate in **Auto-commit** mode.

Info

See more information on [Auto and Manual commit modes](#).

SQL Editor preferences

To configure the SQL Editor, navigate to **Settings -> Preferences** or **Settings -> Administration -> Settings**.

Setting	Description
Insert table aliases (in FROM clause)	Controls how aliases are inserted. Options:
	Disable: No aliases will be automatically inserted.
	my_table mt: Inserts the alias <code>mt</code> for the table <code>my_table</code> .
	my_table AS mt: Inserts <code>AS mt</code> explicitly as the alias.

Setting	Description
Use long object names (schema/catalog)	Includes schema or catalog details in table names for clarity.

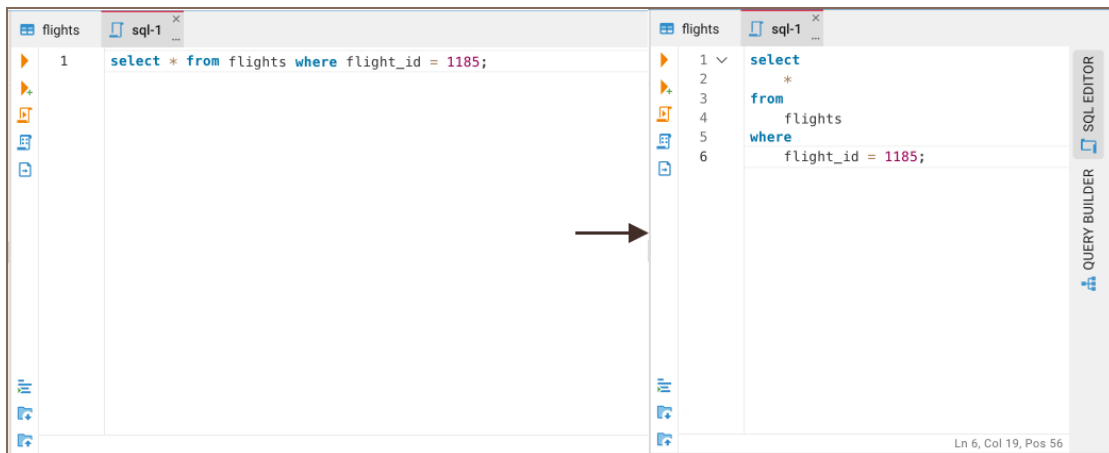
Shortcuts

You can use keyboard shortcuts to speed up your work in the SQL Editor. See the full list in the [Shortcuts](#) article.

SQL formatting

To format SQL text:

1. Select the SQL text you wish to format.
2. Press `Ctrl+Shift+F` or click on the **Format SQL Script** button  located on the left toolbar.



Commenting SQL lines

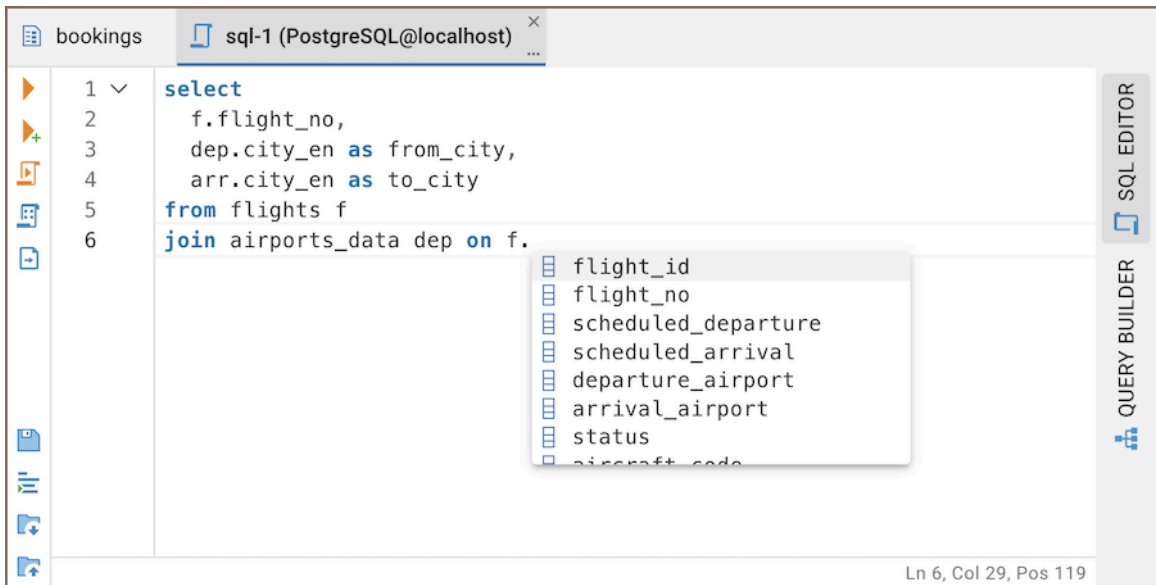
To manage comments in SQL scripts:

- **To comment out an SQL line:**
 - a. Place your cursor on the line you want to comment.
 - b. Press `Ctrl+/*` or `CMD+/*` for macOS.
- **To uncomment a commented line:**
 - a. Place your cursor on the commented line.
 - b. Press `Ctrl+/*` (`CMD+/*` for macOS) again to remove the commenting syntax.

SQL assist and auto complete

CloudBeaver provides the SQL assist feature, which offers auto-completion of database object names, SQL commands, and other keywords in queries.

When you start typing an SQL code, CloudBeaver suggests auto-complete options based on the context.



Completion engines

CloudBeaver offers two SQL completion engines:

- **Semantic (default)** – provides the most accurate and context-aware suggestions by analyzing the entire SQL structure and lexical scopes. It improves completion accuracy but may not work correctly with certain database-specific syntax.
- **Legacy** – a more straightforward completion engine that suggests SQL elements based only on their position in the query. This can sometimes lead to incorrect or incomplete suggestions, especially in subqueries.

Note

The **Semantic** engine is actively being improved, but if you encounter issues, switching to **Legacy** may provide more stable behavior. We still recommend using **Semantic** for the best overall experience.

Change the completion engine

To switch between engines:

- **Profile -> Preferences -> SQL Editor -> Completion engine**
- (For administrators) **Settings -> Administration -> Settings -> SQL Editor -> Completion engine**

Query Execution Plan

If your database driver supports it, you can view the execution plan for a query:

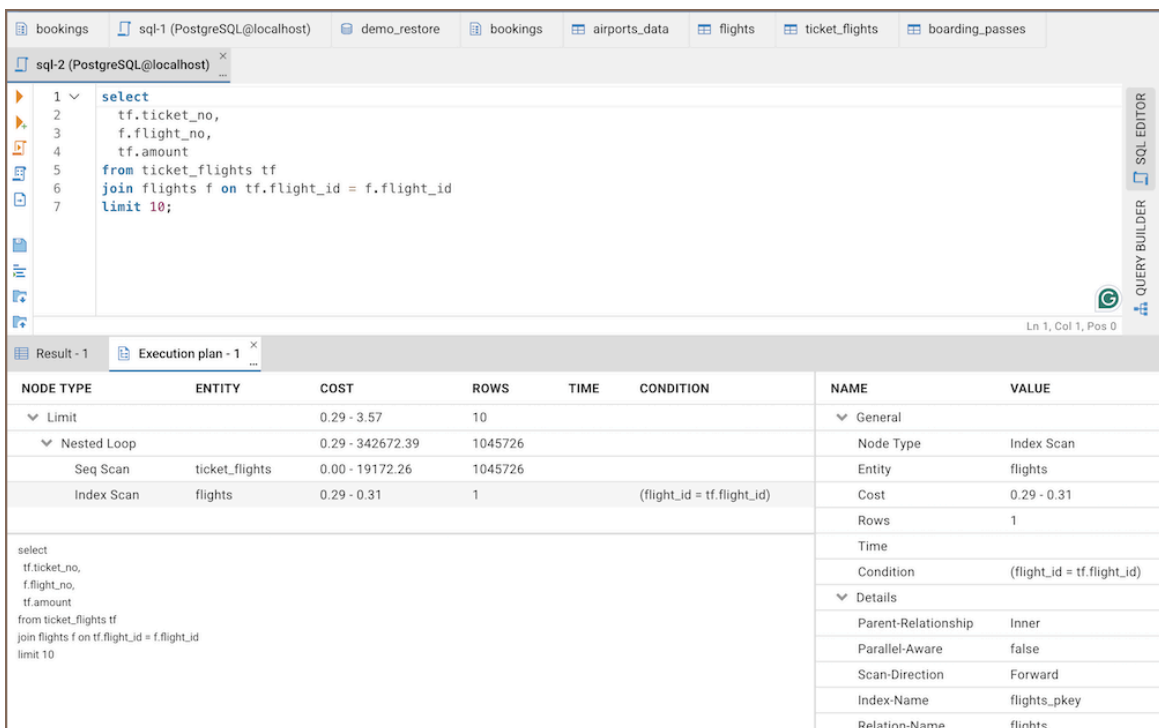
View execution plan

- Press `Shift+Ctrl+E` (`Shift+CMD+E` on macOS)
- Or click **Explain execution plan** () on the main toolbar

The execution plan opens in a new result tab as a query execution tree. It helps you estimate whether the query or script is efficient.

Tip

Click any row in the plan to view detailed statistics in the panel on the right.



The screenshot shows a database IDE interface. At the top, there's a toolbar with various icons. Below it, a tab labeled 'sql-2 (PostgreSQL@localhost)' is active, displaying a SQL query:

```
1 select
2   tf.ticket_no,
3   f.flight_no,
4   tf.amount
5 from ticket_flights tf
6 join flights f on tf.flight_id = f.flight_id
7 limit 10;
```

Below the query editor, there's a tab labeled 'Execution plan - 1'. It displays a query execution tree with the following structure:

NODE TYPE	ENTITY	COST	ROWS	TIME	CONDITION
Limit		0.29 - 3.57	10		
Nested Loop		0.29 - 342672.39	1045726		
Seq Scan	ticket_flights	0.00 - 19172.26	1045726		
Index Scan	flights	0.29 - 0.31	1		(flight_id = tf.flight_id)

On the right side of the IDE, there's a panel titled 'QUERY BUILDER' and 'SQL EDITOR'. It shows detailed statistics for the selected row (Index Scan on flights):

NAME	VALUE
General	
Node Type	Index Scan
Entity	flights
Cost	0.29 - 0.31
Rows	1
Time	
Condition	(flight_id = tf.flight_id)
Details	
Parent-Relationship	Inner
Parallel-Aware	false
Scan-Direction	Forward
Index-Name	flights_pkey
Relation-Name	flights

Visual Query Builder

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

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

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

The screenshot displays the Visual Query Builder interface. On the left, a visual query diagram shows two tables, 'basket_a ba' and 'basket_b bb', connected by a join relationship. The 'basket_a ba' table has columns 'a' (value 123) and 'fruit_a' (value ABC). The 'basket_b bb' table has columns 'b' (value 123) and 'fruit_b' (value ABC). A join relationship is indicated between 'fruit_a' and 'fruit_b'. On the right, the SQL code is displayed in the SQL Editor:

```
SELECT
  ba.a,
  ba.fruit_a,
  bb.fruit_b
FROM
  test_vqb.basket_a AS ba
  INNER JOIN test_vqb.basket_b AS bb ON
    ba.fruit_a = bb.fruit_b
ORDER BY
  ba.fruit_a;
```

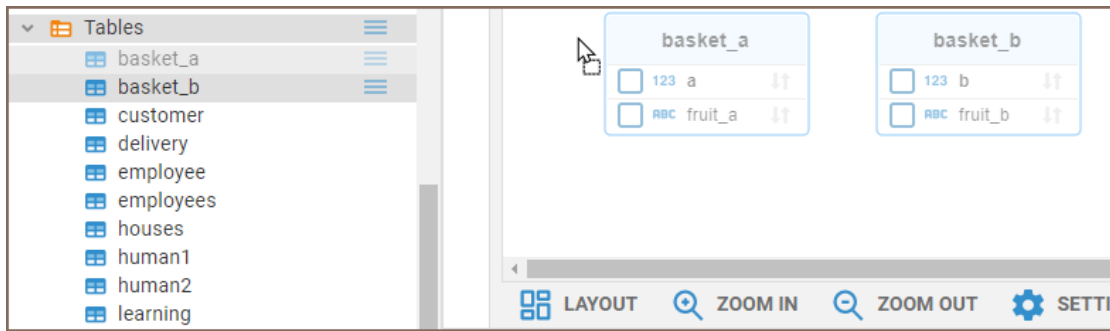
At the bottom, the 'Result - 1' tab shows the query results in a table:

#	123 a	ABC fruit_a	ABC fruit_b
1	1	Apple	Apple
2	2	Orange	Orange

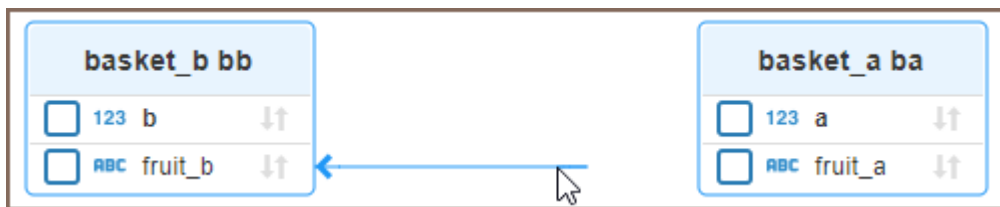
To open the Visual Query Builder, click the Query Builder tab in the SQL Editor right toolbar.

Creating a Visual Query

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



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



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



Filtering

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

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

Filter example:

Filter example: `e.employeeid>2`

SQL Script:

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

Result - 1

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

Sorting

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



Executing a Visual Query

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

Shortcuts

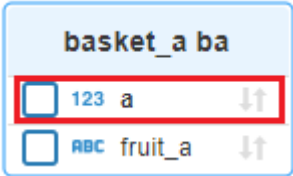
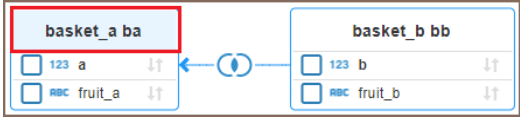
You can use the same shortcuts as in the SQL Editor to execute the Visual Query.

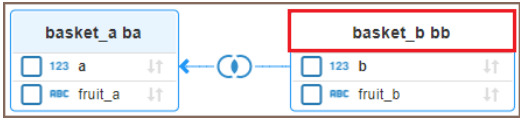
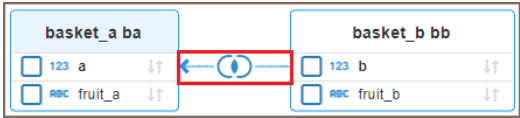
Key	Description
Ctrl+Enter	Execute the SQL statement
Ctrl+\ or Ctrl+Shift+Enter	Execute the SQL statement in a new tab

The Visual Query Builder symbols

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

Table symbols

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

Symbol	Description
	Colorless header marks a joined table in your Join Statement.
	Line goes from the joined table to the first table.

Join symbols

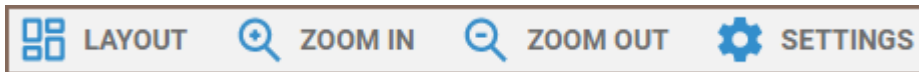
Available Join types are described in the table below. The Visual Query Builder can show results only for those types of Joins that are supported by your database.

Symbol	Description
	Inner Join
	Left Join
	Left Outer Join
	Right Join
	Right Outer Join
	Full Join
	Full Outer Join

Symbol	Description
	Cross Join

Settings

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



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

Visualization of an existing SQL query

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

Note

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

Client-side commands

You can use special commands in the [SQL Editor](#).

Info

These commands are executed on CloudBeaver's side, not on the server-side.

Supported commands

CloudBeaver supports the following commands:

Command	Database	Description
<code>@ai message</code> ★	All	Sends a natural-language message to the AI assistant. The AI processes the message and generates an appropriate SQL query or action based on your input. For more details, see AI command .

Auto and Manual commit modes

By default, the SQL Editor uses **Auto-commit** mode, automatically saving changes after each SQL statement. Switch to **Manual commit** for more control over transactions.

Manual commit mode

In **Manual commit** mode, you can commit or rollback transactions manually. Use it for multiple queries or to review changes before saving.

Action	Icon	Description
Switch to manual commit		Enables manual transaction control. You must Commit or Roll back changes manually.
Commit		Saves changes to the database.
Rollback		Reverts changes in the current transaction.
Switch to auto-commit		Restores Auto-commit mode, where changes are saved automatically.
Transaction log		Opens the Transaction Log .

Info

After committing in **Manual commit** mode, click **Refresh** to update the Result tab.

Transaction log

The **Transaction Log** shows all transactions (queries of the transaction type, such as `INSERT` , `DELETE` , `UPDATE` , and others) made during the current CloudBeaver session. To open the **Transaction Log**, click **Open transaction log** in the toolbar.

Info

The **Open transaction log** button also displays the number of uncommitted queries.

Transaction log (PostgreSQL@postgres) ✕

Time	Type	Query	Duration (ms)	Rows	Result
4:06:05 PM	SQL / User	DELETE FROM postgres.authors WHERE	2	1	Success
4:06:02 PM	SQL / User	INSERT INTO postgres.authors (authori	12	1	Success
4:06:02 PM	SQL / User	INSERT INTO postgres.authors (authori	15	1	Success

CANCEL
ROLLBACK
COMMIT

In the **Transaction log** window, you can interact with the following fields:

Field	Description
Time	Shows when the query was executed.
Type	Identifies the type of query (e.g., SQL/User).
Query	Lists all uncommitted queries.
Duration	Displays how long each query took to execute.
Rows	Indicates how many rows each query affected.
Result	Shows the query status.

Tip

Double-click a **Query** field to view the full query.

Limitations of Rollback capabilities

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

Important

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

Resource Manager

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

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

Administrators can also create and manage scripts in the Shared project. All users with access to this project can see, open and run the project scripts. Scripts can be associated with a specific connection. When a user opens such a script, the specified connection is displayed in the SQL Editor top toolbar and the user can immediately execute the script for the connection.

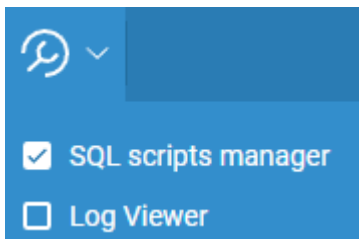
Note

scripts can only be associated with connections from the same project.

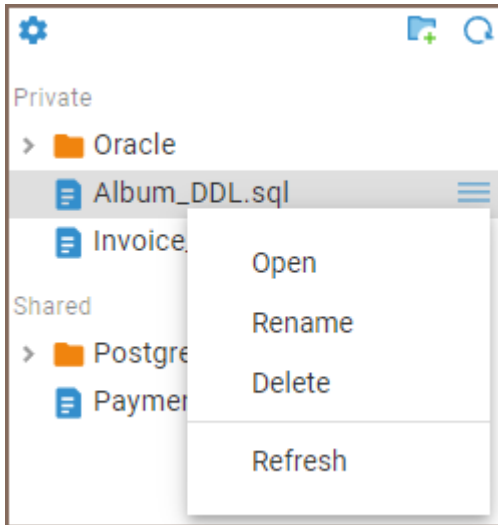
The Resource Manager is only available to logged users. Anonymous users may save scripts on a local machine.

Opening the Resource Manager

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



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



Resource manager settings menu

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

The table below describes each tool within the Resource Manager:

Tool	Description
Filter	Allows for script search within the Resource Manager. Activate the Filter switch to access the filter field and enter the script name to display relevant scripts. The search is limited to scripts that are currently visible.
Show collapsed	An additional setting for the Filter. When enabled, the search will include scripts within collapsed directories that the user has previously expanded.
Save tree state	Preserves the state of the Resource Manager tree view after refreshing the browser page. This setting is enabled by default.
Folders	Displays only the top level of folders. The full path to a folder is shown as breadcrumbs. When a user double-clicks on a script, the path to it is displayed.

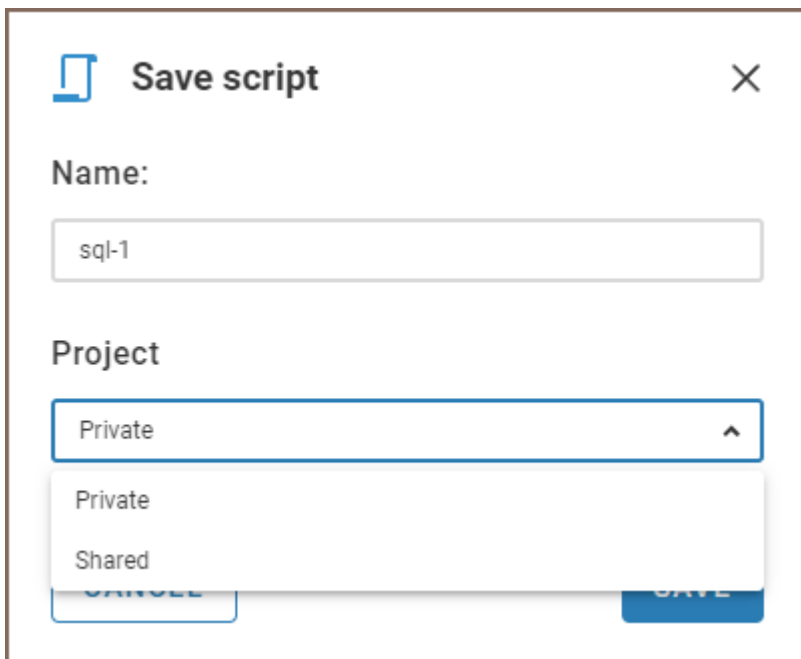
Tool	Description
	d in the Resource Manager tree.
Group by Project	Omits project names from the Resource Manager tree view, simplifying the display of scripts and folders.

Saving a script

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

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

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



The image shows a 'Save script' dialog box. It has a title bar with a document icon and a close button. Inside, there is a 'Name:' label followed by a text input field containing 'sql-1'. Below that is a 'Project' label followed by a dropdown menu. The dropdown menu is open, showing 'Private' as the selected option and 'Shared' as another option. At the bottom of the dialog, there are two buttons: 'CANCEL' and 'SAVE'.

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

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

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

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

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

Opening a script

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

Open

Rename

Delete

Refresh

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

Deleting a script

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

Script size limit

You can configure the size limit for scripts in the Resource Manager. This can be useful if you need to control the amount of space on the CloudBeaver server. The limit also helps to improve the application performance, because opening big scripts can take quite a long time. If a user is saving a

large script, they will see a message that the file size has been exceeded and the script will not be saved.

Info

For more information, see [Resource quotas](#).

Deactivate Resource manager

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

Tip

For more information on how to disable the Resource Manager through the configuration file, see [Server configuration](#).

Server Output

The Server output log lets you view messages generated by database scripts and procedures directly in the SQL Editor. This feature helps debug queries, monitor script execution, and track informational or warning messages from the server.

Open server output log

Before executing scripts, enable the **Server output** log:

1. In the **SQL Editor toolbar**, click **Show server output** button ().
2. Run your SQL queries or procedural scripts.



Tip

Scripts typically use statements like `RAISE` (PostgreSQL) or `PRINT` (SQL Server) to send these messages to the output log. Adjust message visibility by configuring log levels with statements like `SET` to show more detailed information. For details about log message types, see your database system's documentation.

View log messages

After executing the scripts, switch to the **Output** tab.

Search log messages

To quickly find a specific log type your search text into the search field at the top of the **Output** tab. Matching logs appear instantly.

Tip

Clear the search field to see all logs again.

Filter log messages

To simplify viewing specific message types, filter logs:

1. In the **Output** tab toolbar, click **Filter** button ().
2. Select the log message types you want to display:
3. Debug
4. Log
5. Info
6. Notice
7. Warning
8. Error

Info

Filtering only affects displayed logs - it doesn't change what's stored or generated by your database scripts.

Wrap log messages

To wrap long log messages in the **Output** tab:

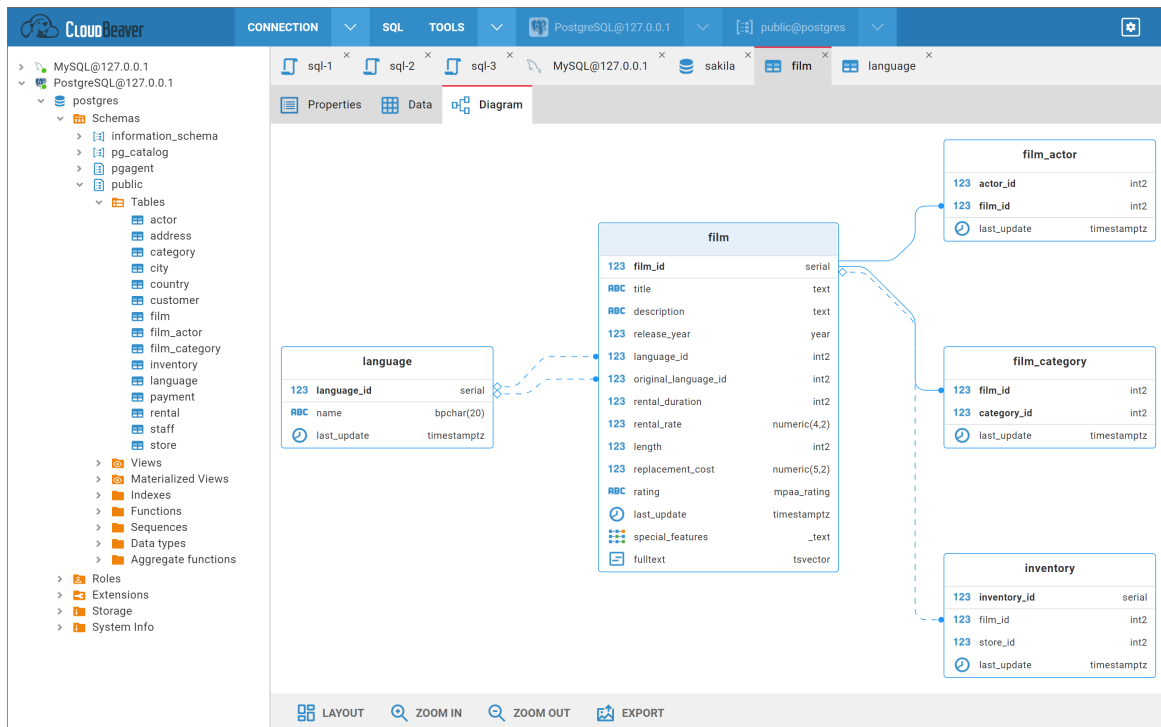
1. In the **Output** tab toolbar, click the settings button ().
2. Select the **Wrap mode** checkbox.

Entity relation diagrams

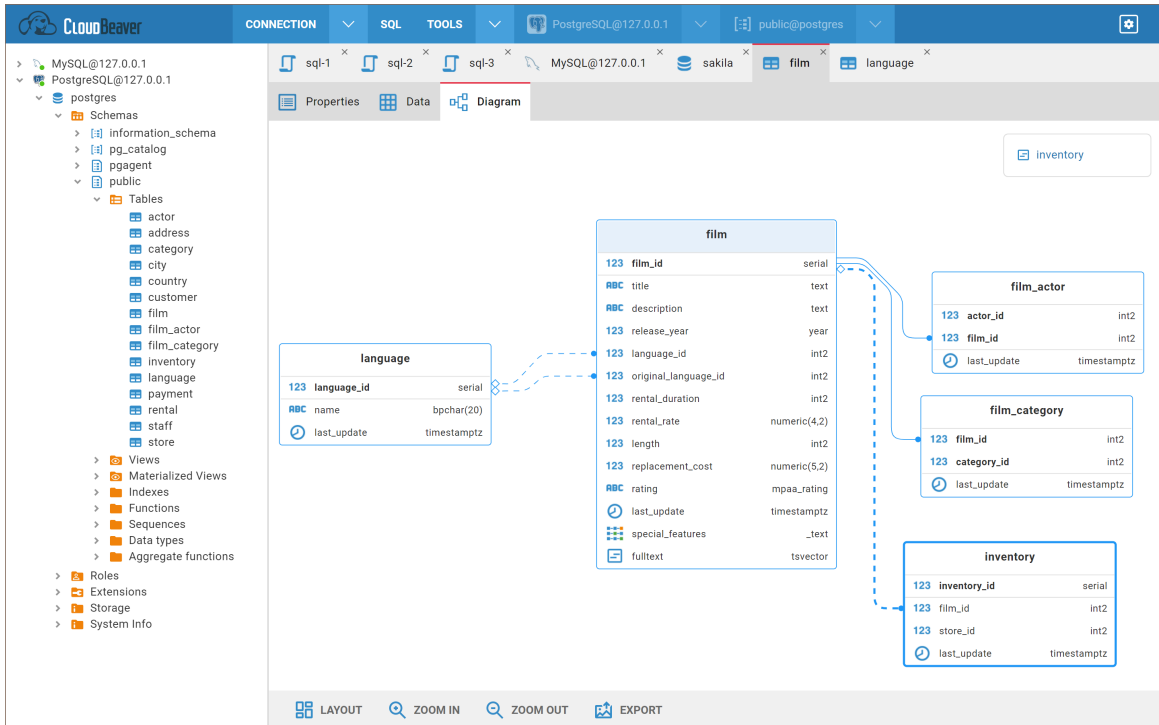
Overview

Display entities with attributes

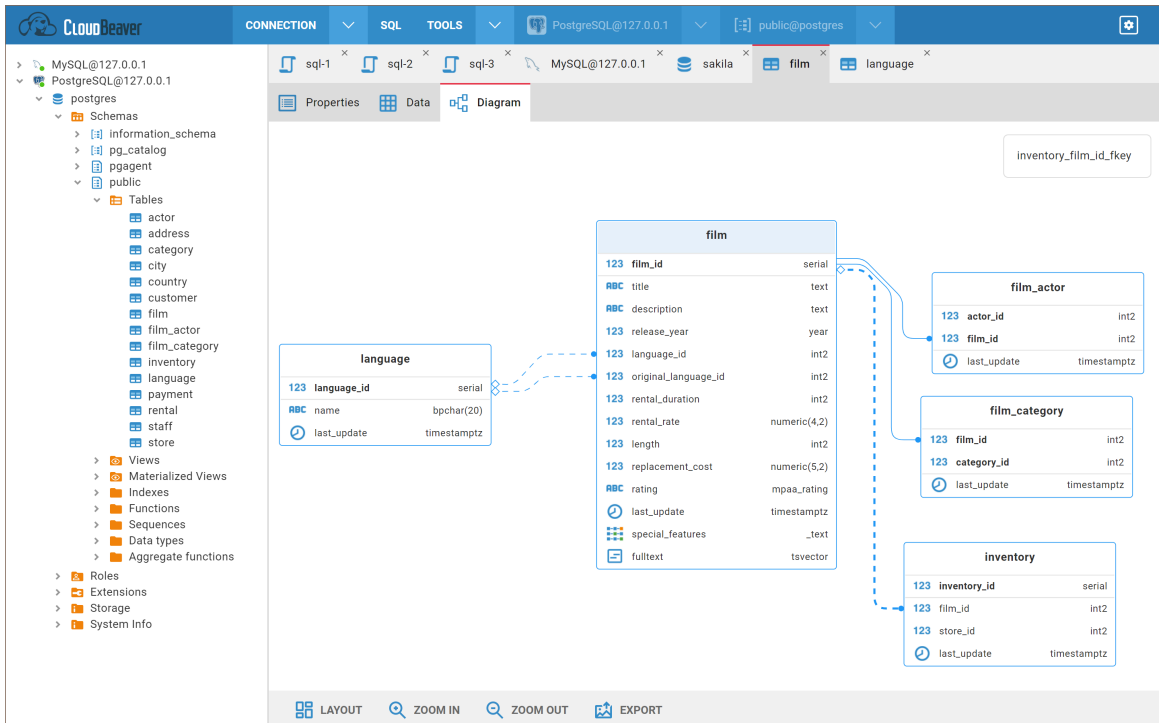
1. Navigate to your connection and open a table or schema
2. Select the "Diagram" tab (if the tab is not presented then the object does not support the diagram presentation)



You can click on an entity to highlight it:



You can click on a relation to get highlight it:

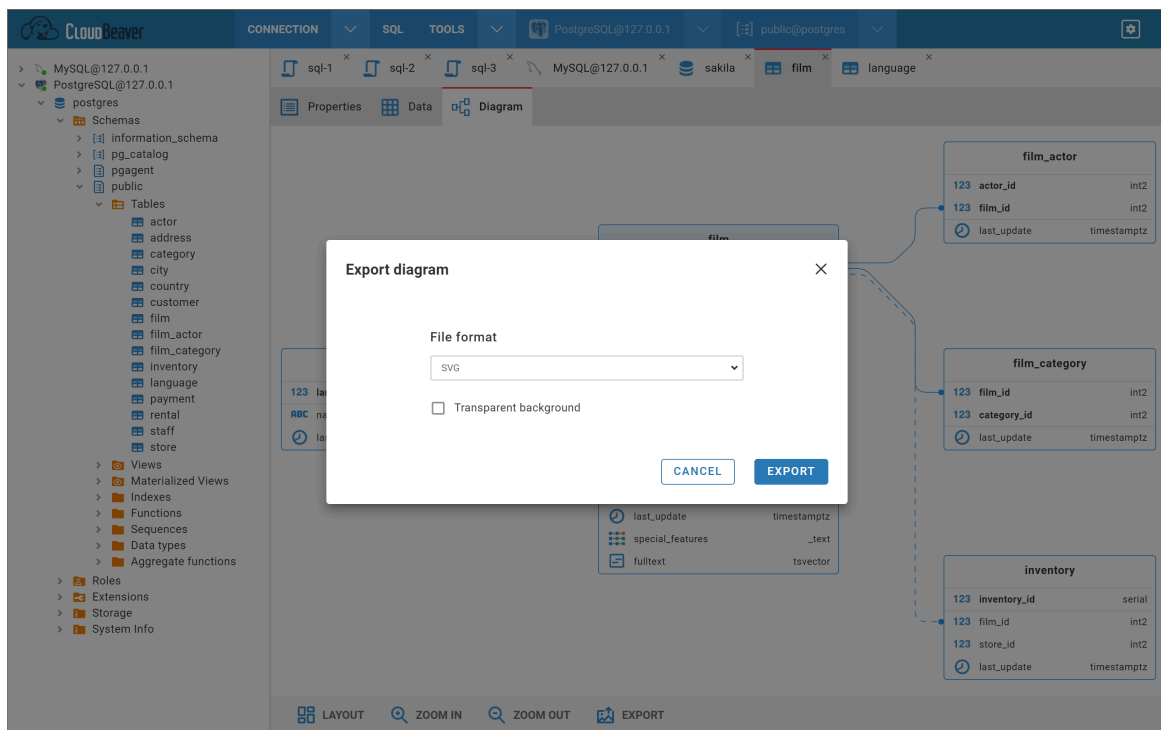


You can open the entity in the metadata editor by:

- 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



Cloud services

Cloud Explorer

Note

This feature is available in Enterprise, AWS, and Team editions only.

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

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

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

Tip

Use Cloud Storage to manage files and work with .sql scripts in your cloud provider.

Supported databases

The **Cloud Explorer** allows you to connect with and manage various databases:

Providers	Databases
Amazon Web Services	PostgreSQL
	MySQL
	Oracle
	Amazon Redshift
	Amazon Athena

Providers	Databases
	Amazon DocumentDB
	Amazon DynamoDB
	Amazon Keyspaces
	Amazon ElastiCache (Redis)
	Amazon Timestream
	Amazon Neptune
Google Cloud Platform	AlloyDB for PostgreSQL
	Microsoft SQL Server
	PostgreSQL
	MySQL
	Spanner
	Firestore
	BigQuery
	Bigtable
Microsoft Entra ID (Azure)	Microsoft SQL Server
	PostgreSQL
	MySQL
	Azure Cosmos DB for MongoDB

Providers	Databases
	Azure Cosmos DB for Cassandra
	Azure Cosmos DB for NoSQL

Cloud configuration

1. As an administrator, go to **Settings -> Administration -> Server Configuration**.
2. In the **Configuration** section, enable the relevant cloud service provider.
3. Save the changes.

Note

In the CloudBeaver AWS Edition, the **AWS** option is enabled by default. The **GCP** and **Microsoft Entra ID (Azure)** configurations are not available. You can use default instance credentials for **Cloud Explorer** authentication. For more details, see [AWS Settings](#).

Identity provider configuration

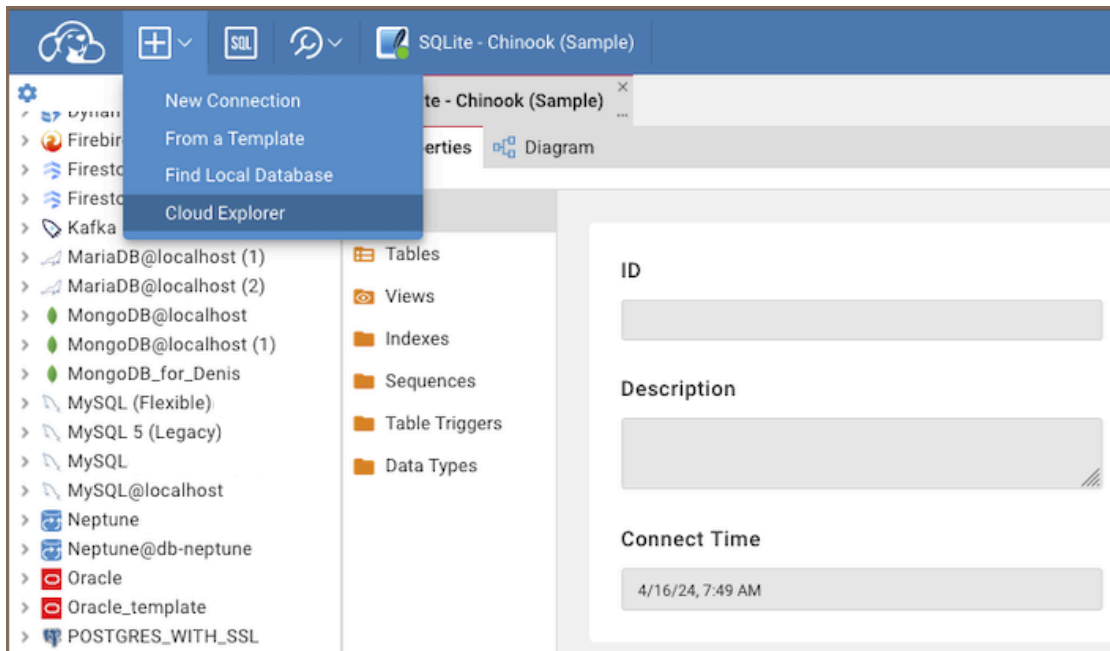
Each cloud service provider requires specific settings for the identity provider. Below is a table outlining the basic guides on configuring these settings.

Provider	Additional article
AWS	AWS Settings .
GCP	Google setup .
Azure	Microsoft Entra ID setup

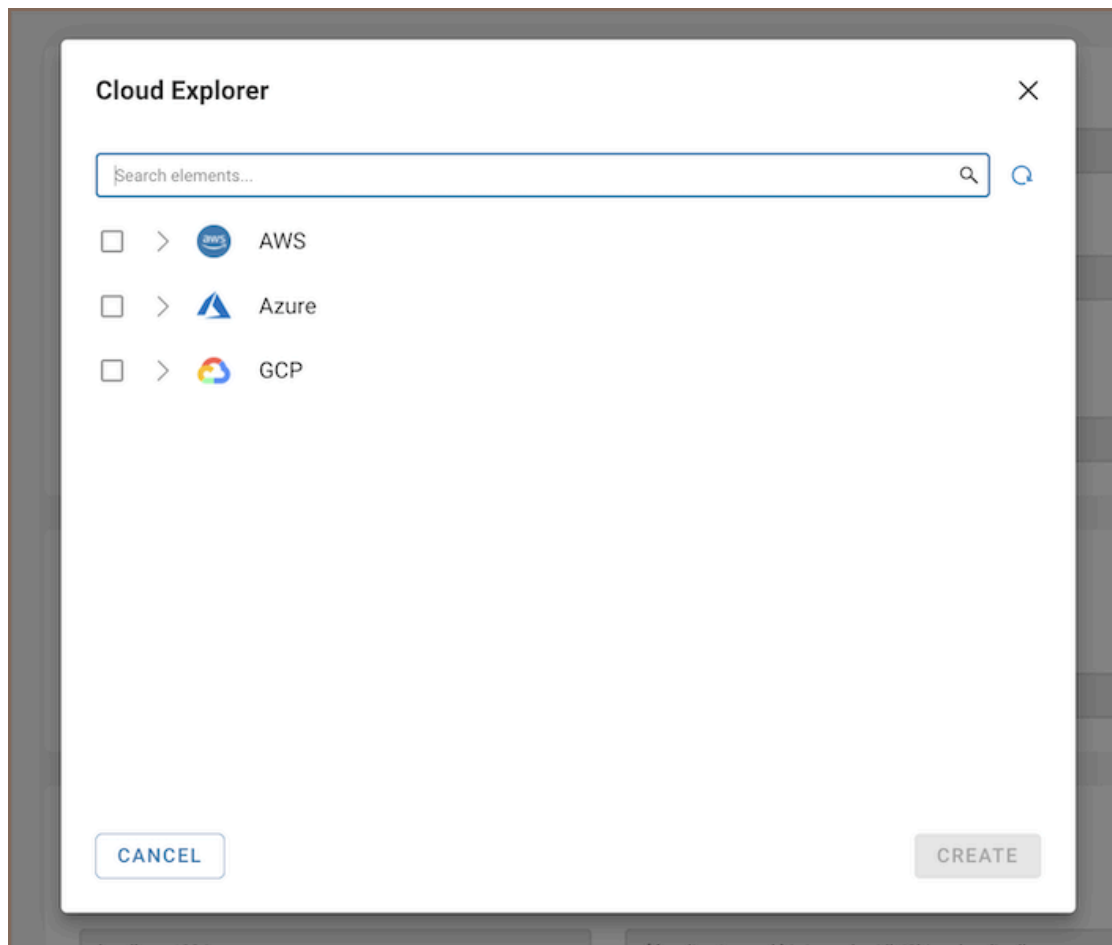
Cloud Explorer

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

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



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



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

ID

Name

 **Additional Authentication**
AWS IAM | AWS IAM

×

AWS IAM

FEDERATED

Static access keys

▼

Access Key

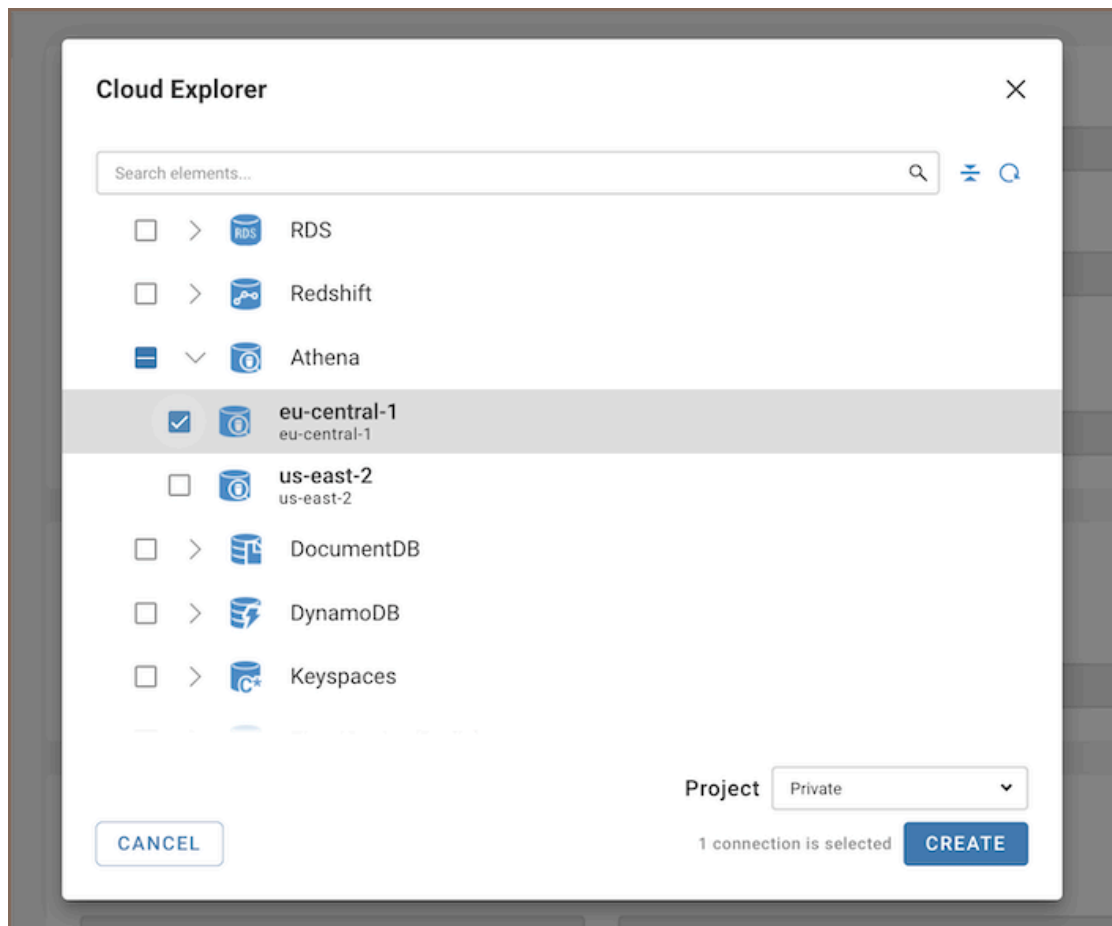
EXAMPLE_KEY

Secret Key

.....

LOGIN

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



Tip

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

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

Database cloud information

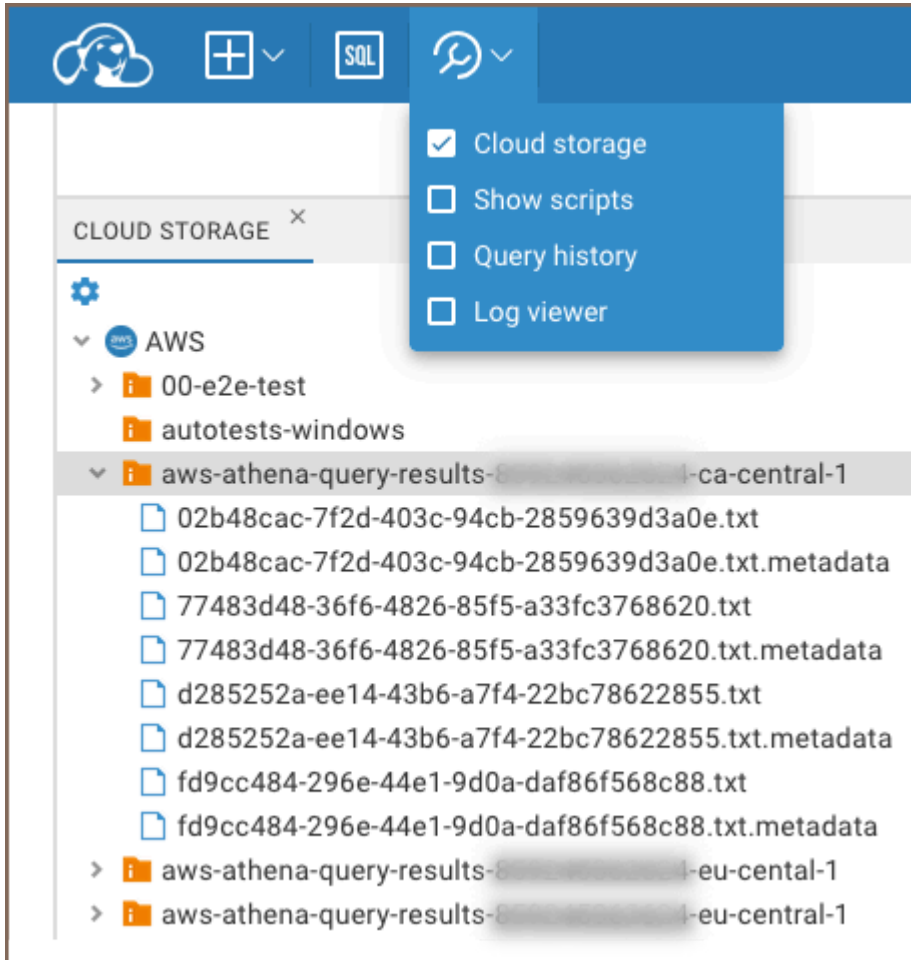
You can access your cloud database configuration directly from the Database Navigator. To do this, click on the menu button next to the database you are interested in and select **Edit Connection**. This action opens a special tab in the connection settings dialog. The information displayed in this tab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

Cloud Storage

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

Cloud Storage lets you manage files and work with `.sql` scripts in **Amazon S3** and **Google Cloud Storage**.



Configuration

1. As an administrator, go to **Settings -> Administration -> Server Configuration**.
2. In the **Configuration** section, enable the relevant cloud service provider.

3. Save the changes.
4. Set up the cloud service provider:
 - [Cloud \(AWS\)](#)
 - [Cloud \(GCP\)](#)

Tip

Use preconfigured authentication, like [AWS OpenID](#), to simplify access. In the CloudBeaver AWS Edition, you can use default instance credentials. For more details, see [AWS Settings](#).

Cloud storage

After configuring your cloud provider, you can work with the data files. Navigate to **Tools -> Cloud Storage** to access it.

In the **Cloud Storage** you can manage files like in a regular file system. Right-click a file or folder to open the context menu and access options such as:

- Upload files from your local storage to a bucket.
- Download files to your local storage.
- Delete or rename files.
- Drag and drop files between buckets or file systems.

Tip

Use the **Settings** button  for additional configuration options. For more details, see the [Settings menu](#).

Work with SQL scripts

Cloud Storage allows you to work with `.sql` files. Double-click a file to open it in the [SQL editor](#), where you can:

- View and edit the script.
- Execute SQL statements.
- Save changes directly back to **Cloud Storage**.

Work with data files

You can also open and query data files stored in your cloud buckets, such as `.csv`, `.json`, `.xlsx`, `.xml`, and `.parquet`. These files can be connected as data sources directly in CloudBeaver.

For details, see:

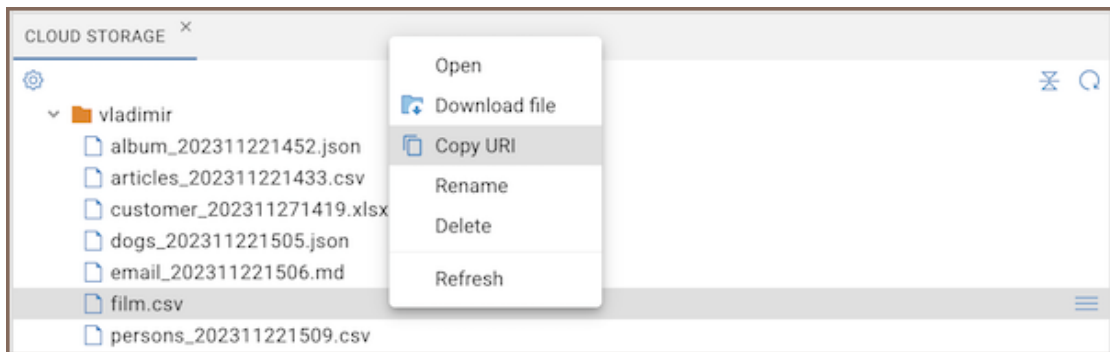
- [Multi Source](#)
- [CSV](#)
- [JSON](#)
- [Parquet](#)
- [XLSX](#)
- [XML](#)

Tip

Files like `.csv`, `.json`, `.xlsx`, `.xml`, and `.parquet` opened from Cloud Storage using the **Open** button create **temporary Multi Source connections**. These connections disappear after you log out.

To create a permanent connection:

1. Click **Copy URI** for the file you need.



2. Create a **new connection** and select the matching file type (for example, **CSV** or **JSON**).
3. Paste the copied URI into the **Database** field.

AWS settings

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

This article guides you through the initial steps to configure AWS integration in CloudBeaver.

Info

Once AWS integration is configured, you can access **Cloud Explorer**, and **Cloud Storage**.

Steps to configure AWS services

Access AWS configuration

1. As an administrator, go to **Settings -> Administration -> Server Configuration -> Configuration**.
2. Find the **Cloud (AWS)** checkbox and activate it.

Tip

For users of the AWS Edition, activating **Cloud (AWS)** is not required as it comes pre-enabled.

Cloudbeaver EE Web Server

https://cloudbeaver.cloudbeaver.com

620

SERVER INFORMATION

Server Name *

Server URL *

Session lifetime, min *

CONFIGURATION

Enable private connections
Allows users to create private connections

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

SERVICES

AI Completion
Enabled AI services in SQL editor. Prompt-based code completion and SQL queries generator.

Cloud (AWS)
Enables AWS cloud services (IAM, databases explorer, file storage)

Cloud (Azure)
Enables Microsoft Azure cloud services (databases explorer, file storage)

Cloud (Google)
Enables Google Cloud services (databases explorer, file storage)

Cloud Storage
Enables cloud file systems (e.g. S3). Users can access files in cloud storages if they have appropriate permissions.

Configure an Identity Provider

Configuring an Identity Provider is essential for authenticating and authorizing user access to AWS resources.

Info

For a list of available Identity Providers and detailed configuration instructions, see [Authentication Methods](#).

The screenshot shows the 'Configuration creation' dialog in the CloudBeaver interface. The left sidebar contains navigation links: Query Manager, Driver Management, Connection Templates, Access Management, Server configuration, AWS Settings, Identity Providers (selected), License, and Version update. The main area is titled 'Configuration creation' and includes a 'OPTIONS' tab. The form contains the following fields:

- Provider ***: AWS OpenID (dropdown)
- ID ***: google-aws (text input)
- Configuration name ***: ASW OpenID Configuration (text input)
- Description**: (text area)
- Icon URL**: (text input)
- Disabled**: ☐ (checkbox)
- Client ID**: :80-m3fsvgt939odaikphv6b.apps.googleusercontent.com (text input)
- Client Secret**: FRDPKM-drdUXXqkKu2n3srPQ1EoYM6RvzE (text input)
- IDP auth endpoint URL**: https://accounts.google.com/o/oauth2/auth (text input)
- IDP token endpoint URL**: https://oauth2.googleapis.com/token (text input)
- Role ARN**: (text input)

Buttons at the top right include '+ ADD', 'REFRESH', 'DELETE', 'CANCEL', and 'CREATE'.

AWS settings for default instance credentials

You can configure how AWS credentials are applied for authentication in Cloud Explorer, and Cloud Storage.

Setting	Description
Use default instance credentials	Enable to use AWS instance credentials for authentication.
Shared Access Role	Specify an IAM role that will be assumed for user access. Requires Use default instance credentials to be enabled.

Tips

- Enable **Use default machine credentials** to simplify authentication by automatically using credentials from your AWS instance.
- Specify a **Shared Access Role** to provide limited permissions for users, separate from the instance role.
- CloudBeaver supports various cloud-based **Secret Providers** to retrieve database credentials.

Configure AWS partitions and regions

You can configure the AWS partitions and regions where your resources are located. This ensures the system connects to the correct environment based on your selection.

1. Go to **Settings -> Administration -> AWS Settings**.
2. In the **Partition** dropdown, select the partition where your resources are located:
 - **AWS Standard**: Standard AWS regions.
 - **AWS China**: Regions compliant with Chinese regulations.
 - **AWS GovCloud (US)**: Regions for sensitive or regulated U.S. government workloads.
3. After selecting a partition, choose one or more AWS regions from the **Region List**.
4. Click **Save** to confirm and apply your settings.

The screenshot shows the 'AWS Settings' interface. On the left is a sidebar with navigation links: Server Configuration, Global Preferences, Users and Teams, Query Manager, Driver Management, AWS Settings (selected), Identity Providers, AI Settings, Domain Manager, and Product Information. The main content area has a top bar with 'SAVE' and 'CANCEL' buttons. Below this is the 'INSTANCE CONFIGURATION' section, which includes 'Instance Region' (set to 'eu-central-1') and 'Instance Role' (set to 'DBeaver'). There is a toggle for 'Use default instance credentials' which is turned on, with a note: 'The credentials will be used for common services (Cloud Explorer, Cloud Storage)'. Below that is a 'Shared Access Role' field. To the right of the instance configuration is the 'REGION LIST' section. It has a 'Partition' dropdown set to 'AWS Standard'. Under the 'AFRICA' heading, there is one checkbox for 'Africa (Cape Town) (af-south-1)'. Under the 'ASIA' heading, there are 14 checkboxes for various regions: Asia Pacific (Hong Kong) (ap-east-1), Asia Pacific (Hyderabad) (ap-south-2), Asia Pacific (Jakarta) (ap-southeast-3), Asia Pacific (Malaysia) (ap-southeast-5), Asia Pacific (Melbourne) (ap-southeast-4), 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), Middle East (Bahrain) (me-south-1), and Middle East (Istanbul) (me-central-1).

Azure permissions

To use CloudBeaver with Azure, certain permissions are required for different actions. Below is a summary of the necessary permissions and their purposes.

Required permissions

Action	Permission	Description
Login	No additional permissions required	Allows users to authenticate in CloudBeaver.
Cloud Explorer	Azure Service Management (user_impersonation)	Grants access to retrieve a list of available user resources using the <code>https://management.azure.com//.default</code> scope.
Database authentication - Azure SQL, MySQL	Azure SQL Database (user_impersonation)	Enables database authentication (Azure SQL, MySQL) using the <code>https://database.windows.net//.default</code> scope.
Database authentication - PostgreSQL, MySQL	Azure OSSRDBMS Database (user_impersonation)	Enables database authentication (PostgreSQL, MySQL) using the <code>https://ossrdbms-aad.database.windows.net/.default</code> scope.
User provisioning	User.ReadBasic.All or User.Read.All	Allows importing users from Azure AD.
Auto-assigned user teams	No additional permissions required, but groups claim must be included in the access or ID token.	Automatically assigns users to teams based on their Azure AD group memberships.

AI

AI Assistant

Note

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

CloudBeaver provides AI powered features for working with databases in natural language, using popular providers. It can [answer questions](#), [troubleshoot errors](#), or [explore database objects](#).

Supported providers

AI Assistant in CloudBeaver works with several AI providers.

- [OpenAI](#)
- [GitHub Copilot](#)
- [Azure OpenAI](#)
- [Google Gemini](#)
- [Ollama](#)
- [Anthropic Claude](#)
- [Amazon Bedrock](#)

CloudBeaver isn't affiliated with these providers. Integration with AI assistance is achieved through the public APIs. To use AI Assistant, register with the respective service provider and [get a secret key](#).

Important

Some paid AI provider plans don't allow access for use in third-party applications. Make sure your subscription includes this capability before connecting.

AI Assistant features

Chat

- [AI Chat](#). Chat naturally with AI to ask questions and get queries.

Work with SQL

- **Explain query.** Run AI to explain what a query does in plain language.
- **Fix SQL errors.** See what went wrong and how to fix a failed query.
- **AI command.** Run AI command directly in the SQL Editor. Type command like `@ai show films in which Grace Mostel starred`, and let AI generate the SQL for you.

Work with objects

- **Describe object with AI.** Let AI describe the purpose of a table, view, or other object.

Best practices for prompt formulation

Follow these guidelines to get the most accurate results from AI:

- **Use English:** While AI supports multiple languages, English usually gives the best results.
- **Know your database:** The better you know your tables, columns, and relationships, the more accurate the generated queries.
- **Be specific:** Include table and column names when possible to help AI produce precise queries.
- **Provide full context:** Instead of refining step by step, write a complete, well-structured prompt that clearly describes your goal.

Disable AI Assistant

Learn more about [disabling AI Assistant](#).

Data privacy

CloudBeaver protect your data and privacy with strict safeguards. AI Assistant works with the minimum information required, and all processing follows provider privacy policies.

Info

Learn more in [Data privacy](#).

Data privacy

Note

This feature is available in Enterprise, AWS, and Team editions only.

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

To enable AI features capabilities, metadata such as table and column names from the current database schema are transmitted to third-party AI services. This step is crucial for accurately translating user requests into SQL queries.

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

Privacy policy

In addition to these general principles, each AI provider applies its own privacy policy:

Provider	Privacy policy
OpenAI	Requests are processed under the OpenAI privacy policy.
GitHub Copilot	Copilot follows the GitHub privacy statement.
Azure OpenAI	Operates under its own privacy policy. It's recommended to review their terms.
Google Gemini	Data is handled according to Gemini privacy measures.
Ollama	Ollama follows a local-first design: all AI processing happens on your machine, and prompts, code, or documents never leave your computer.
Anthropic Claude	Requests are processed according to Anthropic's

Provider	Privacy policy
	privacy policy. Review their terms to understand how your data is handled.

Features

AI Chat

Note

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

To use the **AI Chat**:

1. **Activate** the AI assistant.
2. Select **Tools -> AI Chat** from the top **toolbar**.
 - Alternatively, click on the **AI assistant** icon  located in the left toolbar of the **SQL Editor**.
3. In the chat window that opens, select a **connection** and, optionally, a **scope**.
4. Enter your request in natural language.
5. Press the **Send** icon () to get the SQL translation. You can then choose from the following options:

Option	Icon	Description
Execute		Run the generated SQL query
Insert		Open the query in the SQL Editor and continue working with it directly
Copy query		Copy the query to your clipboard
Copy		Copy the entire AI Chat response to your clipboard

The screenshot displays the AI Chat interface. On the left, a table titled "AI Result (List all flights with f..." shows flight data. The table has columns: "AZ flight_no", "Flight number", "scheduled_departure", and "scheduled_arrival". The data is filtered by "column_name=10". The table contains 14 rows of flight information, including flight numbers like PG0134, PG0052, PG0561, etc., and their scheduled departure and arrival times.

On the right, the chat window shows a message: "List all flights with flight number, scheduled departure, and scheduled arrival". Below this, a text box explains: "To list all flights with flight number, scheduled departure, and scheduled arrival, you can simply query the flights table. Here is the SQL query:". The SQL query is displayed in a code block:

```
sql
SELECT flight_no, scheduled_departure, scheduled_arrival
FROM flights;
```

The chat window also includes a toolbar at the top with a dropdown menu set to "PostgreSQL shared" and a search bar. At the bottom, there is a text input field with a placeholder "Press ⌘+ to submit your message, press Enter to add a new line" and a send button.

AI Chat configuration

Connection field

To use a specific connection in **AI Chat**, select the connection from the toolbar at the top of the chat panel. The selected connection defines which database the AI will use when generating SQL queries.

PostgreSQL shared

Show me all actors whose name starts with the letter...

Show me all actors whose name starts with the letter T

To retrieve all actors whose names start with the letter 'T' from the `actor` table, you can use the following SQL query:

sql

SELECT *

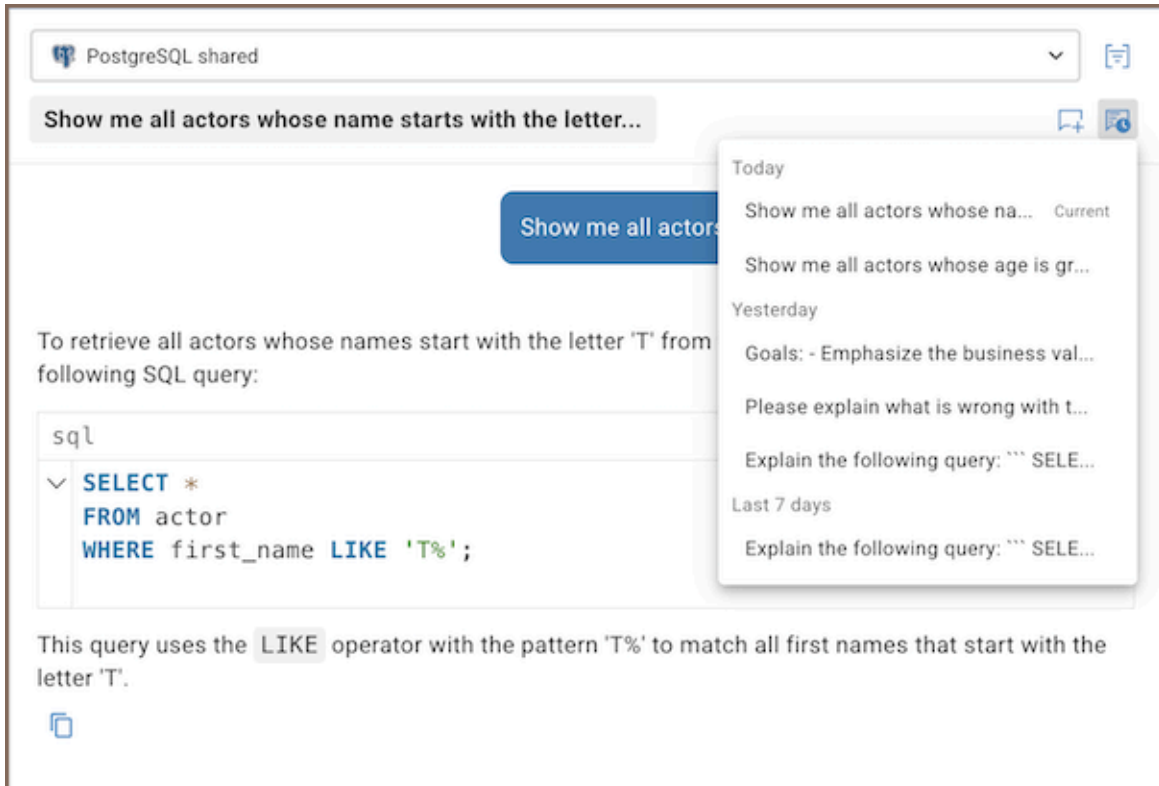
FROM actor

WHERE first_name LIKE 'T%';

This query uses the `LIKE` operator with the pattern 'T%' to match all first names that start with the letter 'T'.

Multiple conversations

You can create multiple chats for different connections.



The first chat is created automatically when you open **AI Chat**. Each chat is named based on your first prompt. Messages related to the same query or object in the same connection are kept together in one conversation.

Action	Icon	How to do it
New conversation		Click the icon to start a new conversation
Conversation history		Use the dropdown menu to switch between existing chats
Delete a chat		Select the chat from the Conversation history dropdown, then click the delete icon

Important

Deleted conversations can't be restored.

Resetting query context

The AI Chat logs your query history, enabling you to reference and expand on prior inputs. Each new or edited entry prompts the AI to generate a revised SQL query. To change the query context in **AI Chat** or clear history up to a specific message, click the Delete message icon () next to the desired message. Deleting a message also removes it and all messages below it.

Defining the scope

To improve accuracy, especially in databases with large schemas, you can use the **Scope** field to limit the context to specific database objects.

1. In the **AI Chat** toolbar, click the **Change scope** icon  to open the context menu.
2. Choose the desired context level:

Scope option	Description
Connection	AI considers all schemas within the current database connection
Current Database	AI is limited to the currently selected database
Current Schema	AI focuses only on the schema you're currently using
Custom	Lets you define a more specific scope, such as a particular table or set of objects

Work with SQL

Explain query

Note

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

You can use **Explain Query with AI** to quickly understand what your SQL query does. The LLM translates your SQL into plain language, helps you catch mistakes, and makes it easier to understand complex logic.

Using Explain Query with AI

1. [Activate](#) the AI assistant.
2. Open your query in the [SQL Editor](#).
3. Select the needed query text.
4. Click **Explain Query with AI** icon () located in the left toolbar of the SQL Editor.
5. In the [AI chat](#) window that opens you will see an explanation of the selected query. You can continue the conversation, ask follow-up questions, and iterate on the query with help from the LLM.

Fix SQL errors

Note

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

AI error explanation helps you understand and fix SQL errors using AI.

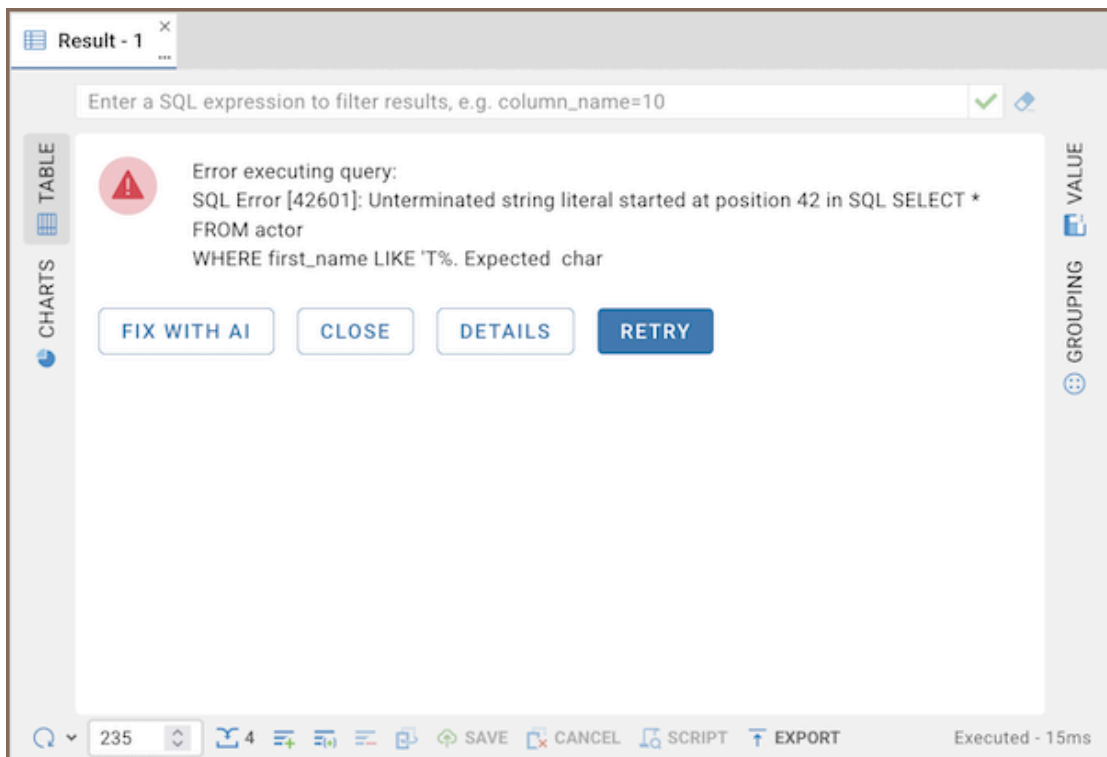
Using AI error explanation

To get help with a SQL error:

1. After running a query, if an error occurs, the **Fix with AI** button appears below the error message in the Results panel.

Note

AI assistant must be activated first. [Learn how to enable it.](#)



2. Click it to open **AI Chat** with an explanation and a suggested fix.

PostgreSQL shared

Please explain what is wrong with the following SQL...

Auto-generated

Please explain what is wrong with the following SQL query and suggest a fix for the error:

```
SELECT *
FROM actor
WHERE first_name LIKE 'T%;
```

The error message is:

```
Error executing query:
SQL Error [42601]: Unterminated string literal s
tarted at position 42 in SQL SELECT *
FROM actor
WHERE first_name LIKE 'T%. Expected char
```

The error is due to the missing closing single quote in the `LIKE` clause. To fix it, you need to close the single quote properly. Here is the corrected query:

sql

```
SELECT *
FROM actor
WHERE first_name LIKE 'T%';
```

3. Use **Execute** icon to apply the fixed query.

AI command

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

The AI command lets you run natural language requests directly in the SQL Editor. Just type `@ai` with your question, and CloudBeaver will generate and execute the SQL for you.

Using AI command

To use the **AI command**:

1. **Activate** the AI assistant.
2. Open the **SQL Editor**.
3. Type your request using the `@ai` command.

Example

```
@ai show all customers with invoices in the last month
```

4. Execute the AI command. The results will appear in the **Result** panel.

Important

The AI command shows a confirmation for any risky query, including `UPDATE` , `DELETE` , and any `DROP TABLE` operation. These commands always prompt you before execution.

Work with objects

Describe object with AI

Note

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

The **Describe Object with AI** command helps you quickly understand the purpose of a table, view, or another object. It uses AI to generate a short, human-readable description based on object metadata.

PostgreSQL shared

Goals: - Emphasize the business value of database...

Auto-generated

Provide a structured, concise and formal description of table 'public.actor' in the database schema.

Table: public.actor

- **Purpose:** The `actor` table stores information about actors involved in films. It contains details such as the actor's first name, last name, age, and other relevant information.
- **Connections:**
 - Connected to the `film_actor` table through the `actor_id` column, establishing a relationship between actors and the films they are part of.
- **Usage:** This table is essential for tracking and managing actor information in the context of film production and casting. It facilitates linking actors to specific films through the `film_actor` table.

Describe an object

To use the Metadata description:

1. **Activate** the AI assistant.
2. Open the **Database Navigator** or **Properties Editor**.
3. Click the cell context menu (three-bar icon) next to a supported object:
 - Schemas
 - Tables
 - Views
 - Indexes
 - Triggers
 - Procedures
4. Select **Describe object with AI** from the context menu.
5. In the **AI chat** window that opens you will see a short description. You can continue the conversation, ask follow-up questions, and iterate on the query with help from the LLM.

Tutorials

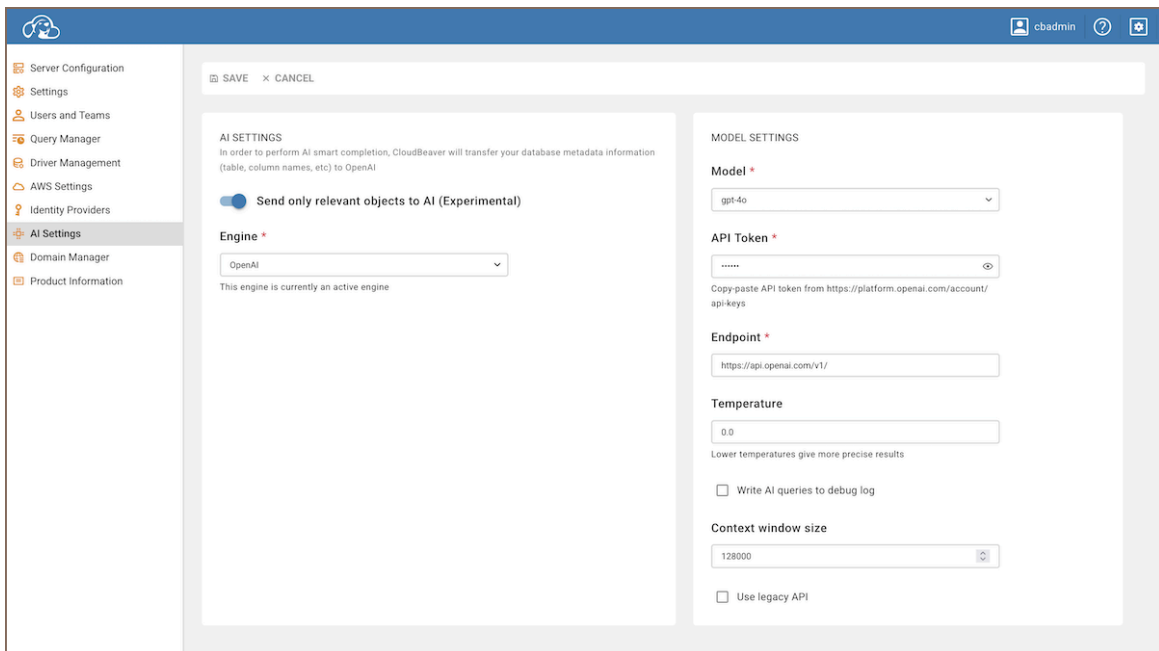
Quick start with AI

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

Work through a typical flow in the SQL editor using AI assistant - from writing a query, to fixing an error, to understanding objects.

To get started, **activate** AI Assistant, open **Settings -> Administration -> AI Settings**, choose a **provider**, and paste your API token.



The screenshot shows the CloudBeaver web interface with the 'AI Settings' panel open. The left sidebar contains a navigation menu with items like 'Server Configuration', 'Settings', 'Users and Teams', 'Query Manager', 'Driver Management', 'AWS Settings', 'Identity Providers', 'AI Settings' (highlighted), 'Domain Manager', and 'Product Information'. The main panel is titled 'AI SETTINGS' and includes a 'SAVE' button and a 'CANCEL' button. It contains two sections: 'AI SETTINGS' and 'MODEL SETTINGS'. The 'AI SETTINGS' section has a toggle for 'Send only relevant objects to AI (Experimental)' which is turned on, and a dropdown for 'Engine' set to 'OpenAI'. The 'MODEL SETTINGS' section includes a dropdown for 'Model' set to 'gpt-4o', a text field for 'API Token' with a copy icon, a text field for 'Endpoint' set to 'https://api.openai.com/v1/', a text field for 'Temperature' set to '0.0', a checkbox for 'Write AI queries to debug log', a text field for 'Context window size' set to '128000', and a checkbox for 'Use legacy API'.

Tip

Different vendors have different limits, models, and regions. Pick what fits your data policy and budget.

Start in the SQL Editor

1. Open the **SQL editor**.
2. Select **Tools -> AI Chat** from the top **toolbar**.

3. Pick a **connection**.

The screenshot shows a web-based SQL editor interface for a 'PostgreSQL shared' connection. At the top, a dropdown menu is set to 'PostgreSQL shared'. Below it, a chat input field contains the text 'Show me all actors whose name starts with the letter...'. A blue button next to it says 'Show me all actors whose name starts with the letter T'. Below the button, a text block explains: 'To retrieve all actors whose names start with the letter 'T' from the actor table, you can use the following SQL query:'. This is followed by a code editor with a light blue background and a 'sql' tab. The code in the editor is:

```
SELECT *  
FROM actor  
WHERE first_name LIKE 'T%';
```

 Below the code editor, another text block explains: 'This query uses the LIKE operator with the pattern 'T%' to match all first names that start with the letter 'T'.'. There are copy and paste icons for the chat input, the SQL query, and the explanation text.

Info

Optionally set a **scope** to improve accuracy on large instances.

4. Type what you need in AI chat.
5. Execute or refine.

For details, see [AI chat](#).

Generate directly in the editor with @ai

Want to stay focused in the SQL editor? Use `@ai` when you don't need the full chat flow.

1. In the SQL editor, enter `@ai your request...`
2. Execute the command. The results will appear in the **Result** panel.

For details, see [AI command](#).

Understand what the query is doing

Need a clear explanation, risks, or edge cases?

1. Select the SQL
2. Click **Explain Query with AI** icon () located in the left toolbar of the SQL Editor.



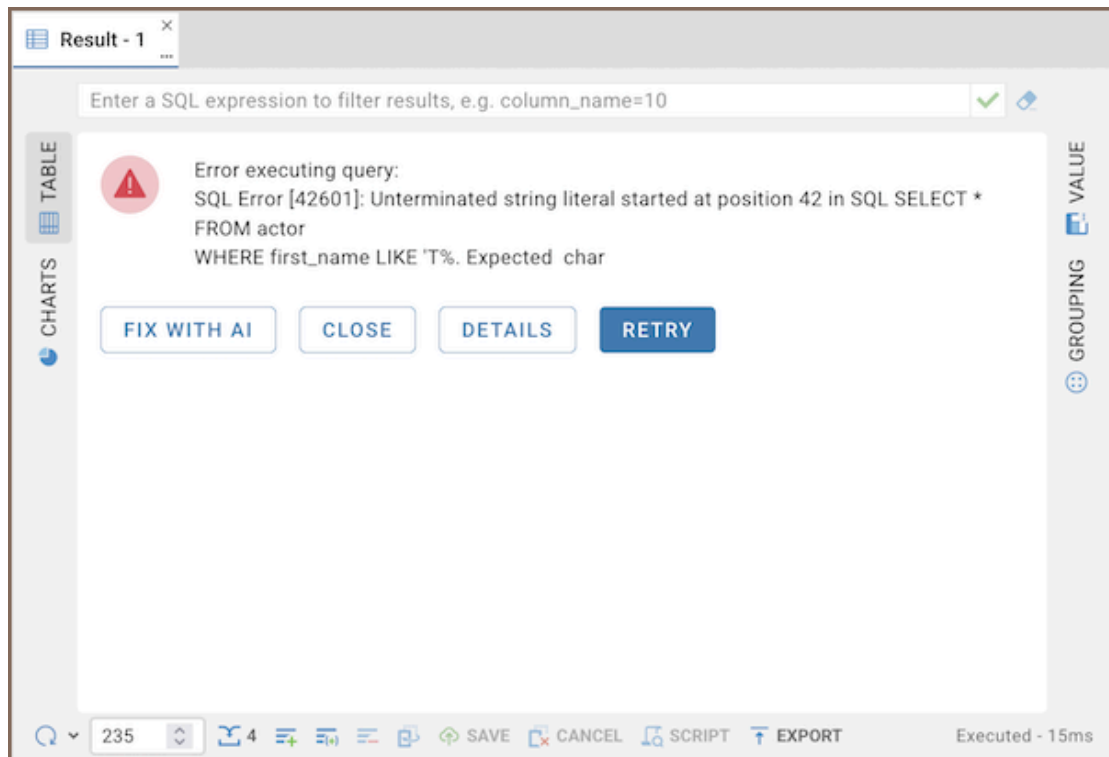
3. Read the explanation in AI chat, and ask follow-ups

For details, see [Explain query using AI](#).

Fix an error when the query fails

If the SQL fails, use AI to explain and repair it in one pass.

1. Click **Explain and Fix**



2. Apply the fix

PostgreSQL shared

Please explain what is wrong with the following SQL...

Auto-generated

Please explain what is wrong with the following SQL query and suggest a fix for the error:

```
SELECT *
FROM actor
WHERE first_name LIKE 'T%;
```

The error message is:

```
Error executing query:
SQL Error [42601]: Unterminated string literal s
tarted at position 42 in SQL SELECT *
FROM actor
WHERE first_name LIKE 'T%. Expected char
```

The error is due to the missing closing single quote in the `LIKE` clause. To fix it, you need to close the single quote properly. Here is the corrected query:

sql

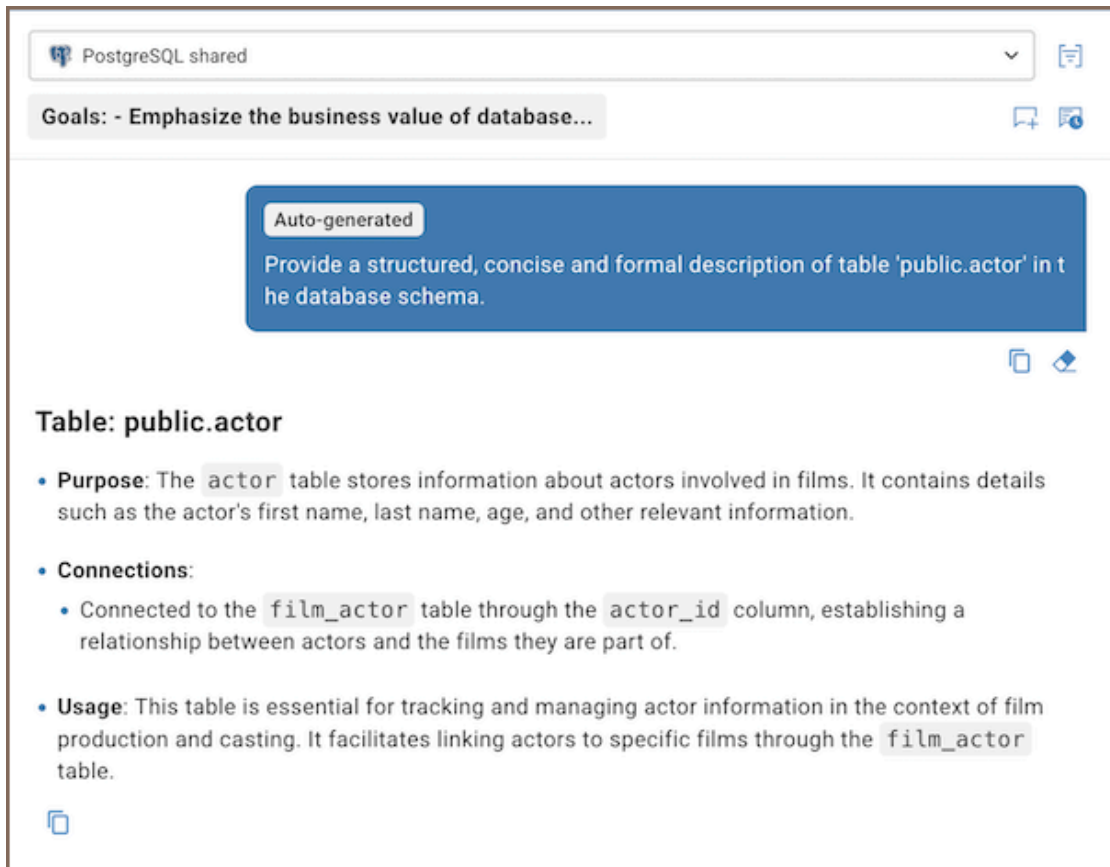
```
SELECT *
FROM actor
WHERE first_name LIKE 'T%';
```

For the flow, see [AI error explanation](#).

Describe a database object when you need context

Ask AI to describe a table, view, or other object in plain language. The result opens in AI chat.

1. Open [Database Navigator](#) or [Properties editor](#)
2. Click the cell context menu (three-bar icon)
3. Select **Describe object with AI** from the context menu.
4. Review the description in AI chat, and ask follow-ups



The screenshot shows a software interface for a PostgreSQL database. At the top, there's a dropdown menu set to 'PostgreSQL shared'. Below it, a grey bar contains the text 'Goals: - Emphasize the business value of database...'. The main area features a blue box with the label 'Auto-generated' and the instruction: 'Provide a structured, concise and formal description of table 'public.actor' in the database schema.' Below this, the title 'Table: public.actor' is displayed. The description follows in a bulleted list: 'Purpose: The actor table stores information about actors involved in films. It contains details such as the actor's first name, last name, age, and other relevant information.'; 'Connections: Connected to the film_actor table through the actor_id column, establishing a relationship between actors and the films they are part of.'; and 'Usage: This table is essential for tracking and managing actor information in the context of film production and casting. It facilitates linking actors to specific films through the film_actor table.'

PostgreSQL shared

Goals: - Emphasize the business value of database...

Auto-generated

Provide a structured, concise and formal description of table 'public.actor' in the database schema.

Table: public.actor

- **Purpose:** The `actor` table stores information about actors involved in films. It contains details such as the actor's first name, last name, age, and other relevant information.
- **Connections:**
 - Connected to the `film_actor` table through the `actor_id` column, establishing a relationship between actors and the films they are part of.
- **Usage:** This table is essential for tracking and managing actor information in the context of film production and casting. It facilitates linking actors to specific films through the `film_actor` table.

For supported objects and flow, see [Describe database object](#).

Settings

AI Assistant settings

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

Use this page to configure the general behavior of AI in CloudBeaver.

The screenshot shows the CloudBeaver interface with the 'AI Settings' tab selected in the left sidebar. The main content area is divided into two sections: 'AI SETTINGS' and 'MODEL SETTINGS'. The 'AI SETTINGS' section includes a toggle for 'Send only relevant objects to AI (Experimental)' and a dropdown menu for 'Engine' set to 'OpenAI'. The 'MODEL SETTINGS' section includes fields for 'Model' (gpt-4o), 'API Token' (masked), 'Endpoint' (https://api.openai.com/v1/), 'Temperature' (0.0), 'Context window size' (128000), and checkboxes for 'Write AI queries to debug log' and 'Use legacy API'.

AI Assistant setup

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

1. Navigate to **Settings -> Administration -> Server Configuration**
2. Ensure the **AI** option is activated
3. Navigate to **AI Settings** tab
4. Use the **Engine** dropdown menu to switch between and set up supported providers:
 - **OpenAI**

- [GitHub Copilot](#)
- [Azure OpenAI](#)
- [Google Gemini](#)
- [Ollama](#)
- [Anthropic Claude](#)
- [Amazon Bedrock](#)

Send only relevant objects to AI

 Available since 25.3

This option enables Retrieval-Augmented Generation behavior (RAG). When it's on, CloudBeaver looks at your request and sends only the tables, columns, or other objects that are actually needed for the answer. This keeps responses faster, reduces token usage, and helps the assistant build replies based on real, relevant data instead of your full schema.

Note

RAG works only with the [OpenAI](#) provider, and only when the server runs on a PostgreSQL database. For information on setting up the server database, see [Server database](#).

Disable AI Assistant

To disable AI Assistant features:

1. Navigate to **Settings -> Administration -> Server Configuration -> Services section**
2. Deselect **AI option**
3. Save changes

Providers

OpenAI

Note

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

To use OpenAI as an AI provider in CloudBeaver, configure it as follows:

1. As an administrator, sign up on the [OpenAI platform](#).
2. Navigate to the [API Keys section](#) and generate a new secret key.
3. Insert this key into CloudBeaver's **Model Settings**.
4. Choose the model.
5. Save changes.

Note

OpenAI services are available in specific countries. Consult the [supported countries](#) list to verify availability in your location.

Model settings

Setting	Description	Default
API token	Input your secret key from the OpenAI platform.	
Endpoint	The address where requests to the AI service are sent. Normally this is filled automatically for the selected provider. Change it only if you use a custom endpoint.	https://api.openai.com/v1/
Model	Choose the AI model.	<code>gpt-4o</code>
Temperature	Control AI's creativity from	<code>0.0</code>

Setting	Description	Default
	<code>0.0</code> (more precise) to <code>0.9</code> (more diverse). Note that higher temperature can lead to less predictable results.	
Write AI queries to debug log	Logs your AI requests. These entries go to the server log.	false
Context window size	Choose the context size between <code>2048</code> and <code>32768</code> . A larger number allows the AI to use more data for better answers but may slow down response time. Choose based on your balance of accuracy and speed.	<code>12800</code>
Use legacy API	Use the legacy OpenAI chat API (<code>/v1/chat/completions</code>). May be useful if you're using an OpenAI-compatible engine that doesn't support the newer <code>/v1/responses</code> endpoint.	false

GitHub Copilot

Note

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

To use GitHub Copilot as an AI provider in CloudBeaver, configure it as follows:

1. As an administrator, sign in to your [GitHub account](#)
2. Go to the [GitHub Copilot page](#) and activate a free trial or subscription
3. Get your Copilot access token
4. Paste the token into CloudBeaver's Model Settings
5. Choose the model
6. Save changes

Model settings

Setting	Description	default
Copilot Access token	Input your secret key from the OpenAI platform.	
Model	Choose the AI model.	<code>gpt-3.5-turbo</code>
Temperature	Control AI's creativity from <code>0.0</code> (more precise) to <code>0.9</code> (more diverse). Note that higher temperature can lead to less predictable results.	<code>0.0</code>
Context window size	Choose the context size between <code>2048</code> and <code>32768</code> . A larger number allows the AI to use more data for better answers but may slow down response time. Choose based on your balance of accuracy and speed.	<code>16384</code>
Write AI queries to debug log	Logs your AI requests. These e	false

Setting	Description	default
	entries go to the server log.	

Azure OpenAI

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

To use Azure OpenAI as an AI provider in CloudBeaver, configure it as follows:

1. As an administrator, sign in to the **Azure Portal**
2. Click **Create a resource** in the menu
3. Open the **AI Apps and Agents** category
4. Select **Azure OpenAI** from the list and click **Create**
5. Generate and copy the credentials for the newly created service
6. Insert these credentials into CloudBeaver's **Model Settings**
7. Save changes

Model settings

Setting	Description	default
API token	Input your secret key from the OpenAI platform.	
Endpoint	Configure a custom endpoint URL for Azure OpenAPI interactions.	
Deployment	Specify the deployment name chosen during model deployment.	
Temperature	Control AI's creativity from 0.0 (more precise) to 0.9 (more diverse). Note that higher temperature can lead to less predictable results.	4.0
Write AI queries to debug log	Logs your AI requests. These e	false

Setting	Description	default
	entries go to the server log.	
Context window size	Choose the context size between 2048 and 32768 . A larger number allows the AI to use more data for better answers but may slow down response time. Choose based on your balance of accuracy and speed.	

Google Gemini

Note

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

To use Gemini as an AI provider in CloudBeaver, configure it as follows:

1. As an administrator, sign up on the [Google Cloud Platform](#)
2. Navigate to the [Google Cloud Console](#) and create a new project
3. Enable the Gemini API for your project by searching for the Gemini API in the marketplace and clicking **Enable**
4. Create credentials for your project by navigating to the **Credentials** page under **APIs & Services**. Choose **Create credentials** and select the appropriate type for your Gemini integration
5. Insert these credentials into CloudBeaver's **Model Settings**.
6. Save changes

Note

Google Gemini services are subject to regional availability. Check the list of [available regions](#) to ensure access in your area.

Model settings

Setting	Description	default
API token	Input your secret key from the OpenAI platform.	
Model	Choose the AI model.	
Temperature	Control AI's creativity from <code>0.0</code> (more precise) to <code>0.9</code> (more diverse). Note that higher temperature can lead to less predictable results.	<code>0.0</code>
Write AI queries to debug log	Logs your AI requests. These e	false

Setting	Description	default
	entries go to the server log.	

Ollama

Note

This feature is available in Enterprise, AWS, and Team editions only.

Ensure that Ollama is already installed and running on a server. You will need the host address where Ollama is installed to proceed.

To use Ollama as an AI provider in CloudBeaver, configure it as follows:

1. As an administrator, specify the host address of your Ollama server in the **Hostname** field, ensuring it follows the format `http://host:port`
2. Insert the **Model**, **Context Size**, and **Temperature** you need for your integration
3. Save changes

Model settings

Setting	Description	default
Hostname	Specify the address of the Ollama server in the format <code>http://host:port</code> . This is required for connecting and loading available models.	
Model	Choose the AI model.	<code>llama2:latest</code>
Context size	Choose the context size between <code>2048</code> and <code>32768</code> . A larger number allows the AI to use more data for better answers but may slow down response time. Choose based on your balance of accuracy and speed.	<code>3000</code>
Temperature	Control AI's creativity from <code>0.0</code> (more precise) to <code>0.9</code> (more diverse). Note that higher temperature can lead to less predictable results.	<code>0.0</code>

Setting	Description	default
Write AI queries to debug log	Logs your AI requests. These entries go to the server log.	false

Anthropic Claude

Note

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

To use Anthropic Claude as an AI provider in CloudBeaver, configure it as follows:

1. As an administrator, sign up on the [Anthropic Console](#)
2. Navigate to the [Dashboard](#) and generate a new API key
3. In CloudBeaver, go to **Model Settings** and insert this key into the **API token** field
4. Choose the model
5. Apply the changes

Model settings

Setting	Description	default
API token	Input your secret key from the OpenAI platform.	
Model	Choose the AI model.	claude-opus-4-1-20250805
Temperature	Control AI's creativity from 0.0 (more precise) to 0.9 (more diverse). Note that higher temperature can lead to less predictable results.	0.0
Context window size	Choose the context size between 2048 and 32768. A larger number allows the AI to use more data for better answers but may slow down response time. Choose based on your balance of accuracy and speed.	
Write AI queries to debug log	Logs your AI requests. These e	false

Setting	Description	default
	entries go to the server log.	

Amazon Bedrock

 Since 25.3

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

To use Amazon Bedrock as an AI provider in CloudBeaver, set it up like this:

1. As an administrator, open the **AWS console** and go to IAM (Identity and Access Management)
2. Create or use an IAM user/account that has permissions to use Bedrock, and generate an Access Key ID and Secret Access Key for this user
3. Insert credentials into CloudBeaver's **Model Settings**
4. Enter the **Region** and **Model**
5. Save changes

Model settings

Setting	Description	default
AWS Access key	Your AWS access key with permissions to call Amazon Bedrock.	
AWS Secret key	Your AWS secret key.	
Region	AWS region where Bedrock is enabled for your account, such as <code>us-east-1</code> or <code>eu-central-1</code> .	
Model	Choose the AI model. Use Load model list to refresh the available models. You can also type a model name manually.	
Context size	A larger number allows the AI to use more data for better answers but may slow down response time.	

Setting	Description	default
Temperature	Control AI's creativity from 0.0 (more precise) to 0.9 (more diverse). Note that higher temperature can lead to less predictable results.	0.0

Data transfer

Data export

CloudBeaver supports exporting data from database tables in various formats. Users can initiate data exports from the [Database Navigator](#), Metadata Editor, [Data Editor](#), or directly from the [SQL Editor's ResultSet](#).

Supported formats

CloudBeaver supports multiple output formats for data export:

- CSV.
- DBUnit.
- JSON.
- Markdown.
- Source Code.
- SQL.
- TXT.
- XML.
- Parquet. ★
- XLSX.
- HTML.

Steps to export data

To export data from CloudBeaver, follow these steps:

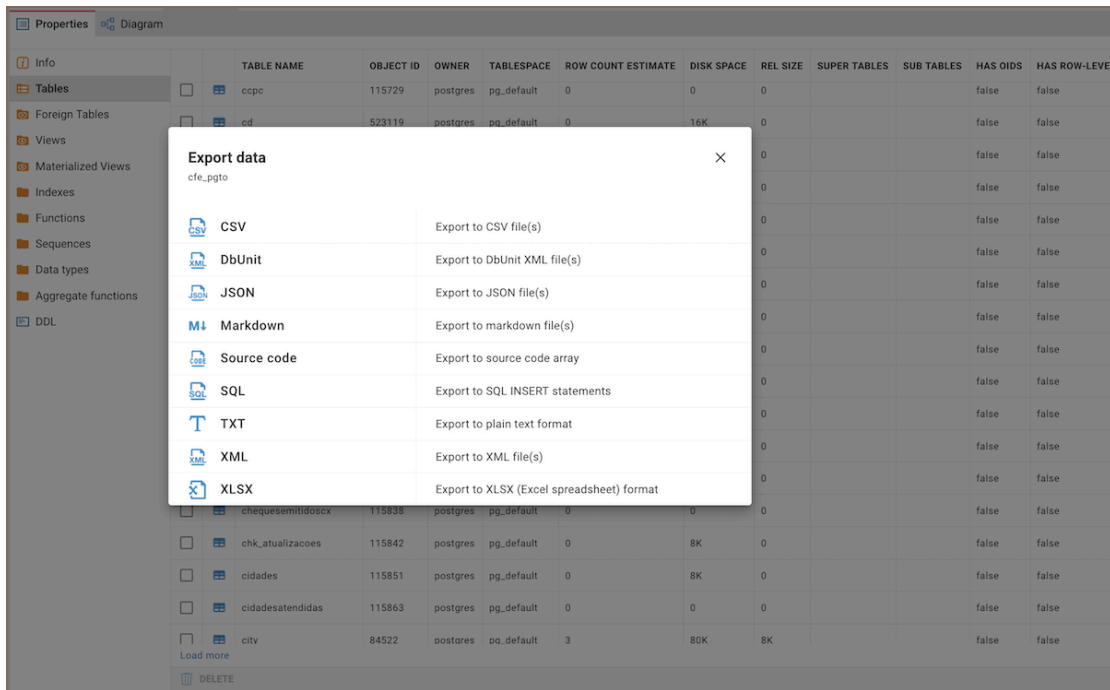
1. Choose the table you wish to export.

Properties	Diagram			TABLE NAME	OBJECT ID	OWNER	TABLESPACE	ROW COUNT ESTIMATE	DISK SPACE	REL SIZE	SUPER TABLES	SUB TABLES	HAS OIDS	HAS ROW-LEVEL
Info														
Tables				ccpc	115729	postgres	pg_default	0	0	0			false	false
Foreign Tables				cd	523119	postgres	pg_default	0	16K	0			false	false
Views				cest	115735	postgres	pg_default	0	0	0			false	false
Materialized Views														
Indexes				cf_referenciados	115740	postgres	pg_default	0	0	0			false	false
Functions				cfe	115746	postgres	pg_default	0	8K	0			false	false
Sequences				cfe_items	115760	postgres	pg_default	0	8K	0			false	false
Data types				cfe_items_obs_fisco	115768	postgres	pg_default	0	8K	0			false	false
Aggregate functions				cfe_pgto	115776	postgres	pg_default	0	8K	0			false	false
DDL														
				cfe_status			pg_default	0	8K	0			false	false
				cfopimpostos			pg_default	0	8K	0			false	false
				chat			pg_default	0	8K	0			false	false
				cheques			pg_default	0	8K	0			false	false
				chequescx			pg_default	0	8K	0			false	false
				chequesemitidoscx	115838	postgres	pg_default	0	0	0			false	false
				chk_atualizacoes	115842	postgres	pg_default	0	8K	0			false	false
				idades	115851	postgres	pg_default	0	8K	0			false	false
				idadesatendidas	115863	postgres	pg_default	0	0	0			false	false
				civ	84522	postgres	pg_default	3	80K	8K			false	false

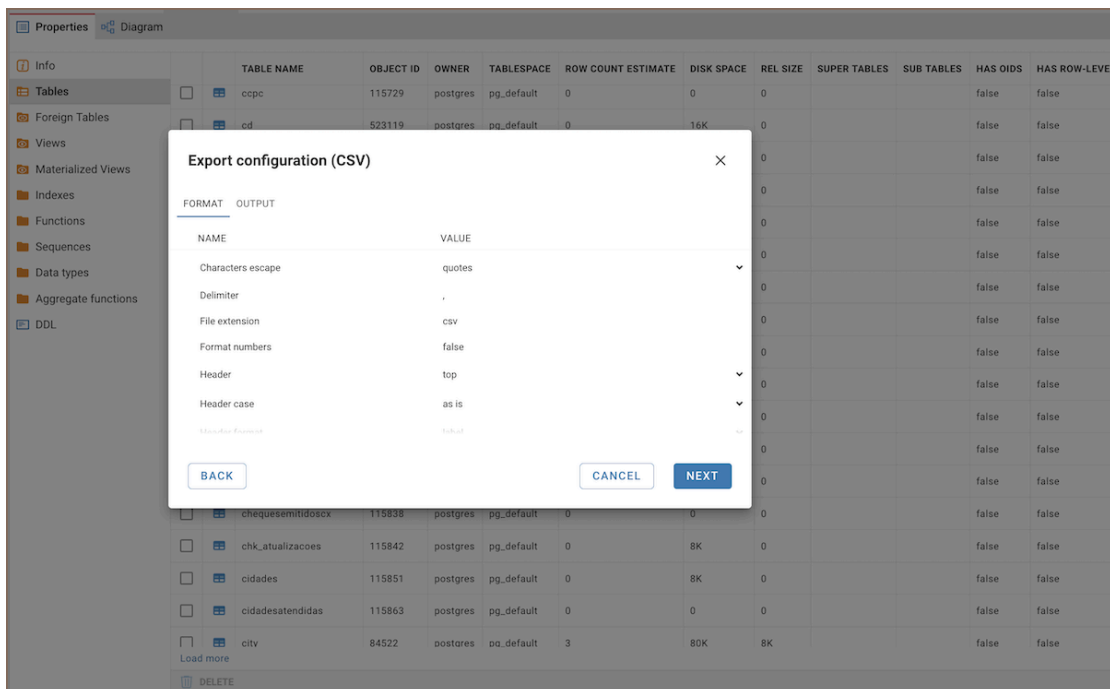
Info

You can initiate an export also from the Database Navigator, Data Editor after querying data, or directly from the ResultSet in the SQL Editor.

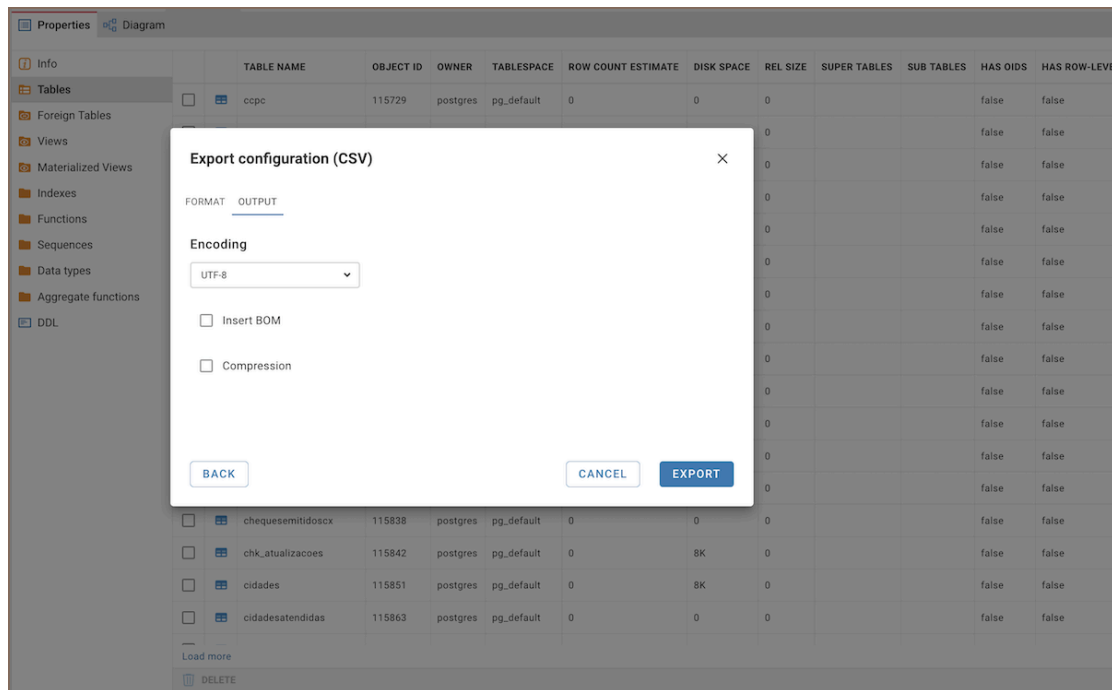
2. Select your desired format from the available options.



- Set the necessary configuration options based on the format you have selected. Each format may have **specific configuration settings** that can be adjusted to meet your needs.



- Once you have set the necessary configuration options for your selected format, move to the "Output" tab to configure the additional settings.



Setting name	Description
Encoding	Set the character encoding for the exported file.
Insert BOM	Choose whether to include a Byte Order Mark in the file.
Compression	Specifies if the export file should be compressed into a <code>.zip</code> archive.

After configuring the settings, click on **Export** to initiate the export process.

- Once you click the **Export** button, a pop-up dialog titled "File is Ready to Download" will appear. You can cancel the process by clicking **Cancel** or proceed with the download by clicking **Download**. Selecting **Download** will initiate the file download to your local system.

Tip

It is advisable not to make changes to the data while the export is in progress to prevent any potential data integrity issues.

Specific configuration settings

In this section, you will find specific configuration settings for each supported export format. These settings allow you to customize the export process according to the requirements of the data format you choose.

CSV

Setting name	Description	Available options
Characters escape	Bad characters escaping model (surrounded with quotes or escaped with '\ character).	quotes / escape
Delimiter	Column delimiter.	You can use special characters like \t, \n, and \r.
File extension	The default file extension for the exported file.	csv
Format numbers	Format numeric values using locale settings.	true / false
Header	CSV header settings.	none / top / bottom
Header case	You can choose lower or upper case for column names or descriptions in the header.	as is / upper / lower
Header format	Defines the formatting of the header.	label / description / both
NULL string	String which will be used instead of NULL values.	

Setting name	Description	Available options
Quote always	Quote all cell values. Cannot be used with "Quote Never".	disabled / all / strings / all but numbers / all but nulls
Quote character	Character which will be used to quote strings (space means no quote).	
Quote never	Do not quote cell values. Cannot be used with "Quote Always".	true / false
Row delimiter	Row delimiter. Default is system-specific line feed delimiter.	default / \n / \r / \r\n / \n\r

DBUnit

Setting name	Description	Available options
File extension	The default file extension for the exported file.	xml
Force upper case column names	Convert all column names to upper case.	true / false
Force upper case table name	Convert the table name to upper case.	true / false
Include NULL values in export	Include NULL values in the exported data.	true / false
Replace NULL values with	Specify a string to replace NULL values in the export.	

JSON

Setting name	Description	Available options
File extension	The default file extension for the exported file.	<code>json</code>
Format dates in ISO 8601	Convert all date values to ISO 8601 format.	<code>true / false</code>
Print table name	Include the table name in the exported file.	<code>true / false</code>
Export JSON values as	Export values as text or as JSON objects.	<code>string / JSON</code>

Markdown

Setting name	Description	Available Options
Confluence format	Enable Confluence-specific Markdown formatting.	<code>true / false</code>
File extension	The default file extension for the exported file.	<code>md</code>
Format numbers	Format numeric values using locale settings.	<code>true / false</code>
NULL string	String to represent NULL values in the export.	
Show header separator	Include a separator line below the header.	<code>true / false</code>

Source code

Setting name	Description	Available options
File extension	The default file extension for the exported file.	<code>php</code>
Format dates in ISO 8601	Convert all date values to ISO 8601 format.	<code>true / false</code>
Language	Specifies the PHP version compatibility for the generated code.	<code>PHP < 5.4</code> , <code>PHP 5.4+</code>
Quote character	Character which will be used to quote strings.	<code>"</code> , <code>'</code>
Row delimiter	Row delimiter. Default is system-specific line feed delimiter.	<code>default</code> , <code>\n</code> , <code>\r</code> , <code>\r\n</code> , <code>\n\r</code>

SQL

Setting name	Description	Available options
Data rows per statement	Specifies the number of data rows in a single insert statement.	<code>integer</code>
File extension	The default file extension for the exported file.	<code>sql</code>
Identifier case	Allows selection of lower or upper keyword case for table and column names.	<code>as is</code> / <code>upper</code> / <code>lower</code>
Include generated columns	Specifies whether to include auto-generated columns (e.g., auto-increment) in SQL INSERT statements.	<code>true / false</code>

Setting name	Description	Available options
Insert line before rows	Specifies inserting a line feed before values in multi-row inserts.	true / false
Keyword case	Allows selection of lower or upper keyword case.	upper / lower
Native date/time format	Specifies using native date/time format in INSERT statements.	true / false
Omit schema name	Specifies omitting schema/catalog name in INSERT statements.	true / false
On conflict expression	Provides an expression for the end of the statement. This setting is specific to the database.	
Target table name	Allows specification of the target table name to generate an INSERT statement.	
Upsert keyword	Allows selection of different upsert keywords.	INSERT / INSERT ALL / UPDATE OR / UPSERT INTO / REPLACE INTO / ON DUPLICATE KEY UPDATE / ON CONFLICT

TXT

Setting name	Description	Available options
Batch size	Specifies the number of records per batch.	integer
File extension	Specifies the file type for output.	txt

Setting name	Description	Available options
In-between delimiter	Adds a custom character between data values.	
Max column length	Specifies the maximum length of data in a column; longer values will be cropped.	integer
Min column length	Specifies the minimum length of data in a column; shorter values will be padded with spaces.	integer
Print header	Specifies whether to print column names at the top of the file.	true / false
Show header delimiter	Adds hyphen characters either in the first row without a header or between the header and data.	true / false
Show leading delimiter	Adds a pipe character at the start of the row.	true / false
Show NULLs	Controls the display of NULL values in the output.	true / false
Show trailing delimiter	Adds a pipe character at the end of the row.	true / false

XML

Setting name	Description	Available options
File extension	Specifies the file type for output.	xml
Include DOCTYPE declaration	Specifies whether to include the DOCTYPE declaration in the XML file export.	true / false

XLSX

Setting name	Description	Available options
Append strategy	Strategy used when appending data to an existing file.	create new sheets / use existing sheets
Boolean string FALSE	String that replaces FALSE boolean values in the exported file.	true / false
Boolean string TRUE	String that replaces TRUE boolean values in the exported file.	true / false
Border style	Style of cell borders in the exported file.	NONE / THIN / THICK
Excel date format	Date and time format used in the Excel file (modifiable in Excel application).	m/d/yy / d-mmm-yy / d-mm-m / mmm-yy / h:mm AM/PM / h:mm:ss AM/PM / h:mm / h:mm:ss / m/d/yy h:mm
Column group	Column number used for grouping rows on a sheet by column value.	integer
Export SQL	Specifies whether to export SQL to a second sheet in the Excel file.	true / false
File extension	File extension for the exported document.	xlsx
Header format	Format of the header in the exported file.	label / description / both / none
Header row font	Font styling for the first row in the exported file.	NONE / BOLD / ITALIC / STRIKEOUT / UNDERLINE
Max row on sheet	Maximum number of rows allowed on a single sheet, after which	integer

Setting name	Description	Available options
	ch data will split into another sheet.	
NULL string	String used instead of displaying NULL values in the exported file.	
Row number(s)	Includes the row index as the first column in the exported file.	true / false
Split SQL Text	Splits exported SQL into rows by CR (Carriage Return) in the exported file.	true / false
Trim strings	Removes extra leading and trailing spaces from all string values in the exported file.	true / false

HTML

Setting name	Description	Available options
File extension	File extension for the exported document.	html
Images	Extracts images to graphic files.	true / false
Output column headers	Outputs column names as an additional row in the generated table.	true / false
Output table header	Outputs the query or table name as the first row in the generated table.	true / false

Data import

You can import data into CloudBeaver from various file formats. Start the import process from the [Data Editor](#).

Supported formats

CloudBeaver supports the following formats for data import:

- CSV
- XML ★
- XLSX ★

Note

The Community Edition supports only CSV files.

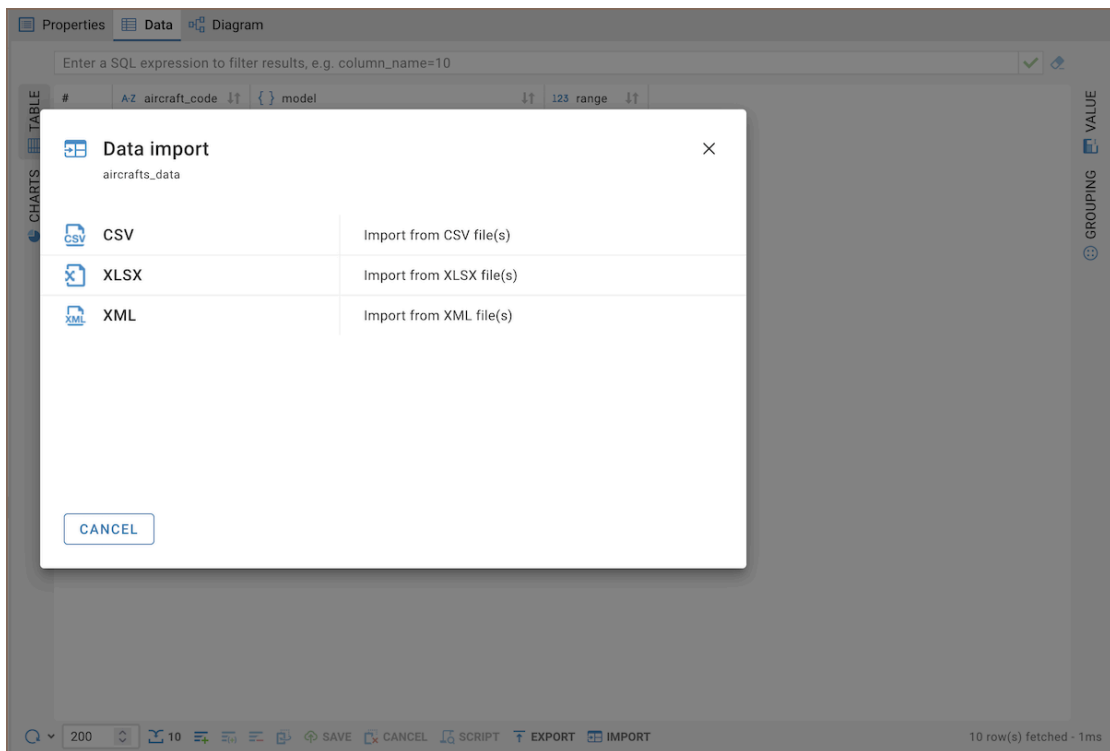
Steps to import data

To import data to the CloudBeaver, follow these steps:

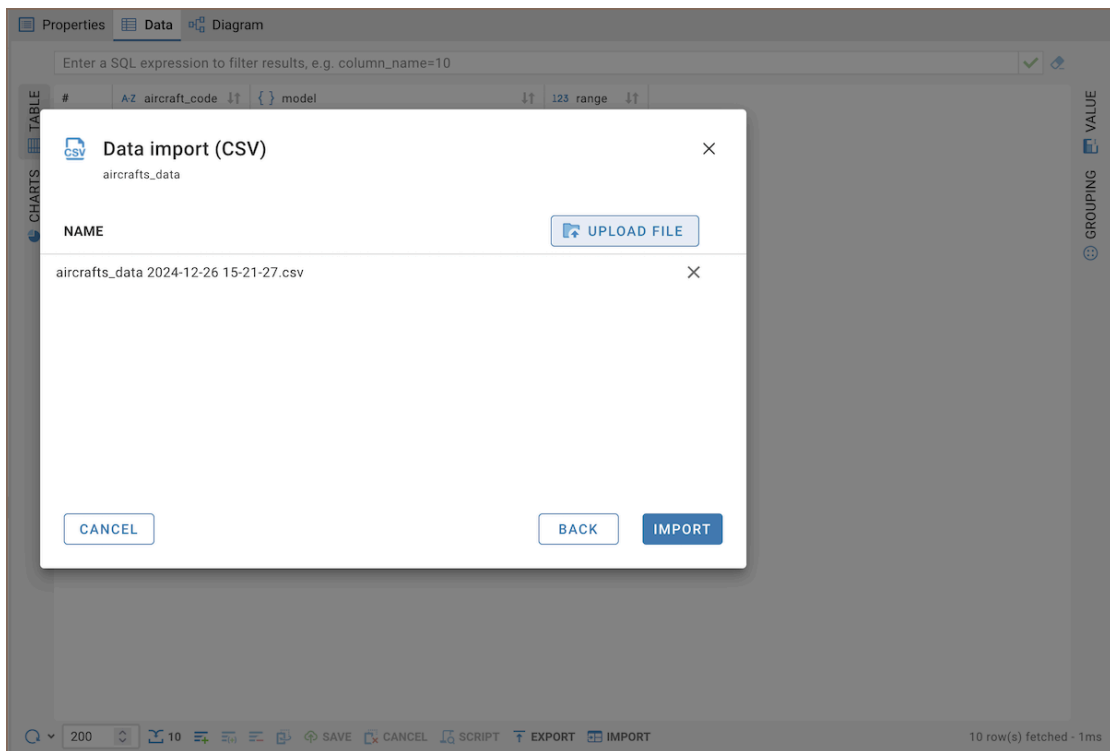
1. Select the table where you want to import data and click **Import** in the bottom toolbar.



2. Pick the format you need from the list of available options.



3. Upload the file from your local storage, then click **Import**.



Note

The import runs in the background, so you can keep working without interruptions.

4. Check for a confirmation message indicating the import was successful.

Configuration details

When importing data:

- Use **CSV** files with **comma delimiters**.
- Make sure the table structure (columns) in your file matches the target table. Mismatched columns will cause errors.
- Ensure primary key values are **unique** to avoid duplicate entries and maintain data integrity.

General user guide

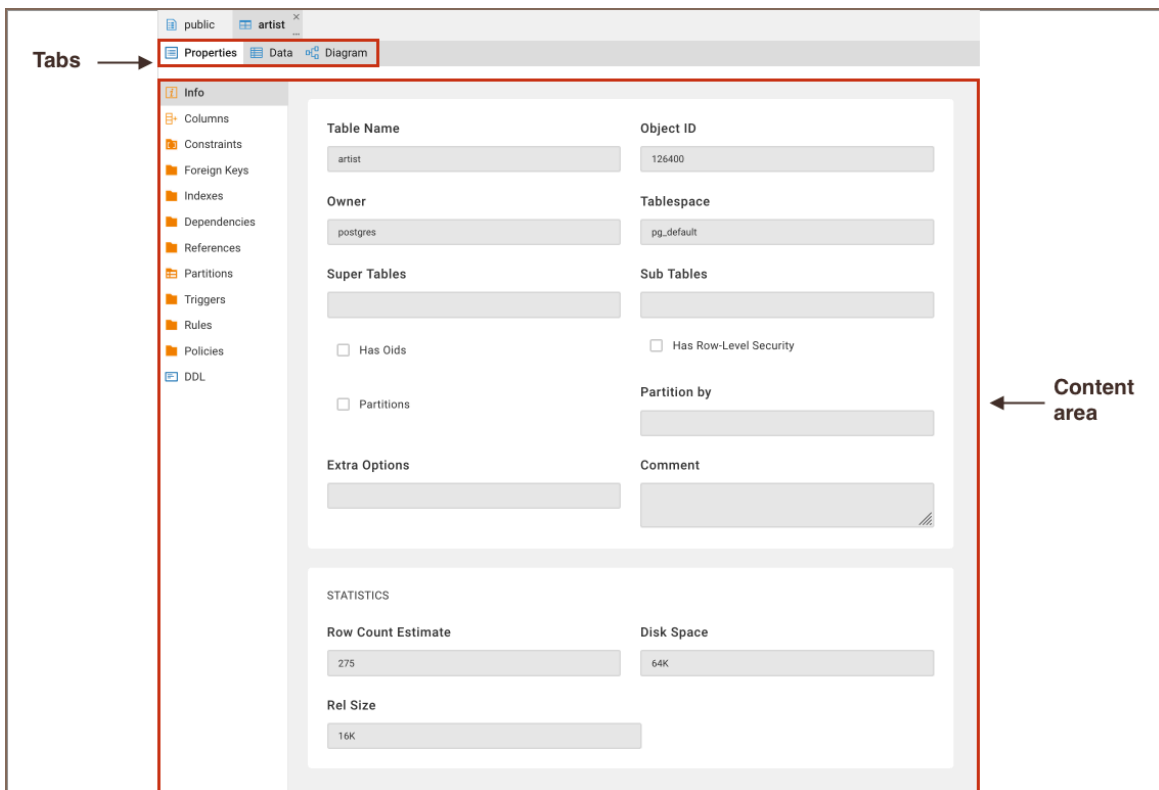
Database Object Editor

The Database object editor, or Metadata editor, is a general-purpose tool that provides a comprehensive interface for working with various database objects such as tables, views, and schemas. It includes several key components that help you interact with database objects:

- **Tabs:** Provide access to specific editors, such as Properties, Data, and Diagrams.
- **Content area:** Shows detailed information about the selected object.
- **Toolbar:** Offers tools specific to each tab to streamline tasks.

To open the Metadata editor for an object, in the **Database Navigator**:

- Double-click the database object.
- Click the database object and choose **Open**.



Tabs

Tab	Description
Properties	Available for all objects, shows the properties of the database object and its sub-entities.
Data	Appears for tables and views, represents the Data Editor.
ER Diagram ★	Appears for tables and schemas, displays ERD (Entity-Relationship Diagrams).

Content area

The Content Area is key part of the Metadata editor. The Content Area shows information about a database object and things like its sub-parts or DDL.

Managing preferences

Managing preferences

You can configure how the application looks and works on two levels:

- **Settings** - managed only by administrators and applied to all users.
- **Preferences** - available to individual users and applied only to their own session.

Settings set the defaults for everyone, while Preferences let each user adjust the application to their needs. Preferences always override global ones for the current user.

In the tables below you can see all available settings.

Tip

Use the **Admin only** column to understand which settings are global (Yes) or local (No).

Interface

Setting	Description	Admin only	Variable values
Language	Sets the application's language.	No	
	Available languages:		
	- English		en
	- Arabic		ar
	- French		fr
	- German		de
	- Italian		it
	- Japanese		jp

Setting	Description	Admin only	Variable values
	- Korean		ko
	- Portuguese (BR)		pt_BR
	- Romanian		ro
	- Russian		ru
	- Simplified Chinese		zh
	- Spanish		es
	- Traditional Chinese		tw
	- Ukrainian		uk
	- Vietnamese		vi
Theme	Application color theme. Available themes:	No	
	- Light		light
	- Dark		dark
	- Device (default)		system
Multiple rows for tabs	Show SQL Editor tabs in multiple rows instead of a single scrollable row	No	

Tip

Variables are part of **Product settings parameters**. For more details, see [Product settings parameters](#).

Data viewer

Setting	Description	Admin only
Show column description	Shows column description in the Data Editor grid .	No
Default fetch size	Default number of rows to fetch.	No
Maximum fetch size	Maximum number of rows to fetch.	Yes

Global permissions for Data viewer

Note

In Enterprise and AWS editions, you can restrict specific actions (for example, copy, edit, import, or export) using global permission settings. For details, see [Global permissions](#).

Setting	Description	Admin only
Disable Import	Disables importing of data in Data Viewer for non-admin users.	Yes
Disable Copy	Disables copying of data in Data Viewer.	Yes
Disable Export	Disables exporting of data in Data Viewer for non-admin users.	Yes
Disable Edit	Disables editing of data in Data Viewer.	Yes

SQL Editor

Setting	Description	Admin only
Autosave script files	Automatically save the script files.	No
Import limit (KB)	Sets the maximum file size that can be imported (in KB).	No
Request dangerous queries confirmation	Warn before running UPDATE or DELETE queries without a WHERE clause.	No
Request DROP TABLE queries confirmation	Warn before running any DROP TABLE query.	No
Completion engine	Selects the engine used for SQL auto-completion. Options:	No
	- Semantic (default)	
	- Legacy	
Insert table aliases	Automatically inserts table aliases in the FROM clause of a SQL query. Options:	No
	- Disable: Do not insert any aliases automatically.	
	- my_table mt: Use a short alias for tables.	
	- my_table AS mt: Use the AS keyword with a short alias for tables.	

Setting	Description	Admin only
Use long object names	Always use schema/catalog for object names.	No

Info

Read more about [SQL Editor](#).

Tools

Setting	Description	Admin only
Disable log viewer	Disables the ability to view logs for all users.	Yes
Disable Tools	Removes Tools menu from the top menu bar.	Yes

Diagram

Setting	Description	Admin only
Max elements to display	Sets the maximum number of elements (tables, columns, etc.) shown in the ERD viewer.	No

Info

Read more about [ERD viewer](#)

Administration

Query Manager

Setting	Description	Admin only
Automatic history cleanup	Automatically deletes old query history once a day. If turned off, history is kept forever. Cleanup removes entire log files based on their last modification date, not individual records. A new log file is created on each server restart. Only complete files are deleted - partial cleanup in side files isn't supported.	Yes
Days to store history	Number of days to keep query history. Files older than the set number of days (based on last modification date) are deleted daily.	Yes
Number of days to retain record files	Deletes record files once per day if they were last modified earlier than the specified number of days.	Yes
Save records to a file	Saves query manager records to a file in the metadata folder (<code>/opt/workspace/.metadata</code>). Customize the path using a <code>.env</code> variable. Read more about environment file configuration.	Yes

Info

Read more about [Query Manager](#).

Logger

Setting	Description	Admin only
Audit all API requests to the database ★	Stores API call logs in the <code>database</code> (<code>qm_api_call_log</code> table, QM schema). Audit logs include: - session ID and User ID - <code>user</code> creation, modification, and logout - <code>resource manager</code> activity - <code>S3/cloud storage</code> actions - all GraphQL and administrative <code>API</code> parameters	Yes
Audit log retention (days) ★	Number of days to keep API call logs. Older logs are deleted automatically once per day. If the field is empty, logs are stored indefinitely.	Yes
Log GraphQL requests to console	Enables detailed logging of GraphQL queries in the server log, including all provided variables.	Yes

Info

Sensitive information such as passwords, private keys, credentials, and tokens is never logged.

Accessing audit logs

You can access logs in two ways:

- **Database** - stored in the `QM_API_CALL_LOG` table of the internal query manager database. For more details, see [How to access the query manager database](#).
- **Browser** - open `http://<your-host>:<your-port>/api/apiCallLog?from=2025-06-26T20:00:00&to=2025-06-26T23:59:59`

Tip

Always apply filters (`from` , `to` , etc.) to avoid overloading the server. You must be logged in as an administrator to view logs in the browser.

See also:

- [Log viewer](#)
- [Log customization](#)
- [Log viewer configuration](#) (server-side settings)

Navigation tree

Setting	Description	Admin only
Amount of nodes to load	Loads this number of child items in the database navigator, scripts, and datasets. Default is <code>1000</code> , maximum is <code>10000</code> .	No

Preferences

Preferences are available to every user. They affect only the current user's session and do not impact others.

Access Preferences

- Go to **Settings** -> **Preferences**, or
- Navigate to **Profile** -> **Preferences**

Preferences are typically used to configure:

- interface language and theme
- SQL Editor options (auto-completion, formatting, aliases)
- data viewer behavior (fetch size, show descriptions)
- navigation tree settings

Tip

For the full list of available settings, see [Preferences](#). Settings marked as **Admin only = No** are local.

Settings

Settings are available only to administrators. They apply to the entire application and affect all users.

Access Settings

- Go to **Settings** -> **Administration** -> **Settings**

Settings are often used to configure:

- security and permissions
- query manager and logging
- default behavior for data viewing and editing

Tip

For the full list of available settings, see [Preferences](#). Settings marked as **Admin only = Yes** are global.

Global permissions

Some actions in the **Data viewer** or **SQL Editor** can be restricted globally for non-admin users. For details, see [Permission management](#).

Accessibility guide

CloudBeaver offers you the ability to use most of its features with a keyboard and screen readers. Here you will find guidelines on how to utilize CloudBeaver effectively without a mouse, ensuring accessibility for all users. This includes options for theme customization, shortcuts and support for screen readers.

Theme settings

You can customize the appearance of CloudBeaver by adjusting the theme settings. This allows you to synchronize the theme with your system settings or choose from a light or dark theme based on your preference. For more information, see [Preferences](#).

Tip

Consider using browser extensions for high contrast themes for even more customization options.

Shortcuts

Use keyboard shortcuts to improve your efficiency in CloudBeaver. For a detailed list of shortcuts, see [Shortcuts](#).

Screen readers support

CloudBeaver is compatible with screen readers, which helps users navigate and interact with the interface effectively. The platform adheres to web accessibility guidelines and supports major screen readers.

To navigate with a screen reader, use the `Tab` key to move the focus forward between elements. Use `Shift + Tab` to move the focus backward.

Note

CloudBeaver is tested primarily with screen readers like VoiceOver and JAWS.

Structure and navigation

CloudBeaver is structured into several components, referred to as Views, such as the **Database Navigator**, **SQL Editor**, and **Data Editor**. Each View contains a **Toolbar** with multiple buttons and features horizontal and vertical tabs for navigating between and within these Views.

To navigate within CloudBeaver using a screen reader, use the `Tab` key to move focus across interactive elements like buttons and tabs. Use arrow keys to move horizontally or vertically between tabs and toolbar options.

When you focus on any interface object, such as a table, column, tab, or cell, you can open the Context menu to access available actions for that object. To open the Context menu, navigate to the context menu button using the `Tab` key and press `Enter`.

When you are within a text area, such as the **SQL Editor** or a **Value Panel**, the behavior of the `Tab` key changes. Instead of navigating to the next interactive element, pressing `Tab` will insert an indent in the typed text. To navigate out of the text area, press `Tab` + `Esc`.

CloudBeaver Views and wizards

Learn how to use the most popular Views in CloudBeaver, focusing on navigation and operation with keyboard shortcuts and screen readers.

Database Navigator

The **Database Navigator** lets you see and manage your database connections and their objects. Use the arrow keys to move between items and press `Enter` to open or close them.

See [Database Navigator shortcuts](#).

New connection

To create a **New Connection**, navigate to **Connection -> New Connection** in the top toolbar using keyboard, and select **Database Connection**.

Properties Editor

The **Properties Editor** shows information about the selected database object. You can move through the properties using the `Tab` key or the arrow keys.

SQL Editor

The **SQL Editor** is where you write and run your SQL scripts. Open it by choosing **Open SQL Editor** from the top toolbar and execute commands with `Ctrl + Enter`.

See [SQL Editor shortcuts](#).

Data Editor

When you focus on the first cell of the **Data Editor** in the result set, screen readers will tell you the table and column name. As you move to different cells, it will announce any change in column names.

See [Data Viewer shortcuts](#).

ER Diagrams

ER Diagrams are readable with screen readers, which will give you feedback as you move through different parts of the diagram. For more information, see [ER Diagrams](#).

Keyboard shortcuts

CloudBeaver offers keyboard shortcuts to help you work more efficiently. You can find these shortcuts within the web application by clicking on the Shortcuts icon  in the top toolbar. You can also refer to the detailed list provided in this article below.

Data Viewer

Action	Shortcut for macOS	Shortcut for Windows/Linux
Revert Inline Editor changes	ESCAPE	ESCAPE
Add a new Row	OPTION + R	ALT + R
Duplicate row	SHIFT + OPTION + R	SHIFT + ALT + R
Delete row	DELETE	DELETE

SQL Editor

Action	Shortcut for macOS	Shortcut for Windows/Linux
Execute SQL statement	CMD + ENTER	CTRL + ENTER
Execute SQL statement in the new Tab	CMD + \ or SHIFT + CMD + ENTER	CTRL + \ or SHIFT + CTRL + ENTER
Execute script	OPTION + X	ALT + X
Show Execution plan	SHIFT + CMD + E	SHIFT + CTRL + E
Format script	SHIFT + CMD + F	SHIFT + CTRL + F
Save as script	CMD + S	CTRL + S

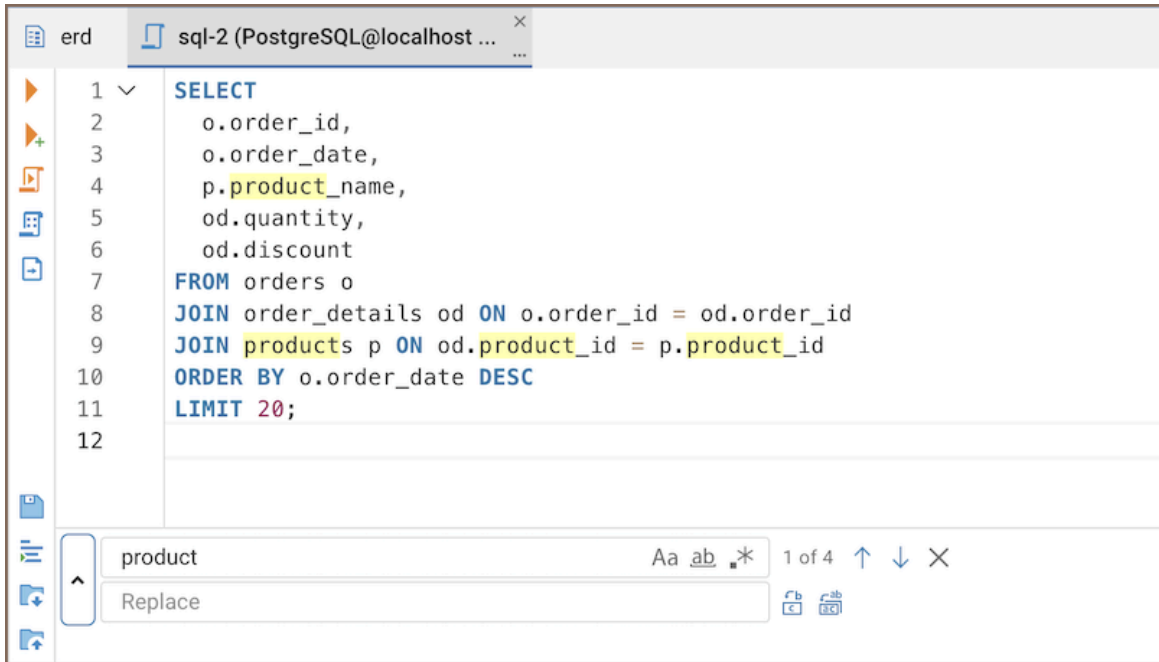
Action	Shortcut for macOS	Shortcut for Windows/Linux
Undo	<code>CMD + Z</code>	<code>CTRL + Z</code>
Redo	<code>CMD + Y</code> or <code>CMD + SHIFT + Z</code>	<code>CTRL + Y</code> or <code>CTRL + SHIFT + Z</code>
Find	<code>CMD + F</code>	<code>CTRL + F</code>
Comment/Uncomment selection	<code>CMD + /</code>	<code>CTRL + /</code>
Open SQL Editor in the separate browser Tab	<code>OPTION + T</code>	<code>ALT + T</code>

Database Navigator

Action	Shortcut for macOS	Shortcut for Windows/Linux
Enable filtering	<code>CMD + F</code>	<code>CTRL + F</code>
Collapse all	<code>CMD + SHIFT + /</code>	<code>CTRL + SHIFT + /</code>
Link with editor	<code>CMD + SHIFT + ,</code>	<code>CTRL + SHIFT + ,</code>

Search

When you're working with data, search is the essential feature. It helps you move faster - whether you're editing SQL queries or updating values.



Search support

In CloudBeaver, you can use it in the following areas:

- [SQL Editor](#)
- [Value panel](#)
- [Visual Query Builder](#) ★

Use search

1. Press `Ctrl+F` (Windows/Linux) or `Cmd+F` (macOS).

Tip

To close the panel, press `Esc`.

2. Fine-tune your search.

Use the toolbar to fine-tune your search and perform replacements:

Button	Description	Shortcut
Aa	Toggle case sensitivity.	
ab	Match whole word.	
.	Enable regular expressions.	
↑	Find previous search result.	Shift+Enter
↓	Find next search result.	Enter
▼	Toggle replace.	
Replace	Replace the selected result.	
Replace all	Replace all results in the current context.	

Administration

Server configuration

Server configuration

This article provides a guide to configuring CloudBeaver, using the main server configuration file (`cloudbeaver.conf`) and additional runtime settings. It explains how to set up various server parameters, including database connections, authentication settings, and resource quotas.

Note

Some configuration values support variables that can be set using command line arguments, optional settings when running Docker Desktop, or defined within the [Docker Compose file](#).

Main server configuration

The main configuration file is `cloudbeaver.conf` . By default, you'll find it in the `/opt/cloudbeaver/` folder. If you want to use a different file, you can specify its location with the command line parameter `-web-config <config-file-path>` .

Info

Use `/opt/cloudbeaver/workspace/.data/.cloudbeaver.runtime.conf` when the server is already running - it has higher priority and overrides settings from `cloudbeaver.conf` .

The server configuration uses a `JSONC` format (like `JSON` , but with comments and no extra quotes), so it's easy for most `JSON` parsers to handle.

Tip

If you're using Docker, the `.cloudbeaver.runtime.conf` file can be especially useful for customizing configurations across different containers easily. All paths can be absolute or relative to the server start directory (or current directory).

Example configuration:

```
{
  forceHttps: "${CLOUDBEAVER_FORCE_HTTPS:false}",
```

```

supportedHosts: [],
server: {
  serverName: "CloudBeaver CE Server",
  serverPort: "${CLOUDBEAVER_WEB_SERVER_PORT:8978}",
  workspaceLocation: "${CLOUDBEAVER_WORKSPACE_LOCATION:workspace}",
  contentRoot: "web",
  driversLocation: "drivers",
  sslConfigurationPath: "${CLOUDBEAVER_SSL_CONF_PATH:workspace/.data/ssl-
config.xml}",
  rootURI: "${CLOUDBEAVER_ROOT_URI:/}",
  serviceURI: "/api/",
  expireSessionAfterPeriod:
"${CLOUDBEAVER_EXPIRE_SESSION_AFTER_PERIOD:1800000}",
  bindSessionToIp: "${CLOUDBEAVER_BIND_SESSION_TO_IP:dbeaverDomains}",
  develMode: "${CLOUDBEAVER_DEVEL_MODE:true}",
  productSettings: {
    "core.localization.language":
"${CLOUDBEAVER_CORE_LOCALIZATION:en}",
    "core.theming.theme": "${CLOUDBEAVER_CORE_THEMING_THEME:light}",
    "plugin.log-viewer.disabled":
"${CLOUDBEAVER_LOG_VIEWER_DISABLED:false}",
    "plugin.log-viewer.logBatchSize":
"${CLOUDBEAVER_LOG_VIEWER_LOG_BATCH_SIZE:1000}",
    "plugin.log-viewer.maxLogRecords":
"${CLOUDBEAVER_LOG_VIEWER_MAX_LOG_RECORDS:2000}",
    "plugin.sql-editor.autoSave":
"${CLOUDBEAVER_SQL_EDITOR_AUTOSAVE:true}",
    "plugin.sql-editor.disabled":
"${CLOUDBEAVER_SQL_EDITOR_DISABLED:false}",
    "plugin.sql-editor.maxFileSize":
"${CLOUDBEAVER_SQL_EDITOR_MAX_FILE_SIZE:10240}",
    "sql.proposals.insert.table.alias":
"${CLOUDBEAVER_SQL_PROPOSALS_INSERT_TABLE_ALIAS:PLAIN}"
  },
  database: {
    driver: "${CLOUDBEAVER_DB_DRIVER:h2_embedded_v2}",
    url: "${CLOUDBEAVER_DB_URL:jdbc:h2:${workspace}/.data/
cb.h2v2.dat}",
    schema: "${CLOUDBEAVER_DB_SCHEMA:''}",
    user: "${CLOUDBEAVER_DB_USER:''}",
    password: "${CLOUDBEAVER_DB_PASSWORD:''}",
    initialDataConfiguration: "${CLOUDBEAVER_DB_INITIAL_DATA:conf/
initial-data.conf}",
    pool: {
      minIdleConnections: "${CLOUDBEAVER_DB_MIN_IDLE_CONNECTIONS:4}",
      maxIdleConnections:
"${CLOUDBEAVER_DB_MAX_IDLE_CONNECTIONS:10}",

```

```

        maxConnections: "${CLOUDBEAVER_DB_MAX_CONNECTIONS:100}",
        validationQuery: "${CLOUDBEAVER_DB_VALIDATION_QUERY:SELECT 1}"
    },
    backupEnabled: "${CLOUDBEAVER_DB_BACKUP_ENABLED:true}"
},
sm: {
    enableBruteForceProtection:
"${CLOUDBEAVER_BRUTE_FORCE_PROTECTION_ENABLED:true}",
    expiredAuthAttemptInfoTtl:
"${CLOUDBEAVER_EXPIRED_AUTH_ATTEMPT_INFO_TTL:60}",
    maxFailedLogin: "${CLOUDBEAVER_MAX_FAILED_LOGINS:10}",
    minimumLoginTimeout: "${CLOUDBEAVER_MINIMUM_LOGIN_TIMEOUT:1}",
    blockLoginPeriod: "${CLOUDBEAVER_BLOCK_PERIOD:300}",
    connectionBruteForce: {
        enabled:
"${CLOUDBEAVER_CONNECTION_BF_PROTECTION_ENABLED:false}",
        maxFailedConnectAttempts:
"${CLOUDBEAVER_CONNECTION_BF_PROTECTION_MAX_ATTEMPTS:5}",
        errorAttemptsPeriodInMinutes:
"${CLOUDBEAVER_CONNECTION_BF_PROTECTION_ERROR_ATTEMPTS_PERIOD:1440}",
        blockTimeInMinutes:
"${CLOUDBEAVER_CONNECTION_BF_BLOCK_PERIOD:60}"
    },
    passwordPolicy: {
        minLength: "${CLOUDBEAVER_POLICY_MIN_LENGTH:8}",
        requireMixedCase:
"${CLOUDBEAVER_POLICY_REQUIRE_MIXED_CASE:true}",
        minNumberCount: "${CLOUDBEAVER_POLICY_MIN_NUMBER_COUNT:1}",
        minSymbolCount: "${CLOUDBEAVER_POLICY_MIN_SYMBOL_COUNT:0}"
    }
},
},
qm: {
    driver: "${CLOUDBEAVER_QM_DB_DRIVER:h2_embedded_v2}",
    url: "${CLOUDBEAVER_QM_DB_URL:jdbc:h2:${workspace}/.metadata/qmdb/
qmdb}",
    user: "${CLOUDBEAVER_QM_DB_USER:''}",
    password: "${CLOUDBEAVER_QM_DB_PASSWORD:''}",
    schema: "${CLOUDBEAVER_QM_DB_SCHEMA:''}",
    pool: {
        minIdleConnections: "${CLOUDBEAVER_QM_DB_MIN_IDLE_CONNECTIONS:4}",
        maxIdleConnections: "${CLOUDBEAVER_QM_DB_MAX_IDLE_CONNECTIONS:10}",
        maxConnections: "${CLOUDBEAVER_QM_DB_MAX_CONNECTIONS:100}",
        validationQuery: "${CLOUDBEAVER_QM_DB_VALIDATION_QUERY:SELECT 1}"
    },
    backupEnabled: "${CLOUDBEAVER_QM_DB_BACKUP_ENABLED:true}"
},
},

```



```

    app: {
      anonymousAccessEnabled:
"${CLOUDBEAVER_APP_ANONYMOUS_ACCESS_ENABLED:false}",
      anonymousUserRole: "user",
      supportsCustomConnections:
"${CLOUDBEAVER_APP_SUPPORTS_CUSTOM_CONNECTIONS:false}",
      publicCredentialsSaveEnabled:
"${CLOUDBEAVER_APP_PUBLIC_CREDENTIALS_SAVE_ENABLED:true}",
      adminCredentialsSaveEnabled:
"${CLOUDBEAVER_APP_ADMIN_CREDENTIALS_SAVE_ENABLED:true}",
      enableReverseProxyAuth: false,
      forwardProxy: "${CLOUDBEAVER_APP_FORWARD_PROXY:true}",
      linkExternalCredentialsWithUser: true,
      redirectOnFederatedAuth: false,
      resourceManagerEnabled:
"${CLOUDBEAVER_APP_RESOURCE_MANAGER_ENABLED:true}",
      showReadOnlyConnectionInfo:
"${CLOUDBEAVER_APP_READ_ONLY_CONNECTION_INFO:false}",
      grantConnectionsAccessToAnonymousTeam:
"${CLOUDBEAVER_APP_GRANT_CONNECTIONS_ACCESS_TO_ANONYMOUS_TEAM:false}",
      systemVariablesResolvingEnabled:
"${CLOUDBEAVER_SYSTEM_VARIABLES_RESOLVING_ENABLED:false}",
      defaultUserTeam: "${CLOUDBEAVER_APP_DEFAULT_USER_TEAM:user}",
      apiCallLoggingEnabled: "${CLOUDBEAVER_API_CALL_LOGGING_ENABLED:false}",
      resourceQuotas: {
        resourceManagerFileSizeLimit:
"${CLOUDBEAVER_RESOURCE_QUOTA_RESOURCE_MANAGER_FILE_SIZE_LIMIT:500000}",
        sqlMaxRunningQueries:
"${CLOUDBEAVER_RESOURCE_QUOTA_SQL_MAX_RUNNING_QUERIES:100}",
        sqlResultSetRowsLimit:
"${CLOUDBEAVER_RESOURCE_QUOTA_SQL_RESULT_SET_ROWS_LIMIT:100000}",
        sqlTextPreviewMaxLength:
"${CLOUDBEAVER_RESOURCE_QUOTA_SQL_TEXT_PREVIEW_MAX_LENGTH:4096}",
        sqlBinaryPreviewMaxLength:
"${CLOUDBEAVER_RESOURCE_QUOTA_SQL_BINARY_PREVIEW_MAX_LENGTH:261120}"
      },
      defaultNavigatorSettings: {
        showOnlyEntities: true,
        hideFolders: true,
        hideVirtualModel: true
      },
      enabledFeatures: [],
      enabledAuthProvider: [
        "local"
      ],
      enabledDrivers: [],
      disabledDrivers: [

```

```

        "h2:h2_embedded",
        "h2:h2_embedded_v2",
        "clickhouse:yandex_clickhouse",
        "generic:duckdb_jdbc",
        "sqlite:sqlite_jdbc"
    ]
}
}

```

Server parameters

Configure main server settings directly in the file or using environment variables (when available).

Name	Environment variable	Default value	Units	Description
<code>serverPort</code>	<code>CLOUDBEAVER_WEB_SERVER_PORT</code>	<code>8978</code>	port number	Port CloudBeaver server listens on.
<code>workspaceLocation</code>	<code>CLOUDBEAVER_WORKSPACE_LOCATION</code>	<code>workspace</code>	URL	Root folder for projects.
<code>contentRoot</code>	Not available	<code>web</code>	path	Path to directory with static content.
<code>driversLocation</code>	Not available	<code>drivers</code>	path	Optional path for driver <code>.jar</code> files.
<code>rootURI</code>	<code>CLOUDBEAVER_ROOT_URI</code>	<code>/</code>	URI path	Web application URI prefix. <code>/</code> by default.
<code>serviceURI</code>	Not available	<code>/api/</code>	URI path	Services API URL prefix (relative to <code>rootURI</code>). <code>/api/</code> by default. For more information on Server

Name	Environment variable	Default value	Units	Description
				API, see API explorer .
<code>productConfiguration</code>	Not available	-	file path	Path to product (web interface) configuration file (JSON).
<code>sslConfigurationPath</code>	<code>CLOUDBEAVER_SSL_CONF_PATH</code>	<code>workspace/.data/ssl-config.xml</code>	file path	Path to the SSL configuration file for HTTPS connections.
<code>devMode</code>	<code>CLOUDBEAVER_DEVEL_MODE</code>	<code>true</code>	boolean	When set to true, extra debug information is printed in logs, and the GraphQL console is enabled on the server.
<code>expireSessionAfterPeriod</code>	<code>CLOUDBEAVER_EXPIRE_SESSION_AFTER_PERIOD</code>	<code>1800000</code>	milliseconds	Maximum idle time after which the user's session will be closed.
<code>forceHttps</code>	<code>CLOUDBEAVER_FORCE_HTTPS</code>	<code>false</code>	boolean	Enforces secure connections. For details, see Force HTTPS mode .
<code>bindSessionToIp</code>	<code>CLOUDBEAVER_BIND_SESSION_TO_IP</code>	<code>dbeaverDomains</code>	string	Bind user sessions to client IP addresses. For details, see Bind session IDs to IP addresses .

WebSockets

CloudBeaver uses a persistent WebSocket for real-time events and async task status in the application. The socket shares the same host and port as the site and switches to `wss` when HTTPS is enabled. No extra domains or ports are required.

For configuration details and troubleshooting, see [WebSockets](#).

Product settings

For details on configuring `productSettings`, see the [Product settings parameters](#).

Database configuration

Configures the database for storing users, credentials, and permissions in `server.database`.

Name	Environment variable	Default value	Units	Description
<code>driver</code>	<code>CLOUDBEAVER_DB_DRIVER</code>	<code>h2_embedded_v2</code>	driver ID	Database driver (e.g., <code>h2_embedded</code> , <code>postgresql-jdbc</code>).
<code>url</code>	<code>CLOUDBEAVER_DB_URL</code>	<code>jdbc:h2:\${workspace}/.data/cb.h2v2.dat</code>	JDBC URL	Database JDBC URL (e.g., <code>jdbc:postgresql://localhost:5432/cb</code>).
<code>schema</code>	<code>CLOUDBEAVER_DB_SCHEMA</code>	-	string	Specifies the database schema where CloudBeaver will store its data (e.g., <code>public</code>).
<code>user</code>	<code>CLOUDBEAVER_DB_USER</code>	-	string	Database user name.
<code>password</code>	<code>CLOUDBEAVER_DB_PASSWORD</code>	-	string	Database user password.

Name	Environment variable	Default value	Units	Description
<code>initialDataConfiguration</code>	<code>CLOUDBEAVER_DB_INITIAL_DATA</code>	<code>conf/initial-data.conf</code>	file path	Path to the admin configuration and team data file (<code>initial-data.conf</code>) that will be loaded the first time the server runs. For more details, see Initial data configuration guide .
<code>backupEnabled</code>	<code>CLOUDBEAVER_DB_BACKUP_ENABLED</code>	<code>true</code>	boolean	Enables or disables backups of the database.

Info

For more information on CloudBeaver's database, see [Server Database](#)

Database connection pool configuration

Configure the connection pool used by the CloudBeaver database in `server.database.pool` :

Name	Environment variable	Default value	Units	Description
<code>validationQuery</code>	<code>CLOUDBEAVER_DB_VALIDATION_QUERY</code>	<code>SELECT 1</code>	SQL query	Query that checks the successful connection to the database.
<code>minIdleConnections</code>	<code>CLOUDBEAVER_DB_MIN_IDLE_CONNECTIONS</code>	<code>4</code>	connections	Minimum number of idle connections that should be kept in the pool.

Name	Environment variable	Default value	Units	Description
<code>maxIdleConnections</code>	<code>CLOUDBEAVER_DB_MAX_IDLE_CONNECTIONS</code>	<code>10</code>	connections	Maximum number of idle connections that should be kept in the pool.
<code>maxConnections</code>	<code>CLOUDBEAVER_DB_MAX_CONNECTIONS</code>	<code>100</code>	connections	Maximum number of idle and active connections that should be kept in the pool.

Application parameters

In the section `app` :

Name	Environment variable	Default value	Units	Description
<code>anonymousAccessEnabled</code>	<code>CLOUDBEAVER_APP_ANONYMOUS_ACCESS_ENABLED</code>	<code>false</code>	boolean	Allows anonymous access. Anonymous users are assigned to the <code>anonymousUserTeam</code> .
<code>anonymousUserRole</code>	Not available	<code>user</code>	string	Defines which role anonymous users receive by default.
<code>supportsCustomConnections</code>	<code>CLOUDBEAVER_APP_SUPPORTS_CUSTOM_CONNECTIONS</code>	<code>false</code>	boolean	Allows users to create custom connections to any databases. Otherwise, only the CloudBeaver administrator can create/edit c

Name	Environment variable	Default value	Units	Description
				connections.
publicCredentialsSaveEnabled	CLOUDBEAV R_APP_PUBLI C_CREDENTIAL S_SAVE_ENABL ED	true	boolean	Allows saving us er database cre dentials in a loc al cache.
adminCredentialsSaveEnabled	CLOUDBEAV R_APP_ADMIN C_CREDENTIAL S_SAVE_ENABL ED	true	boolean	Allows saving gl obal database cr edentials in a lo cal cache.
forwardProxy	CLOUDBEAV R_APP_FORWAR D_PROXY	true	boolean	Identifies the or iginating IP addr ess and other he aders of a client connecting thro ugh an HTTP pr oxy.
redirectOnFederatedAuth	Not available	false	boolean	If there's only o ne federation au thentication con figuration, login attempts autom atically when th e application op ens.
resourceManagerEnabled	CLOUDBEAV R_APP_RESOUR CE_MANAGER_E NABLED	true	boolean	Enables or disab les the resource manager functio nality in the app lication.
showReadOnlyConnectionInfo	CLOUDBEAV R_APP_READ_O NLY_CONNECTI	false	boolean	Shows connecti on information i f the user has re

Name	Environment variable	Default value	Units	Description
	ON_INFO			ad-only permission for it.
grantConnectionsAccessToAnonymousTeam	CLOUDBEAV R_APP_GRAN T_CONNECTION S_ACCESS_T O_ANONYMOU S_TEAM	false	boolean	Provides access to predefined shared connections for the anonymous team.
systemVariablesResolvingEnabled	CLOUDBEAV R_SYSTEM_VAR IABLES_RESOL VING_ENABLED	false	boolean	Enables using environment variables in the connection configuration.
defaultUserTeam	CLOUDBEAV R_APP_DEFAULT USER_TEAM	user	string	The team assigned to the anonymous user, <code>user</code> by default.
enabledAuthProvider	Not available	["local"]	list of strings	List of allowed authorization providers. If absent, all providers are allowed. For more information on providers, see Identity providers .
enabledDrivers	Not available	[]	list of strings	List of allowed drivers. If the list is empty, all drivers are allowed. For more details on drivers, see Driver management .

Name	Environment variable	Default value	Units	Description
<code>disabledDrivers</code>	Not available	<code>["h2:h2_embedded", "h2:h2_embedded_v2", "clickhouse:yandex_clickhouse", "generic:duckdb_jdbc", "sqlite:sqlite_jdbc"]</code>	list of strings	List of prohibited drivers. If the list is empty, all drivers are allowed.
<code>defaultNavigatorSettings</code>	Not available	<code>showOnlyEntities: true, hideFolders: true, hideVirtualModel: true</code>	object	Default database navigator settings. For more information, see Navigator settings .
<code>apiCallLoggingEnabled</code>	<code>CLOUDBEAVR_API_CALL_LOGGING_ENABLED</code>	<code>false</code>	boolean	Enables writing API call logs into the internal QM database (<code>qm_api_call_log</code> table). For details, see Logger settings .

Resource quotas

Configure the following resource quotas in the `app.resourceQuotas` section:

Name	Environment variable	Default value	Units	Description
<code>resourceManagerFileSizeLimit</code>	<code>\${CLOUDBEAVR_RESOURCE_QUOTA_RESOURCE_MANAGER_FILE_SIZE_LIMIT}</code>	<code>500000</code>	bytes	Maximum file size saved in the resource manager.

Name	Environment variable	Default value	Units	Description
<code>sqlMaxRunningQueries</code>	<code>\${CLOUDBEAV R_RESOURCE_Q UOTA_SQL_MA X_RUNNING_QU ERIES}</code>	100	queries	Maximum number of simultaneous queries for a single user session, including data read queries (e.g., table data view).
<code>sqlResultSetRowsLimit</code>	<code>\${CLOUDBEAV R_RESOURCE_Q UOTA_SQL_RES ULT_SET_ROW S_LIMIT}</code>	100000	rows	Maximum number of rows to select from a table or query.
<code>sqlTextPreviewMaxLength</code>	<code>\${CLOUDBEAV R_RESOURCE_Q UOTA_SQL_TEX T_PREVIEW_MA X_LENGTH}</code>	4096	bytes	Maximum size for a text file shown in the value panel.
<code>sqlBinaryPreviewMaxLength</code>	<code>\${CLOUDBEAV R_RESOURCE_Q UOTA_SQL_BIN ARY_PREVIEW W_MAX_LENGTH}</code>	261120	bytes	Maximum size for a binary file (affects JSON in SQLite) shown in the value panel.

Info

For more information on Resource Manager, see [our documentation](#).

Navigator settings

Configure the following properties in the `app.defaultNavigatorSettings` section:

Name	Default value	Units	Description
<code>showSystemObject</code>	-	boolean	Show system objects.

Name	Default value	Units	Description
<code>s</code>			
<code>showUtilityObjects</code>	-	boolean	Show "utility" objects.
<code>showOnlyEntities</code>	<code>true</code>	boolean	Only show schemas and tables.
<code>mergeEntities</code>	-	boolean	Show all types of database objects in one list (e.g., tables, sequences).
<code>hideFolders</code>	<code>true</code>	boolean	Hide folders like "Tables", "Schemas", "Columns".
<code>hideSchemas</code>	-	boolean	Do not show schemas (all tables in one list).

Example configuration:

```
defaultNavigatorSettings: {
  showOnlyEntities: true,
  hideFolders: true,
  hideVirtualModel: true
}
```

Info

For more information on Database Navigator, see [our documentation](#).

Initial data

For details on configuring initial data, see the [Initial data configuration guide](#).

Password policy

To configure password policies, refer to the [Password policy guide](#).

Datasource configuration

For information on configuring server datasources, visit the [Datasource configuration guide](#).

Query manager database configuration

To learn about configuring the query manager database, see the [Query manager database configuration documentation](#).

Automatic server configuration

When you start the CloudBeaver server for the first time, an administrator interface guides you through the server setup. However, in some cases-like when running in a Kubernetes environment-you might need to configure the server automatically.

To set up the server automatically, specify the following parameters:

Name	Description	Example
<code>CB_SERVER_NAME</code>	The server's name.	<code>Test Server</code>
<code>CB_SERVER_URL</code>	The base URL for the server.	<code>https://cloudbeaver.domain.com:10000/</code>
<code>CB_ADMIN_NAME</code>	Administrator username.	<code>administrator</code>
<code>CB_ADMIN_PASSWORD</code>	Administrator password.	<code>S0mePazzword</code>

How to specify these parameters:

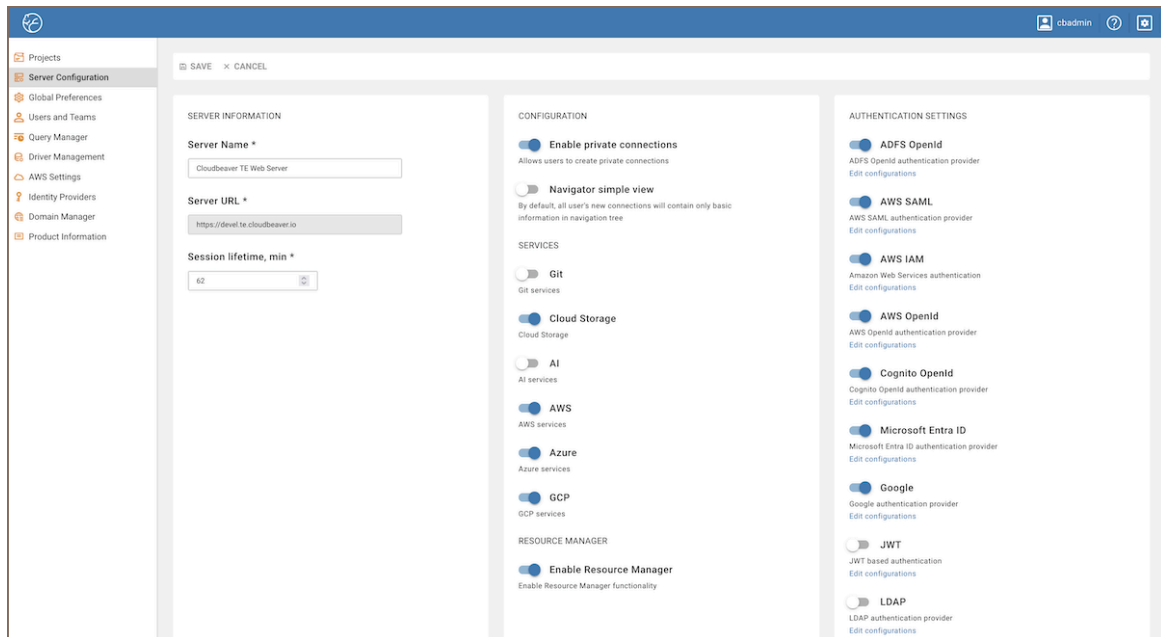
- **OS environment variables:** Set these variables directly in your system's environment settings.
- **Configuration file:** Add them to a `.cloudbeaver.auto.conf` file, located in the same directory as the `cloudbeaver.conf` file.

Web server configuration

Overview

The Server Configuration panel in CloudBeaver enables administrators to manage server settings important for the server. The panel is structured into distinct sections, each dedicated to specific aspects of server management.

To access the Server Configuration, navigate as administrator to **Settings -> Administration -> Server Configuration**.



The sections within the Server Configuration include:

- **Server Information:** Specifies essential server details.
- **Configuration:** Manages general configurations.
- **Authentication Settings:** Controls the methods for authentication.
- **Security:** Manages the storage of credentials.
- **Disabled drivers:** Allows to disable specific drivers on the server.

Note

The availability and specific options of the settings in the Server Configuration panel may vary between different versions of CloudBeaver.

Server information

Setting name	Description
Server Name	The name assigned to the server for identification purposes.
Server URL	The URL through which the server can be accessed.
Session lifetime	Specify the duration, in minutes, for which a session remains active before expiring.

Configuration

Setting name	Description	Additional article
Enable private connections	Allows users to create private connections that are not visible to other users.	Anonymous access configuration.
Navigator simple view	Shows only basic information in new connections within the navigation tree.	Simple and advanced view.

Services

Setting name	Description	Additional article
AI Integration	Enables AI-powered features throughout the application, including AI Chat, database object descriptions, intelligent SQL query optimization, and error correction.	AI smart assistance
Allow drivers downloading	Enables downloading JDBC drivers	

Setting name	Description	Additional article
	vers from external sources (Maven, GitHub).	
Cloud (AWS)	Enables AWS cloud services.	AWS Settings
Cloud (Azure)	Enables Microsoft Azure cloud services.	Cloud Explorer
Cloud (Google)	Enables Google Cloud services.	Cloud Explorer
Cloud Storage	Enables cloud file systems (e.g. S3). Users can access files in cloud storages if they have appropriate permissions.	Cloud Storage
DBeaver Proxy Driver	Adds DBeaver Proxy Driver protocol support. It allows using JDBC proxy to connect to databases accessible to DBeaver server.	

Resource management

Setting name	Description	Additional article
Enable Resource Manager	Activates the Resource Manager, allowing users to store and manage scripts on the server.	Resource manager.
Secret Management 	Enables Secret Management functionality.	Secret Management

Authentication settings

The Authentication Settings section allows enabling various authentication methods that can later be configured in the **Identity Provider** tab.

Tip

For more information, see [Authentication methods](#).

Security

Setting name	Description
Save credentials	Allows saving credentials for pre-configured data bases.
Save users credentials	Allows saving credentials for non-admin users.
Bind session to IP	Binds user sessions to client IP addresses. For details, see section below .

Bind session to IP

Binding sessions to IPs links each user session to their client's IP address. This prevents a stolen session cookie from being reused from another network and makes account sharing harder.

This setting links a user session to the client's IP address. If binding is enabled, users may be logged out when their IP address changes (for example, behind NAT or when switching networks).

Info

You can also configure this option with the `CLOUDBEAVER_BIND_SESSION_TO_IP` environment variable. For more information, see [Environment variables configuration](#).

Value	Environment variable	Description
Enable	<code>true</code>	Always bind sessions to IP addresses
Disable	<code>false</code>	Never bind
Enable for databases.team	<code>dbeaverDomains</code>	Used only for <code>*.databases.team</code> deployments (generated by the Domain Manager)

Note

In Community Edition, this option can be configured only with the `CLOUDBEAVER_BIND_SESSION_TO_IP` environment variable (default is `false`). It's not available in the web client.

Disabled drivers

The Disabled Drivers section allows administrators to control the availability of database drivers on the server.

- **Search for the driver:** Enter the name of a specific driver in the search field to quickly locate it.
- **Disabling a driver:** Select the desired driver from the search results to disable it.
- **Enabling a driver:** To re-enable a disabled driver, click on the `X` next to its name in the list.

Tip

For more information on how to manage these settings through the configuration file, see [Server configuration](#).

How to start

Upon launching the application for the first time, it necessitates an initial setup process. This setup primarily involves configuring specific aspects of the application to ensure optimal functionality.

Initial server configuration

The initial configuration is a vital step to ensure that the application functions correctly and according to your preferences.

The base application settings require specific essential configurations on initial launch. This includes the following components: administrator account, server info, and base application behavior.

The screenshot displays the 'INITIAL SERVER CONFIGURATION' window. The left sidebar shows a navigation menu with 'Welcome', 'Server configuration' (selected), and 'Confirmation'. The main content area is titled 'Main server configuration' and includes a 'BACK' button and a 'NEXT' button. A warning message states: 'You will be able to add additional services after the server configuration. Administrator is a super user who can configure server, set databases connections, manage other users and much more. Please, remember the entered password. It is not possible to recover administrator password automatically.'

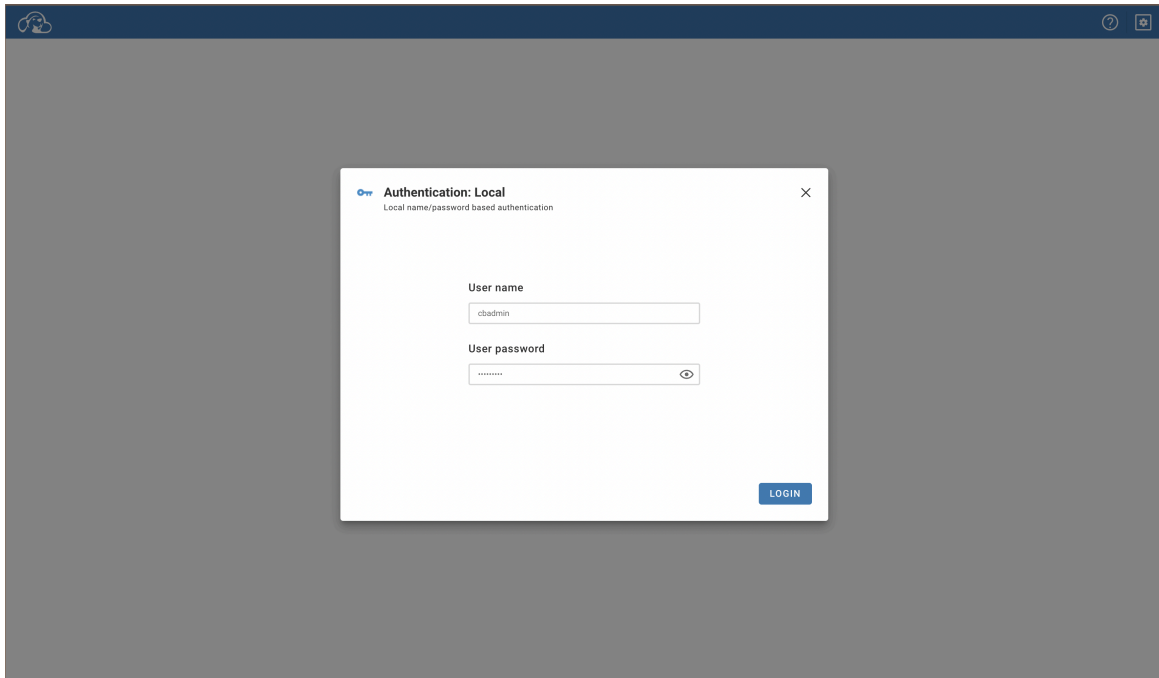
The configuration is divided into several sections:

- SERVER INFORMATION:** Includes fields for 'Server Name *' (pre-filled with 'CloudBeaver CE Server'), 'Server URL *' (pre-filled with 'http://localhost:3100'), and 'Session lifetime, min *' (pre-filled with '30').
- CONFIGURATION:** Includes toggle switches for 'Enable private connections' (disabled), 'Navigator simple view' (disabled), and 'Enable Resource Manager' (enabled).
- ADMINISTRATOR CREDENTIALS:** Includes fields for 'Login *' (pre-filled with 'cbadmin') and 'Password *' (with a password strength indicator).
- AUTHENTICATION SETTINGS:** Includes toggle switches for 'Allow anonymous access' (disabled) and 'Local' (enabled).
- DISABLED DRIVERS:** Includes a search bar and a list of disabled drivers: 'SQLite X', 'H2 Embedded X', 'H2 Embedded V2 X', and 'ClickHouse (Legs... X'.

Entering administration page

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

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



If you see the error "Unsupported authentication provider: local" then you need to enable local authentication in the server configuration and restart the server: [Server Configuration Documentation](#):

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

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

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

Access Management

On the administration page, select the **Users and Teams** tab.

- In the **Users** tab, you can create, delete, and edit users. For details, see [Users](#).
- In the **Teams** tab, you can manage team memberships and permissions. For details, see [Teams](#).

Server configuration

On the administration page, select the **Server configuration** tab. On the **Server configuration** tab, you can edit the base application settings. You can read more about it [here](#).

Preferences

On the administration page, select the **Preferences** tab. On the **Preferences** tab, you can customize settings for the interface, Data Viewer, SQL Editor, and Tools. You can find initial guidelines in [Preferences](#) article.

Server architecture

The CloudBeaver server is a Java application. It is based on the OSGI framework and utilizes several Eclipse bundles.

Important

CloudBeaver includes OpenJDK, which is an open-source version of Java. This means you do not need to purchase a separate Java license to use CloudBeaver.

Back-end parts depend on [DBeaiver](#).

DBeaiver is a multiplatform desktop application. It is also based on OSGI/Eclipse. CloudBeaver uses a "headless" part of it - everything related to database connectivity and project management.

Libraries

- [Equinox](#) as OSGi container.
- [Eclipse Platform](#) for virtual file system and plugin management.
- [Jetty](#) as a web server.
- [Java GraphQL](#) as a front-end communication API.
- [DBeaiver](#) as database management platform.
- Various 3rd party bundles with database drivers

Supported databases

In the CloudBeaver Community we support only free and open-source databases. For more information, see [Supported databases](#)

WebSockets

CloudBeaver uses a persistent WebSocket for real-time events. With a healthy socket, the UI updates live, SSO completes, and background jobs report progress. If the socket is blocked, the UI may look normal but never refresh.

Important

Default [deployments](#) already include WebSocket support. For requirements and troubleshooting, see [WebSockets](#).

Server database

Note

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

By default, the H2 database is used as the internal database for the Community, Enterprise, and AWS Editions of CloudBeaver. The Team Edition employs PostgreSQL.

Starting with version `24.2.0`, PostgreSQL becomes the default database in the Docker Compose configurations we provide for the Enterprise and AWS Editions. The Community Edition continues to use the H2 database.

If you have previously used the H2 database and are transitioning to new Docker Compose setup, an automatic migration to PostgreSQL will be triggered. This migration is designed to seamlessly transfer all server data from H2 to PostgreSQL with minimal intervention required.

Warning

Update only the database login and password. Any other changes will break the migration.

Automated migration from H2 to PostgreSQL

To ensure the automated migration from H2 to PostgreSQL occurs, you need to use the version `24.2.0` or later. This migration is necessary for older installations. If it's a new deployment, migration is not required because there is no old data and PostgreSQL will be used immediately.

Info

For deployment details, refer to the repository: [CloudBeaver deploy](#).

Migration conditions

For the migration to be initiated, the following conditions must be met:

- CloudBeaver must be set to use the standard H2 configuration.
 - `${workspace}/.data/cb.h2v2.dat` should be a valid standard database.
 - `${workspace}/.data/my_db_name.h2v2.dat` is a custom H2 database which will not trigger migration as it indicates manual customization by the administrator.
 - `${workspace}/.data111111/cb.h2v2.dat` is also considered a custom database.

Environment configuration

Standard environment values for database connection must be in place:

- **Driver:** `CLOUDBEAVER_DB_DRIVER=postgres-jdbc`
- **URL:** `CLOUDBEAVER_DB_URL=jdbc:postgresql://postgres:5432/cloudbeaver` (changes to the host or database name prevent migration).
- **Schema:** `CLOUDBEAVER_DB_SCHEMA=cb` (must not be changed).

The following Query Manager (QM) environment settings should match:

- `CLOUDBEAVER_QM_DB_DRIVER=${CLOUDBEAVER_DB_DRIVER}`
- `CLOUDBEAVER_QM_DB_URL=${CLOUDBEAVER_DB_URL}`
- `CLOUDBEAVER_QM_DB_SCHEMA=qm`

However, it is advisable to change the PostgreSQL username and password for security reasons:

- `CLOUDBEAVER_DB_USER=postgres`
- `CLOUDBEAVER_DB_PASSWORD=StR0NgP2s`

During migration

- The application will not be accessible, and an error message will be displayed across the screen to prevent any operations that might disrupt the process.
- Migration includes the internal database with user lists and access rights and the Query Manager (QM) with the history of executed queries.
- If the QM has a substantial amount of data, the migration might take an extended period, during which it is advised not to turn off the server.

Post-migration

- Once migration is successful, the old databases are backed up and are not directly accessible; they will be stored with new names in their original locations.
- Reverting to the old configuration is possible only through manual intervention.

Domain manager

Note

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

Domain Manager enables you to configure a domain name and associated SSL certificate for your CloudBeaver instance. You can choose to automatically generate a domain at no cost or use your existing domain during [product installation](#) or later through the [administration interface](#).

Domain configuration during initial setup

When setting up your server, you have two options for configuring the domain: automatic generation and manual setup. Follow the instructions below based on your preferred method.

Automatic domain generation

When using automatic generation, the domain you create will have the format `<deployment_subdomain>.<organization_subdomain>.databases.team`. The `<organization_subdomain>` is initially generated from your company's name as specified in the license, but it can be customized if needed. Also, you can choose any name for the `<deployment_subdomain>`.

Example

If your organization name is `example-corp` and you choose `production` as your deployment subdomain, your domain would be `production.example-corp.databases.team`. [Learn more about domain configuration fields](#).

In addition to the domain, you receive an SSL certificate, automatically renewed every three months.

To automatically generate a domain for your web server within the local network using a Let's Encrypt certificate, follow these steps:

1. Navigate to the **Domain Manager** tab during the initial server configuration.
2. Click the **Generate Automatically** button.
3. Fill in the necessary fields as detailed in the [table](#) below.
4. Submit your settings and wait for the domain registration to complete.
5. Once the domain is active, you will be redirected to your server at the newly constructed domain, `<deployment_subdomain>.<organization_subdomain>.databases.team`.

Note

Domain registration and certificate generation can take time. DNS updates might take more than a minute.

The screenshot shows the 'INITIAL SERVER CONFIGURATION' window with the 'Domain configuration' tab selected. The left sidebar lists the configuration steps: Welcome, License, Domain Manager (selected), Server configuration, and Confirmation. The main content area displays the 'DOMAIN' configuration form. It includes a 'Finished' status indicator, an 'Organization subdomain' field with the value 'dbeaver', a 'Deployment subdomain' field with the value 'test', and a 'Server Address' field with the value 'localhost'. A note below the 'Server Address' field states: 'You should use an address in your corporate network. Usage of publicly accessible addresses is strongly not recommended.' The 'Deployment Domain' field shows the generated domain 'test.dbeaver.databases.te' with a copy icon. A 'NEXT' button is visible at the bottom of the form.

Tip

For more information on deployment, see [CloudBeaver Enterprise deployment repository](#).

Configuration fields

Field/Button	Description
Organization Subdomain	Specify the Organization subdomain. Automatically assigned during initial setup based on your license.
Deployment Subdomain (Optional)	Specify the subdomain for your specific CloudBeaver server instance. Instances can be named uniquely, for example <code>production</code> or <code>test</code> .
Server Address	Provide the IP address where your server is hosted.

Field/Button	Description
Delete	Delete the domain.

Tip

Once your domain is active, you can safely **enable Force HTTPS**. Domain Manager issues and renews Let's Encrypt certificates automatically, so you don't need extra setup.

Manual domain configuration

If you prefer to set up the domain manually, using a custom SSL certificate, follow these steps:

1. Navigate to the **Domain Manager** tab during the initial server configuration.
2. Click the **Set Up Manually** button.

The screenshot shows the 'INITIAL SERVER CONFIGURATION' window with the 'Domain configuration' tab active. The sidebar on the left lists 'Welcome', 'License', 'Domain Manager' (selected), 'Server Configuration', and 'Confirmation'. The main content area has a 'CUSTOM SSL CERTIFICATE' form. It includes a notification: 'There are unsaved changes. You can apply them by pressing the "Add certificate" button'. The form fields are: 'Domain name' (text input with 'team.yourdomain.com'), 'Certificate' (text area with a sample certificate, a 'FILE' button, and a green checkmark), 'Private key' (text area with a sample private key, a 'FILE' button), and 'Certificate chain' (text area with a sample chain, a 'FILE' button). A blue 'ADD CERTIFICATE' button is at the bottom.

3. Fill in the necessary fields.

Field	Description
Domain Name	Enter your custom domain name.
Certificate	Upload the SSL certificate file.

Field	Description
Private Key	Upload the private key file associated with your SSL certificate.
Certificate Chain (optional)	Upload the SSL certificate chain if applicable.

> **Tip**: After uploading a valid certificate and key, enable [Force HTTPS](Proxy-Configuration.md#force-https-mode){ data-preview } to ensure all connections are encrypted.

1. Ensure that the necessary DNS records are set to redirect traffic to your server.
2. Click the **Add Certificate** button to finalize the configuration.

Tip

Provided certificates will be stored in the appropriate directory (e.g., `/etc/nginx/ssl/live/` on your server).

Adjusting domain settings in admin panel

Once the application is operational, you can modify the domain settings.

To modify your domain settings, follow these steps:

1. Log in as an administrator.
2. Navigate to **Settings -> Administration -> Domain Manager**.
3. Select either the automatic or manual configuration method.

Info

Detailed steps for each method can be found in the sections: - [Automatic domain generation](#) - [Manual domain configuration](#)

Deleting the domain

To delete a domain, follow these steps:

1. Log in as an administrator.
2. Navigate to **Settings -> Administration -> Domain Manager**.

3. Select the **Delete** option.

Notes

- If you're using a Let's Encrypt certificate, you don't need to delete the Organization subdomain to assign a new one. Simply update the subdomain name and click **Save** to apply the changes.
- Ensure that **local access** is configured, as changes might disrupt SSO functionalities.
- After deleting the domain, this server will be accessible with the Server Address (IP).

Modifying the domain

To make changes to a domain:

1. Log in as an administrator.
2. Navigate to **Settings -> Administration -> Domain Manager**.
3. Update the necessary fields. If your domain uses a custom certificate, click the **Update Certificate** button to make changes to your SSL certificate.

For existing domains, whether set up automatically or manually, you can switch between the automatic and manual setup types by clicking the **Change Type** button.

Modifications for automatically generated domain

If your domain was configured using the automatic domain generation, consider the following additional steps and restrictions:

1. Modify the fields under the Deployment or Organization subdomains, respecting the limits on changes.

Note

You can change the Deployment Subdomain up to three times per calendar month.

Projects
Server Configuration
Global Preferences
Users and Teams
Query Manager
Driver Management
AWS Settings
Identity Providers
AI Settings
Domain Manager
Product Information

SAVE X CANCEL

Domain registration may take some time. Please wait until the domain is available.

LET'S ENCRYPT CERTIFICATE X
The certificate will be generated on your server. It expires after three months and will be automatically renewed by the server. The domain can be changed only three times a month.

Organization subdomain *
dbeaver

Deployment subdomain
test1-te DELETE

Server Address *
192.168.1.100
You should use an address in your corporate network. Usage of publicly accessible addresses is strongly not recommended.

Deployment Domain
192.168.1.100
After this configuration, your domain will be accessible at the address above.

Valid until
11/26/2024, 1:00:00 AM
Will be renewed automatically
CHANGE TYPE

2. Submit your settings and wait for the domain registration to complete. There may be a waiting period.
3. Once the domain is active, you will be redirected to your server at the newly constructed domain, e.g., `<deployment_subdomain>.<organization_subdomain>.databases.team`. If the organizational domain has been deleted, you will be redirected to your Server's Address.

Product configuration parameters

Product configuration parameters

This document provides guidelines for configuring core product settings in CloudBeaver. These settings control localization, themes, logging, the SQL editor, and more.

Info

For more details on configuration, see the [CloudBeaver server configuration](#).

Example configuration

The Product settings configuration is located under the `productSettings` section of the main server configuration file.

```
productSettings: {
  "core.localization.language": "${CLOUDBEAVER_CORE_LOCALIZATION:en}",
  "core.theming.theme": "${CLOUDBEAVER_CORE_THEMING_THEME:light}",
  "plugin.log-viewer.disabled": "${CLOUDBEAVER_LOG_VIEWER_DISABLED:false}",
  "plugin.log-viewer.logBatchSize":
    "${CLOUDBEAVER_LOG_VIEWER_LOG_BATCH_SIZE:2}",
  "plugin.log-viewer.maxLogRecords":
    "${CLOUDBEAVER_LOG_VIEWER_MAX_LOG_RECORDS:2000}",
  "plugin.sql-editor.autoSave": "${CLOUDBEAVER_SQL_EDITOR_AUTOSAVE:true}",
  "plugin.sql-editor.disabled": "${CLOUDBEAVER_SQL_EDITOR_DISABLED:false}",
  "plugin.sql-editor.maxFileSize":
    "${CLOUDBEAVER_SQL_EDITOR_MAX_FILE_SIZE:10240}",
  "sql.proposals.insert.table.alias":
    "${CLOUDBEAVER_SQL_PROPOSALS_INSERT_TABLE_ALIAS:PLAIN}"
},
```

Configuration properties

Interface

Name	Environment variable	Default value	Units	Description	Allowed values
<code>core.localization.language</code>	<code>CLOUDBEAV ER_CORE_L OCALIZATI ON</code>	<code>en</code>	string	Default language for the interface.	See Languages .
<code>core.theming.theme</code>	<code>CLOUDBEAV ER_CORE_T HEMING_TH EME</code>	<code>light</code>	string	Default application theme.	See Themes .

Log viewer

Name	Environment variable	Default value	Units	Description	Allowed values
<code>plugin.log-viewer.disabled</code>	<code>CLOUDBEAV ER_LOG_VI EWER_DISA BLED</code>	<code>false</code>	boolean	Disables the log viewer plugin.	<code>true</code> , <code>false</code>
<code>plugin.log-viewer.logBatchSize</code>	<code>CLOUDBEAV ER_LOG_VI EWER_LO G_BATCH_S IZE</code>	<code>2</code>	integer	Number of log entries retrieved per request.	Integer
<code>plugin.log-viewer.maxLogRecords</code>	<code>CLOUDBEAV ER_LOG_VI EWER_MA X_LOG_REC ORDS</code>	<code>2000</code>	integer	Maximum number of log records stored.	Integer

Info

For details on customizing logging level and pattern, see [Log customization](#).

SQL editor

Name	Environment variable	Default value	Units	Description	Allowed values
<code>plugin.sql-editor.autoSave</code>	<code>CLOUDBEAV ER_SQL_EDITOR_AUTO SAVE</code>	<code>true</code>	boolean	Enables auto-saving for SQL editor sessions.	<code>true</code> , <code>false</code>
<code>plugin.sql-editor.disabled</code>	<code>CLOUDBEAV ER_SQL_EDITOR_DISABLED</code>	<code>false</code>	boolean	Disables the SQL editor plugin.	<code>true</code> , <code>false</code>
<code>plugin.sql-editor.maxFileSize</code>	<code>CLOUDBEAV ER_SQL_EDITOR_MAX_FILE_SIZE</code>	<code>10240</code>	integer	Maximum file size for importing files in the SQL editor.	Integer

Info

For more details on SQL Editor, see [our documentation](#).

SQL proposals

Name	Environment variable	Default value	Units	Description	Allowed values
<code>sql.proposals.insert.table.alias</code>	<code>CLOUDBEAV ER_SQL_PROPOSALS_INSERT_TABLE_ALIAS</code>	<code>PLAIN</code>	string	Default alias format for table insert proposals.	<code>PLAIN</code> , <code>SHORT</code> , <code>FULL</code>

Additional product settings

- [Authentication settings](#)
- [Connection settings](#)
- [Data export settings](#)
- [Data spreadsheet settings](#)
- [ERD settings](#)
- [Navigation tree settings](#)

Authentication settings

Product configuration parameters

Variables

Variable	~~Deprecated~~	Value	Description
core.authentication.disableAnonymousAccess	core.authentication.disableAnonymousAccess	false	Disable anonymous access in administration

Navigation tree settings

Product configuration parameters

Variables

Variable	~~Deprecate d~~	Allowed values	Limit	Description
core.navigation-tree.childrenLimit	core.app.navigationTree.childrenLimit	integer	1000	Sets navigator tree level fetch size
core.navigation-tree.editing	core.app.metadata.editing	boolean		Enables nodes renaming
core.navigation-tree.deleting	core.app.metadata.deleting	boolean		Enables nodes deletion

Data spreadsheet settings

Product configuration parameters

Variables

Variable	~~Deprecated~~	Value	Description
plugin.data-viewer.disableEdit	core.app.dataViewer.disableEdit	false	Disables editing for non-admin users
plugin.data-viewer.disableCopyData		false	Disables copying of data for non-admin users
plugin.data-viewer.fetchMin		100	Minimum amount of fetched rows
plugin.data-viewer.fetchMax		5000	Maximum amount of fetched rows
plugin.data-viewer.fetchDefault		200	Default amount of fetched rows
plugin.data-spreadsheet.hidden	plugin_data_spreadsheet_new.hidden	false	Hides data editor from ui

Data export settings

Product configuration parameters

Variables

Variable	Deprecated	Value	Description
plugin.data-export.disabled	~~plugin_data_export.disabled~~	false	Disables data export

ERD settings

Product configuration parameters

Variables

This parameters only available in Enterprise and AWS products

Variable	~~Deprecated~~	Value	Description
plugin.erd-viewer.maxColumnsToDisplay	plugin_erd_viewer.maxColumnsToDisplay	15000	Sets the maximum displayed objects count

Connection settings

Product configuration parameters

Variables

Variable	Value	Description
plugin.connections.hideConne ctionViewForUsers	false	Hides "Connection view" conte xt menu item for non admin us ers

Query manager database configuration

This document provides guidelines for configuring the Query Manager database in CloudBeaver.

Info

For more information on Query Manager, see [Query Manager](#). For more details on configuration, see the [CloudBeaver server configuration](#).

Example configuration

The Query Manager database configuration is located under the `qm` section of the main server configuration file.

```
qm: {
  driver: "${CLOUDBEAVER_QM_DB_DRIVER:h2_embedded_v2}",
  url: "${CLOUDBEAVER_QM_DB_URL:jdbc:h2:${workspace}/.metadata/qmdb/qmdb}",
  user: "${CLOUDBEAVER_QM_DB_USER:''}",
  password: "${CLOUDBEAVER_QM_DB_PASSWORD:''}",
  schema: "${CLOUDBEAVER_QM_DB_SCHEMA:''}",
  pool: {
    minIdleConnections: "${CLOUDBEAVER_QM_DB_MIN_IDLE_CONNECTIONS:4}",
    maxIdleConnections: "${CLOUDBEAVER_QM_DB_MAX_IDLE_CONNECTIONS:10}",
    maxConnections: "${CLOUDBEAVER_QM_DB_MAX_CONNECTIONS:100}",
    validationQuery: "${CLOUDBEAVER_QM_DB_VALIDATION_QUERY:SELECT 1}"
  },
  backupEnabled: "${CLOUDBEAVER_QM_DB_BACKUP_ENABLED:true}"
}
```

Configuration properties

Name	Environment variable	Default value	Units	Description
<code>driver</code>	<code>CLOUDBEAVER_QM_DB_DRIVER</code>	<code>h2_embedded_v2</code>	string	Database driver (e.g., <code>postgresql-jdbc</code> , <code>oracle-thin</code> , etc.).
<code>url</code>	<code>CLOUDBEAVER_QM_DB_URL</code>	<code>jdbc:h2:\${workspace}/.metadata/qmdb/qmdb</code>	string	Database JDBC

Name	Environment variable	Default value	Units	Description
	R_QM_DB_URL	rkspace}/.metadata/qmdb/qmdb		URL (e.g., jdbc:postgresql://localhost:5432/qm).
user	CLOUDBEAVE R_QM_DB_USER	' '	string	Database user name.
password	CLOUDBEAVE R_QM_DB_PASSWORD	' '	string	Database user password.
schema	CLOUDBEAVE R_QM_DB_SCHEMA	' '	string	Database schema.
minIdleConnections	CLOUDBEAVE R_QM_DB_MIN_IDLE_CONNECTIONS	4	number	Minimum idle connections for the connection pool.
maxIdleConnections	CLOUDBEAVE R_QM_DB_MAX_IDLE_CONNECTIONS	10	number	Maximum idle connections for the connection pool.
maxConnections	CLOUDBEAVE R_QM_DB_MAX_CONNECTIONS	100	number	Maximum total connections in the pool.
validationQuery	CLOUDBEAVE R_QM_DB_VALIDATION_QUERY	SELECT 1	string	Query to validate connections.
backupEnabled	CLOUDBEAVE R_QM_DB_BACKUP_ENABLED	true	boolean	Enables or disables backup of the Query Manager database.

Security and access configuration

Server security and access configuration

In CloudBeaver, you can manage security and access on several levels. These settings let you control how users sign in, who can access the system, how database connections are secured, and how the server is protected on the network.

You can configure:

- **Authentication** - sign-in methods for users
- **Access management** - permissions and policies for accounts and teams
- **Connection network options** - encryption and tunnels for database connections
- **Network and security settings** - domains, proxies, certificates, and secure storage
- **Additional security options** - password recovery, credentials storage, provisioning, and restrictions

Authentication

Configure how users sign in. Choose local accounts, enterprise identity providers, or cloud-specific services.

- [Anonymous access](#)
- [Local authentication](#)
- [LDAP](#)
- [SAML](#)
- [OpenID](#)
- [NTLM](#)
- [JWT](#)
- [AWS IAM](#)
- [Microsoft Entra ID](#), and others

Info

See [Authentication methods](#) for a full overview.

Brute force protection

CloudBeaver can protect against brute force attacks on both application login and database connections.

- **Login protection** - block users after several failed login attempts
- **Connection protection** - temporarily block the ability to connect after repeated failed attempts (for example, wrong database password)

Info

For more details, see [Password policy](#)

Access management

Control who can sign in and what they can do. Enforce password rules, assign permissions, and organize users into teams.

- [Users](#) and [Teams](#)
- [Password policy](#)

Connection network options

Secure database connections at the connection level. Enable SSL, set up SSH tunnels, and manage keys.

- [SSL configuration](#) - encrypt traffic with CA-signed or self-signed certificates
- [SSH configuration](#) - protect connections with SSH tunnels and key authentication

Network and security settings

Manage how the server is exposed on the network. Configure domains, proxies, TLS, and secure storage for sensitive data.

Setting	Description	Reference
Domain manager	Configure a custom domain, and apply SSL certificates	Domain manager

Setting	Description	Reference
Reverse proxy	Control and filter incoming traffic, enable auth via headers, offload TLS, enforce HTTPS, and configure multiple server URLs	Proxy configuration
Secret management	Store and retrieve credentials from providers like AWS Secrets or Vault	Secret providers
Java security properties	Override default JVM crypto properties for compliance and security policies	Java security properties

Additional security options

CloudBeaver also provides extra security features that complement authentication, access control, and connection settings.

Setting	Description	Reference
Password recovery	Restore access if the main password is lost	Admin password recovery
User credentials storage	Control how saved credentials are stored and protected	User credentials storage
Pre-configured permissions	Define default permissions for new database connections	Pre-configured permissions
User provisioning	Automate account creation and updates with provisioning systems	Provisioning users
Proxy header authentication	Authenticate users via trusted reverse proxy headers	Reverse proxy header authentication

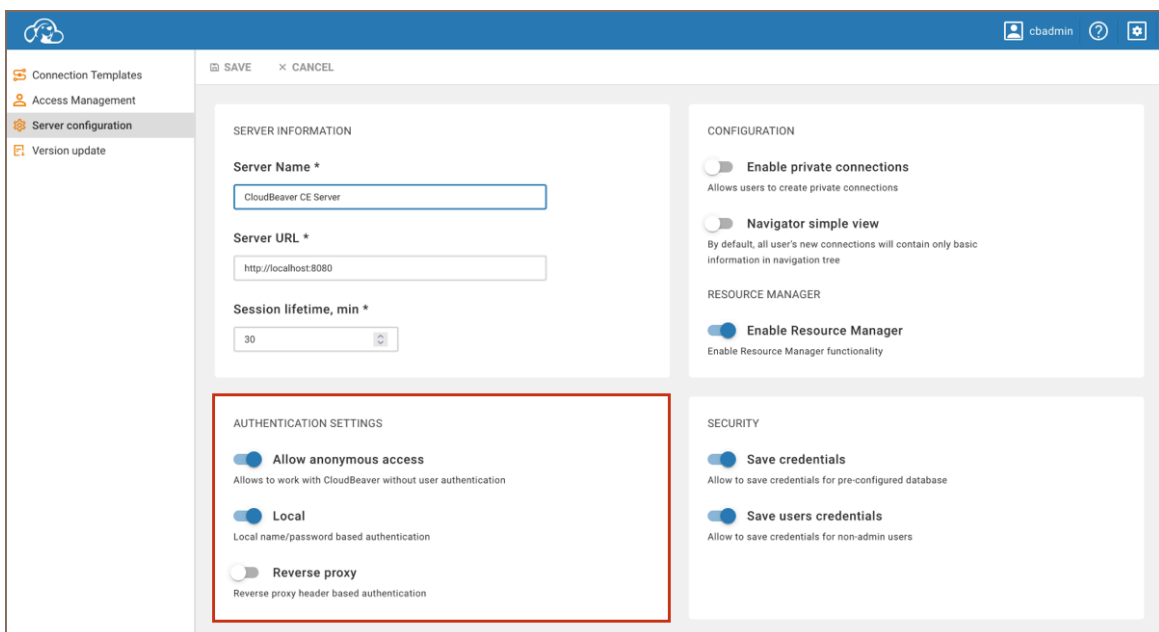
Authentication

Authentication methods

CloudBeaver provides various authentication methods, configurable by administrators in the **Settings -> Administration -> Server Configuration**. These settings apply to logging into the CloudBeaver interface.

CloudBeaver Community Edition authentication types

The Community Edition (CE) of CloudBeaver supports various types of authentication. Detailed information about each type is available in the sections below.



The screenshot displays the CloudBeaver administration interface. On the left, a sidebar contains navigation links: Connection Templates, Access Management, Server configuration (highlighted), and Version update. The main content area is titled 'SERVER CONFIGURATION' and includes a 'SAVE' button and a 'CANCEL' button. The configuration is divided into several sections:

- SERVER INFORMATION**:
 - Server Name ***: A text input field containing 'CloudBeaver CE Server'.
 - Server URL ***: A text input field containing 'http://localhost:8080'.
 - Session lifetime, min ***: A numeric input field set to '30'.
- AUTHENTICATION SETTINGS** (highlighted with a red border):
 - Allow anonymous access**: A toggle switch that is turned on. Below it, the text reads: 'Allows to work with CloudBeaver without user authentication'.
 - Local**: A toggle switch that is turned on. Below it, the text reads: 'Local name/password based authentication'.
 - Reverse proxy**: A toggle switch that is turned off. Below it, the text reads: 'Reverse proxy header based authentication'.
- CONFIGURATION**:
 - Enable private connections**: A toggle switch that is turned off. Below it, the text reads: 'Allows users to create private connections'.
 - Navigator simple view**: A toggle switch that is turned off. Below it, the text reads: 'By default, all user's new connections will contain only basic information in navigation tree'.
- RESOURCE MANAGER**:
 - Enable Resource Manager**: A toggle switch that is turned on. Below it, the text reads: 'Enable Resource Manager functionality'.
- SECURITY**:
 - Save credentials**: A toggle switch that is turned on. Below it, the text reads: 'Allow to save credentials for pre-configured database'.
 - Save users credentials**: A toggle switch that is turned on. Below it, the text reads: 'Allow to save credentials for non-admin users'.

Anonymous access

This method allows users to interact with CloudBeaver without needing to authenticate. Further details can be found in the [Anonymous access configuration](#) article.

Local access

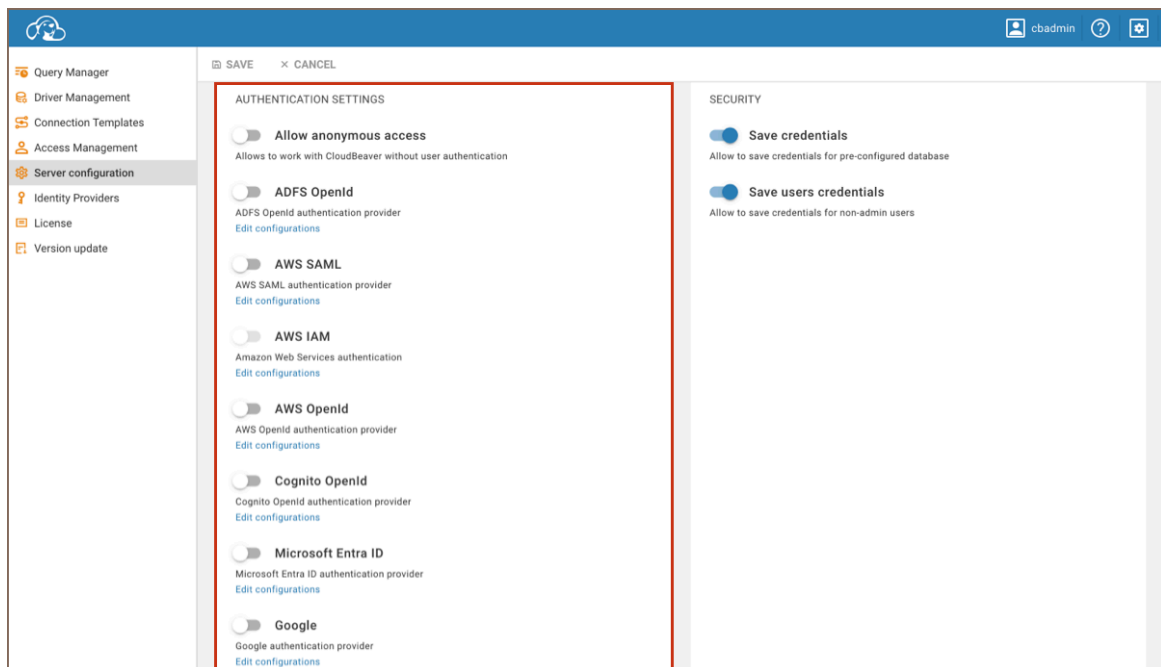
Local access requires users to authenticate using a username and password. Administrators create user accounts and assign roles to define user permissions. For more detailed information, see the [Local access authentication](#) article.

Reverse proxy access

This authentication method is based on `HTTP` request headers. For more detailed information, see the [Reverse Proxy Header Authentication](#) article.

CloudBeaver Enterprise Edition authentication types

In addition to the authentication methods supported by the Community Edition (CE), CloudBeaver Enterprise Edition (EE) offers a broader range of authentication options. For more detailed information on each method, please refer to the subsequent sections.



AWS IAM access

AWS IAM authentication allows logging in with IAM credentials, automatically assigning a 'User' role in CloudBeaver. For more details, see the [AWS IAM](#) article.

SSO 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).

OpenID

OpenID Connect (OIDC) authentication enables users to log in using credentials from an OpenID Connect identity provider, simplifying the login process across different services. For further details, please consult the [OpenID authentication](#) article.

AWS OpenID

AWS OpenID Connect (OIDC) authentication integrates with AWS Identity and Access Management (IAM) to enable signing in using an OpenID Connect identity provider. This approach allows for the secure delegation of permissions to AWS resources based on the identity established by the provider. For more information, please refer to the [AWS OpenID authentication](#) article.

Okta OpenID

Okta OpenID Connect (OIDC) authentication allows applications to authenticate users via Okta's OIDC provider. This process enables secure and streamlined user access to applications and services without requiring them to manage multiple passwords. For more information, refer to the [Okta OpenID Authentication](#) article.

Cognito OpenID

Cognito OpenID is a feature of Amazon Cognito that allows you to use Cognito as an OpenID Connect (OIDC) identity provider. By configuring Cognito as an OIDC provider, users can sign in to CloudBeaver using their existing accounts with OIDC identity providers. For more information, refer to the [Cognito OpenID Authentication](#) article.

Microsoft Entra ID

Microsoft Entra ID, previously known as Azure AD, is a service that allows you to use Microsoft's cloud-based identity and access management service as an identity provider. By integrating Microsoft Entra ID with CloudBeaver, users can sign in using their Microsoft accounts, leveraging

Azure Active Directory (Azure AD) for authentication. For more details, see the [Microsoft Entra ID Authentication](#) article.

Google

Google Authentication is a method that allows users to sign in to CloudBeaver using their Google accounts. This integration leverages Google as an OpenID Connect (OIDC) identity provider, enabling users to authenticate with their existing Google credentials. For more information, refer to the [Google Authentication](#) article.

JWT

JWT (JSON Web Tokens) Authentication is a secure method to transmit information between parties as a JSON object. In the context of CloudBeaver, JWT authentication can be configured to verify the identity of users. For more information, refer to the [JWT Authentication](#) article.

NTLM

NTLM (New Technology LAN Manager), is a suite of Microsoft security protocols used for user authentication and data protection in Windows networks. It operates on a challenge-response mechanism to verify user identity without transmitting passwords directly. For more information on configuration, refer to the [NTLM Authentication](#) article.

Secure Storage with Secret Providers

CloudBeaver supports various cloud-based secret providers to retrieve database credentials. For detailed setup instructions, see [Secret Providers](#).

CloudBeaver Enterprise Edition for AWS

CloudBeaver Enterprise Edition for AWS offers compatibility with a range of authentication methods that facilitate integration with AWS services and other identity solutions. Below is the list of the supported authentication methods:

- [Anonymous access configuration](#)
- [Local access](#)
- ADFS OpenID
- AWS SAML
- [AWS IAM](#)

- AWS OpenID
- Cognito OpenID
- Microsoft Entra ID
- Reverse Proxy Header Authentication
- Google
- JWT
- NTLM
- Okta OpenID
- OpenID
- SAML

Anonymous access

Anonymous access in CloudBeaver lets users connect to databases without authentication. Before configuring any anonymous connections, you must first enable anonymous access in the server settings.

Enabling anonymous access

You can turn on anonymous access in two ways: through the UI or by using server configuration.

Use the web UI

1. Sign in as an administrator.
2. Go to **Administration -> Settings -> Server configuration**.
3. Select the **Allow anonymous access** checkbox.
4. Click **Save** to apply the changes.

Use server configuration

Set the `CLOUDBEAVER_APP_ANONYMOUS_ACCESS_ENABLED` environment variable and restart the server.

Info

For details on where to define server parameters and how configuration files work, see [Server configuration](#)

Once anonymous access is enabled, you can set it up in one of two ways:

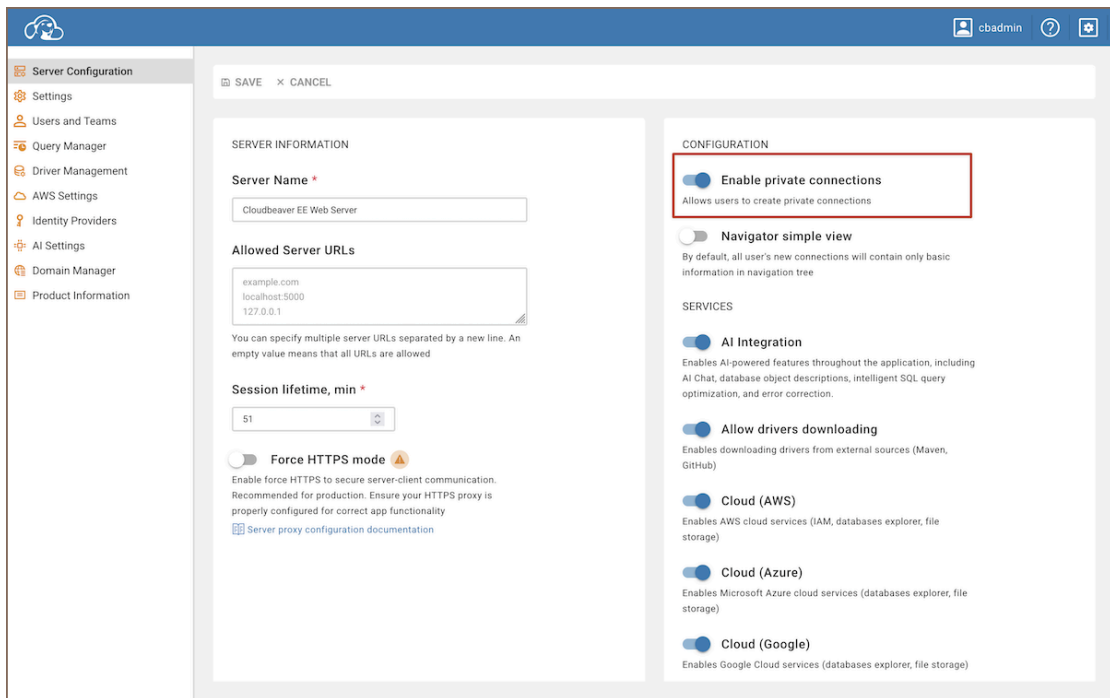
- **Administrator defined connections:** Administrators can configure connections that are accessible anonymously, allowing access to these predefined connections.
- **Private connections:** When enabled, this feature lets anonymous users configure their private connections from the main page. These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

Configuration methods

Enabling private connections for anonymous users

1. As an administrator, navigate to the **Settings -> Server configuration**.

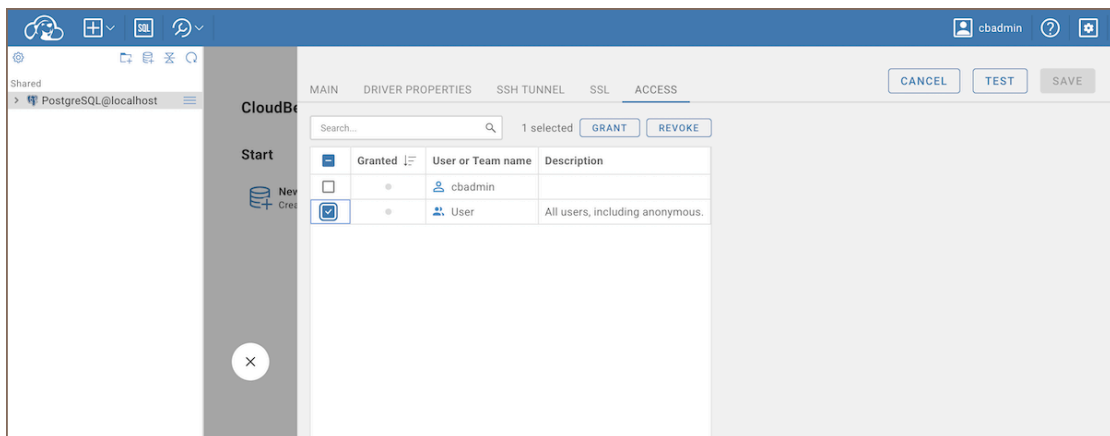
2. Locate the **Enable private connections** option and activate this setting to allow the creation of connections by anonymous users.



3. Save changes.

Administrator defined anonymous access

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



3. Select the `user` entry (representing all users, including anonymous) by checking the box next to it, then click **Grant** to assign access.

Tip

If you want anonymous users to have a different default role, change the `anonymousUserRole` value in your [server configuration](#).

4. When you've finished configuring the connection, click **Save** to apply the changes.

Anonymous access limits

CloudBeaver **Enterprise** restricts the number of simultaneous anonymous sessions to prevent license overuse. The system applies the following rules:

- The allowed number of anonymous sessions is calculated as `Anonymous limit = Total licensed users - Currently logged-in users`.
- If all licensed users are logged in, anonymous mode is disabled automatically.
- Each anonymous session ends automatically after **30 minutes of inactivity**.
- When the limit is exceeded, an error appears in the login window, and the event is recorded in the server log.

Local access authentication

Note

This feature is available in **Enterprise** and **Team** editions only.

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

Configuration steps

Step 1 Enabling local authentication

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

The screenshot displays the 'Server configuration' settings page in the CloudBeaver CE interface. The left sidebar shows the navigation menu with 'Server configuration' selected. The main content area is divided into several sections: 'SERVER INFORMATION' with fields for 'Server Name' (CloudBeaver CE Server), 'Server URL' (http://localhost:8090), and 'Session lifetime, min' (30); 'AUTHENTICATION SETTINGS' with three options: 'Allow anonymous access' (checked), 'Local' (checked and highlighted with a red box), and 'Reverse proxy' (unchecked); 'CONFIGURATION' with 'Enable private connections' (unchecked) and 'Navigator simple view' (checked); 'RESOURCE MANAGER' with 'Enable Resource Manager' (checked); and 'SECURITY' with 'Save credentials' (checked) and 'Save users credentials' (checked). The top bar includes a 'SAVE' button and a 'CANCEL' button.

3. Save changes.

Step 2 Granting local access

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

The screenshot displays the CloudBeaver web interface for user management. The top navigation bar shows the user 'cbadmin'. The left sidebar lists 'Connection Templates', 'Access Management' (selected), 'Server configuration', and 'Version update'. The main content area is titled 'USERS' and 'TEAMS'. It features a '+ CREATE' and 'REFRESH' button, a search bar, and a 'User Creation' form. The form has two tabs: 'INFO' (selected) and 'CONNECTIONS ACCESS'. The 'INFO' tab is divided into three sections: 'CREDENTIALS' with fields for 'Username *', 'User password *', and 'Repeat password *'; 'USER STATUS' with a checked 'Enabled' checkbox and a 'USER TEAM' section with 'admin' (unchecked) and 'user' (checked) options; and 'PARAMETERS' with 'First Name' and 'Last Name' fields. 'CANCEL' and 'CREATE' buttons are at the top right of the form.

Recovering from disabled authentication methods

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

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

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

4. Restart the server to apply the changes.

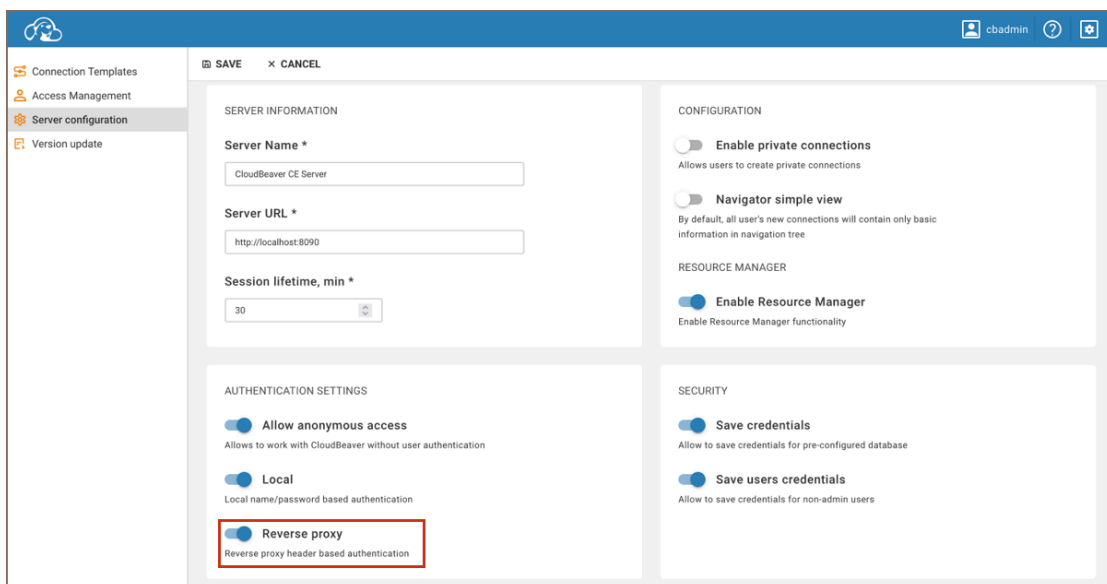
Reverse proxy header authentication

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

Configuration Steps

Enabling Reverse proxy authentication

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



3. Save changes.

Reverse proxy identity provider configuration in Community Edition

To configure reverse proxy authentication, follow these steps:

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

```

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

```

Important

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

Reverse proxy identity provider configuration

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

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

Configuring default HTTP header fields

Configure the standard HTTP header fields as follows:

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

Header example

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

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

Tip

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

LDAP authentication

LDAP (Lightweight Directory Access Protocol) is a protocol designed to manage and access distributed directory information services over an Internet Protocol network. LDAP is used to store and retrieve data in a hierarchical directory structure, such as usernames and passwords, organizational units, and configuration settings. It facilitates directory management and authentication processes, allowing for a centralized approach to maintaining user credentials and policy settings.

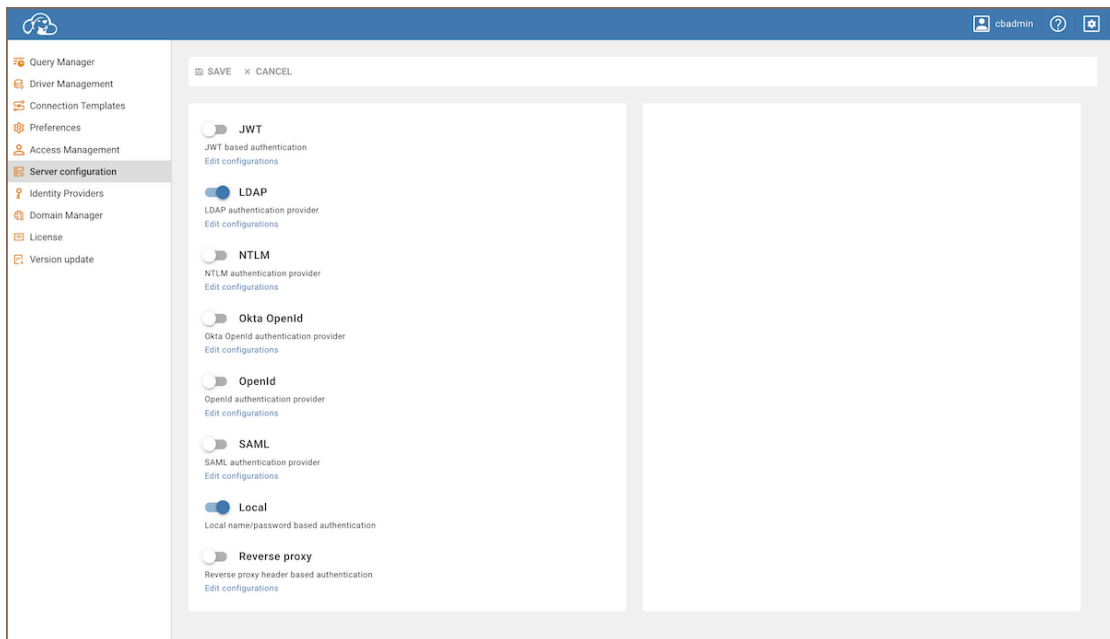
Info

For comprehensive guidance on configuring and managing LDAP, refer to the [official LDAP documentation](#).

Configuration steps

Step 1: Enabling LDAP Authentication

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



3. Save the changes.

Step 2: Adding an Identity Provider

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

The screenshot shows the 'CONFIGURATION CREATION' page for adding an LDAP identity provider. The left sidebar contains navigation links: Server Configuration, Global Preferences, Users and Teams, Query Manager, Driver Management, AWS Settings, Identity Providers (selected), Domain Manager, and Product Information. The main area has a header with '+ ADD', 'REFRESH', and 'DELETE' buttons. Below the header, a message states: 'You can configure the list of enabled Identity Providers in the Server Configuration settings'. There are 'CANCEL' and 'CREATE' buttons. The form is divided into two sections: 'OPTIONS' and 'USER PROVISIONING INTEGRATION'. The 'OPTIONS' section includes fields for 'Provider *' (a dropdown menu with 'LDAP' selected), 'ID *' (a text field with 'LDAP'), 'Configuration Name *' (a text field with 'LDAP'), 'Description' (a text area), and 'Icon URL' (a text field). There is also a 'Disabled' checkbox. The 'USER PROVISIONING INTEGRATION' section includes fields for 'Host *' (a text field with 'ldap.example.com'), 'Port *' (a dropdown menu with '389'), 'User identifier attribute *' (a text field with 'cn'), 'Base Distinguished Name' (a text field with 'ou=users,dc=example,dc=com'), 'Bind User DN' (a text field with 'cn=serviceAccount1,ou=serviceAccounts'), 'Bind User Password' (a text field), 'First name field' (a text field with 'fn'), 'Last name field' (a text field with 'ln'), and 'Role field' (a text field with 'department').

Field	Description	Parameters used to configure LDAP in the CloudBeaver Community
Provider	Select LDAP from the dropdown menu.	
ID	Enter a unique identifier for the configuration.	
Configuration name	Enter a descriptive name for this configuration.	
Description	(Optional) Provide a brief description of this identity provider configuration.	

Field	Description	Parameters used to configure LDAP in the CloudBeaver Community
Icon URL	(Optional) Enter the URL of an icon to represent this provider.	
Disabled	(Optional) Leave unchecked to enable this identity provider.	
Host	Enter the server hostname or IP address where LDAP authentication is handled.	<code>ldap-host : ""</code>
Port	Specify the port number used for LDAP communication. Default port is 389 for standard mode, 636 for SSL.	<code>ldap-port : ""</code>
User identifier attribute	Enter the attribute used to uniquely identify LDAP users. Default value in PRO version is <code>cn</code> .	<code>ldap-identifier-attr : ""</code>
Base Distinguished Name	(Optional) Enter the Base Distinguished Name (DN) for LDAP queries. It should be in the format appropriate for your LDAP directory, such as <code>dc=example,dc=com</code> . If not explicitly specified, this value will be automatically appended to the end of the User DN during authorization.	<code>ldap-dn : ""</code>
Bind User DN	(Optional) Enter the Distinguished Name (DN) of the user who has permissions to search for entries (e.g., users) in the directory. This user will be used to verify access to the ap	<code>ldap-bind-user : ""</code>

Field	Description	Parameters used to configure LDAP in the CloudBeaver Community
	plication with the specified filter. The bind user is required if you want to log in with something other than the full DN.	
Bind User Password	(Optional) Enter the password associated with the Bind User DN account. Bind User DN and Bind User Password are both mandatory if you want to log in with something other than the full DN.	<code>ldap-bind-user-pwd : ""</code>
User Filter	(Optional) Specify criteria for searching specific entries (users, groups, attributes). The filter must always be written in parentheses. Logical operators are supported. Bind User DN and Bind User Password must be configured for this field to work.	<code>ldap-filter : "("</code>
User login parameter	(Optional) Enter the LDAP attribute to be used as the user login. The attribute must be unique. This parameter enables users to log in with the specified LDAP attribute <code>sAMAccountName</code> . Bind User DN and Bind User Password must be configured for this field to work.	<code>ldap-login : ""</code>
Enable SSL	(Optional) Set to <code>true</code> to enable SSL encryption for LDAP communication.	<code>ldap-enable-ssl : true</code>
Certificate (PEM format)	(Optional) Provide the SSL certificate	<code>ldap-ssl-cert : ""</code>

Field	Description	Parameters used to configure LDAP in the CloudBeaver Community
	rtificate in <code>base64</code> format to trust the LDAP server.	
Referral handling	(Optional) Define how LDAP referrals to other servers are handled, for example when part of your directory (e.g., <code>dc=us</code>) is stored on another domain controller. Use <code>follow</code> to follow referrals to other domains, or <code>ignore</code> to skip them. Default is <code>ignore</code> .	"ldap-referral": "ignore"
First name field ★	(Optional) Enter the name of the LDAP attribute used for the user's first name (e.g., <code>fn</code>). This attribute will be automatically detected during user provisioning.	
Last name field ★	(Optional) Enter the name of the LDAP attribute used for the user's last name (e.g., <code>ln</code>). This attribute will be automatically detected during user provisioning.	

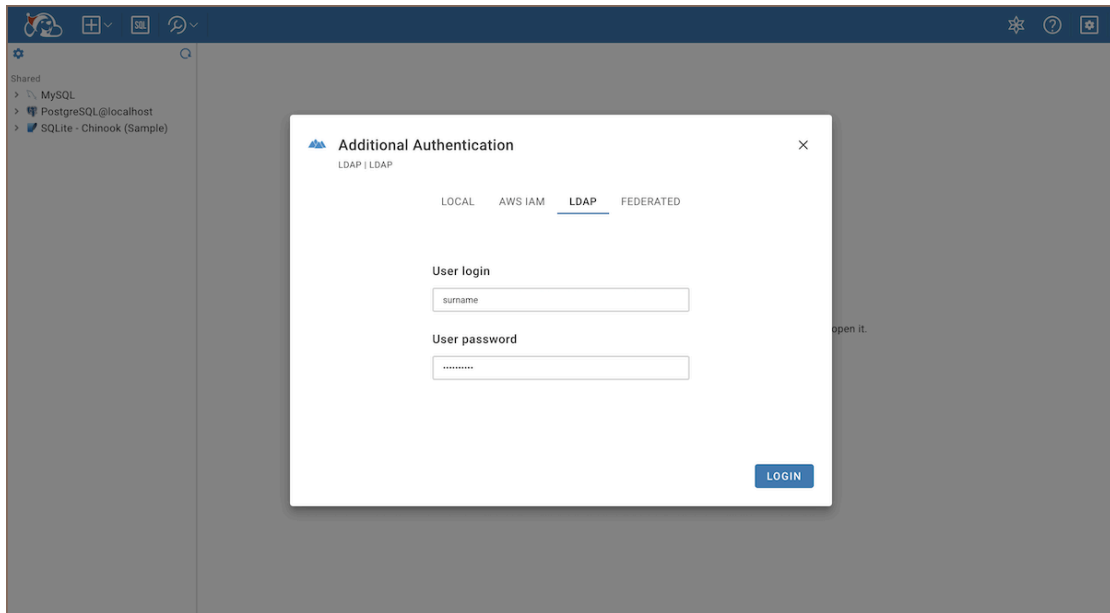
- Click on the **Create** button.

Tip

You can specify organizational units within the Base Distinguished Name to streamline access, such as `ou=unit1,dc=example,dc=com`. These can be set during the Identity Provider setup or during login in CloudBeaver. For automatic mapping of **Teams** to LDAP groups, fill in the **LDAP Group name** field. Use the `cn` attribute containing one of the following: `memberOf`, `member`, `uniqueMember`, or `gidNumber`. For example, if you specify `cn=groupName` in **LDAP Group name**, only users whose `member` attribute is listed in `cn=groupName` will be mapped.

Step 3: Logging in

1. With the **LDAP** configuration now established, proceed to the login screen.
2. Select the **LDAP** authentication method.



3. In the **User login** field, enter the Distinguished Name (DN) of the user who is logging in. This specifies the exact entry within the LDAP directory associated with your user account.

Tip

If the **User login parameter** was configured during the setup, you can alternatively enter the value of the specified login attribute instead of the full **User DN**.

4. Enter your **User password** in the corresponding field to authenticate.

Note

If the **Base Distinguished Name** was specified during the setup of your LDAP configuration, it will be automatically appended to the **User DN** if not explicitly included. This can simplify the login process, especially when managing multiple users.

5. After entering the required information, click **Login** to access the application.

Configuration steps for CloudBeaver Community Edition

In CloudBeaver Community Edition, configuration settings are not accessible through the user interface (UI). Instead, settings must be configured directly within the workspace configuration file.

Steps to configure LDAP in Community Edition:

1. As an administrator, go to **Settings -> Server Configuration** and activate the **LDAP** option in the Configuration section.
2. Locate either the `.cloudbeaver.runtime.conf` file to configure LDAP for the current workspace or the `.cloudbeaver.conf` file to set the LDAP provider for all newly created workspaces. More details can be found in the [server configuration guide](#).
3. Open the file with a text editor to modify or add LDAP configurations.
4. Insert the following configuration snippet into your configuration file under the `app.authConfigurations` section. Adjust the parameters according to your LDAP server details.

```
"authConfigurations": [  
  {  
    "id": "ldap",  
    "provider": "ldap",  
    "displayName": "LDAP",  
    "disabled": false,  
    "iconURL": "",  
    "description": "",  
    "parameters": {  
      "ldap-host": "ldap.example.com",  
      "ldap-port": "1389",  
      "ldap-login": "AttributeName"  
      "ldap-dn": "ou=users,dc=company,dc=com",  
      "ldap-enable-ssl": true,  
      "ldap-ssl-cert": "your_cert_in_base64_format",  
      "ldap-referral": "ignore",  
      "ldap-identifier-attr": "cn",  
      "ldap-bind-user":  
      "cn=serviceAccount1,ou=serviceAccounts,dc=company,dc=com",  
      "ldap-bind-user-pwd": "serviceAccount1password",  
      "ldap-filter": "(givenName=*)"  
    }  
  }  
]
```

Important

Ensure that the `provider` is set to `ldap`.

5. After editing the configuration file, restart CloudBeaver for the changes to take effect.
6. **Log in.**

Single sign-on (SSO)

Note

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

CloudBeaver supports Single Sign-On (SSO), allowing users to access the application with a single set of credentials.

Use single sign-on

For a list of available Identity Providers and detailed configuration instructions, see [Authentication Methods](#).

Note

CloudBeaver does not store authentication information on the server-side or in configuration files. When your session expires, you will need to re-authenticate. When you log out of the application, CloudBeaver also logs you out from the Identity Provider.

SAML authentication

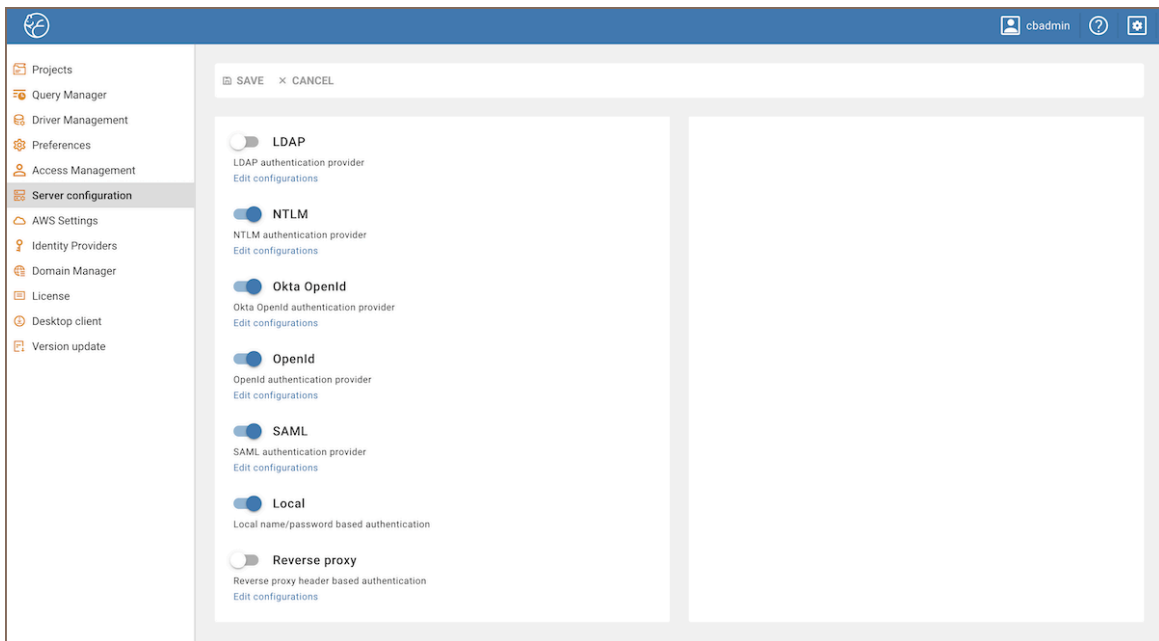
Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

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.



Configuring an external identity provider

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

CloudBeaver Identity Providers configuration page. The form includes the following fields:

- Provider ***: SAML (dropdown)
- ID ***: SAML-TEST
- Configuration name ***: SAML-TEST
- Description**: (text input)
- Icon URL**: (text input)
- Disabled**: ☐
- SAML**
 - IDP Entity ID**: https://example.com/123
 - IDP sign-on URL**: https://example.com/123/saml2/saml2
 - IDP logout URL**: https://example.com/123/saml2/saml2
 - IDP x509 certificate**: (text area with BEGIN/END markers and a download icon)
 - User group mapping attribute**: (text input)
- OPTIONAL**
 - Public SAML x509 certificate**: (text input)
 - Private SAML x509 key**: (text input)

Buttons: CANCEL, CREATE

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 copy the **ACS** and **Entity ID** links.

CloudBeaver 'EDIT SAML' configuration page. The form includes the same fields as the first screenshot. A red box highlights the **LINKS** section on the right, which contains the following links:

- Sign in**: (link with copy icon)
- Sign out**: (link with copy icon)
- ACS**: (link with copy icon)
- Entity ID**: (link with copy icon)
- Download metadata file**: (link)

Buttons: CANCEL, SAVE

2. Go to the SAML IdP website and add the **ACS** and **Entity ID** links 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 sign on URL
AWS SSO sign-out URL	IDP logout URL
AWS SSO issuer URL	IDP Entity ID

3. Specify parameters:

Parameter	Value
Application ACS URL	ACS link from CloudBeaver
Application SAML audience	Entity ID link from CloudBeaver

Attributes

Attributes explanation:

Attribute	Value	Meaning
Subject	<code>\${user:email}</code>	User unique identifier (nameid). It is usually an email address.

Attribute	Value	Meaning
https://aws.amazon.com/SAML/Attributes/SessionDuration	1800	Session duration in seconds. 1800 (30 minutes) is the default value
https://aws.amazon.com/SAML/Attributes/Role	roleARN, idpARN	IAM role identifier

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

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

Testing SAML authentication

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

OpenID authentication

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

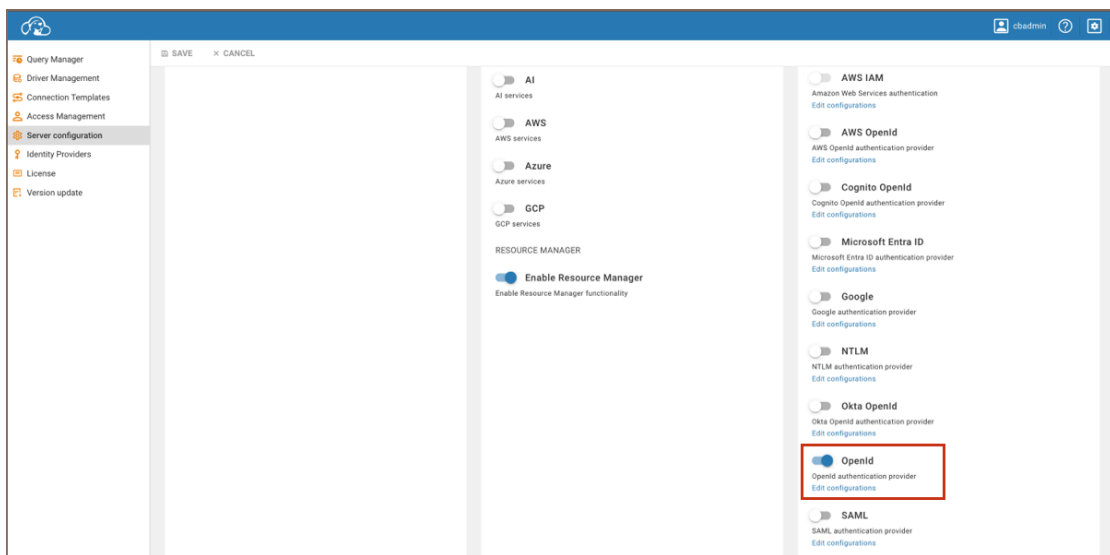
OpenID is an authentication protocol that allows users to authenticate by leveraging their existing identities from an OpenID provider. OpenID is designed for integration with third-party services, making all Identity Providers (IdPs) inherently built for integration with external services. While popular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

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

Configuration steps

Enabling OpenID Authentication

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



3. Save the changes.

Adding an Identity Provider

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

The screenshot shows the 'CONFIGURATION CREATION' form in the Identity Providers settings. The form is divided into two main sections: 'OPTIONS' and 'ENDPOINTS'. The 'OPTIONS' section includes fields for 'Provider *' (a dropdown menu with 'OpenID' selected), 'ID *' (a text field with 'google-openid' entered), 'Configuration name *' (a text field with 'OpenID' entered), 'Description' (a text area), 'Icon URL' (a text field), and a 'Disabled' checkbox. The 'ENDPOINTS' section includes fields for 'Client ID *', 'Client Secret', 'IDP auth endpoint URL *', 'IDP token endpoint URL *', 'IDP userinfo endpoint URL', and 'Name of the user groups attribute'. The 'IDP auth endpoint URL *' field contains the value 'https://accounts.google.com/o/oauth2/auth' and the 'IDP token endpoint URL *' field contains the value 'https://oauth2.googleapis.com/token'. The 'Name of an AWS role claim' field is also present but empty. The form has 'CANCEL' and 'CREATE' buttons at the top right.

Field	Description
Provider	Select <code>OpenID</code> from the dropdown menu.
ID	Enter a unique identifier for the configuration.
Configuration name	Enter a descriptive name for this configuration.
Description	Provide a brief description of this identity provider configuration.
Icon URL	Enter the URL of an icon to represent this provider.
Disabled	Leave unchecked to enable this identity provider.

Field	Description
Client ID	The client identifier provided by the OpenID Connect provider.
Client Secret	A secret key associated with the client ID for authentication.
IDP auth endpoint URL	The endpoint for initiating the authentication process.
IDP token endpoint URL	The endpoint for obtaining access and refresh tokens.
Custom scopes	The custom scopes. Use with <code>;</code> delimiter.
Name of the user groups attribute	Name of the attribute which includes user groups.
Name of the user id attribute	Name of the attribute containing the user id value. Use if your identity provider doesn't pass the user id to the standard claims <code>[email, preferred_username, sub]</code> or you want to use a custom field for the user id.
Read user info	Check this box to read user profile data using the <code>userinfo</code> endpoint URL.
Name of an AWS role claim	Name of the AWS role claim that contains the name of the AWS role.

Note

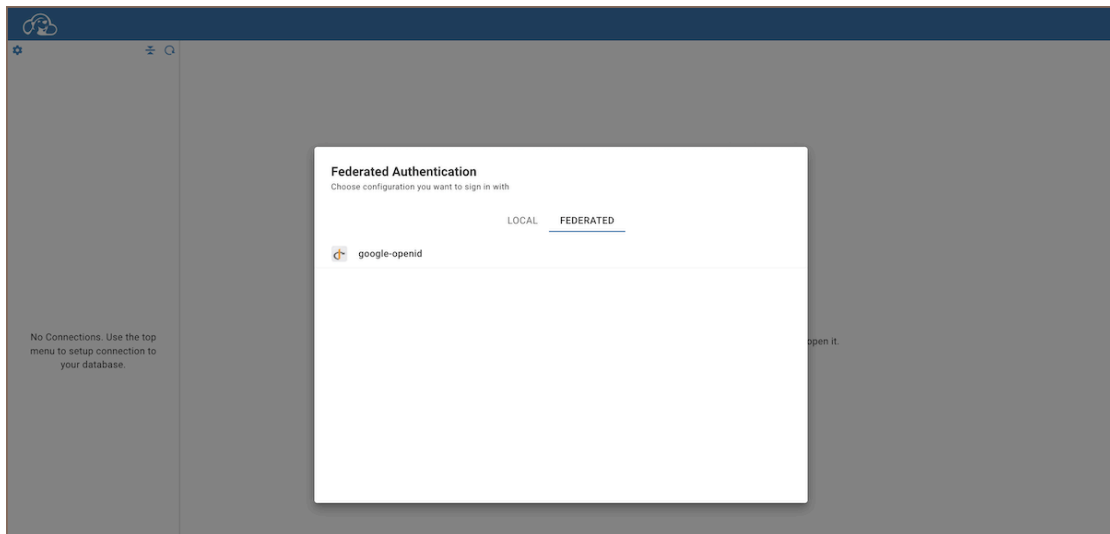
The values for the **Client ID**, **Client Secret**, **IDP auth endpoint URL**, and **IDP token endpoint URL** depend on the specific OpenID Connect provider being used.

1. Click on the **Create** button.
2. **Copy Redirect and Sign out Links:**
 - a. Enter the newly created identity provider.
 - b. Copy the **Redirect** link and the **Sign out** link.

3. Update Redirect URIs in your service.

Logging in

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



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

AWS OpenID authentication

Note

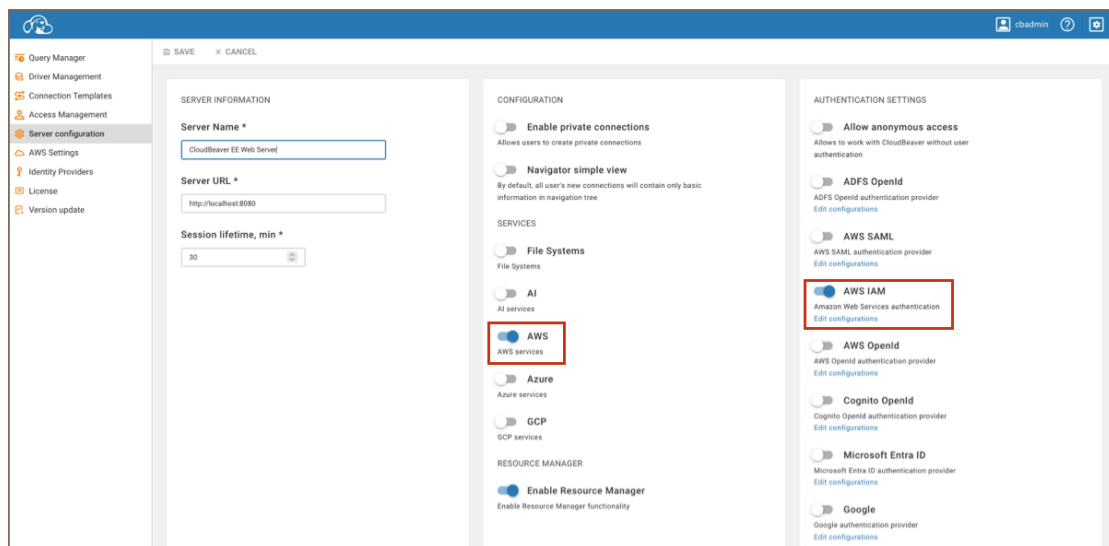
This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

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

Configuration steps

Step 1: Enabling AWS OpenID Authentication

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



Note

In CloudBeaver AWS Edition, the **Cloud (AWS)** option is enabled by default.

1. Save the changes.

Info

For instructions on configuring AWS Regions, see [AWS Settings](#).

Step 2: Adding an Identity Provider

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

The screenshot shows the 'CONFIGURATION CREATION' form for adding an identity provider. The form is divided into two main sections: 'OPTIONS' and 'CONFIGURATION CREATION'. The 'OPTIONS' section includes a 'Provider' dropdown menu (set to 'AWS OpenID'), an 'ID' field (containing 'aws-openid'), a 'Configuration name' field (containing 'aws-openid'), a 'Description' field, an 'Icon URL' field, and a 'Disabled' checkbox. The 'CONFIGURATION CREATION' section includes fields for 'Client ID', 'Client Secret', 'IDP auth endpoint URL', 'IDP token endpoint URL', 'IDP userinfo endpoint URL', 'Read user info' checkbox, 'Custom scopes' checkbox, 'Role ARN', and 'Name of an AWS role claim'. The 'Client ID' field contains '80-mf3ftrgs9980dalkphv6b.apps.googleusercontent.com'. The 'IDP auth endpoint URL' field contains 'https://accounts.google.com/o/oauth2/auth'. The 'IDP token endpoint URL' field contains 'https://oauth2.googleapis.com/token'. The 'Role ARN' field is empty. The 'Name of an AWS role claim' field is empty. The 'Disabled' checkbox is unchecked. The 'Read user info' checkbox is unchecked. The 'Custom scopes' checkbox is unchecked. The 'Cancel' and 'Create' buttons are visible at the top right of the form.

Field	Description
Provider	Select AWS OpenID from the dropdown menu.
ID	Enter a unique identifier for the configuration.
Configuration name	Enter a descriptive name for this configuration.
Description	Provide a brief description of this identity provider configuration.

Field	Description
Icon URL	Enter the URL of an icon to represent this provider.
Disabled	Leave unchecked to enable this identity provider.
Client ID	The client identifier provided by the OpenID Connect provider.
Client Secret	A secret key associated with the client ID for authentication.
IDP auth endpoint URL	The endpoint for initiating the authentication process.
IDP token endpoint URL	The endpoint for obtaining access and refresh tokens.
Custom scopes	The custom scopes. Use with ; delimiter.
Role ARN	Enter the ARN for the WebIdentity role from AWS.
Name of an AWS role claim	Name of the AWS role claim that contains the name of the AWS role.

Important

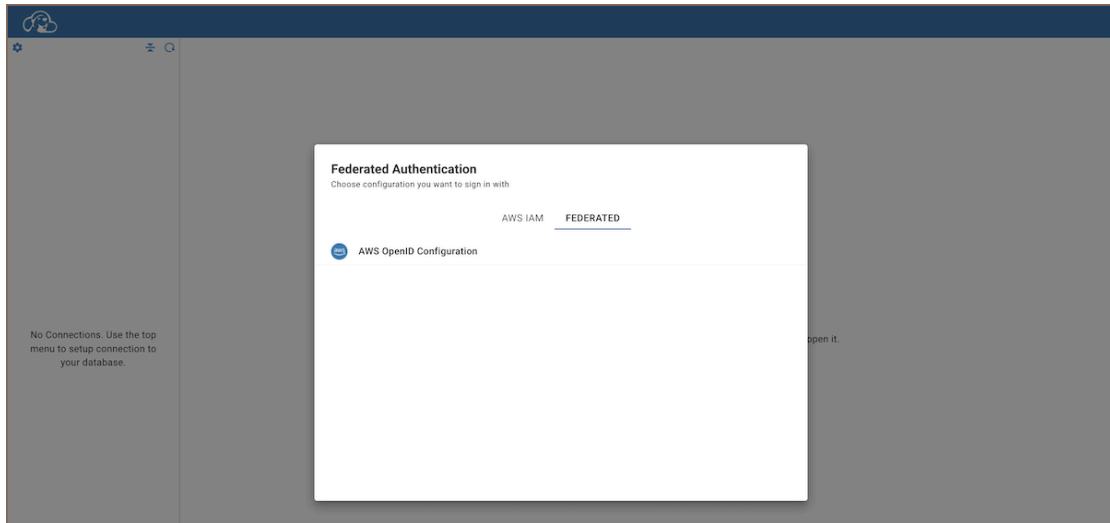
The **Role ARN** added during this step acts as the default role. It's not advisable to use an administrator role at this step. It is recommended to use a role with minimum privileges during provider setup. After the provider is configured, you will see an **AWS Role ARN** field for each user, where you can specify a role with higher privileges, if necessary.

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

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

Step 3: Logging in

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



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

AWS SAML database access via Okta

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

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

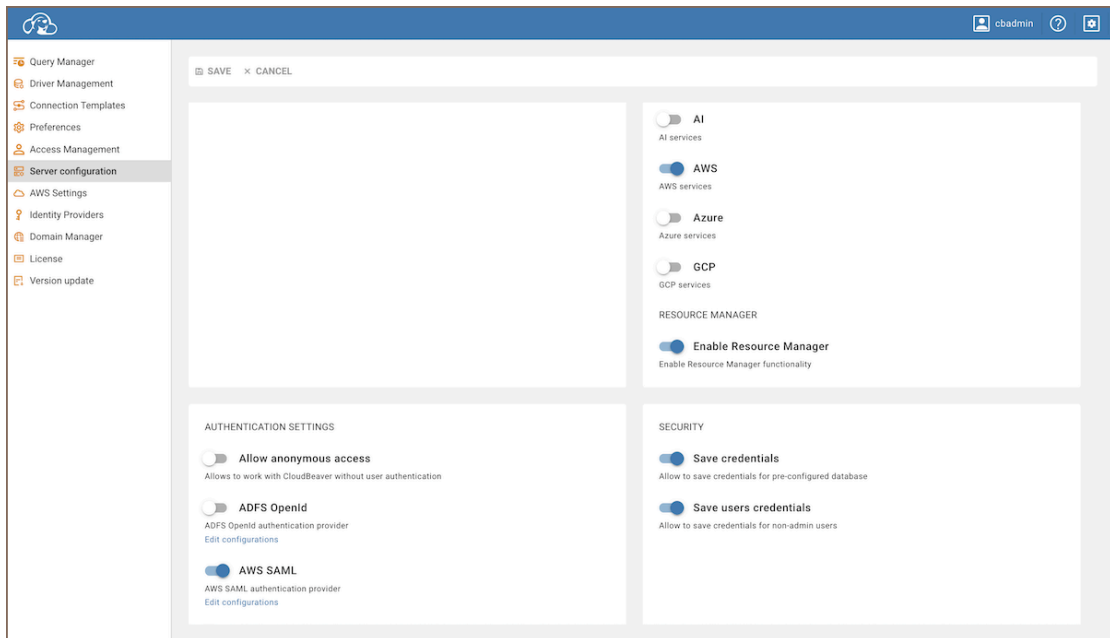
Prerequisites

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

For this type of authentication, proxy access keys and secret keys are not required. This means there is no need to generate static keys in AWS.

Configuration Steps

1. **Log in to CloudBeaver as an Administrator.**
2. **Enable AWS services and AWS SAML provider:**
 - Navigate to **Settings -> Administration -> Server Configuration** and select the checkboxes for both **Cloud (AWS)** and **AWS SAML**.



3. Create a blank Identity Provider:

- Go to **Settings -> Administration -> Identity Providers**.
- Click on the **+ Add** button to create a draft of a new identity provider.
- Initially, fill in only the necessary fields to generate the **Sign-in/Sign-out URLs, ACS** and **Entity ID** links. The remaining fields will be completed later.
- After filling in the fields, click on the **Create** button to complete the blank identity provider configuration.

Info

For instructions on configuring AWS Regions, see [AWS Settings](#).

4. Copy links:

- a. Enter the newly created identity provider.
- b. Copy the **Sign in**, **Sign out**, **ACS** and **Entity ID** links.

5. Create SAML 2.0 application in Okta:

- Initiate the process by creating SAML 2.0 application in Okta. For detailed steps, consult the official [Okta documentation on application creation](#).

- Under **Single Sign-on URL**, paste the copied from CloudBeaver application **Sign in** link.
- Locate the **Audience URI (SP Entity ID)** and paste the **Entity ID** link.
- In the same section, locate the **Show Advanced Settings** button and click to open additional settings.
 - Find **Other Requestable SSO URIs** and click the **Add Another +** button. Paste the **ACS** link and enter `0` in the **Index** field.
 - Insert `http://www.okta.com/${org.externalKey}` into the **SAML User ID** field. This

is your provider ID value, which is automatically populated.

- Click **Next** and then **Finish** to finalize these configurations.

Create SAML Integration

1 General Settings 2 **Configure SAML** 3 Feedback

A SAML Settings

General

Single sign-on URL
☒ Use this for Recipient URL and Destination URL

Audience URI (SP Entity ID)

Default RelayState
If no value is set, a blank RelayState is sent

Name ID format

Application username

Update application username on

[Show Advanced Settings](#)

What does this form do?
This form generates the XML needed for the app's SAML request.

Where do I find the info this form needs?
The app you're trying to integrate with should have its own documentation on using SAML. You'll need to find that doc, and it should outline what information you need to specify in this form.

6. Configure SAML settings in CloudBeaver:

- Access **Settings** -> **Administration** -> **Identity Providers** in CloudBeaver.
- Click on the identity provider that was set up for SAML integration.
- Retrieve SAML configuration URLs from Okta:**
 - In Okta, navigate to the **Sign On** settings of your SAML application.
 - Expand the **Hide details** section to view the necessary URLs.
- Enter SAML details into CloudBeaver:**
 - Copy the **SSO sign-on URL**, **SSO sign-out URL**, and the **Issuer URL** from Okta.
 - Paste these URLs into the corresponding fields in CloudBeaver's identity provider settings.
 - Upload the **IDP x509 certificate**, which can be downloaded from Okta, into the respective field in CloudBeaver.
 - i. **Complete additional fields**

Field Name	Description
Description (Optional)	Provide a brief description of the identity provider.

Field Name	Description
Icon URL (Optional)	Enter the URL of an icon to represent this identity provider in CloudBeaver.
User group mapping attribute (Optional)	Specify the attribute that will be used for mapping user groups between CloudBeaver and the identity provider.

- e. After entering all necessary information, save the settings to finalize the configuration of your SAML identity provider.

The screenshot displays the 'EDIT "AWS OKTA SAML"' configuration page in CloudBeaver. The interface is divided into several sections:

- Provider *:** A dropdown menu showing 'AWS SAML'.
- ID *:** A text field containing 'aws-okta-saml'.
- Configuration Name *:** A text field containing 'AWS Okta SAML'.
- Description:** A text area for additional details.
- Icon URL:** A text field for an optional icon.
- User group mapping attribute:** A text field for mapping user groups.
- SAML Section:** Includes fields for 'SSO issuer URL', 'SSO sign-on URL', 'SSO sign-out URL', and 'IDP x509 certificate'.
- AWS Section:** Includes checkboxes for 'Use federated session', 'Proxy user access key', and 'Proxy user secret key'.
- LINKS Section:** Includes fields for 'Sign in', 'Sign out', 'ACS', and 'Entity ID', each with a copy icon.
- Buttons:** 'CANCEL' and 'SAVE' buttons are located at the top right.
- Footer:** A 'Download metadata file' link is present at the bottom right.

7. **Add users to application in Okta:**
8. In your newly created Okta application, assign people or groups. For detail steps, consult the official [Okta documentation on assign applications to users](#).
9. **Add Identity Provider in AWS IAM:**
10. Add a SAML identity provider in AWS IAM. This allows AWS to authenticate users managed by Okta. For comprehensive instructions, refer to the official [AWS documentation on identity providers](#).

The screenshot shows the AWS IAM console interface for adding an identity provider. The breadcrumb trail is IAM > Identity providers > Create Identity Provider. The main heading is 'Add an Identity provider'. Under the 'Configure provider' section, 'SAML' is selected as the provider type. The 'Provider name' field is empty, and the 'Metadata document' section has a 'Choose file' button. The 'Add tags - optional' section is at the bottom.

Hint to obtain the Metadata document

- Navigate to Okta and select your application under **Applications**.
- Proceed to the **Sign On** tab, locate the **SAML 2.0** section, and click on the **Metadata URL**.
- Then, paste this URL into a new browser tab to open the XML content. Copy the XML content displayed in your browser into a new file.

1. **Configure a role for SAML 2.0 federation:**
2. The next step involves configuring an AWS role for SAML. This role will be used to grant permissions based on the authenticated identity from Okta. For a detailed walkthrough, visit the official [AWS documentation on creating roles for SAML 2.0 federation](#).

SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

SAML 2.0-based provider
Okta_SAML

Create new

Access to be allowed
☒ Allow programmatic access only
☐ Allow programmatic and AWS Management Console access

Attribute
SAML:iss

Value
http://www.okta.com/oxk

Condition - (optional)
Add condition

Hints for filling fields on the initial page when creating a role

- **Attribute:** You can choose `SAML : iss`.
- **Value:** To find Issuer, navigate to Okta and select your application under **Applications**. Then proceed to the **Sign On** tab, locate the **SAML 2.0** section, and there you will find the **Issuer URL** to copy.

1. Configure SAML attributes in Okta for AWS:

To ensure proper session management and permissions during SAML authentication with AWS, configure the following SAML attributes in Okta:

- Steps to Configure SAML Attributes:
- Go to **Applications** in Okta, select your application, and edit the **SAML Settings** under **General**.
- Scroll down to find the **SAML Settings** section and click **Edit** to modify the configurations.
- After entering the edit mode, proceed to the **Configure SAML** section.
- Locate and fill in the **Attribute statements (Optional)** settings.

Dashboard

Directory

Applications

Security

Authentication Policies

API

Reports

Settings

Attribute Statements (optional)

Name

Name format (optional)

Value

https://aws.amazon.com/SAML/Attributes/SessionDuration

Basic

1000

https://aws.amazon.com/SAML/Attributes/Role

Basic

https://aws.amazon.com/SAML/Attributes/RoleSessionName

Basic

user.email

Add Another

Group Attribute Statements (optional)

Name

Name format (optional)

Filter

Unspecified

Starts with

Add Another

Attribute Name	Name Format	Value Example	Notes
https://aws.amazon.com/SAML/Attributes/SessionDuration	Basic	3600	Set in seconds, range: 900 to 43200 seconds.
https://aws.amazon.com/SAML/Attributes/Role	Basic	arn:aws:iam::123456789012:role/SAMLRole, arn:aws:iam::123456789012:saml-provider/Okta	Format: {roleArn}, {identityProviderArn}.
https://aws.amazon.com/SAML/Attributes/RoleSessionName	Basic	user.email	Use a unique identifier like user.email or user.id.

Note

In the **Value** for the **Role** attribute, ensure the ARNs are correctly copied from AWS IAM. The `roleArn` is the ARN for the role created for SAML integration, and `identityProviderArn` is the ARN for the identity provider (Okta) configured in AWS IAM.

Set a default role and AWS session lifetime in the SAML configuration for consistent permissions, or customize via attribute mapping for user-specific roles. See [documentation](#) for details.

2. Now you can log in through the newly created Federated authentication method in CloudBeaver, using the **Configuration name** you assigned during the setup.
3. **Verify the integration of AWS SAML and Okta**
 - a. Once logged in, click on your username in CloudBeaver and navigate to the **User Info** tab.
 - b. Here, you should see two AUTH tokens. Their presence indicates that the integration of AWS OpenID and Okta has been successfully completed, and CloudBeaver has access to the necessary credentials.
4. Following successful login, you can access the databases listed in **Cloud Explorer** that are integrated with your AWS account.

AWS IAM authentication

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

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

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

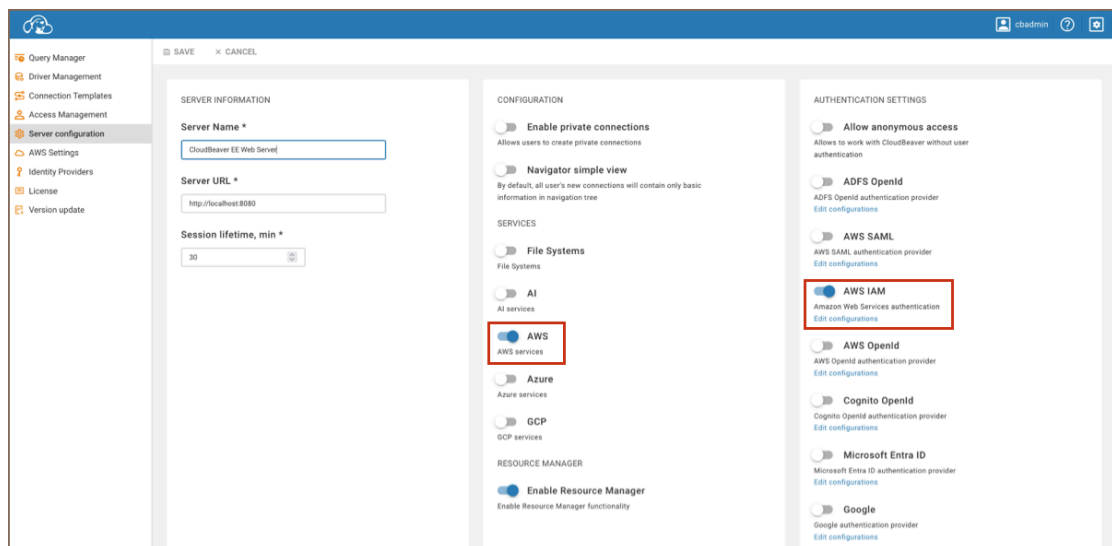
For comprehensive setup information of AWS IAM itself, refer to the [official AWS IAM documentation](#).

Configuration steps

For CloudBeaver

Step 1: Enabling AWS Authentication

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



3. Save the changes.

Info

For instructions on configuring AWS Regions, see [AWS Settings](#).

Step 2: Adding an Identity Provider

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

The screenshot shows the 'Configuration creation' dialog in the AWS IAM console. The left sidebar contains navigation links: Query Manager, Driver Management, Connection Templates, Access Management, Server configuration, AWS Settings, Identity Providers (selected), License, and Version update. The main area has a header with '+ ADD', 'REFRESH', and 'DELETE' buttons. Below the header is a 'Configuration creation' title bar with a close button. The 'OPTIONS' tab is active, showing a form with the following fields: 'Provider *' (a dropdown menu with 'AWS IAM' selected), 'ID *' (a text input field containing 'aws_iam'), 'Configuration name *' (a text input field containing 'aws config'), 'Description' (a text area), 'Icon URL' (a text input field), and a 'Disabled' checkbox. To the right of the form is a 'AWS SETTINGS' section with 'Account IDs' (a text input field containing '859875062624') and 'AWS session lifetime' (a text input field). At the bottom right of the dialog are 'CANCEL' and 'CREATE' buttons.

Field	Description
Provider	Select AWS IAM from the dropdown menu.
ID	Enter a unique identifier for the configuration.
Configuration name	Enter a descriptive name for this configuration.
Description	Provide a brief description of this identity provider configuration.
Icon URL	Enter the URL of an icon to represent this provider.
Disabled	Leave unchecked to enable this identity provider.

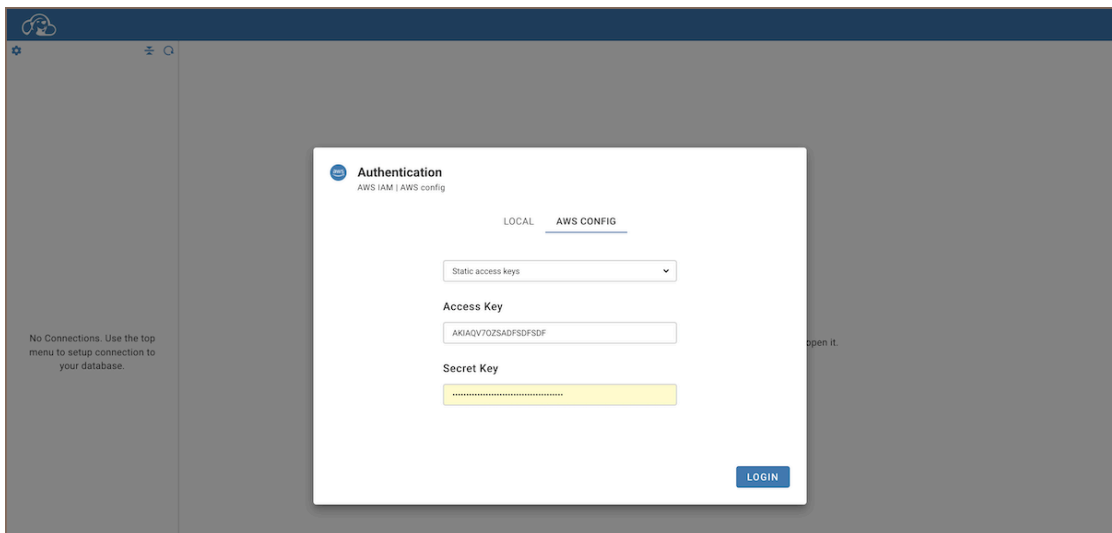
Field	Description
Account IDs	Enter AWS Account IDs, separated by commas. Only users from these accounts are allowed.
AWS session lifetime	Specify the duration for the AWS session in seconds.

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

1. Click on the **Create** button.

Step 3: Logging in

1. With the AWS configuration now established, proceed to the login screen.
2. You will be presented with two options for key types:
3. For permanent credentials, select **Static access keys**.
4. If using credentials that change regularly, select **Temporary access keys**.
5. Input your **Access Key** and **Secret Key** in the respective fields.
6. If you selected **Temporary access keys**, enter your **Session Token** in the additional field that appears.
7. Confirm your details and click the **LOGIN** button to authenticate.



The screenshot shows a web application interface with a dark blue header and a grey sidebar. The main content area is white and displays an 'Authentication' dialog box. The dialog box has a title bar with 'AWS IAM | AWS config' and two tabs: 'LOCAL' and 'AWS CONFIG'. The 'AWS CONFIG' tab is selected. Below the tabs, there is a dropdown menu with 'Static access keys' selected. Underneath, there are two input fields: 'Access Key' with the value 'AKIAQV7OZSADPDSDF' and 'Secret Key' with a masked value '.....'. A 'LOGIN' button is located at the bottom right of the dialog box. The sidebar contains a message: 'No Connections. Use the top menu to setup connection to your database.'

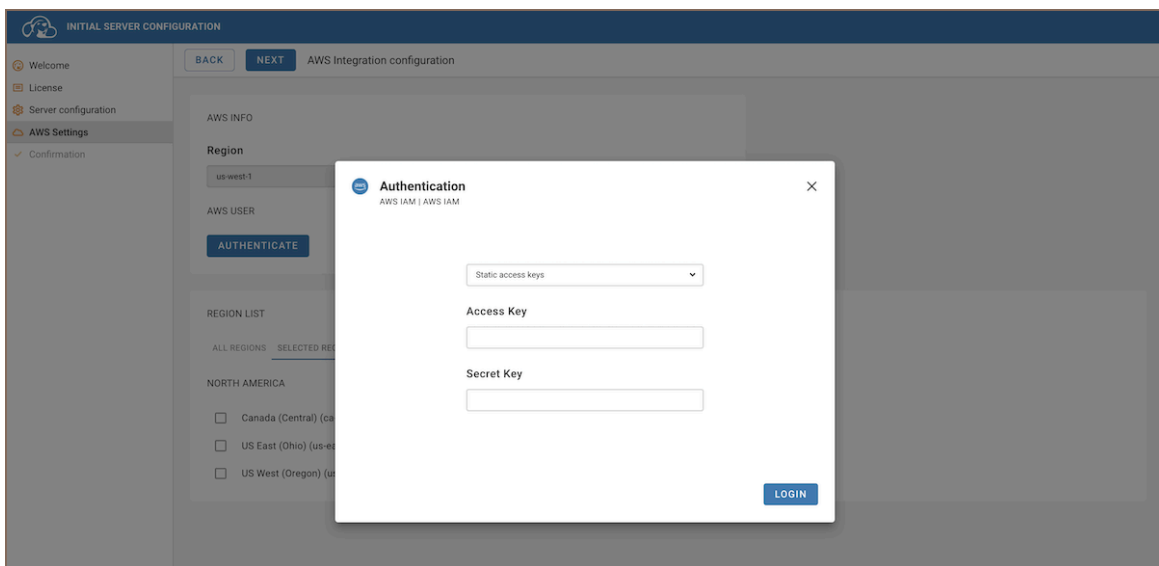
Note

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

For CloudBeaver AWS Edition

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

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



Special characteristics

- **No Server-Side Key Storage:** CloudBeaver is designed not to store AWS access and secret keys on the server, ensuring they are not held in databases or configuration files. This approach is crucial for maintaining the security of your data.
- **Automatic administrator role assignment:** The AWS user responsible for configuring CloudBeaver automatically receives administrator privileges in the CloudBeaver instance. This user will have comprehensive control over the instance's settings and configurations.
- **AWS account association:** Upon completing the server configuration, the AWS account of the administrator is associated with the CloudBeaver instance. This means that only AWS users belonging to this specific account can authenticate and access this instance of CloudBeaver.

- **User management within AWS scope:** CloudBeaver for AWS does not support the creation of new users within its platform. It solely operates with existing AWS user accounts. Consequently, every user who needs access must authenticate through their AWS account.
- **Database Authentication Requirements:** AWS databases typically have their own authorization mechanisms, requiring additional authentication parameters such as a username and password. For RDS/Aurora databases using IAM authentication, you may only need to provide the database username, leaving the password field empty.

IAM permissions

CloudBeaver 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 uses native database clients to connect and operate with most databases. It uses AWS services only to find database instances and configure database connection. The only exception is the DynamoDB service which is a database driver by itself. You can limit DynamoDB access directly in the AWS console.

AWS OpenID database access via Okta

Note

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

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

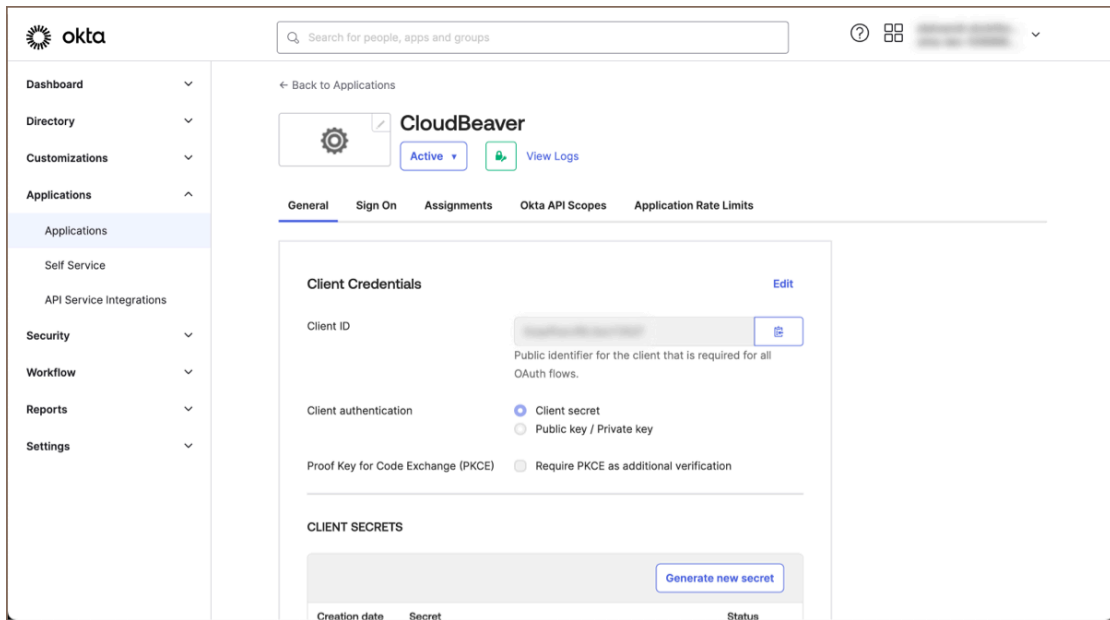
Prerequisites

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

Configuration steps

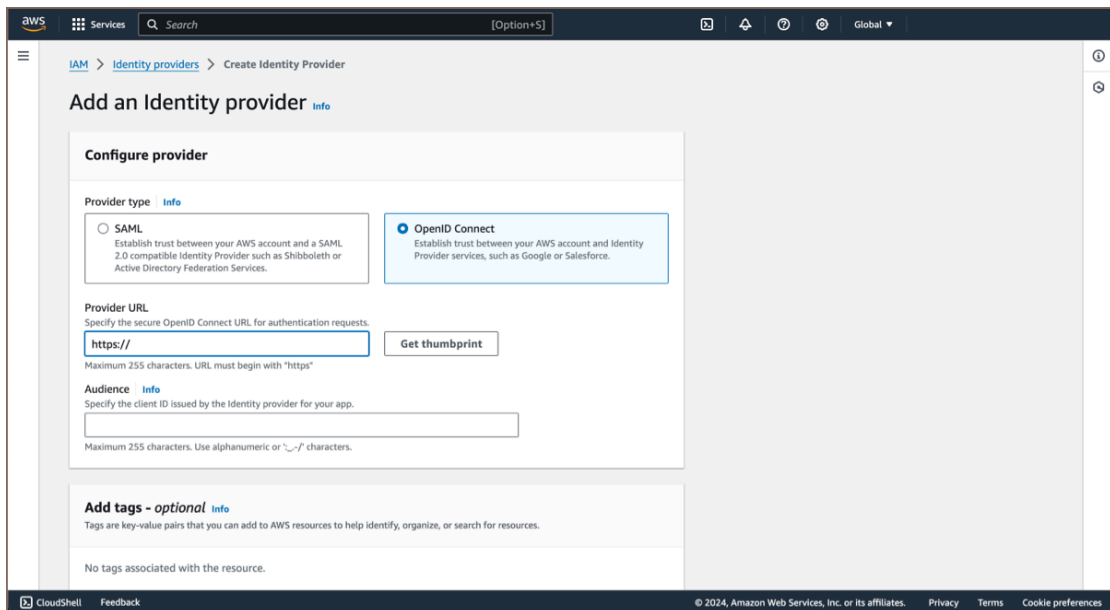
1. Create an Application in Okta:

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



2. Add Identity Provider in AWS IAM:

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

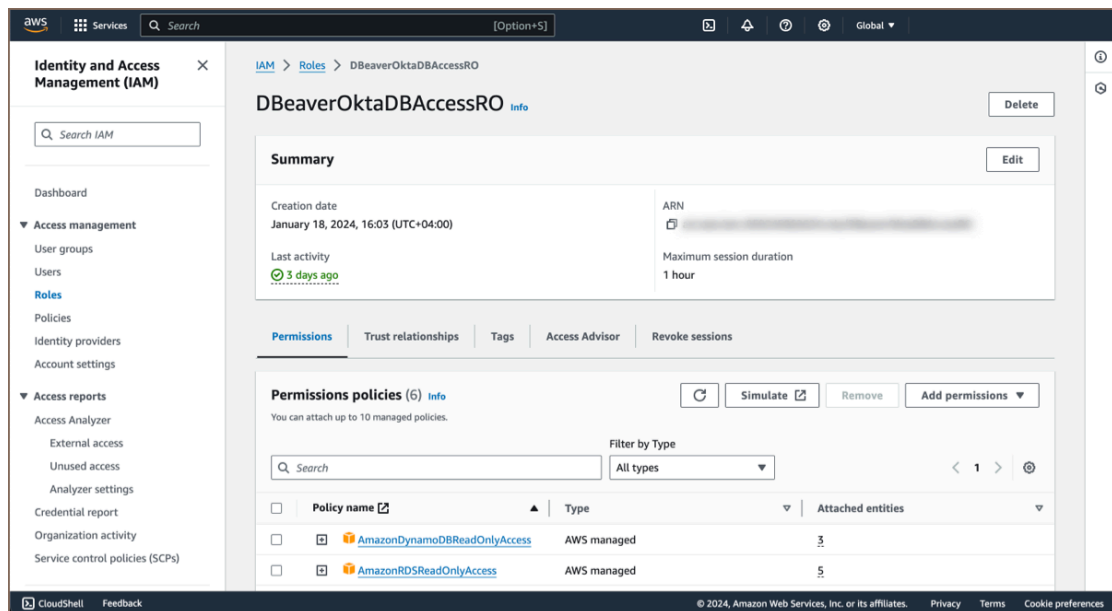


Hints for AWS IAM Identity Provider configuration

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

3. Configure a Role for Web Identity in AWS:

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



4. Log in to CloudBeaver as an Administrator.

5. Enable AWS Services and AWS OpenID Provider:

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

7. Configure Identity Provider

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

The screenshot shows the 'CONFIGURATION CREATION' page in the CloudBeaver interface. The left sidebar contains navigation links: Server Configuration, Global Preferences, Users and Teams, Connection Templates, Query Manager, Driver Management, AWS Settings, Identity Providers (selected), AI Settings, Domain Manager, and Product Information. The main content area has a 'CONFIGURATION CREATION' header with '+ ADD', 'REFRESH', and 'DELETE' buttons. Below the header is the 'OPTIONS' section with the following fields:

- Provider ***: AWS OpenID (dropdown)
- ID ***: aws-openID (text input)
- Configuration name ***: aws-openID (text input)
- Description**: (text input)
- Icon URL**: (text input)
- Client ID ***: 80-mf3fnvgs9390dakphv6b.apps.googleusercontent.com (text input)
- Client Secret**: (text input)
- IDP auth endpoint URL ***: nups/accounts.google.com/o/oauth2/auth (text input)
- IDP token endpoint URL ***: https://oauth2.googleapis.com/token (text input)
- IDP userinfo endpoint URL**: (text input)
- Read user info**: (checkbox)
- Custom scopes**: (text input)
- Role ARN**: (text input)
- Name of an AWS role claim**: (text input)

At the bottom right of the form are 'CANCEL' and 'CREATE' buttons.

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

Field	Description
Provider	Select AWS OpenID from the dropdown menu.
ID	Enter a custom name for the identity provider.
Configuration name	Specify the configuration name.
Description (optional)	Provide a brief description of the identity provider.
Icon URL (optional)	Enter the URL of an icon to represent this identity provider in CloudBeaver.
Client ID	Use the Client ID from the Okta application.
Client Secret	Use the Client Secret from the Okta application.

Field	Description
IDP auth endpoint URL	Format as <code>https://{okta_domain}/oauth2/v1/authorize</code> .
IDP token endpoint URL	Format as <code>https://{okta_domain}/oauth2/v1/token</code> .
Custom scopes	The custom scopes. Use with <code>;</code> delimiter.
Role ARN	Enter the ARN for the WebIdentity role from AWS.
Name of an AWS role claim	Name of the AWS role claim that contains the name of the AWS role.

Important

The **Role ARN** added during this step acts as the default role. It's not advisable to use an administrator role at this step. It is recommended to use a role with minimum privileges during provider setup. After the provider is configured, you will see an **AWS Role ARN** field for each user, where you can specify a role with higher privileges, if necessary.

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

Info

For instructions on configuring AWS Regions, see [AWS Settings](#).

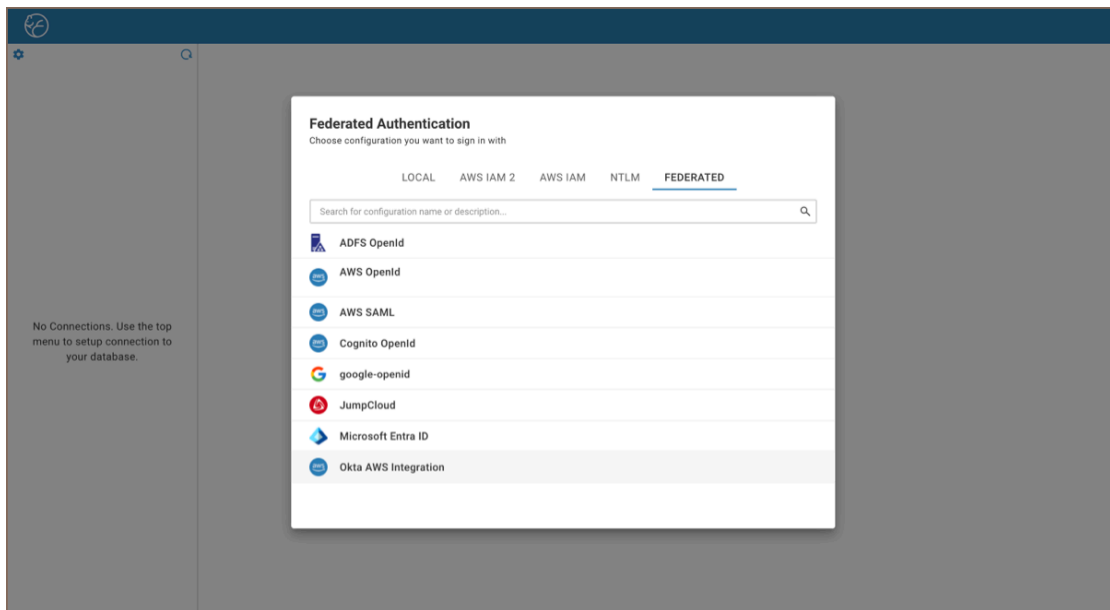
8. **Copy Redirect and Sign out Links:**
9. Enter the newly created identity provider.
10. Copy the **Redirect** link and the **Sign out** link.

The screenshot shows the AWS IAM console interface for configuring an OpenID Connect provider. The left sidebar contains navigation links: Server Configuration, Global Preferences, Users and Teams, Connection Templates, Query Manager, Driver Management, AWS Settings, Identity Providers (selected), AI Settings, Domain Manager, and Product Information. The main content area displays a table with one configuration named 'aws-openID' using the 'aws-openid' provider. Below the table, the 'OPTIONS' tab is active, showing fields for Provider (AWS OpenID), ID (aws-openID), Configuration name (aws-openID), and Description. It also includes fields for Client ID, Client Secret, IDP auth endpoint URL, IDP token endpoint URL, IDP userinfo endpoint URL, Role ARN, Name of an AWS role claim, and a Redirect link. There are checkboxes for 'Disabled' and 'Read user info', and a 'Custom scopes' section. 'CANCEL' and 'SAVE' buttons are at the top right of the options panel.

11. **Update Redirect URIs in Okta:**
12. In your Okta application, navigate to **General -> Login**.
13. Under **Sign-in redirect URIs**, paste the copied **Redirect** link.
14. In the same section, locate **Sign-out redirect URIs** and paste the **Sign out** link there.
15. Click **Save** in Okta to finalize these configurations.

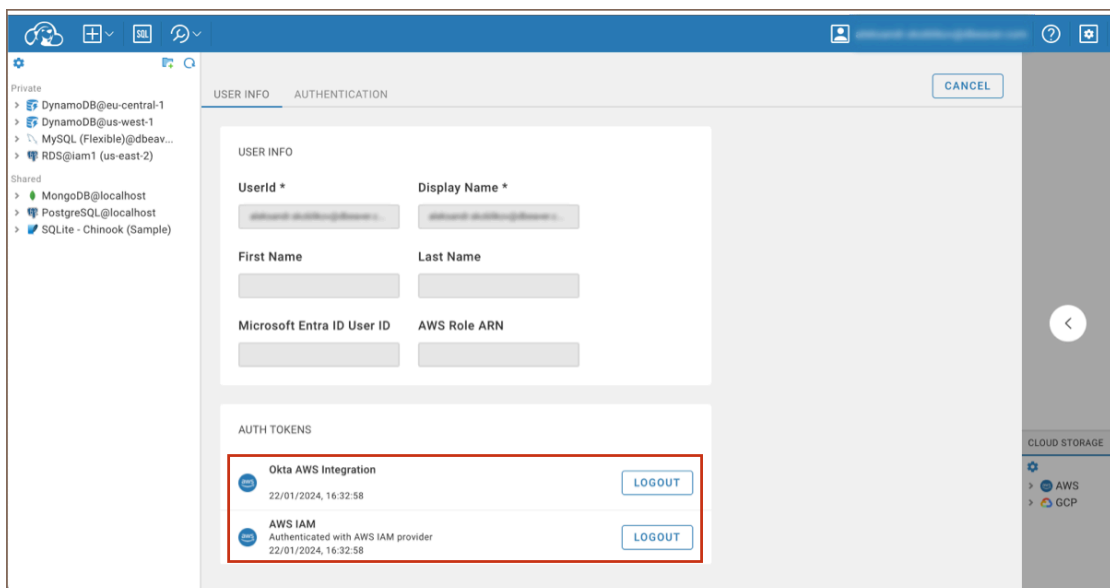
The screenshot shows the Okta application settings page. The left sidebar contains navigation links: Dashboard, Directory, Customizations, Applications (selected), Security, Workflow, Reports, and Settings. The main content area shows the 'Grant type' section with 'Client acting on behalf of a user' selected, and 'Authorization Code' chosen. Below this, the 'Sign-in redirect URIs' section is active, showing a list of URIs with a '+ Add URI' button. The 'Sign-out redirect URIs (Optional)' section is also visible, with a '+ Add URI' button. The 'Trusted Origins' section is at the bottom.

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



17. **Verify the Integration of AWS OpenID and Okta**

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



18. Following successful login, you can access the databases listed in **Cloud Explorer** that are integrated with your AWS account.

Snowflake OpenID database access via Okta

Note

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

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

Note

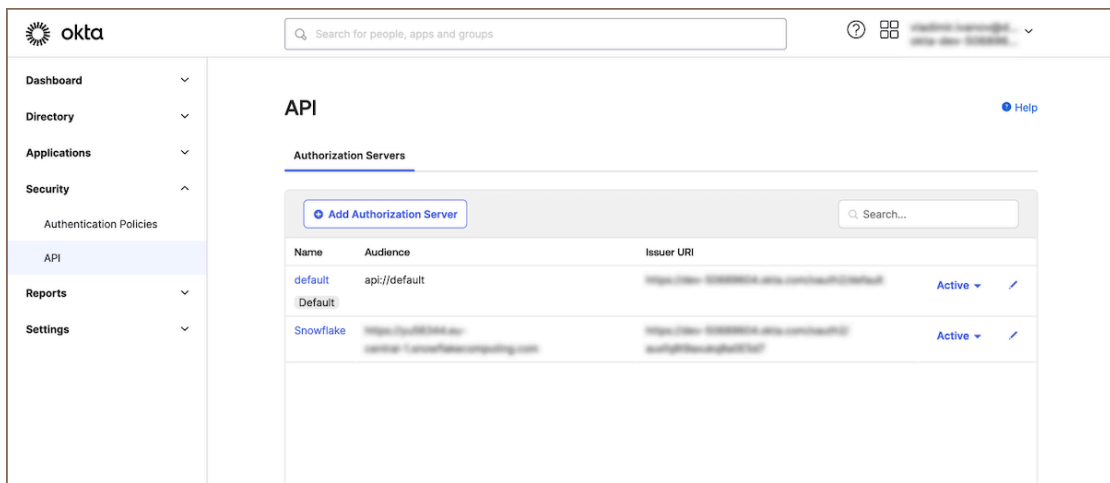
While this article provides an example specifically for Okta, you may use any Snowflake-compatible identity provider.

Prerequisites

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

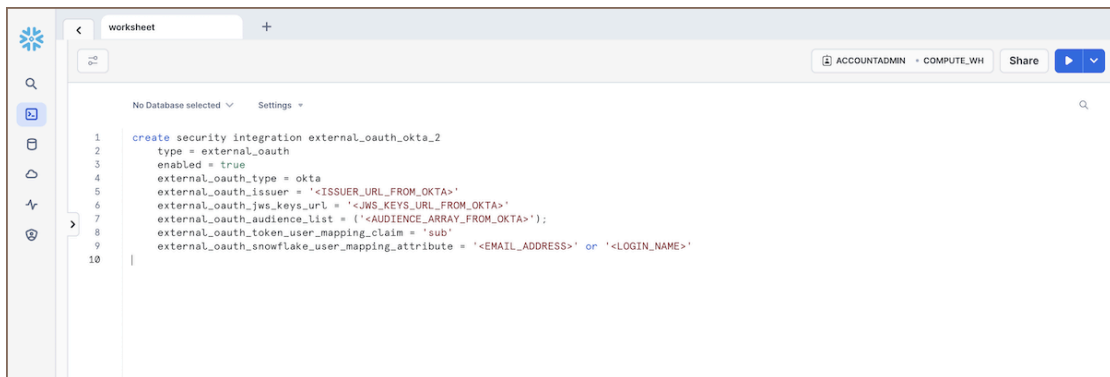
Configuration steps

1. **Create an Authorization Server in Okta:** Begin by setting up an authorization server in Okta. This server will manage the security tokens needed for user authentication and authorization. For a comprehensive guide on how to create and configure an authorization server, refer to the [Okta documentation on custom authorization servers](#).



2. Create a security integration in Snowflake:

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



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

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

'LOGIN_NAME'

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

Info

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

5. Allow dynamic roles:

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

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

```
alter security integration external_oauth_okta_2 SET
  external_oauth_any_role_mode = 'ENABLE'
  external_oauth_allowed_roles_list = ('<SNOWFLAKE_ROLE_1>',
```



```
'<SNOWFLAKE_ROLE_2' )
```

```
grant USE_ANY_ROLE on integration external_oauth_okta_2 to <SNOWFLAKE  
ROLE_1>
```

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

Tip

In addition to predefined roles, you can grant a universal role assignment capability within Snowflake. Use the following command to allow all authenticated users to assume any role: `grant USE_ANY_ROLE on integration external_oauth_okta_2 to <SNOWFLAKE_ROLE_1>;` Use this setting with caution, as it gives broad permissions to all users.

3. Create API access Scopes in Okta:

After establishing the authorization server, the next step involves creating API access scopes within Okta. These scopes define the access permissions that an OAuth token grants in Snowflake, ensuring that users can only perform actions aligned with their roles in Snowflake.

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

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

Follow these guidelines to create API access scopes:

1. Navigate to your Okta web interface.
2. Go to **Security -> API**.

3. Select the authorization server you created earlier.
4. Click on the **Scopes** tab.
5. Here, you can add new scopes that correspond to the roles and permissions you have defined in Snowflake.

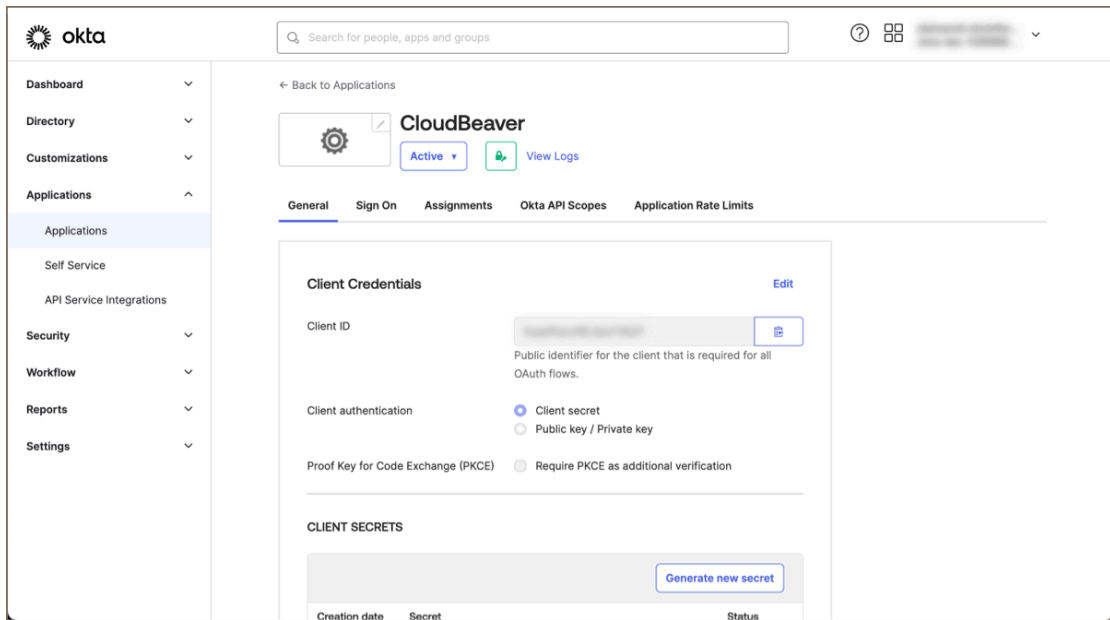
To make sure that users have the right access in Snowflake, match the Okta scopes with Snowflake roles. For instance:

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

```
> For detailed instructions on setting up API access scopes in Okta, consult  
> the [Okta documentation on API scopes](https://help.okta.com/en-us/content/  
topics/security/api-config-scopes.htm).
```

1. Create an Application in Okta:

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



2. **Log in to CloudBeaver as an Administrator.**
3. **Enable OpenID:**
Navigate to **Settings -> Administration -> Server Configuration** and select the checkboxes for **OpenID**.
4. **Configure Identity Provider**
5. Continue to **Settings -> Administration -> Identity Providers**.
6. Click on the **+ Add** button to begin configuring a new identity provider.
7. After filling in the fields, click on the **Create** button to complete the identity provider configuration.

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

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

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

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

The screenshot shows the Okta Identity Providers configuration page for an 'OpenID' provider. The left sidebar contains navigation links: Query Manager, Driver Management, Connection Templates, Preferences, Access Management, Server configuration, AWS Settings, Identity Providers (selected), All Settings, License, and Version update. The main content area has a table with columns: CONFIGURATION NAME, PROVIDER, DESCRIPTION, and DISABLED. Below the table is the 'OPTIONS' section, which is divided into four tabs: Provider *, OPENID, AWS, and LINKS. The 'LINKS' tab is active and contains two fields: 'Sign in' and 'Sign out', both with a copy icon. The 'Sign out' field is highlighted with a red rectangle. The 'Provider *' tab shows fields for ID *, Configuration name *, Description, and Icon URL. The 'OPENID' tab shows fields for Client ID *, Client Secret, IDP auth endpoint URL *, IDP token endpoint URL *, and IDP userinfo endpoint URL. The 'AWS' tab shows a field for Name of an AWS role claim. The 'DISABLED' column has a 'DISABLED' checkbox.

4. **Update Redirect URIs in Okta:**
 - a. In your Okta application, navigate to **General -> Login**.
 - b. Under **Sign-in redirect URIs**, paste the copied **Redirect** link.
 - c. In the same section, locate **Sign-out redirect URIs** and paste the **Sign out** link there.
 - d. Click **Save** in Okta to finalize these configurations.

Okta Admin Console configuration for a new connection:

- Grant type**
 - Client acting on behalf of itself
 - ☐ Client Credentials
 - Client acting on behalf of a user
 - ☒ Authorization Code
 - ☐ Refresh Token
 - ☐ Client-initiated backchannel authentication flow (CIBA)
 - ☐ Implicit (hybrid)
- Sign-in redirect URIs**

Okta sends the authentication response and ID token for the user's sign-in request to these URIs.

☐ Allow wildcard * in sign-in URI redirect.

+ Add URI
- Sign-out redirect URIs (Optional)**

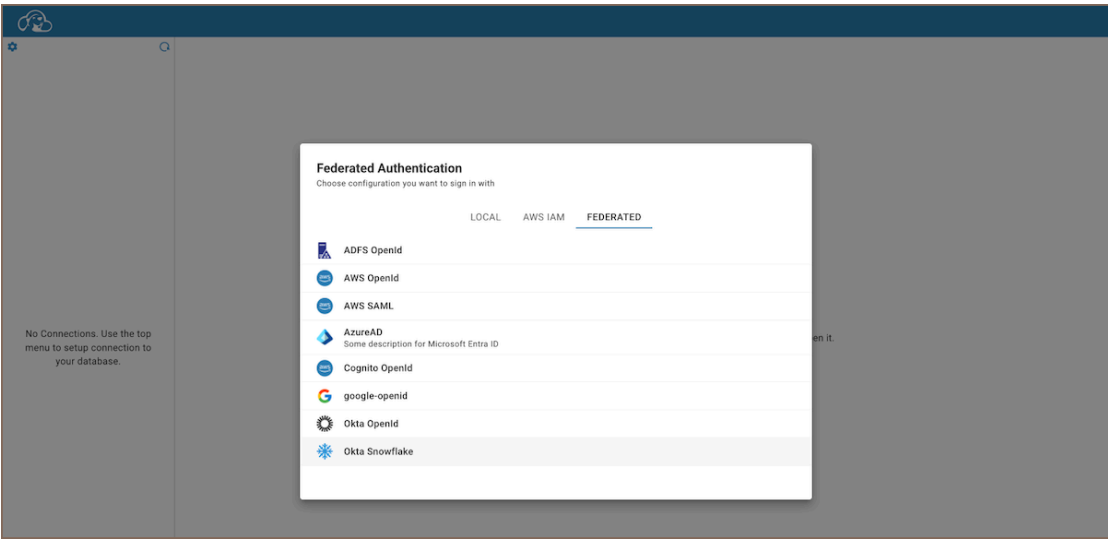
After your application contacts Okta to close the user session, Okta redirects the user to one of these URIs.

+ Add URI
- Trusted Origins**

5. Set SSO Authentication for a New Connection

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

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



Okta OpenID authentication

Note

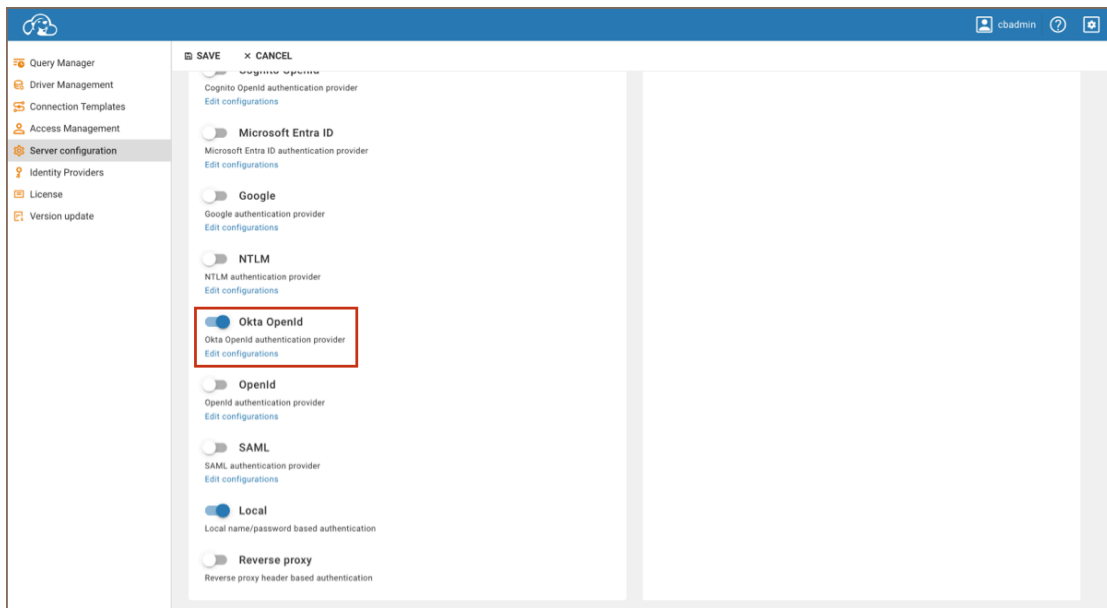
This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

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

Configuration steps

Step 1: Enabling Okta OpenID Authentication

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



3. Save the changes.

Step 2: Adding an Identity Provider

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

The screenshot shows the 'Configuration creation' form in the Identity Providers section. The form is titled 'Configuration creation' and has a 'CANCEL' button and a 'CREATE' button. It is divided into three main sections: 'Provider *', 'OPENID', and 'AWS'. The 'Provider *' section has a dropdown menu for 'Okta OpenID' and a text field for 'ID *' with the value 'okta'. Below this are fields for 'Configuration name *' (Okta OpenID), 'Description' (empty), and 'Icon URL' (empty). There is also a 'Disabled' checkbox. The 'OPENID' section has a 'Client ID *' field with the value '898EXAMPLEdjsokMjdsf8', a 'Client Secret' field (empty), and an 'Okta Domain *' field with the value 'example-98209384092.okta.com'. There are two checkboxes: 'Read Okta group information' and 'Read user info', both unchecked. The 'AWS' section has a 'Name of a AWS role claim' field with the value 'aws-role'.

Field	Description
Provider	Select Okta OpenID from the dropdown menu.
ID	Enter a unique identifier for the configuration.
Configuration name	Enter a descriptive name for this configuration.
Description	Provide a brief description of this identity provider configuration.
Icon URL	Enter the URL of an icon to represent this provider.
Disabled	Leave unchecked to enable this identity provider.
Client ID	The client identifier provided by the OpenID Con

Field	Description
	nect provider.
Client Secret	A secret key associated with the client ID for authentication.
Okta Domain	Organization domain in Okta.
Read Okta group information	If checked than Active Directory user group information will be claimed. May be required for Okta permissions integration.
Read user info	Read user profile data, using <code>userinfo</code> endpoint URL.
Custom scopes	The custom scopes. Use with <code>;</code> delimiter.
Name of an AWS role claim	Name of the AWS role claim that contains the name of the AWS role.

1. Click on the **Create** button.
2. **Copy Redirect and Sign out Links:**
 - a. Enter the newly created identity provider.
 - b. Copy the **Redirect** link and the **Sign out** link.

The screenshot shows the Okta Admin Console interface for configuring an OpenID provider. The left sidebar lists various management tools. The main area displays a table of providers with 'Okta OpenId' selected. Below the table, the 'OPTIONS' tab is active, showing configuration fields for the provider. The 'LINKS' section on the right contains three input fields: 'Sign in', 'Sign out', and 'Redirect'. The 'Sign out' field is highlighted with a red rectangular box.

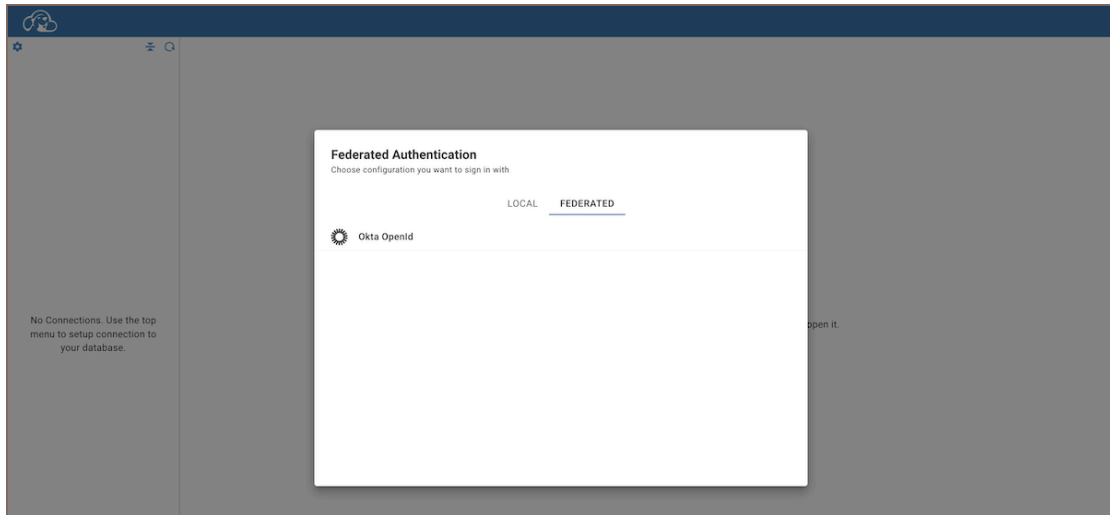
3. Update Redirect URIs in Okta:

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

The screenshot shows the Okta Admin Console interface for configuring an application. The left sidebar lists various management tools. The main area displays the 'Grant type' and 'Sign-in redirect URIs' sections. The 'Sign-in redirect URIs' section has a text input field and an 'Add URI' button. Below it, the 'Sign-out redirect URIs (Optional)' section also has a text input field and an 'Add URI' button. The 'Sign-out redirect URIs' input field is highlighted with a red rectangular box.

Step 3: Logging in

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



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

Cognito OpenID authentication

Note

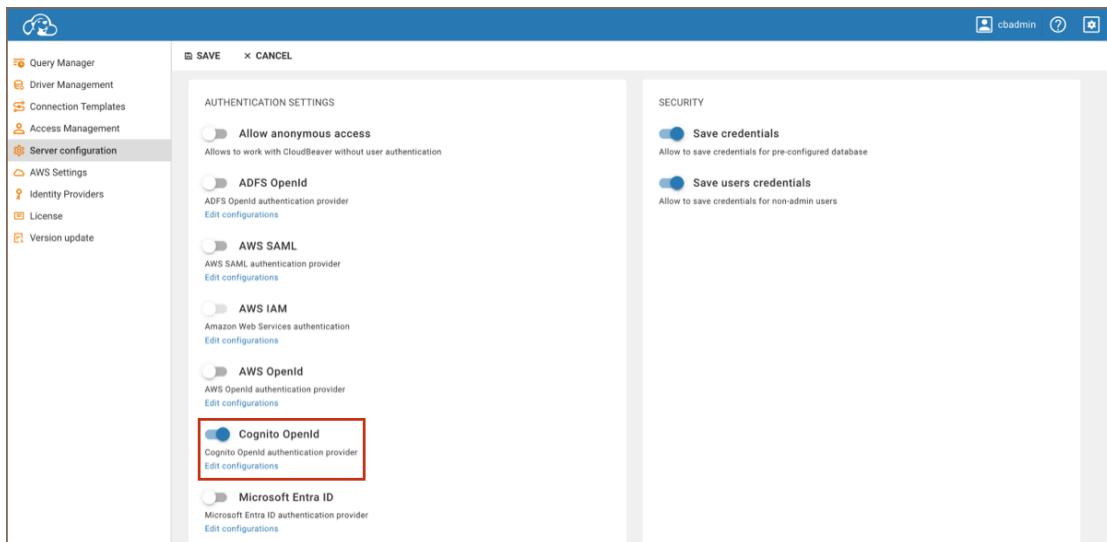
This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

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

Configuration steps

Step 1: Enabling Cognito OpenID Authentication

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



3. Save the changes.

Step 2: Adding an Identity Provider

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

2. Click on the **+ Add** button.
3. Fill in the following fields:

The screenshot shows the 'Configuration creation' page in the AWS IAM console. The left sidebar contains navigation links: Query Manager, Driver Management, Connection Templates, Access Management, Server configuration, AWS Settings, Identity Providers (selected), License, and Version update. The main area has a header with '+ ADD', 'REFRESH', and 'DELETE' buttons. Below the header is a 'Configuration creation' title bar with a close button. The 'OPTIONS' tab is active, showing three columns of fields:

- Provider *:** A dropdown menu with 'Cognito OpenID' selected.
- ID *:** A text field containing 'google-cognito'.
- Configuration name *:** A text field containing 'Cognito OpenID'.
- Description:** An empty text area.
- Icon URL:** An empty text field.
- Disabled:** An unchecked checkbox.
- OPENID:**
 - Client ID:** A text field containing 'j19390d3423469uqakiphvib.apps.googleusercontent.com'.
 - Client Secret:** A text field containing 'WRTSDF-dnRFFqRg3n3rTR1EoER6RuzR'.
 - IDP auth endpoint URL:** A text field containing 'https://accounts.google.com/o/oauth2/auth'.
 - IDP token endpoint URL:** A text field containing 'https://oauth2.googleapis.com/token'.
- COGNITO:**
 - Region:** A dropdown menu with 'us-east-1' selected.
 - Identity pool ID:** A text field containing 'eu-central-1:4534db1e-d766-4b04-34c2-23f9e9078ade'.
 - Custom role ARN:** An empty text field.
 - User pool authentication:** An unchecked checkbox.

At the bottom right of the main area are 'CANCEL' and 'CREATE' buttons.

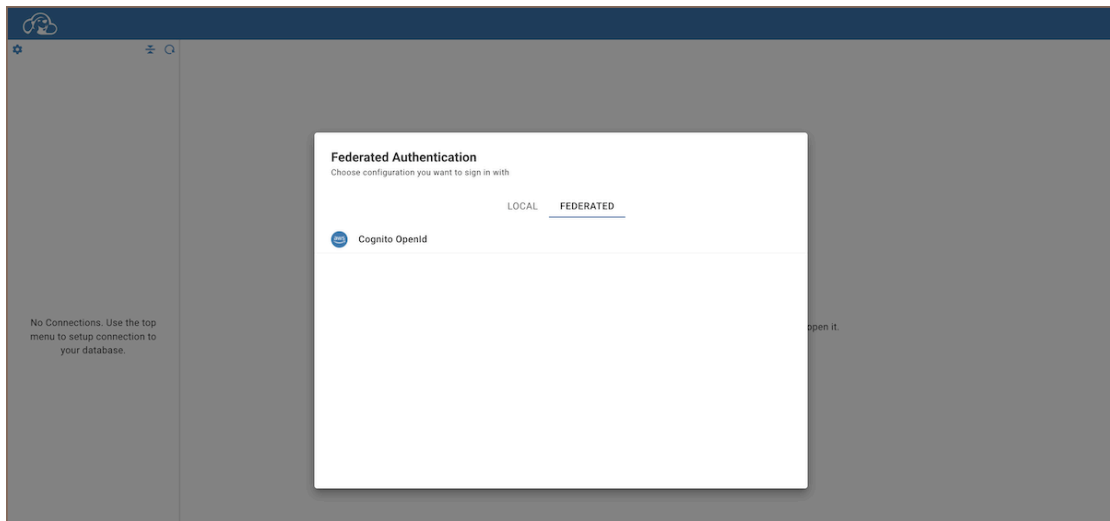
Field	Description
Provider	Select Cognito OpenID from the dropdown menu.
ID	Enter a unique identifier for the configuration.
Configuration name	Enter a descriptive name for this configuration.
Description	Provide a brief description of this identity provider configuration.
Icon URL	Enter the URL of an icon to represent this provider.
Disabled	Leave unchecked to enable this identity provider.
Client ID	The client identifier provided by the OpenID Connect provider.
Client Secret	A secret key associated with the client ID for auth

Field	Description
	entication.
IDP auth endpoint URL	The endpoint for initiating the authentication process.
IDP token endpoint URL	The endpoint for obtaining access and refresh tokens.
IDP userinfo endpoint URL	Fill in with the endpoint URL found in the Amazon Cognito User Pool under "App integration".
Region	Specify the AWS region of your Cognito User Pool.
Identity pool ID	Enter the ID of your Cognito Identity Pool.
Custom role ARN	Provide ARN of the role that will be used by the users during authorization.

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

Step 3: Logging in

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



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

JWT authentication

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

In CloudBeaver, you can use JWT (JSON Web Tokens) authentication to securely exchange claims between the client and the server. This method allows you to authenticate using JWTs, which the server validates to grant access to its resources. This article outlines the basic settings necessary for JWT authentication in CloudBeaver.

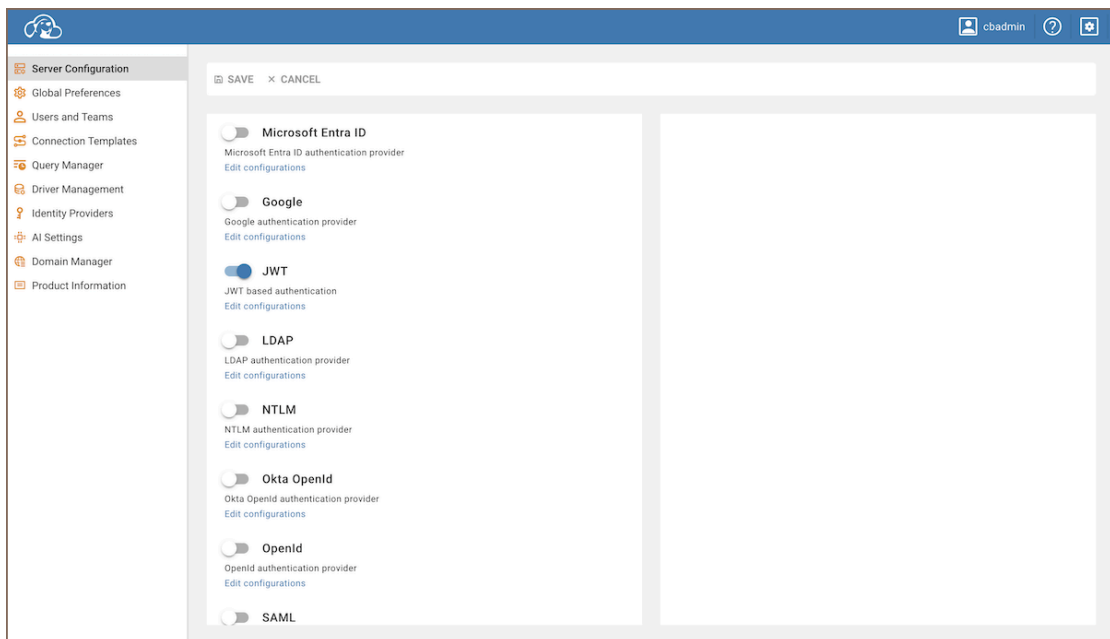
Info

For more information on JWT standards, see [JWT official documentation](#)

Configuration steps

Step 1: Enabling JWT authentication

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



3. Save changes.

Step 2: Create JWT

1. Create the payload of your JWT. Use the template below and modify its values according to your needs:

```
{
  "email": "example@email.com",
  "firstName": "Name",
  "lastName": "Surname",
  "team": "user"
}
```

Tip

For more information on teams, see [Teams](#) article.

Alongside the payload, create a sufficiently large and secure secret key. Here is an example of a robust secret key:

```
8Sa8sdfkj980stCLyV6XICnI2TLoFcgzwtwgX94joMiaMnB8PTft6EDBxBqJpiIieeg7b26b9oiXKUx0
0s9i1lmGD11hpB2e0IiVuLQNxAZ9CMdScnchzprIjgnK0eKaYHjUVZpNlnckxc0zzix4hLPsdfsdfsdf
fq2WyFNJ2Juw10nEedA8MbgzSyfqbL0s02gXHLiRdakxbhsWg6NVvUMMTcsDLQZDxAzvLPD0MtKEIy6Vn
8kSC0icog5Q187Uw0swlxNNAnt82cLcBa0woHUFU0jZet3Rdu
```

2. Generate your JWT.

Tip

You can use [jwt.io](#). Enter the payload data in the `PAYLOAD : DATA` section and your secret key in the `VERIFY SIGNATURE` section. `Jwt . io` will automatically create the encoded JWT for you.

Algorithm

HS256

Encoded

PASTE A TOKEN HERE

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFPbCI6InRlc3RKV1RAZ21haWwuy29tIiwiaWF0IjoiMjc3ODYyOTkxNjE2Iiwicm9vdCI6Imxhc3R5ZWl1Ijo1U3VybmFtZSI6InRlYW0iOj1c2VyIn0.EAlzZtoNY14TelPoTemTz6bihKhNaFTHvtahy2MbZyA

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

{
 "alg": "HS256",
 "typ": "JWT"
}

PAYLOAD: DATA

{
 "email": "testJWT@gmail.com",
 "firstName": "Name",
 "lastName": "Surname",
 "team": "user"
}

VERIFY SIGNATURE

HMACSHA256(
 base64UrlEncode(header) + "." +
 base64UrlEncode(payload),
 ?cLcBa0woHUfU0jZet3Rdu
) ☐ secret base64 encoded

Step 3: Adding an Identity Provider

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

Server Configuration

Global Preferences

Users and Teams

Connection Templates

Query Manager

Driver Management

AWS Settings

Identity Providers

AI Settings

Domain Manager

Product Information

+ ADD

REFRESH

DELETE

CONFIGURATION CREATION

You can configure the list of enabled Identity Providers in the Server Configuration settings

OPTIONS

CANCEL

CREATE

Provider *

JWT

ID *

JWT

Configuration name *

JWT

Description

Icon URL

☐ Disabled

JWT

Cookie name

Email attribute

Team attribute

First name attribute

Last name attribute

Name of an AWS role claim attribute

DECRYPTION

☐ Jwk token

Public key

Secret key

HEADERS

☐ Use additional header

Team header

Public key RSA

Field	Description
Provider	Select <code>JWT</code> from the dropdown menu.
ID	Enter a unique identifier for the configuration.
Configuration name	Enter a descriptive name for this configuration.
Description	Provide a brief description of this identity provider configuration.
Icon URL	Enter the URL of an icon to represent this provider.
Disabled	Leave unchecked to enable this identity provider.
Cookie Name	Set the name of the cookie to store the JWT.
Email Attribute	Specify the custom attribute that represents the user's email. Default value: <code>"email"</code> .

Field	Description
Team Attribute	Specify the custom attribute that represents the user's team. Default value: <code>"team"</code> .
First Name Attribute	Specify the custom attribute that represents the user's first name. Default value: <code>"firstName"</code> .
Last Name Attribute	Specify the custom attribute that represents the user's last name. Default value: <code>"lastName"</code> .
Name of an AWS role claim attribute	Use this attribute to identify the AWS IAM role claim within the JWT.
Jwk token	Enable option for decrypting incoming tokens.
Public Key	Enter the public key for validating the JWT signature.
Secret Key	Enter the secret key used in Step 2 .
Use additional header	Enable if additional headers for the JWT authentication requests.
Team Header	Set the custom header to be used for the team verification.
Decrypt RSA256 Header	Enable if RSA256 decryption of the header is required.
Public Key RSA	Enter the RSA public key for additional security checks.

Tip

You can insert the Public key with the `-----BEGIN PUBLIC KEY-----` and `-----END PUBLIC KEY-----` headers, or without them.

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

Step 4: Insert JWT in browser cookies

You need to insert the JWT into your browser cookies at this step. Below is a straightforward method to achieve this:

1. Access the developer tools in your web browser. This is generally achieved by right-clicking on a web page and selecting **Inspect** or by using shortcuts like `F12`.
2. In the developer tools, go to the **Application** tab. Here, locate the **Cookies** section under the **Storage** header on the left. This section allows you to view and manipulate cookies related to the current website.
3. Add your cookie:
 - Click on the **Add new** button in the cookies panel.
 - In the `Name` field, enter the **Cookie Name**, used in [Step 3](#).
 - In the `Value` field, paste your JWT token, generated in [Step 2](#).
4. After inputting your cookie's name and value, save the changes.

Step 5: Logging in

By entering the JWT token into the cookie, users can automatically authenticate upon visiting the CloudBeaver login page. This method eliminates the need for refreshing the page or manually logging in each time, providing a seamless access experience to the application.

NTLM user authentication

Note

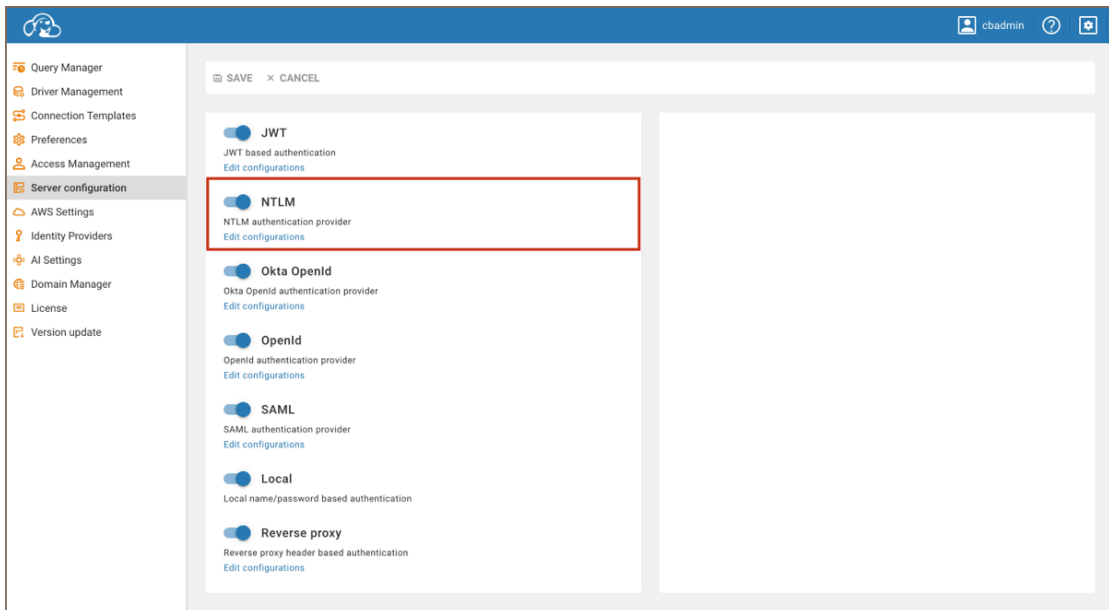
This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

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

Configuration steps

Step 1: Enabling NTLM Authentication

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



3. Save the changes.

Step 2: Adding an Identity Provider

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

2. Click on the **+ Add** button.
3. Fill in the following fields:

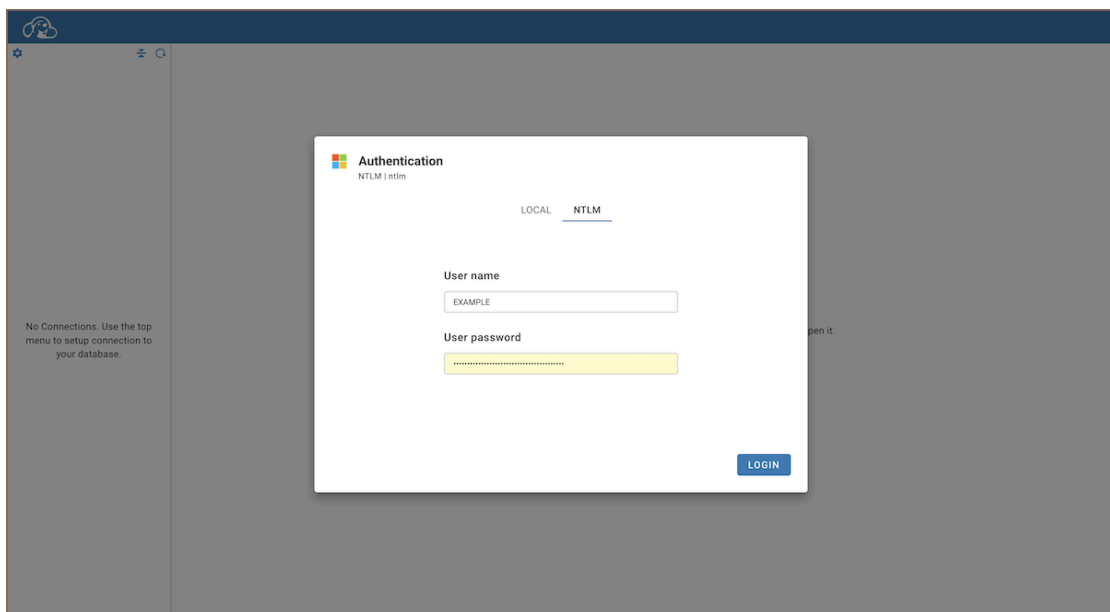
Field	Description
Provider	Select NTLM from the dropdown menu.
ID	Enter a unique identifier for the configuration.
Configuration name	Enter a descriptive name for this configuration.
Description	Provide a brief description of this identity provider configuration.
Icon URL	Enter the URL of an icon to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

Field	Description
Domain	Enter the Windows domain name against which the NTLM server authenticates.

1. Click on the **Create** button.

Step 3: Logging in

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



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

Microsoft Entra ID authentication

Note

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

Microsoft Entra ID (formerly Azure AD) is a cloud-based identity and access management solution. It secures user authentication, enforces access policies, and enables single sign-on (SSO) for CloudBeaver.

Info

For details, see the [Microsoft Entra ID documentation](#).

Prerequisites

Make sure you have:

- An active Azure account with the appropriate permissions.
- Access to the Microsoft Entra admin center.

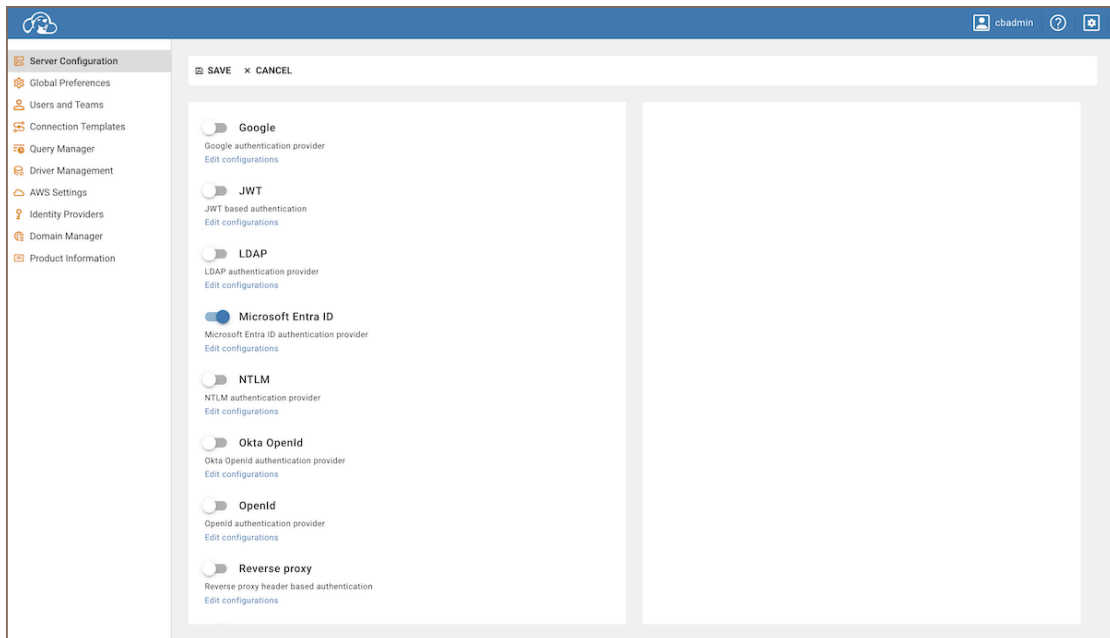
Info

For more details on permissions, see [Azure permissions](#).

Configuration steps

Enable Entra ID Authentication

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



Tip

For more information on Server Configuration, see [Server configuration administration](#)

Microsoft Entra ID configuration

To enable authorization with the Microsoft platform, you need a registered application in Azure. If one doesn't exist, create and configure it as follows:

1. **Register an application** Create a new enterprise application in Microsoft Entra by following the steps in the [official Microsoft documentation](#).
2. **Configure application secrets** CloudBeaver uses the OpenID Connect protocol for authorization with Microsoft Entra ID. To enable this, configure application secrets. Detailed instructions are available in the [official Microsoft documentation](#).

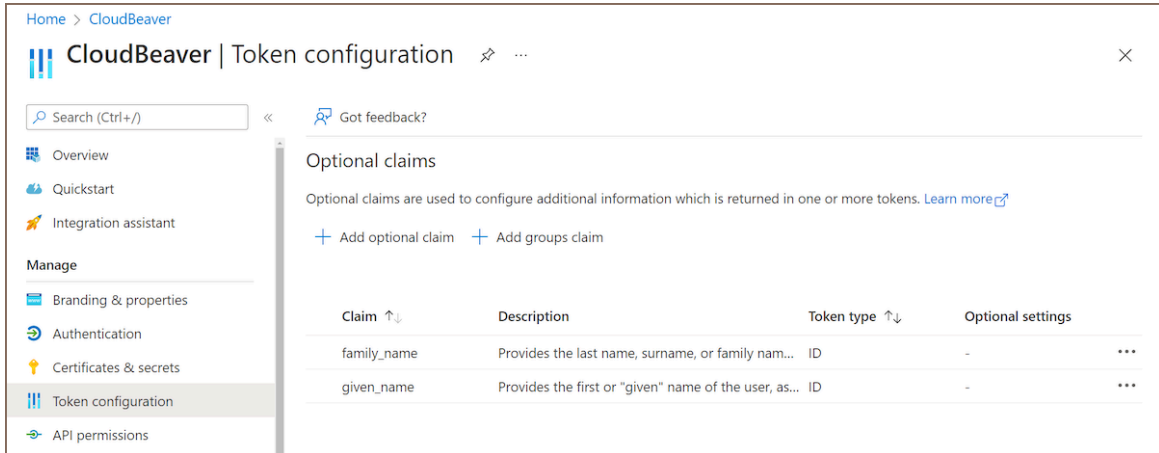
Important

Record the value of the client secret immediately after creating it. It can only be viewed once. If you miss this step, you'll need to create a new secret.

Optional configuration

To enhance functionality, CloudBeaver can read and display the user's first and last name from the OpenID token. To enable this, add the **family_name** and **given_name** fields to the response token.

Follow the steps in the [official Microsoft documentation](#) to configure optional claims.



The screenshot shows the CloudBeaver 'Token configuration' page. On the left is a sidebar with navigation links: Overview, Quickstart, Integration assistant, Manage (Branding & properties, Authentication, Certificates & secrets, Token configuration, API permissions). The main content area is titled 'Optional claims' and includes a search bar, a 'Got feedback?' link, and a description: 'Optional claims are used to configure additional information which is returned in one or more tokens. [Learn more](#)'. Below this are two buttons: '+ Add optional claim' and '+ Add groups claim'. A table lists the configured claims:

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

Tip

Adding these fields ensures a more personalized user experience by displaying the full name in the application.

Add an Identity Provider

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

The screenshot shows a web application interface for configuring identity providers. On the left is a sidebar with navigation links: Server Configuration, Global Preferences, Users and Teams, Connection Templates, Query Manager, Driver Management, Identity Providers (selected), Domain Manager, and Product Information. The main content area is titled 'CONFIGURATION CREATION' and includes a sub-header 'You can configure the list of enabled Identity Providers in the Server Configuration settings'. Below this is an 'OPTIONS' section with a 'CANCEL' button and a 'CREATE' button. The form fields are as follows:

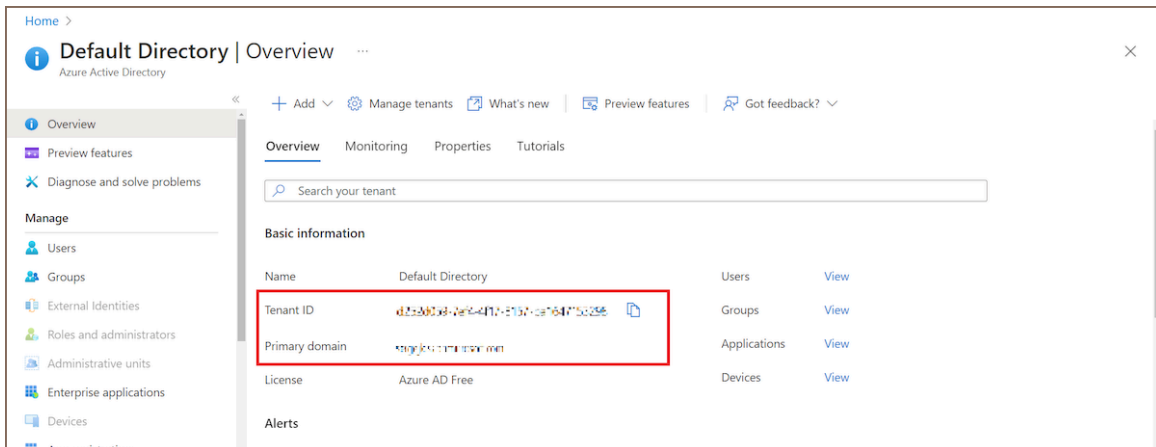
- Provider ***: A dropdown menu with 'Microsoft Entra ID' selected.
- ID ***: A text input field.
- Configuration name ***: A text input field.
- Description**: A text input field.
- Icon URL**: A text input field.
- Domain / Tenant ID ***: A text input field.
- Application (client) ID ***: A text input field.
- Secret Key ***: A text input field.
- Options**: A series of checkboxes:
 - ☐ Provide access to databases from Azure Cloud
 - ☐ Database authentication provider
 - ☐ Read ME-ID group information
 - ☐ Read user info
- Custom scopes**: A text input field.

Field	Description
Provider	Select the <code>Microsoft Entra ID</code> from the drop down menu.
Configuration name	Enter a descriptive name for this configuration.
Description	(Optional) Provide a brief description of the Microsoft Entra identity provider configuration.
Icon URL	(Optional) Enter the URL of an icon to represent this provider in the UI.
Disabled	(Optional) Leave unchecked to enable this identity provider.
Domain / Tenant ID	Enter the Microsoft Entra tenant ID. You can find it in the Azure portal .
Application (client) ID	Enter the application (client) ID of the registered application in Microsoft Entra.

Field	Description
Secret Key	Enter the client secret generated for the application in Microsoft Entra.
Provide access to databases from Azure Cloud	(Optional) Enforces multi-factor authentication (MFA) if it is enabled in Microsoft Entra ID settings.
Database authentication provider	(Optional) Indicates that this configuration can be used as a method for database authorization.
Read ME-ID group information	(Optional) If enabled, retrieves Active Directory group information from Microsoft Entra ID.
Read user info	(Optional) Retrieves user profile data using the <code>userinfo</code> endpoint URL.
Custom scopes	(Optional) The custom scopes. Use with <code>;</code> delimiter.

Tip

Ensure you have configured the application with the required permissions in Microsoft Entra to retrieve user information and group details. For additional details, see the [Microsoft Entra ID documentation](#).



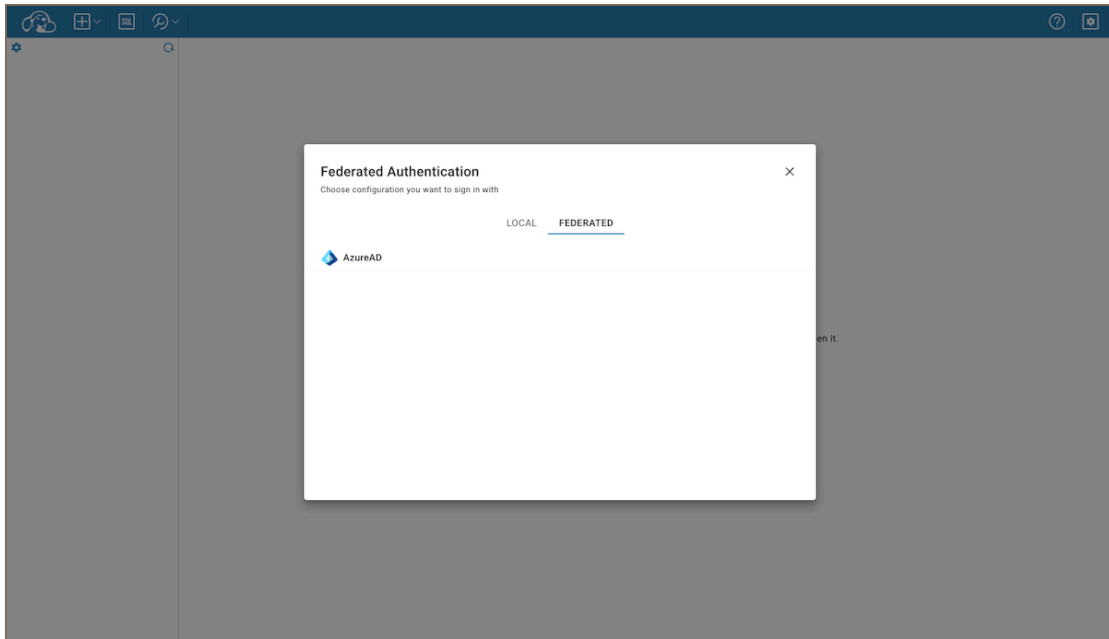
1. Click on the **Create** button.
2. **Copy Redirect link:**

- Enter the newly created identity provider.
- Copy the **Redirect** link.
- Add a Redirect link to the application in Microsoft Entra (select **Web** as platform) - [official Microsoft documentation](#)

The screenshot shows the 'EDIT 'AZUREAD'' configuration page. The left sidebar contains navigation links: Server Configuration, Global Preferences, Users and Teams, Connection Templates, Query Manager, Driver Management, AWS Settings, Identity Providers (selected), Domain Manager, and Product Information. The main content area is titled 'EDIT 'AZUREAD'' and has a 'CANCEL' and 'SAVE' button in the top right. Below the title is an 'OPTIONS' tab. The configuration is divided into several sections: 'Provider *' (Microsoft Entra ID), 'ID *' (AzureAD), 'Configuration name *' (AzureAD), 'Description' (empty text area), 'Icon URL' (empty text area), and 'Disabled' (checkbox). The 'OPENID' section includes 'Domain / Tenant ID *' (a long alphanumeric string), 'Application (client) ID *' (a long alphanumeric string), 'Secret Key *' (a masked field with an eye icon), and a list of checkboxes: 'Provide access to databases from Azure Cloud' (checked), 'Database authentication provider' (checked), 'Read ME-ID group information' (checked), and 'Read user info' (unchecked). The 'AWS' section includes 'Name of an AWS role claim' (empty text field). The 'LINKS' section includes 'Sign in' (a button with a copy icon), 'Sign out' (a button with a copy icon), and 'Redirect' (a button with a copy icon, highlighted with a red box). The right sidebar contains a '+ ADD' button and a list of application icons.

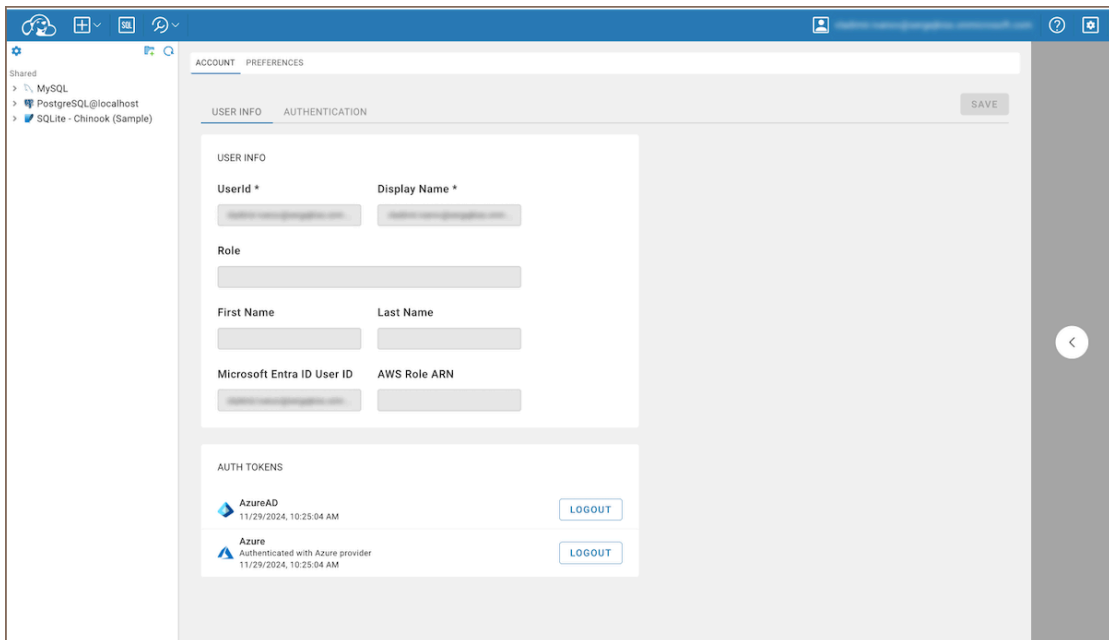
Login

1. Once the **Microsoft Entra ID** configuration is complete, navigate to the login screen to test authentication.
2. Select the Federated authentication method, labeled with the **Configuration name** you specified.



3. Verify the integration of Microsoft Entra ID

- a. Once logged in, click on your username in CloudBeaver and navigate to the **User Info** tab.
- b. Here, you should see tokens. Their presence indicates that the integration of Microsoft Entra ID has been successfully completed, and CloudBeaver has access to the necessary credentials.



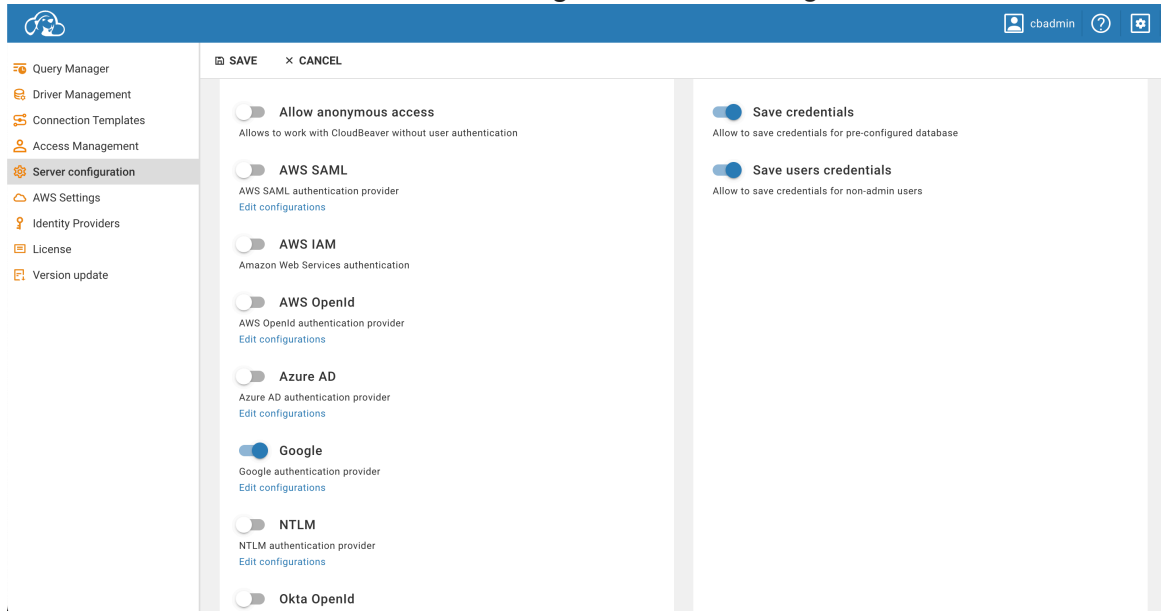
Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

Google configuration

Enabling Google authentication

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



Configuring identity provider

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

The screenshot shows the CloudBeaver web interface. On the left is a sidebar menu with options: Query Manager, Driver Management, Connection Templates, Access Management, Server configuration, AWS Settings, Identity Providers (selected), License, and Version update. The main content area is titled 'Identity Providers' and has a sub-header 'OPTIONS'. At the top of the main area are buttons: '+ ADD', 'REFRESH', and 'DELETE'. Below the sub-header, there are two columns of form fields. The left column contains: 'Provider *' (a dropdown menu with 'Google' selected), 'ID *' (an empty text field), 'Configuration name *' (an empty text field), 'Description' (a larger empty text area), 'Icon URL' (an empty text field), and a 'Disabled' checkbox. The right column is titled 'GOOGLE' and contains: 'Client ID' (an empty text field), 'Client Secret' (an empty text field), an 'Add custom scopes' checkbox, and 'Custom scopes' (a larger empty text area). At the top right of the main area are 'CANCEL' and 'CREATE' buttons.

Client ID - Client ID from your Google OAuth 2.0 application

Client Secret - Client Secret from your Google OAuth 2.0 application

Configuring Google OAuth 2.0 application

Open identity provider in CloudBeaver, copy `Redirect url`, and **set up redirect in Google application**

Testing Google authentication

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

Federated Authentication

Choose configuration you want to sign in with

LOCAL

FEDERATED



GCP configuration

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

Enable GCP Services

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

The screenshot shows the Cloudbeaver Administration interface. On the left is a sidebar menu with options: Query Manager, Driver Management, Connection Templates, Access Management, **Server configuration** (highlighted), AWS Settings, Identity Providers, License, and Version update. The main area is titled 'SAVE' and 'CANCEL'. It is divided into two panels. The left panel, 'SERVER INFORMATION', contains three fields: 'Server Name *' with the value 'Cloudbeaver EE Web Server', 'Server URL *' with the value 'https://cloudbeaver_url', and 'Session lifetime, min *' with the value '30'. The right panel, 'CONFIGURATION', contains several toggle switches: 'Enable private connections' (checked), 'Navigator simple view' (unchecked), 'AWS' (unchecked), 'Azure' (unchecked), 'GCP' (checked), and 'Enable Resource Manager' (checked). Below each toggle is a brief description of its function.

Add custom scopes to the Google Identity Provider

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

```
https://www.googleapis.com/auth/spanner.admin;https://www.googleapis.com/auth/bigquery;https://www.googleapis.com/auth/cloud-platform;https://www.googleapis.com/auth/devstorage.full_control
```

Scope	Description
https://www.googleapis.com/auth/spanner.admin	Allow to administrate Spanner databases
https://www.googleapis.com/auth/bigquery	Allow to view and manage your data in Google Bi gQuery

Scope	Description
https://www.googleapis.com/auth/cloud-platform	Gives access to the GCP and is needed to read a list of databases
https://www.googleapis.com/auth/devstorage.full_control	Allow to manage BigQuery data

CloudBeaver Identity Providers configuration page. The page shows a table of providers with columns: CONFIGURATION NAME, PROVIDER, DESCRIPTION, and DISABLED. The 'Google' provider is selected, and its configuration details are shown in a form. The form includes fields for Provider (Google), ID (google-openid), Configuration name (Google), Description, Icon URL, and a checkbox for Disabled. The 'Client ID' and 'Client Secret' fields are masked. The 'Add custom scopes' checkbox is checked, and a list of custom scopes is shown: https://www.googleapis.com/auth/spanner.admin, https://www.googleapis.com/auth/bigquery, https://www.googleapis.com/auth/cloud-platform, and https://www.googleapis.com/auth/devstorage.full_control. The 'LINKS' section shows 'Sign in', 'Sign out', and 'Redirect' links, all pointing to https://cloudbeaver_url/api/google-openid/go...

Testing

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

databases in it

Cloud Connection



Search elements...



DBeaverGCP



SQL



Spanner



BigQuery

CANCEL

CREATE

Access management

Users

Users

This article gives you an overview of how to create user in CloudBeaver.

Creating user

There are two types of users:

- **Local users:** Created by the Administrator.
- **AWS and Federated users:** These users are managed externally through AWS or federated identity providers and are authorized to access the system via Single Sign-On (SSO). ★

Info

For more information on Identity providers, see [Authentication methods](#).

Local users

To create a new local user, follow these steps:

1. As an administrator, go to **Settings -> Administration -> Users and Teams -> Teams**.
2. Click on the + **Create** button.
3. Fill in the necessary details in the provided fields.

The screenshot shows the 'USER CREATION' page in the AWS IAM console. The page is divided into three main sections: CREDENTIALS, USER STATUS, and PARAMETERS. The CREDENTIALS section includes fields for Username (admin_ee), User password (masked), and Repeat password (masked). The USER STATUS section includes a checkbox for 'Enabled' (checked) and a dropdown for 'USER TEAM' (admin (Admin)). The PARAMETERS section includes fields for First Name (Liam), Last Name, Microsoft Entra ID User ID, and AWS Role ARN (DB3rdParty). The page also has a search bar at the top and a sidebar with navigation links.

Field Name	Description	Additional Info
Username	Enter the desired username for the account.	
User password	Set a password for the account.	The user can change their password after initial setup.
Repeat password	Re-enter the password for verification.	
User Status	Toggle to enable or disable the user.	Default status is Enabled.
User Team	(Optional) Assign the user to one or more teams.	A team defines the permissions a user has within the system. For more information on teams, see Teams .
First Name	(Optional) Provide the user's first name.	
Last Name	(Optional) Provide the user's	

Field Name	Description	Additional Info
	last name.	
AWS Role ARN ★	(Optional) Enter the AWS Role ARN.	For more information about AWS roles, see AWS Settings .
Microsoft Entra ID User ID ★	(Optional) Enter the Microsoft Entra ID.	For more information, see Microsoft Entra ID authentication .

- To complete the process, click on the **Create** button. Once created, the user can be authenticated using local authentication methods. The user's permissions will be determined by their assigned profile.

Global permissions

Every non-administrator user is also subject to [Global permissions](#). These permissions define the baseline rights for all users, such as whether they can copy, edit, import, or export data.

Global permissions are configured by administrators under **Settings -> Administration -> Users and Teams -> Global Permissions**.

Connection access

If necessary, you can provide the user with connection access. This setting can be found and adjusted within the **Connection Access** tab. See the additional information on [Connection management](#).

USER CREATION

INFO

CONNECTIONS ACCESS

CANCEL

CREATE

	CONNECTION NAME	GRANTED BY
☐	 MySQL	
☒	 MySQL@localhost	
☒	 PostgreSQL@localhost	
☐	 SQLite - Chinook (Sample)	

Important

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

Enabling Federated authentication for local users

To allow a local user to authenticate through AWS/Federated auth methods, the local user's username must match the user's email address, which will connect via SSO (Single Sign-On). This alignment is necessary for the federated authentication process to succeed.

Note

This step is crucial during user creation as the username cannot be changed later.

AWS and Federated users

When a user logs in using AWS or Federated authentication for the first time, CloudBeaver automatically creates a user profile assigned to the default team. Administrators can later change this team assignment as necessary.

Note

Administrators cannot create AWS or Federated users directly in the application. CloudBeaver only works with existing AWS and Federated users.

Managing users in CloudBeaver AWS Edition

CloudBeaver AWS Edition is designed to support only **AWS and Federated users**, excluding local user access. Therefore, it is not possible to create local users within this environment. Users must be imported into the system.

Info

For more information, refer to [Administration Users Provisioning](#).

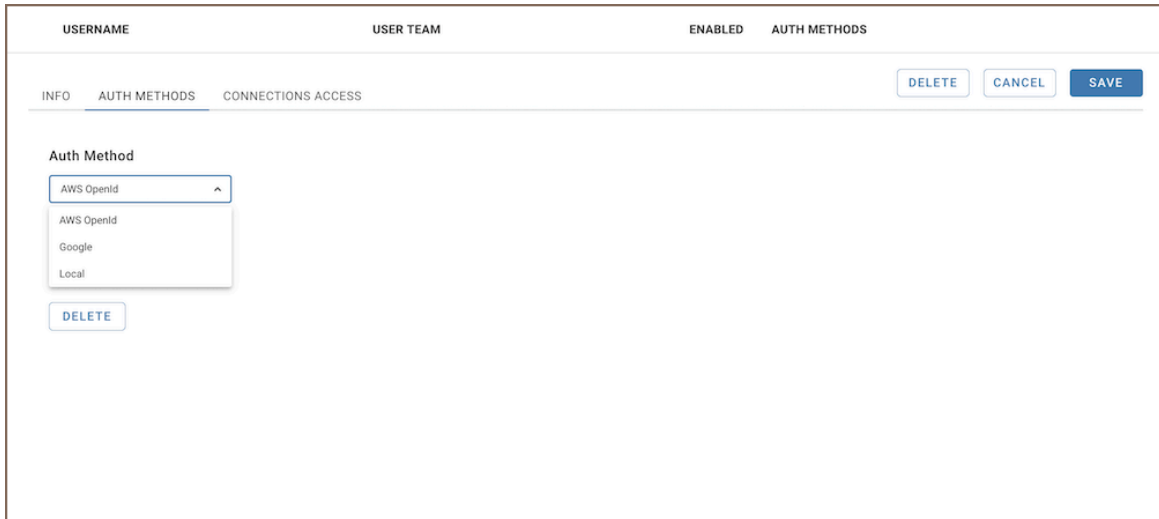
Editing user

The process of editing a user is similar to **creating** one, except you need to access an existing user.

When editing an existing user, you also have additional options:

Auth methods

In the **Auth Methods** tab, administrators can see and remove the authentication methods associated with a user.



The screenshot shows a user profile interface with the following elements:

- Header:** USERNAME, USER TEAM, ENABLED, AUTH METHODS.
- Navigation:** INFO, AUTH METHODS (selected), CONNECTIONS ACCESS.
- Buttons:** DELETE, CANCEL, SAVE.
- Auth Method Section:**
 - Auth Method:** A dropdown menu currently showing "AWS OpenId".
 - Options:** A list of available authentication methods: AWS OpenId, Google, and Local.
 - DELETE:** A button to remove the selected authentication method.

You can remove an existing authentication method:

1. Select the desired method from the dropdown menu.
2. Click the **DELETE** button.

Tip

If you need to restore local authentication for the user, navigate to the **Info** tab and assign a new password to the user. This action will re-enable local authentication.

Enabled and disabled users

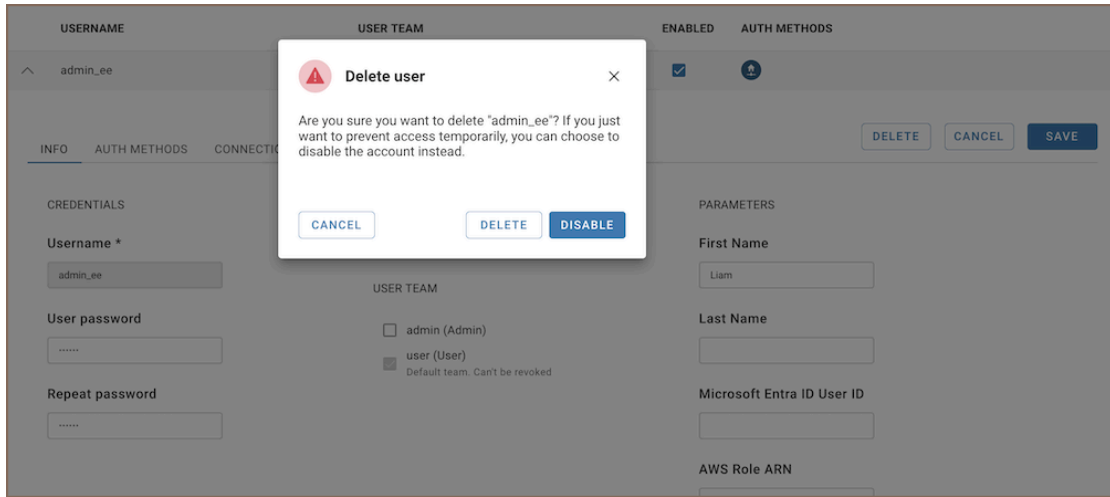
Admins can temporarily disable a user to block access without deleting the account. To re-enable a user, select the **Enabled** checkbox in the user profile.

Disabled users are hidden by default. To view them, set **User status** to **DISABLED** or **ALL** in the filter above the user list.

Deleting a user

If you need to permanently remove a user from the system, you can do so through the **Delete** user option. When you attempt to delete a user, a confirmation dialog will appear to ensure that this action is intentional.

1. To delete a user, select the **Delete** option.
2. Follow the prompts in the dialog to confirm the deletion.



Tip

If you prefer to keep the user but prevent their access, consider using the **Disable** option in the dialog. Alternatively, you can disable a user by selecting the checkbox in the **Info** section of the user profile.

Provisioning users

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

You can import users from a third-party system to preconfigure access.

This feature supports the following identity providers:

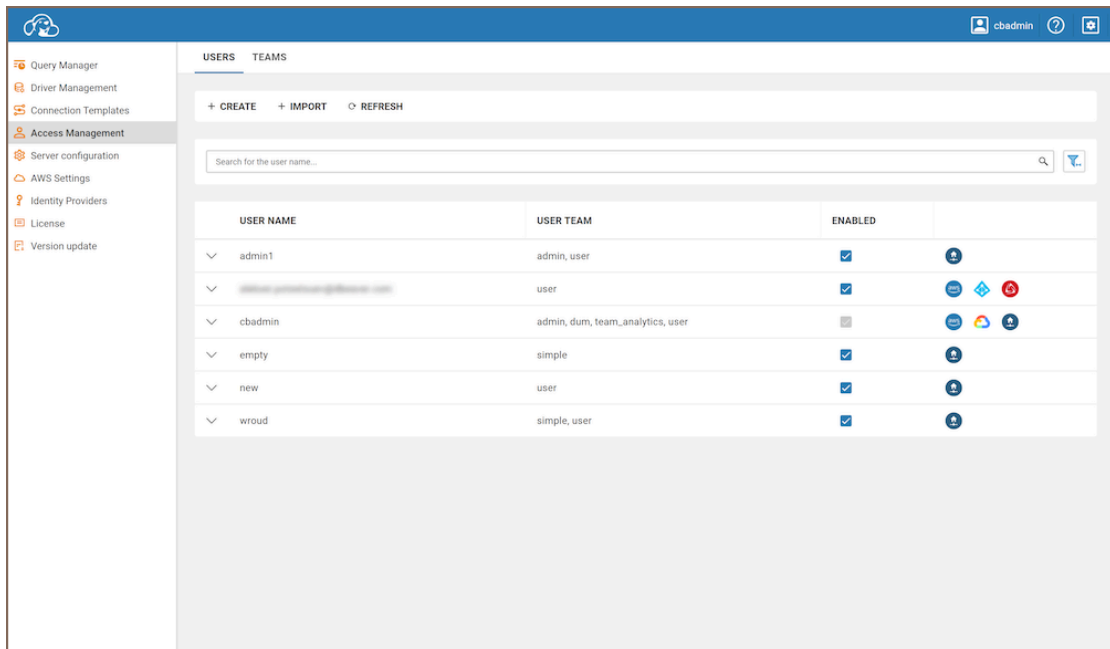
- **Microsoft AzureAD**
- **Okta OpenID**
- **Google**
- **LDAP**

Tip

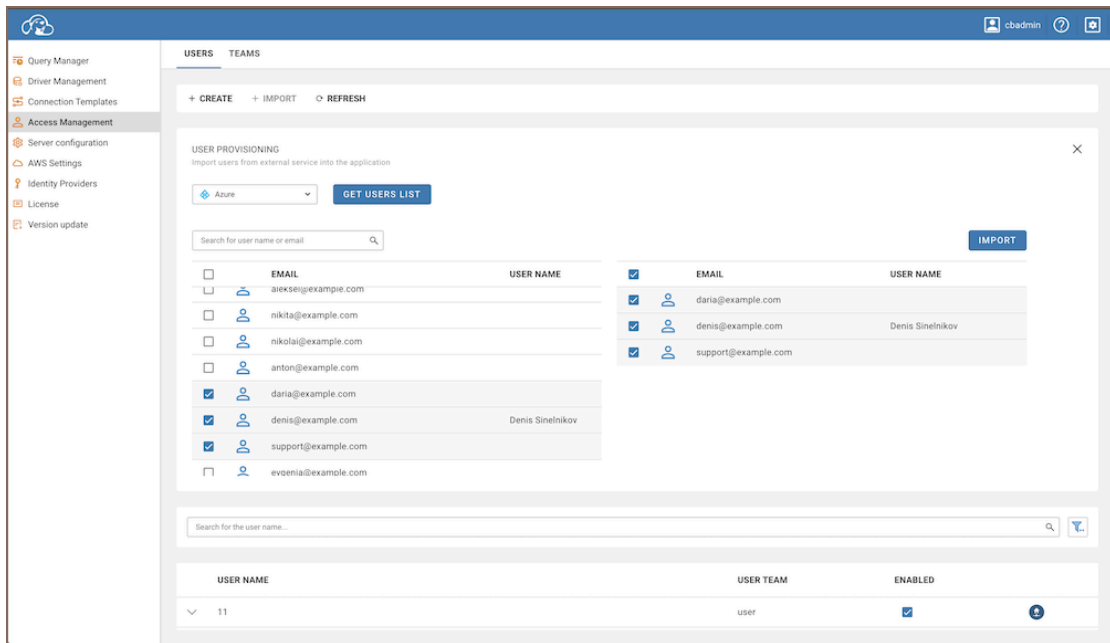
Make sure at least one provider is configured before importing users.

Import users

1. Go to **Settings -> Administration -> Users and Teams**
2. Click **+ Import**. This opens the import form.



3. In the form, choose a provisioning provider.



4. Click **Get Users List**. If not logged in, you'll be prompted to authenticate with the selected provider.
5. After loading, select the users you want to import. The selected users will appear in the table on the right.

6. Click **Import**. The users will be added to the system and shown in the list.

Info

For more details on user management, see [Users](#).

Password policy

This document provides guidelines for configuring the password policy in CloudBeaver. Administrators can define rules for user passwords to enhance security and comply with organizational requirements. These settings, applied globally, ensure passwords meet the specified standards during creation and update processes.

Info

For more details on configuration, see the [CloudBeaver server configuration](#).

Example configuration

The Password Policy configuration is located under the `server.sm` section of the main server configuration file.

```
sm: {
  enableBruteForceProtection:
    "${CLOUDBEAVER_BRUTE_FORCE_PROTECTION_ENABLED:true}",
  expiredAuthAttemptInfoTtl:
    "${CLOUDBEAVER_EXPIRED_AUTH_ATTEMPT_INFO_TTL:60}",
  maxFailedLogin: "${CLOUDBEAVER_MAX_FAILED_LOGINS:10}",
  minimumLoginTimeout: "${CLOUDBEAVER_MINIMUM_LOGIN_TIMEOUT:1}",
  blockLoginPeriod: "${CLOUDBEAVER_BLOCK_PERIOD:300}",
  connectionBruteForce: {
    enabled: "${CLOUDBEAVER_CONNECTION_BF_PROTECTION_ENABLED:false}",
    maxFailedConnectAttempts:
      "${CLOUDBEAVER_CONNECTION_BF_PROTECTION_MAX_ATTEMPTS:5}",
    errorAttemptsPeriodInMinutes:
      "${CLOUDBEAVER_CONNECTION_BF_PROTECTION_ERROR_ATTEMPTS_PERIOD:1440}",
    blockTimeInMinutes: "${CLOUDBEAVER_CONNECTION_BF_BLOCK_PERIOD:60}"
  },
  passwordPolicy: {
    minLength: "${CLOUDBEAVER_POLICY_MIN_LENGTH:8}",
    requireMixedCase: "${CLOUDBEAVER_POLICY_REQUIRE_MIXED_CASE:true}",
    minNumberCount: "${CLOUDBEAVER_POLICY_MIN_NUMBER_COUNT:1}",
    minSymbolCount: "${CLOUDBEAVER_POLICY_MIN_SYMBOL_COUNT:0}"
  }
}
```


Configuration properties

Name	Environment variable	Default value	Units	Description
<code>minLength</code>	<code>CLOUDBEAV R_POLICY_MI N_LENGTH</code>	<code>8</code>	characters	Specifies the minimum length requirement for user passwords.
<code>requireMixedCase</code>	<code>CLOUDBEAV R_POLICY_REQUIRE_MIXED_CASE</code>	<code>true</code>	boolean	Enforces the use of both uppercase and lowercase letters in passwords.
<code>minNumberCount</code>	<code>CLOUDBEAV R_POLICY_MIN_NUMBER_COUNT</code>	<code>1</code>	characters	Sets the minimum number of numeric characters required in passwords.
<code>minSymbolCount</code>	<code>CLOUDBEAV R_POLICY_MIN_SYMBOL_COUNT</code>	<code>0</code>	characters	Defines the minimum number of special symbols required in passwords.
<code>enableBruteForceProtection</code>	<code>CLOUDBEAV R_BRUTE_FORCE_PROTECTION_ENABLED</code>	<code>true</code>	boolean	Enables brute force protection for login attempts.
<code>expiredAuthAttemptInfoTtl</code>	<code>CLOUDBEAV R_EXPIRED_AUTH_ATTEMPT_INFO_TTL</code>	<code>60</code>	minutes	Time-to-live (TTL) for expired authentication attempt information.
<code>maxFailedLogin</code>	<code>CLOUDBEAV R_MAX_FAILURE</code>	<code>10</code>	attempts	Maximum number of allowed failures.

Name	Environment variable	Default value	Units	Description
	D_LOGINS			led login attempts before blocking access.
minimumLoginTimeout	CLOUDBEAV R_MINIMUM_LO GIN_TIMEOUT	1	minutes	Minimum timeo ut duration befo re the next login attempt.
blockLoginPeriod	CLOUDBEAV R_BLOCK_PERI OD	300	seconds	Period for whic h a login is block ed after maximu m failed attempt s.
connectionBruteForce	CLOUDBEAV R_CONNECTIO N_BF_PROTECT ION_ENABLED	false	boolean	Enables brute fo rce protection f or database con nection attempt s.
maxFailedConnectAttempts	CLOUDBEAV R_CONNECTIO N_BF_PROTECT ION_MAX_ATT EMPTS	5	attempts	Number of faile d database conn ection attempts allowed before t he connection is temporarily blo cked. Applies to non-admin user s only. After rea ching the limit, t he connection is blocked for a ti meout period b efore it can be r etried.
errorAttemptsPeriodInMinutes	CLOUDBEAV R_CONNECTIO N_BF_PROTECT ION_ERROR_AT	1440	minutes	Time window fo r checking failed connection atte mpts.

Name	Environment variable	Default value	Units	Description
	TEMPTS_PERIOD			
blockTimeInMinutes	CLOUDBEAVER_CONNECTION_BLOCK_PERIOD	60	minutes	Block duration after exceeding the allowed number of failed connection attempts.

Applying password policy

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

The screenshot shows the CloudBeaver web interface for user management. The left sidebar contains navigation links: Query Manager, Driver Management, Connection Templates, Access Management (selected), Server configuration, Identity Providers, AI Settings, License, and Version update. The main content area is titled 'USERS' and 'TEAMS'. Below this are buttons for '+ CREATE', '+ IMPORT', and 'REFRESH'. A search bar is present with the placeholder 'Search for the user name...'. The 'User Creation' section is active, showing tabs for 'INFO', 'CONNECTIONS', and 'ACCESS'. The 'INFO' tab is selected, displaying three columns: CREDENTIALS, USER STATUS, and PARAMETERS. In the CREDENTIALS column, the 'Username' field contains 'user' and the 'User password' field contains 'password'. A warning message is displayed below the password field: 'Password must contain both upper and lower case letters'. The USER STATUS column shows 'Enabled' checked and 'USER TEAM' with 'admin' unchecked and 'user' checked. The PARAMETERS column has 'First Name' and 'Last Name' fields. At the top right of the form are 'CANCEL' and 'CREATE' buttons.

Brute force protection policy

CloudBeaver supports two types of brute force protection:

- **Login brute force protection** - limits failed login attempts to the CloudBeaver
- **Connection brute force protection** ★ - limits failed database connection attempts through CloudBeaver

These policies help prevent unauthorized access by temporarily blocking logins or database connections after repeated failures.

When the policy is enabled and the failed attempts threshold is reached, CloudBeaver:

- For **login brute force** - the user is temporarily blocked and must wait before trying again
- For **connection brute force** - the database connection is blocked for a timeout period (applies to non-admin users only)

The admin panel logs when the block happened and the reason.

Info

To allow a user to connect again, wait for the timeout to expire or adjust the limits in the configuration.

Login brute force protection

This policy controls how CloudBeaver reacts to multiple failed login attempts.

When enabled, it tracks incorrect passwords during user authentication. After a set number of failures, the user is temporarily blocked and must wait before trying again.

Connection brute force protection

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

This policy tracks failed attempts to connect to a database through CloudBeaver.

If a non-admin user reaches the allowed number of failed connection attempts (for example, when connecting to Redshift with a wrong password), CloudBeaver blocks the **connection** for the configured timeout.

During the block period, the user cannot retry the same connection. After the timeout expires, they can attempt again.

Supported databases:

- Amazon Redshift
- Db2
- Microsoft SQL Server
- MySQL
- Oracle Database
- PostgreSQL

User credentials storage

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`.

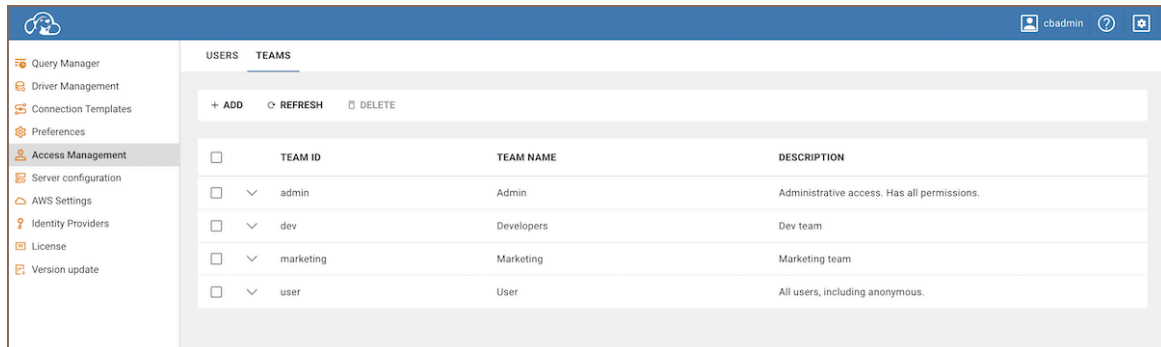
Potentially, if an intruder/malware software will get access to CloudBeaver server filesystem, then it may get access to all stored user credentials. To increase security it is recommended to configure the server to keep workspace on a shared encrypted network folder (e.g. S3, see [S3 Server-side encryption](#)).

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

Teams

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

Teams can also be linked to external identity providers, enabling automatic user assignment based on group information received during login.



USERS TEAMS			
+ ADD ◁ REFRESH □ DELETE			
☐	TEAM ID	TEAM NAME	DESCRIPTION
☐	admin	Admin	Administrative access. Has all permissions.
☐	dev	Developers	Dev team
☐	marketing	Marketing	Marketing team
☐	user	User	All users, including anonymous.

Team creation

To create a new team, follow these steps:

1. Navigate to the **Settings -> Administration -> Users and Teams -> Teams**.
2. Click on the **+ Add** button.
3. Fill in the necessary details in the provided fields.

- Server Configuration
- Settings
- Users and Teams**
- Query Manager
- Driver Management
- AWS Settings
- Identity Providers
- Secret Management
- AI Settings
- Domain Manager
- Product Information

USERS **TEAMS** GLOBAL PERMISSIONS

+ CREATE ◁ REFRESH ▢ DELETE

TEAM CREATION

OPTIONS **USERS** PERMISSIONS CONNECTIONS

Team ID *

Team Name *

PERMISSIONS

☐ Admin Full Access
Provides access to the CloudBeaver configuration

PARAMETERS

AWS Role ARN

LDAP Group name

Microsoft Entra ID Group ID

OKTA Group ID

OpenID Group ID

Description

CANCEL

CREATE

Field Name	Description	Additional Info
Team ID	A unique identifier for the team.	
Team Name	The name of the team.	
Description	A brief description of the team and its purpose.	
Permissions	Access level for team members	Admin Full Access grants full access to CloudBeaver settings
Parameters ★	For group mapping and other metadata, e.g. AWS roles	See Integration with identity providers

Note

The **Parameters** section is available only after you configure at least one **identity provider**.

Predefined Team types

CloudBeaver includes two predefined Team types:

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

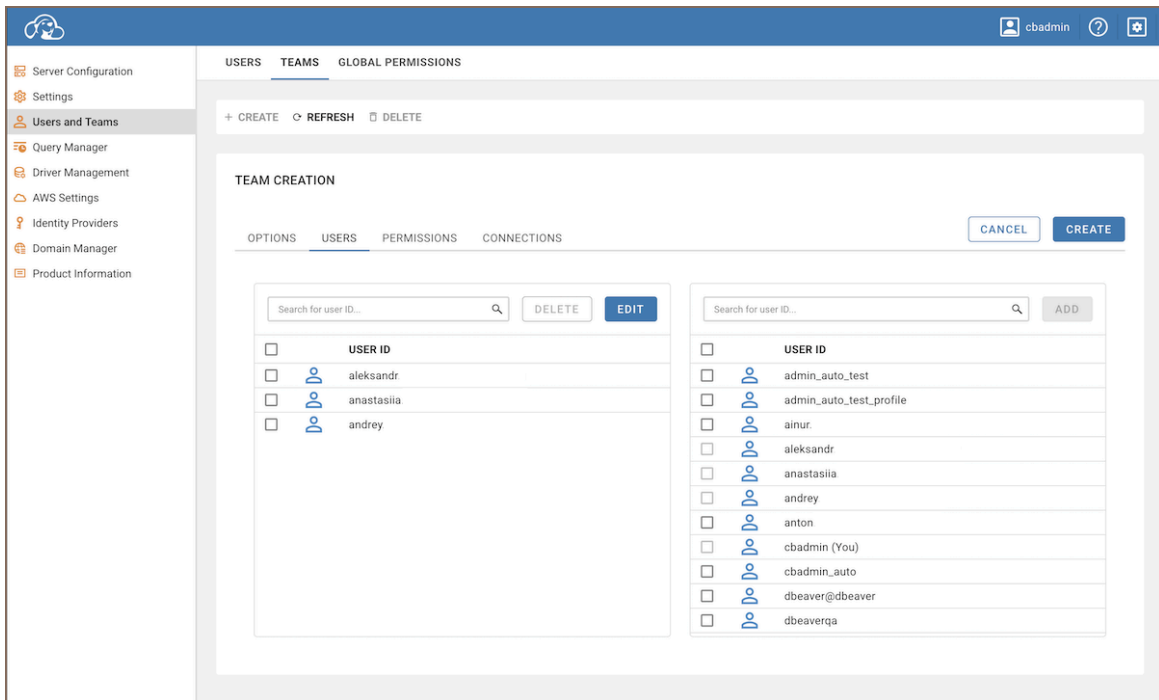
Tip

Administrators can configure the predefined types in the Server Settings. For more information, see [Initial data configuration](#).

User management

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

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



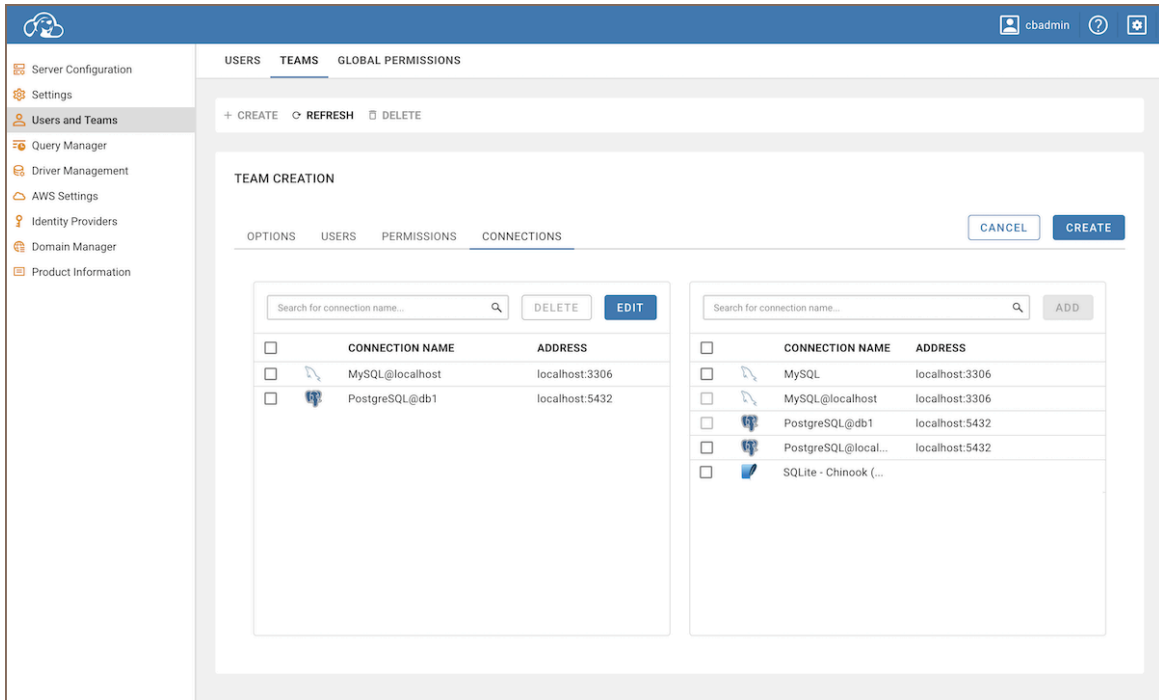
Tip

One user can be a member of a multiple Teams.

Connection management

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

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



Permission management

Permissions control what actions users can perform in the **Data Editor** and **SQL Editor**.

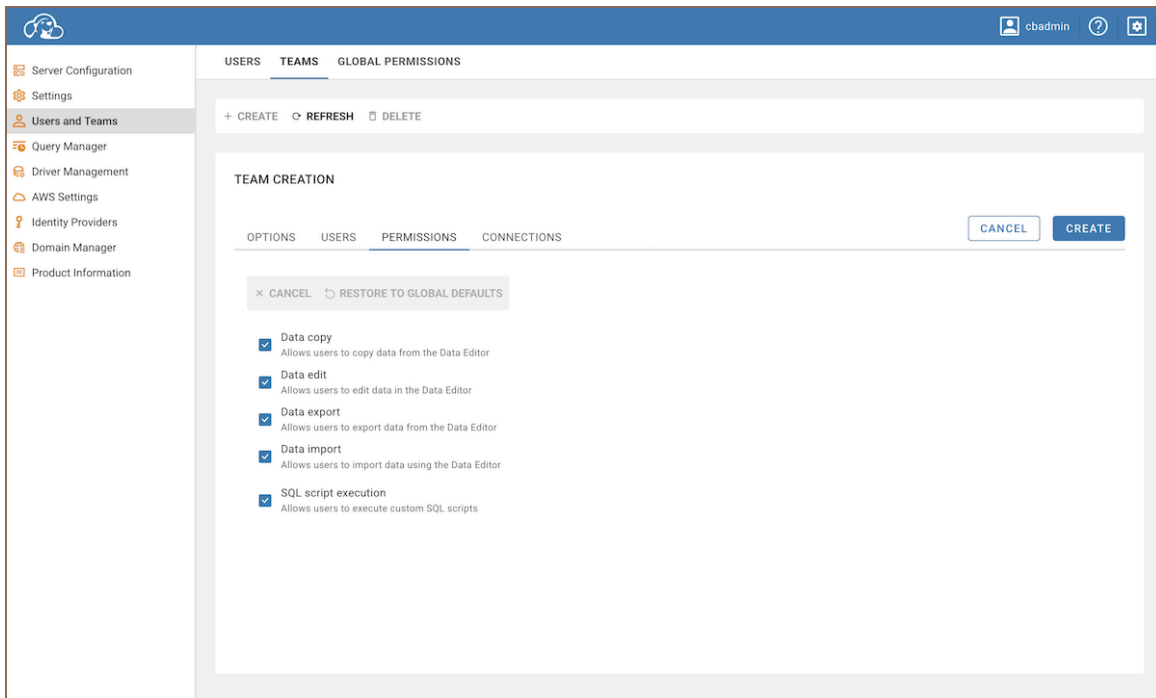
There are two levels of permissions:

- **Global permissions** - define the default rights for all teams and users, except administrators.
- **Team permissions** - allow you to customize rights for each team individually. Team permissions override the global defaults.

In the **Permissions** tab of a team, administrators can configure what actions team members are allowed to perform:

Permission	Description	Related articles
Data copy	Allows users to copy data from the Data Editor	Data Editor
Data edit	Allows users to edit data in the Data Editor	Data Editor
Data export	Allows users to export data fro	Data export

Permission	Description	Related articles
	m the Data Editor	
Data import	Allows users to import data using the Data Editor	Data import
SQL script execution	Allows users to execute custom SQL scripts	SQL Editor



Team permissions have priority over global permissions. Each team can have its own set of permissions, managed independently.

When you change the global defaults, these updates are applied automatically to all teams that are still using the defaults. Teams with custom permissions set by an administrator are not affected.

Auto mapping of users from identity providers

You can link Teams to external identity providers. This allows CloudBeaver to assign users to Teams automatically based on group or role information provided at login.

Team membership is managed entirely by your identity provider. There's no need to manually assign users - CloudBeaver reads group claims from the authentication token and maps users to Teams based on matching parameters.

Configuration steps

1. In the Identity Providers settings, specify the attribute or value used to identify user groups or roles.

Important

CloudBeaver reads this exact attribute during login. If it's missing or misconfigured, Team assignment won't work.

CONFIGURATION CREATION

You can configure the list of enabled Identity Providers in the Server Configuration settings

CANCEL

CREATE

OPTIONS

Provider *	OPENID	AWS	LINKS
AWS Openid	Client ID *	Role ARN	Sign in
ID *	example123kdjkj		{origin}/api/aws-openid/okta-aws/sig
okta-aws	Client Secret	Name of an AWS role claim	Sign out
Configuration Name *		groups	{origin}/api/aws-openid/okta-aws/log
Okta AWS Integration	IDP auth endpoint URL *		Redirect
Description	https://dev-example.okta.com/oauth2/v1...		{origin}/api/aws-openid/okta-aws/call
	IDP token endpoint URL *		
Icon URL	https://dev-example.okta.com/oauth2/v1...		
	IDP userinfo endpoint URL		

2. In the Team's **Parameters** settings, enter the exact group ID you expect from the provider. Use your provider's tools to inspect the token and find the actual group IDs being sent. Some providers return non-human-readable IDs (e.g., UUIDs).

USERS

TEAMS

+ CREATE

REFRESH

DELETE

TEAM CREATION

OPTIONS

USERS

CONNECTIONS

CANCEL

CREATE

Team ID *

dev

Team Name *

Developers

PERMISSIONS

☐ Admin Full Access
Provides access to the CloudBeaver configuration

☐ Admin Full Access
Provides access to the CloudBeaver configuration

PARAMETERS

AWS Role ARN

siam:123456789012:role/DBRole

Description

Provider parameter	Expected value	Example value	Related articles
Microsoft Entra ID	UUID of group	73ab4df0-b4b4-4449-a2cd-1874923871049278	Microsoft Entra ID
AWS Role ARN	ARN of assumed role	arn:aws:iam::123456789012:role/DBRole	AWS OpenID, via Okta
SAML Group ID	External group name or ID	Admins or b274cd83-331a-4e4e-9b8f-92f123456789	SAML configuration
Okta Group ID	Group name or internal ID	00g631wg11qHdgDfZ0x7	Okta OpenID
LDAP Group name	Group common name	cn=Developers	LDAP
OpenID Group ID	Group name or internal group ID	dev-team or 8e4b9ff4-11f3-4f17-902e-e9e5c1c2043a	OpenID configuration

- When a user logs in through the identity provider (e.g., Okta, Entra ID), CloudBeaver receives an authentication token. This token includes user details, such as group or role claims. If any value

in the configured attribute (like `groups`) matches a Team **Parameter**, the user is automatically added to that Team and inherits its permissions.

Important

- Attribute name (e.g., `groups`) must be configured in the identity provider settings
- The value in the Team must match the token value exactly
- Group IDs may not be human-readable - use display names with caution
- Any string format is accepted
- Some providers (e.g., Azure) may limit the number of groups sent in the token

Updating Team memberships

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

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

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

Troubleshooting mismatches

If users aren't assigned to a Team after login, check the following:

- **Incorrect group ID:** The value in Team parameters must exactly match the one sent in the token. Display names usually won't work—use raw IDs (e.g., UUID, ARN).
- **Wrong attribute name:** Make sure the identity provider is configured to include the correct attribute (e.g., `groups` , `roles`) in the token.
- **Missing group claim:** Some providers don't send group claims by default. You may need to explicitly add them in the provider settings.
- **Group limit:** Azure AD and some other providers may limit the number of groups included in the token (e.g., 150). Users beyond the limit won't be mapped.
- **Session not refreshed:** Changes in group membership take effect only after the user logs out and logs in again (or after session timeout).

Admin password recovery

If you've forgotten the admin password for CloudBeaver, follow these steps to reset it:

Locate the configuration file

1. Access the **workspace directory** used by the Docker container

Tip

For more information on locating the workspace directory, see [Workspace location](#).

2. Look for the file named `.cloudbeaver.runtime.conf` in the `.data` subdirectory

Edit the configuration file

1. Open `.cloudbeaver.runtime.conf` in a text editor
2. Locate the line that begins with `"serverName": ...`

Info

For details on configuration file, see [Server-configuration](#).

3. Delete this line completely
4. Save and close the file

Restart CloudBeaver

Run these commands:

```
docker-compose stop
docker-compose start
```

Tip

For more details on Docker Compose commands, see the [Docker Compose CLI reference](#).

Reset the admin password

1. Access the CloudBeaver setup wizard in your browser at `http://<server-ip>:8978`
2. The initial server setup wizard will open, allowing you to configure the admin password along with other settings

Info

Your license, configured connections, and other settings will remain unchanged.

3. Once complete, the admin password is reset and your server will be ready for use

Initial data configuration

This document provides guidelines for configuring administrator credentials and predefined teams in the CloudBeaver, including their IDs, names, descriptions, and assigned permissions. The configuration is applied during the first server initialization. If you need to add new teams later, create them through the [UI](#).

Tip
By default, the file is located at `{WORKSPACE}/conf/initial-data.conf` (see [Workspace location](#) for details).
For more details on configuration, see the [CloudBeaver server configuration](#).

Example configuration

```
{
  adminName: "cbadmin",
  adminPassword: "cbadmin_password",
  teams: [
    {
      subjectId: "admin",
      teamName: "Admin",
      description: "Administrative access. Has total and full
authority.",
      permissions: ["admin"]
    },
    {
      subjectId: "user",
      teamName: "User",
      description: "Standard user",
      permissions: []
    }
  ]
}
```

Configuration properties

Name	Description
<code>subjectId</code>	ID for the team.

Name	Description
<code>teamName</code>	Name for the team.
<code>description</code>	Team description.
<code>adminName</code>	Admin name.
<code>adminPassword</code>	Admin password.
<code>permissions</code>	Permissions for the team. Only two roles are available: administrators and other users. To assign admin access, specify <code>"admin"</code> .

Note

By default, admin configurations are not created automatically, but you can create them manually if needed.

Global Permissions

Note

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

Global permissions define the default rights applied to all non-administrator users. They control which actions are allowed in the Data viewer, such as copying, editing, or importing data.

Administrators can manage global permissions in **Settings -> Administration -> Users and Teams -> Global Permissions**.

Global permissions apply to all teams and users (except administrators) and act as the baseline. If a team has custom permissions set by an administrator, those team permissions override the global defaults. Only teams that use the defaults will be affected when global permissions are changed.

Tip

To learn how to configure permissions for specific teams, see [Permission management](#).

Available permissions

The following actions can be enabled or disabled globally:

Permission	Description
Data copy	Allows users to copy data from the Data Editor.
Data edit	Allows users to edit data in the Data Editor.
Data export	Allows users to export data from the Data Editor.
Data import	Allows users to import data into the Data Editor.
SQL script execution	Allows users to execute custom SQL scripts.

Restoring defaults

At any time, administrators can click **Restore defaults** to reset global permissions to their original values.

Proxy configuration

CloudBeaver includes a built-in reverse proxy for secure access and load balancing.

This article explains how to set up a proxy in the CloudBeaver Compose setup.

- CloudBeaver AWS uses the default pre-configured proxy setup as the Enterprise Edition.
- Community Edition users need to [configure the proxy manually](#).

For proxy configuration in other deployment types, see [README](#) in the [cloudbeaver-deploy](#) repository.

Configure proxy server

Note

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

Starting in version `25.1`, CloudBeaver supports two proxy types: **Nginx** and **HAProxy**. Set the proxy type in your `.env` file:

```
PROXY_TYPE=nginx # Options: nginx, haproxy
```

If you previously used Nginx and want to switch to HAProxy, you can do it without losing configuration or SSL certificates - shared Docker volumes handle it.

Info

For details on environment configuration, see [CloudBeaver Enterprise deployment from Docker Compose](#)

Proxy ports

When using Docker Compose with `network_mode: host`, set ports in your `.env` file:

```
LISTEN_PORT_HTTP=80  
LISTEN_PORT_HTTPS=443
```

These control which ports the proxy listens to inside the container.

WebSockets

For real-time updates and SSO, CloudBeaver requires sockets. For configuration details and troubleshooting, see [WebSockets](#).

Important

Default **deployments** are preconfigured. The socket uses the same host and port, and switches to `wss` when the server uses HTTPS.

Extra step for Nginx

If you use Nginx and set a custom `COMPOSE_PROJECT_NAME` in your `.env`, pass it explicitly in the container's `environment` block:

```
environment:
  - COMPOSE_PROJECT_NAME=${COMPOSE_PROJECT_NAME}
```

Note

This isn't needed for HAProxy - it resolves service names via Docker DNS.

Force HTTPS mode

Force HTTPS enforces secure connections across the entire application. When this option is enabled, the server automatically redirects all HTTP requests to HTTPS and sets cookies with the **Secure** flag, so they're never sent over unencrypted channels.

Why enable force HTTPS

- **Protect sensitive data** - ensures that credentials, tokens, and query results are never transmitted in plain text
- **Stay compliant** - many organizations, regulations, and audits require HTTPS-only communication
- **Remove inconsistencies** - even if users try to connect via `http://`, they're redirected to `https://`
- **Harden authentication** - secure cookies prevent hijacking over unencrypted channels

Tip

Browsers like Chrome may auto-upgrade some HTTP connections, but this isn't guaranteed for all users or environments. Force HTTPS ensures the same security policy everywhere.

Turn on force HTTPS

1. You can enable Force HTTPS in one of the following ways:
 - during initial setup, in the **Easy Config** wizard when starting the application
 - later, in **Settings -> Administration -> Server configuration**
 - or, by setting the environment variable:

```
CLOUDBEAVER_FORCE_HTTPS=true
```

Info

For details, see [Server configuration](#).

2. Confirm that your proxy already has valid SSL certificates (see [SSL certificates](#))

Once enabled, all requests over port 80 will be redirected to 443, and session cookies will be marked secure.

Warning

If SSL isn't configured in the proxy, enabling force HTTPS will make the application unreachable.

Multiple server URLs

You can allow access from several domains instead of a single entry point. This improves SSO, migration, and high-availability scenarios.

For details, see [Multiple server URLs](#).

Nginx configuration in CloudBeaver Community

In the Community Edition, you can configure Nginx manually to set up a web server in front of CloudBeaver.

If you prefer, you can configure HTTPS directly for the Jetty server. For details, see [Jetty configuration](#).

1. **Install Nginx on your server.** Refer to the [official Nginx documentation](#).
2. **Add a proxy configuration:**
 - a. Open the main Nginx configuration file in a text editor.
 - b. In your Nginx configuration file inside the `http` section, add the following:

```
server {
    listen 80;
    server_name _;
    client_max_body_size 500M;
    location / {
        proxy_pass http://127.0.0.1:8978;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}
```

To ensure the client's real IP address is forwarded when using Nginx, add the `forwardProxy:true` parameter to your [server configuration](#).

3. **Apply the changes:**
 - **Reload** the Nginx configuration.
 - Open CloudBeaver in your browser using `http://127.0.0.1`.

WebSockets

WebSockets are critical for CloudBeaver. If they are blocked, the UI may look normal but never refresh, SSO may hang, and query results may not appear.

Typical traffic over the socket includes:

- the SSO login flow
- metadata changes (scripts, connections)
- SQL execution

Requirements for WebSockets

Important

CloudBeaver **deployments** are preconfigured. The socket uses the same host and port, and switches to `wss` when the server uses HTTPS. For details on enabling HTTPS in CloudBeaver, see [Force HTTPS mode](#).

If you use a different proxy or network stack, make sure it meets these requirements. No extra ports or domains are needed:

Area	Requirement
Protocol	Use HTTP/1.1 to upstream, do not downgrade to HTTP/1.0
Upgrade	Pass <code>Upgrade</code> and <code>Connection: upgrade</code> headers unchanged
Connection stability	Keep the socket open with generous read/idle timeouts
Scheme	Match the server scheme: <code>ws://</code> over HTTP, <code>wss://</code> over HTTPS
Endpoint	<code>/api/ws</code> on the same host and port (e.g. <code>wss://<your-domain>/api/ws</code>)
TCP port	Same as the server - no extra port required

Area	Requirement
Metadata	Preserve <code>Host</code> , <code>X-Forwarded-Proto</code> , and related headers
Proxy behavior	Do not rewrite, buffer, or compress WebSocket frames
Load balancing	Pin a client to the same backend for the socket's lifetime
Network filtering	Ensure firewalls, VPNs, and security proxies allow WebSocket traffic
CDN/WAF	Some providers (e.g., Cloudflare) block WebSockets by default – enable them

Quick checklist

1. Open the browser devtools Network tab, filter by `ws` . A single `ws` or `wss` connection should establish with HTTP 101 and stay open
2. If the server is HTTPS, the socket must be `wss` too. Mixed content will be blocked by the browser
3. On the proxy, confirm HTTP/1.1 to upstream and upgrade headers are passed
4. Increase idle timeouts so long tasks do not drop the socket
5. Check corporate VPNs or filtering proxies that block or rewrite WebSockets

Tip

If the socket connects, then closes after a minute or two, it is usually a proxy or load balancer timeout. Increase the read timeout, and keep HTTP/1.1 with upgrade headers.

Multiple Server URLs

Multiple server URLs let you control from which domains users can access the application. Administrators can list several allowed domains instead of relying on a single entry point.

Why enable multiple URLs

- **Internal vs external access** - give employees a private domain while exposing a separate one for customers or partners
- **Single sign-on requirements** - many SSO providers bind logins to a single domain; with multiple URLs you can support both internal SSO and external SSO flows
- **Security hardening** - redirect users who open the app from unapproved domains to the primary URL
- **Business continuity** - keep backup domains to handle DNS outages or corporate network routing issues
- **Smooth migrations** - support both old and new domains during a transition period without breaking access

Tip

Always configure **Force HTTPS** first, so all listed domains redirect consistently to HTTPS.

How it works

- add one or more domains in **Allowed server URLs** (line-separated)
- the **first domain** becomes the **primary entry point** - all redirects point there
- users opening the app from unlisted domains get redirected to the primary URL
- if no domains are listed, the app accepts requests from any domain

Configure multiple URLs

1. Go to **Settings -> Administration -> Server configuration**
2. Find **Allowed server URLs**
3. Enter domains, one per line
4. Press Save

Info

You can also add domains directly in the server configuration file under the `supportedHosts` list. When a domain is generated through the CloudBeaver (with **Domain Manager**), it's added to **Allowed server URLs** automatically. For details, see [Server configuration](#).

Input rules

- enter **hostnames only**, without scheme or path, for example:

Example

```
app.example.com
app-ext.example.com
```

- if you paste a full URL, the server **keeps only the host** during parsing
- separate entries with **line feeds**
- the field is **editable** and **optional**
- values are validated - extra symbols, paths, and wildcards are rejected

Warning

The server never redirects to a host that isn't on this list.

Force HTTPS with multiple URLs

- HTTPS enforcement is controlled by **Force HTTPS**
- when Force HTTPS is on, the app redirects `http://` to `https://`
- when domains are created automatically by **Domain Manager**, the flag is also switched to HTTPS

How Force HTTPS and multiple URLs work together

- **Force HTTPS** - ensures all traffic is encrypted and cookies are secure
- **Multiple server URLs** - restricts access to approved domains and sets a primary redirect target

Combined, they guarantee secure and consistent access.

SSO and identity providers

- during OAuth and other SSO flows, the app uses the **current request URL**, not a legacy single “server URL” value
- restricting the list helps avoid redirects to unapproved domains during login

Work with domain manager

- if you use **Domain Manager**, add new domains **to the end of the list**
- don't replace existing entries - keep historical domains during migrations until traffic fully moves

Best practices

- keep 3-5 domains max
- prefer DNS names
- put the domain you want in browser address bars **first**
- review proxy rules for nginx or haproxy so external HTTPS requests are passed correctly

Secret management

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

CloudBeaver supports external secret managers to securely store and access credentials.

Secret manager	Description
AWS Secrets Manager	Managed service from AWS for storing and retrieving secrets securely. Requires IAM permissions.
HashiCorp Vault	Open-source tool for managing secrets and protecting sensitive data. Supports fine-grained access control.
CyberArk	Enterprise-grade security platform designed for managing privileged access and secrets.

Note

If a secret is configured, it takes priority over other authentication settings.

Configure secret provider

Before adding a secret provider in CloudBeaver, make sure:

- the provider is set up and accessible
- you have the required permissions
- secret management is enabled in the server configuration

Turn on secret management

1. Go to **Settings -> Administration -> Server Configuration**
2. In the **Configuration** section, enable **Secret Management**
3. Click **Save**

Once enabled, a new **Secret Management** tab appears in the interface.

Add a secret provider

1. Open the **Secret Management** tab
2. Click **+ Add**
3. Choose a provider from the dropdown list

AWS secrets

1. Fill in the required fields

The screenshot shows the 'CONFIGURATION CREATION' form in the CloudBeaver interface. The left sidebar contains a navigation menu with items like 'Server Configuration', 'Global Preferences', 'Users and Teams', 'Connection Templates', 'Query Manager', 'Driver Management', 'AWS Settings', 'Identity Providers', 'Secret Management' (highlighted), 'Domain Manager', and 'Product Information'. The main panel has a header with '+ ADD', 'REFRESH', and 'DELETE' buttons. Below the header, there's a 'CONFIGURATION CREATION' section with a 'CANCELED' button and a 'CREATE' button. The form is divided into two main sections: 'OPTIONS' and 'PARAMETERS'. Under 'OPTIONS', there are fields for 'Provider *' (a dropdown menu with 'AWS secrets' selected), 'ID *' (a text input with 'aws-secret'), 'Configuration name *' (a text input with 'AWS-Secret'), and 'Description' (a text area with 'Read datasource parameters from AWS secrets'). Under 'PARAMETERS', there are fields for 'Region *' (a text input with 'eu-central-1') and 'CloudId *' (a text input with 'AWS').

Field	Description
ID	Enter a unique identifier for the configuration.
Configuration name	Enter a descriptive name for this configuration.
Description (Optional)	Provide a brief description of this secret provider configuration.
CloudId	AWS configuration settings. For more details on configuring AWS Cloud in CloudBeaver, see AWS Cloud Explorer .

Field	Description
Region	AWS region where secrets are stored.

2. Click **Create** to save the configuration

Vault secrets

1. Fill in the required fields

The screenshot shows the 'CONFIGURATION CREATION' form in the 'Secret Management' section. The form is divided into two main sections: 'OPTIONS' and 'PARAMETERS'. The 'OPTIONS' section includes fields for 'Provider *' (set to 'Vault secrets'), 'ID *' (set to 'vault-secret'), 'Configuration name *' (set to 'Vault'), and 'Description' (set to 'Read datasource parameters from AWS secrets'). The 'PARAMETERS' section includes fields for 'Vault server URL *' (set to 'https://<vault-server-ip>:8200'), 'Authentication Type *' (set to 'TOKEN'), 'Vault JWT Provider ID' (set to 'vault-cloudbeaver'), 'Vault Role Claim' (set to 'user1'), and 'Token' (empty). The 'CREATE' button is visible in the top right corner.

Field	Description
ID	Enter a unique identifier for the configuration.
Configuration name	Enter a descriptive name for this configuration.
Description (Optional)	Provide a brief description of this secret provider.
Authentication Type	Select an authentication type (TOKEN , JWT , USERNAME_PASSWORD).

Note

When using `USERNAME_PASSWORD` or `TOKEN` authentication, all users will have the same level of access to all secrets. If you need access control with different permissions for users, use `JWT` authentication.

2. Click **Create** to save the configuration

CyberArk

1. Fill in the required fields

The screenshot shows the 'CONFIGURATION CREATION' form in the CloudBeaver interface. The form is divided into two main sections: 'OPTIONS' and 'PARAMETERS'. The 'OPTIONS' section includes fields for 'Provider' (set to 'CyberArk'), 'ID' (set to 'arc'), 'Configuration name' (set to 'CyberArc'), and 'Description' (set to 'Read datasource parameters from AWS secrets'). The 'PARAMETERS' section includes fields for 'Server URL' (set to 'https://cyberark.example.com/'), 'Account' (set to 'myorg'), 'Login' (set to 'admin'), and 'API Key' (masked). The form also has 'CANCEL' and 'CREATE' buttons. The sidebar on the left shows navigation options like 'Server Configuration', 'Settings', 'Users and Teams', 'Query Manager', 'Driver Management', 'AWS Settings', 'Identity Providers', 'Secret Management', 'AI Settings', 'Domain Manager', and 'Product Information'. The top right shows a user profile 'cbadmin' and a help icon.

Field	Description
ID	Enter a unique identifier for the configuration.
Configuration name	Unique identifier for the secret configuration in CloudBeaver.
Description (Optional)	Provide a brief description of this secret provider.
Server URL	URL of the Conjur OSS server (e.g., <code>https://conjur.example.com</code>).

Field	Description
Account	Conjur tenant or organization account name (e.g., <code>myorg</code>).
Login	Conjur user ID used for authentication.
API Key	API key associated with the Conjur user for authentication login.
Policy branch	Path to the policy namespace where secrets are stored (e.g., <code>dev/app</code>).
Trust CyberArk SSL Certificate (Optional)	Trust the CyberArk server's SSL certificate without manual import. Enable this if you're using a self-signed certificate or connecting to a server whose certificate isn't trusted by default.

Info

For more details on authentication, policy structure, and SSL configuration, [CyberArk official documentation](#).

2. Click **Create** to save the configuration

Authentication types

Username/password authentication

Use a username and password to authenticate with the Vault server.

Field	Description
Username	Username for authentication.
Password	Password for authentication.

Token authentication

Use an authentication token to access the Vault server.

Field	Description
Token	Authentication token required to access the Vault server.

JWT authentication

Use a JSON Web Token (JWT) for authentication.

Field	Description
Vault server URL	URL of the Vault server used to retrieve secrets.
Vault JWT Provider ID	Identifier for the JWT provider created on the Vault side (Auth Method in Vault).
Vault Role Claim	The value of the role name in the JWT token where the custom role is stored. See the configuration steps below.

Configuring JWT authentication

JWT authentication requires an external **Single Sign-On (SSO) provider** (e.g., [Okta](#)) because Vault does not generate JWTs internally. It works with any **OpenID Connect (OIDC) provider** that issues JWT tokens.

This configuration happens on the Vault side and includes setting up authentication methods, validation rules, and role mappings.

Note

You must be logged into CloudBeaver with the same authentication provider that Vault is configured with. For instance, if Vault is set up with Okta, you must log in through Okta—logging in through a different provider, like Google, will not work.

1. Enable the JWT authentication method in Vault

- enable JWT authentication
- configure JWT validation settings, such as issuer and audience

2. Create a role in Vault

- vault does not assign roles automatically
- use the [API](#) to create a role

- define policies and specify claims for user identification
- a. **Configure role assignment in the SSO provider**
- ensure your SSO provider includes user roles in the JWT.

Info

Use the SSO provider's documentation to configure role claims.

b. **Match the Vault Role Claim with a JWT claim**

- the **Vault Role Claim** field should match the custom claim in the JWT token
- if it does not match, authentication will fail

Info

See the [Vault documentation](#) for more details.

Creating a connection

1. Start [creating a connection](#)
2. In the **Create Connection** wizard, go to the **Main** tab, choose a **Secret Configuration ID** and enter the **Secret Name** in the **Secret Parameters** section.

Info

The secret name must match the name stored in your secret provider:

- for **HashiCorp Vault**, use `secret/your_secret_name`
- for **AWS Secrets Manager**, use `your_secret_name`
- for **CyberArk**, use the full variable path, for example: `app/dev/db/password`. In Conjur, secret names reflect the full variable ID defined in your policy. For more details, see [Conjur documentation](#).

3. If the secret doesn't include all required details (such as the `host` or `port`), enter them manually in the connection settings.
4. Click the **Test** button to verify your settings (if configured correctly, CloudBeaver will establish a connection using the **Secret Provider**)
5. Click **Create** to save the connection

Pre-configured database connections

Pre-configured database connections

This article explains how to set up and manage pre-configured database connections for the CloudBeaver server. These connections are defined by the administrator and stored on the server, making them unmodifiable by end users. Users can choose from these pre-configured connections directly through the main CloudBeaver toolbar and only need to provide their username and password to access the datasource, no other parameters are necessary.

Info

For information about server and workspace configuration, see [Server configuration](#).

Datasources configuration file

All project-level configurations are located in the `${WORKSPACE}/workspace/GlobalConfiguration/.dbeaver` folder. The datasources themselves are defined within the `data-sources.json` file (see [Workspace location](#) for details).

This file contains the settings for each pre-configured connection, allowing the CloudBeaver server to manage and display these connections.

Example configuration

```
{
  "folders": {
  },
  "connections": {
    "dbeaver-sample-database-sqlite-1": {
      "provider": "sqlite",
      "driver": "sqlite_jdbc",
      "name": "DBeaver Sample Database (SQLite)",
      "save-password": true,
      "show-system-objects": true,
      "configuration": {
        "database": "/Users/user/Library/DBeaverData/workspace6/.metadata/sample-database-sqlite-1/Chinook.db",
        "url": "jdbc:sqlite:/Users/user/Library/DBeaverData/workspace6/.metadata/sample-database-sqlite-1/Chinook.db",

```

```
    "configurationType": "MANUAL",  
    "type": "dev",  
    "closeIdleConnection": true  
  }  
}  
}
```

Tip

This file follows the same format as the **DBeaver** datasources configuration file. You can create this configuration in DBeaver and then copy it to your CloudBeaver server.

Pre-configured permissions for connections

You can control access to **pre-configured connections** for specific users and teams using a permissions configuration file.

The configuration is located in `\${WORKSPACE}/workspace/GlobalConfiguration/.dbeaver/data-sources-permissions.json` (see **Workspace location** for details).

Important

The permissions configuration is applied on the first server start and on each restart. Make sure the required **users** and **teams** already exist before assigning permissions. For details on creating predefined teams, see **Initial data configuration**.

Example configuration

```
{
  "cql-19a9cde51ac-5136c4907ffd12f4": [ "test_team" ],
  "postgres-jdbc-19a9ccb2c0e-201a227bc74f6a48": [ "test_team", "user" ]
}
```

- "cql-19a9cde51ac-5136c4907ffd12f4" and "postgres-jdbc-19a9ccb2c0e-201a227bc74f6a48" - connection IDs. You can get them from the **connection configuration file**.
- ["test_team", "user"] - list of users and teams who will have access to the connection.

Logs

Log viewer

CloudBeaver supports viewing and analyzing user logs through the built-in **Log Viewer** tool. You can use it to quickly identify issues and get detailed error information.

Open the Log Viewer

1. Go to the **Tools** menu on the top **toolbar**.
2. Select **Log Viewer**.

View log details

Logs appear with timestamps and brief descriptions. To get detailed info on a specific error:

1. Click the error link in the log.
2. View the detailed information in the panel displayed on the right.

LOG VIEWER ×

TIME	MESSAGE	🗑️ CLEAR
	5:53:45 PM Error executing query: SQL Error [42703]: ERROR: column "artisti" does not exist	
5:48:53 PM	Load Schema	
5:48:53 PM	Load items ...	
5:48:53 PM	Load Roles	
5:48:53 PM	Load System Info	
5:48:53 PM	Load Storage	
5:48:53 PM	Load Extensions	
5:48:53 PM	Load Event Triggers	
5:48:53 PM	Load Schemas	
5:48:53 PM	Load items ...	
5:48:53 PM	Load Database	
5:48:53 PM	Load items ...	
5:48:52 PM	Load System Info	
5:48:52 PM	Load Administer	

COPY CLOSE

ERROR 2025-03-18T16:53:45.000Z

Error executing query:
SQL Error [42703]: ERROR: column "artisti" does not exist
Position: 8

```
io.cloudbeaver.DBWebException: Error executing query:
SQL Error [42703]: ERROR: column "artisti" does not exist
Position: 8
    at
    io.cloudbeaver.service.sql.WebSQLProcessor.processQuery(WebSQLProc
essor.java:273)
    at
    io.cloudbeaver.service.sql.impl.WebServiceSQL$2.run(WebServiceSQL.jav
a:472)
    at
    io.cloudbeaver.model.session.WebSession$1.run(WebSession.java:571)
    at
    org.jkiss.dbeaver.model.runtime.AbstractJob.run(AbstractJob.java:119)
    at org.eclipse.core.internal.jobs.Worker.run(Worker.java:63)
```

Tip

You can customize the default logging settings for your server. For information on configuring logs, see [Product settings parameters](#).

Disable log viewer for users

To prevent users from viewing logs:

1. Open **Settings -> Administration -> Settings**.
2. Under **Tools**, select **Disable log viewer**.
3. Save your changes.

Info

When this option is enabled, the **Log viewer** tab is hidden for all users, including administrators. Administrators can still access logs through other ways - on the server, in audit logs, or in the [Query Manager database](#).

Log customization

CloudBeaver uses `logback.xml` to configure server logging. This file controls log level and log message pattern.

Info

For details on viewing logs directly in the CloudBeaver, see [Log Viewer](#).

Default logging configuration

CloudBeaver's default `logback.xml` :

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>

    <appender name="CONSOLE" class="ch.qos.logback.core.ConsoleAppender">
        <layout class="ch.qos.logback.classic.PatternLayout">
            <Pattern>
                %d{dd-MM-yyyy HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n
            </Pattern>
        </layout>
    </appender>

    <statusListener class="ch.qos.logback.core.status.NopStatusListener" />
    <logger name="org.jkiss" level="DEBUG" />
    <logger name="com.dbeaver" level="DEBUG" />
    <logger name="io.cloudbeaver" level="DEBUG" />
    <logger name="org.apache.kafka" level="WARN" />

    <root level="INFO">
        <appender-ref ref="CONSOLE" />
    </root>

</configuration>
```

Customize logging level and pattern

Customize CloudBeaver logging by creating a custom `logback.xml` file:

1. Create and save your custom `logback.xml` file on your server.
2. Mount your custom directory to the container's `/opt/cloudbeaver/conf/custom` folder.

3. Restart CloudBeaver to apply your new logging configuration.

Info

For mounting details, see [Docker volumes documentation](#).

Logging level

Set the logging level in the `<root>` element's `level` attribute:

- `ERROR` : logs errors only.
- `WARN` : logs warnings and errors.
- `INFO` : standard logs (recommended for everyday use).
- `DEBUG` : detailed debugging info.
- `TRACE` : maximum verbosity for troubleshooting.

Example:

```
<root level="WARN">
  <appender-ref ref="CONSOLE"/>
</root>
```

Info

To enable detailed logging of GraphQL queries, turn it on in [Global preferences](#).

Logging pattern

Define custom log message pattern in the `<Pattern>` element.

Example (time only):

```
<Pattern>
  %d{HH:mm:ss} [%t] %-5level - %msg%n
</Pattern>
```

Customize logging in Preferences

In addition to editing `logback.xml`, you can configure logging directly in the CloudBeaver preferences. These options let you:

- save all API requests in the Query Manager database
- choose how many days to keep logs
- log GraphQL requests to console

For details, see [Preferences - Logger settings](#).

Query manager

Query manager

Note

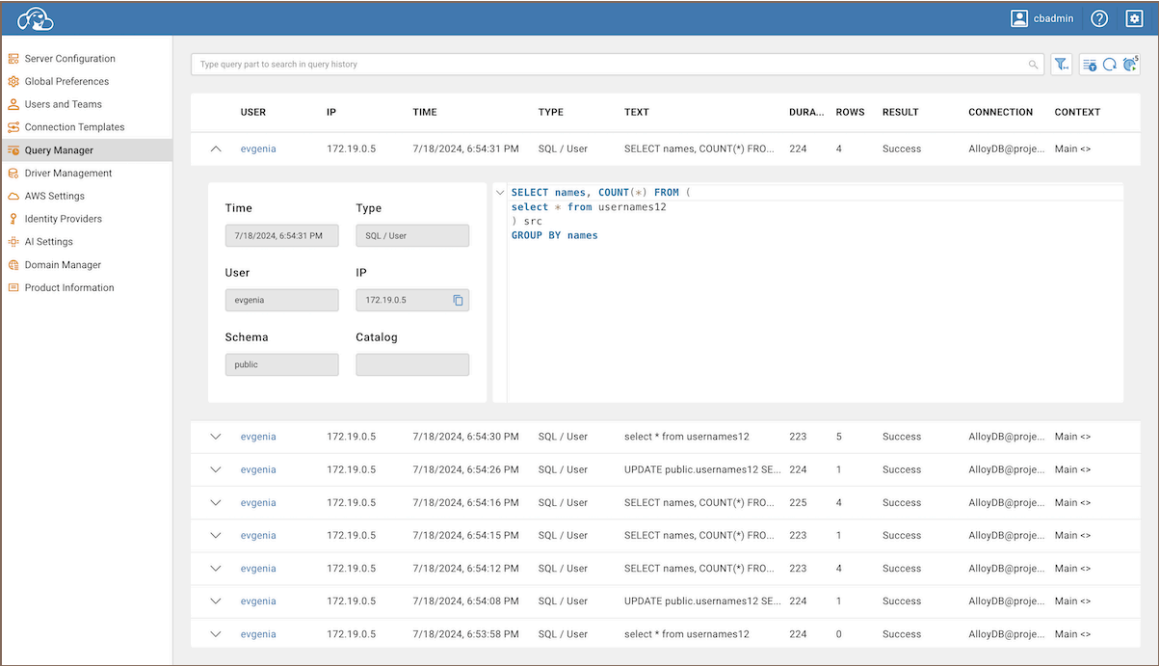
This feature is available in Enterprise, AWS, and Team editions only.

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

Tip

For more information on how to adjust Query Manager settings through the configuration file, see Query Manager configuration.

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



The following table describes each field in the Query Manager:

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

The Query Manager offers several features to enhance usability:

- **Search Functionality:** Use the **Search** function to locate specific queries by entering SQL text in the Search field above the table.

- **Query Manager Options:** Customize view and behavior settings by clicking the **Query Manager Options** button . For more information, see [Query Manager Options](#).
- **Column Manager:** Customize columns using the **Column Manager** button .
- **Refresh Query Manager:** Manually update the data displayed in the Query Manager by clicking the **Refresh** button  located in the toolbar.
- **Auto Refresh:** The Query Manager can automatically update the query list. You can disable this feature by toggling the **Auto-Refresh** button  in the toolbar. This setting is adjustable, for information on how to customize it, see the section on [Auto Refresh](#).

Query Manager Options

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

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

Query types

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

Option	Description
User queries	Show all queries directly executed by users.
Filtered user queries	Show user queries that meet specific criteria.
User scripts	Show batches of queries executed as scripts.
Utility functions	Show system-level utility function calls.
Metadata read	Show queries that read database metadata.
Metadata write (DDL)	Show queries that modify database structure.
Query status	Allows to filter queries based on their execution s

Option	Description
	tatus. The available options are:
	All: Displays all queries regardless of their execution outcome.
	Fail: Shows only the queries that have failed.
	Success: Displays only the queries that have executed successfully.

Object types

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

Option	Description
Sessions	Include queries related to user sessions.
Queries	Include individual query executions.

Filter by date

Specify the date range to view queries:

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

Filter by users

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

Filter by drivers

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

Sorting and Settings

Adjust how query results are sorted and displayed:

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

Auto Refresh

Configure automatic refresh of query information:

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

Restore defaults

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

Cleanup old records

CloudBeaver can automatically remove old query history records. This helps keep your query history clean and manageable by deleting outdated entries based on a set number of days.

Info

For more details, see the [Preferences](#).

Query history

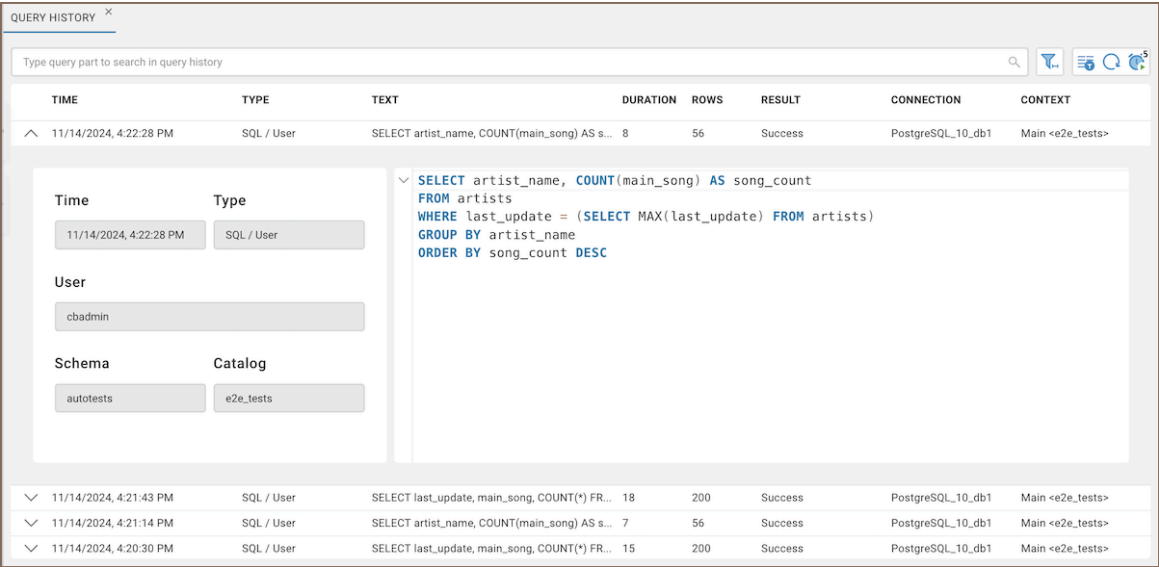
Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

CloudBeaver stores all executed queries in the internal database. You can view all your queries in the Query History.

Info

For more information on an internal database, see **Server Database**. To access the **Query History**, navigate to the **Tools** menu in the top toolbar and select the **Query History** option.



The following table describes each field in the **Query History**:

Field	Description
User	Displays the name of the person who executed the query.
Time	Shows when the query was executed.
Type	Identifies whether the query was initiated by a user or generated by the system.

Field	Description
Text	Contains the SQL code that was executed.
Duration	Measures how long the query took to complete, in milliseconds.
Rows	Counts how many rows were affected or returned by the query.
Result	Describes the outcome of the query, such as "Success" or any errors that occurred.
Connection	Specifies which database connection was used to execute the query.
Context	Provides additional information about the environment or session in which the query was executed.
Schema	Indicates the database schema used for the query.
Catalog	Represents the database catalog being accessed during the query.

Tip

You can access the internal database with all queries from **Query History**. For more details, see [How to access query history database](#).

Capabilities

The Query History offers several features to enhance usability:

- **Search Functionality:** Use the **Search** function to locate specific queries by entering SQL text in the Search field above the table.
- **Query History Options:** Customize view and behavior settings by clicking the **Query History Options** button .
- **Column Manager:** Customize columns using the **Column Manager** button .

- **Refresh Query History:** Manually update the data displayed in the Query History by clicking the **Refresh** button  located in the toolbar.
- **Auto Refresh:** The Query History can automatically update the query list. You can disable this feature by toggling the **Auto-Refresh** button  in the toolbar.

Tip

The **Query History** has similar settings to the Query Manager. For a comprehensive guide on the functionality, see [Query Manager options](#).

Query manager database

Note

This feature is available in **Enterprise** and **AWS** editions only.

The system uses an internal database to store query history, API call logs, and other service data.

Accessing the database

1. Create the target folder inside the container.

```
docker compose exec cloudbeaver mkdir -p /opt/cloudbeaver/workspace/sample
```

2. Copy the database folder.

```
docker compose exec cloudbeaver cp -r /opt/cloudbeaver/workspace/.metadata/qmdb/ /opt/cloudbeaver/workspace/sample/
```

3. **Connect** to the **internal server database** used in your deployment (for example, **H2 Embedded V.2**):

- Set the path to `${workspace}/sample/qmdb/qmdb`.
- Default login and password aren't required unless changed in the **Query manager database configuration**.

The screenshot shows the 'DRIVER PROPERTIES' tab in the Query Manager configuration. The interface is divided into several sections:

- Driver:** A dropdown menu showing 'H2 Embedded V.2'.
- Database:** A text input field containing the path `${workspace}/sample/qmdb/qmdb`.
- Authentication:** Fields for 'User name' and 'User password', both empty. Below them is a checkbox labeled 'Save credentials for all users with access'.
- Connection name *:** A text input field containing 'H2 Embedded V.2@localhost'.
- Description:** An empty text input field.
- EXPERT SETTINGS:** A section with a right-pointing arrow and the text 'EXPERT SETTINGS'.

Info

The `H2 Embedded V.2` driver might be disabled by default. To enable it, see [Server configuration and administration](#).

4. Explore the internal tables to view stored data:

Table	Description
<code>qm_query</code>	Contains query history
<code>qm_api_call_log</code>	Contains API call logs. For configuration details, see Logger settings .

Workspace location

By default, CloudBeaver stores all its files (configurations, scripts, etc.) in the `/opt/cloudbeaver/` on the host machine.

Folder	Description
<code>workspace</code>	Workspace files for CloudBeaver.
<code>drivers</code>	Auto downloaded database drivers.
<code>conf</code>	Configuration files for CloudBeaver. Learn more

Access workspace

1. Locate the name of the running container:
2. Open a terminal on the host machine.
3. Run the following command to list all running containers in the Compose project:

```
docker-compose ps
```

4. Identify the service name and open a shell inside the container:

```
docker-compose exec <service_name> /bin/bash
```

Info

Replace `<service_name>` with the actual name of the service from your `docker-compose.yml` file.

5. After entering the container, navigate to the workspace directory:

```
cd workspace/
```


Workspace in Amazon S3

CloudBeaver supports storing its workspace in an AWS S3 bucket. To enable this, update your `docker-compose.yml` and configure the correct environment variables.

Info

For more details on AWS S3 configuration, including setting up buckets, permissions, and best practices, see the official [Amazon S3 Documentation](#)

Update docker compose

Make sure your CloudBeaver service includes the following environment variables:

```
services:
  cloudbever:
    environment:
      - AWS_ACCESS_KEY_ID=${AWS_ACCESS_KEY_ID}
      - AWS_SECRET_ACCESS_KEY=${AWS_SECRET_ACCESS_KEY}
      - AWS_REGION=${AWS_REGION}
      - CLOUDBEAVER_WORKSPACE_LOCATION=${CLOUDBEAVER_WORKSPACE_LOCATION}
```

Configure S3 workspace

Define these variables in your `.env` file:

```
AWS_ACCESS_KEY_ID=your-access-key
AWS_SECRET_ACCESS_KEY=your-secret-key
AWS_REGION=your-region
CLOUDBEAVER_WORKSPACE_LOCATION=s3:///dbeaver-downloads/test_workspace
```

Important

- The `CLOUDBEAVER_WORKSPACE_LOCATION` path must use **triple slashes (`s3:///`)** before the bucket name. This is required for proper S3 path handling.
- Replace `dbeaver-downloads` with your actual S3 bucket name.
- `test_workspace` is the subfolder where CloudBeaver will store workspace data.

Limitations of using S3 Workspace

1. **No embedded databases**
2. CloudBeaver **cannot use embedded databases** (such as H2) with an external S3-based workspace.
3. Storing an embedded database in S3 would cause **severe performance issues**.
4. **Separate database node required**
5. To use an S3 workspace, you **must** configure an external database such as **PostgreSQL, MySQL, or another supported DB**.
6. Make sure the database is properly defined in `docker-compose.yml` and `CLOUDBEAVER_DB_*` environment variables.

Info

For more information on CloudBeaver's database, see [Server Database](#).

Workspace migration guide

This article provides a guide on how to migrate the CloudBeaver Enterprise Edition workspace. The process involves stopping the server, copying the workspace folder, and then restarting the server.

Tip

For detailed deployment instructions, refer to the [Docker image deployment](#).

Migration process

Step 1: Stop the CloudBeaver EE server

First, stop the CloudBeaver server to ensure that no data is being written to the workspace during the migration. This can typically be done by terminating the running server instance.

Step 2: Copy the workspace folder

Create a copy of the workspace folder. This folder is typically located in the directory specified when the Docker container is launched. For example, to start the container with a specific workspace directory, you might use the command:

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

You will need to copy all files from the workspace. Make sure to transfer all files to the new location where you wish to migrate your CloudBeaver workspace.

Tip

For more details, see [Workspace location](#).

Step 3: Start the CloudBeaver EE server

After relocating the workspace, restart the CloudBeaver Docker container at the new location. When launching the Docker container, it is essential to specify the correct path to the moved workspace to ensure proper configuration. Use the Docker run command with the updated volume path, like so:

```
docker run --name cloudbeaver-ee --rm -ti -p 8978:8978 -v [new-workspace-path]:/opt/cloudbeaver/workspace dbeaver/cloudbeaver-ee:latest
```

This command ensures that the CloudBeaver server operates with the workspace in its new environment, maintaining all your configurations.

Command line parameters

There are two ways to pass command line parameters to the CloudBeaver server.

Modify the run script

Modify the `run-cloudbeaver-server.sh` script and add extra **parameters** after the `java` command in the last line.

For example, add a memory limit and UI language parameter:

```
java -Xmx2048M -jar ${launcherJar} \  
-product io.cloudbeaver.product.ce.product \  
-web-config conf/cloudbeaver.conf \  
-nl en -registryMultiLanguage
```

Then restart the server:

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

Note

To modify the script, you must **build CloudBeaver from sources**.

Pass parameters using environment variables

You can set the `JAVA_OPTS` environment variable with JVM parameters. This works for both manual startup and Docker containers.

Manual start

```
export JAVA_OPTS=-Xmx1000m  
./run-cloudbeaver-server.sh
```

Docker

Pass `JAVA_OPTS` to the container using the `-e` option:

```
docker run -d --name cloudbeaaver-ee -p 8978:8978 \
-e JAVA_OPTS="-Xmx2048M" \
-v /var/cloudbeaaver-ee/workspace:/opt/cloudbeaaver/workspace \
dbeaver/cloudbeaaver-ee:latest
```

Note

only JVM parameters can be passed with `JAVA_OPTS`.

System parameters

Name	Value	Example
<code>-nl</code>	UI language and locale (affects UI translation, number/date formats). For supported values, see Supported languages .	<code>-nl "en"</code>
<code>-data</code>	Workspace path.	<code>-data "/opt/cloudbeaaver/"</code>
<code>-clean</code>	Clears all Eclipse caches. Use it if CloudBeaver fails to start after it upgrades.	
<code>-vmargs</code>	Passes VM parameters.	See VM arguments table

Important

The `-data` parameter only affects the **workspace** location. Other folders are created in the **default location**.

VM parameters

You can pass any advanced Java parameters supported by your local JVM. Parameters supported by HotSpot JVM (21): <https://docs.oracle.com/en/java/javase/21/docs/specs/man/java.html>

Parameters supported by all JVMs:

Name	Value	Example
<code>-Xms</code>	Sets initial memory available for CloudBeaver.	<code>-Xmx1000m</code>
<code>-Xmx</code>	Sets maximum memory available for CloudBeaver.	<code>-Xmx4000m</code>

Session manager

The **Session Manager** in CloudBeaver allows you to view and search for active database sessions.

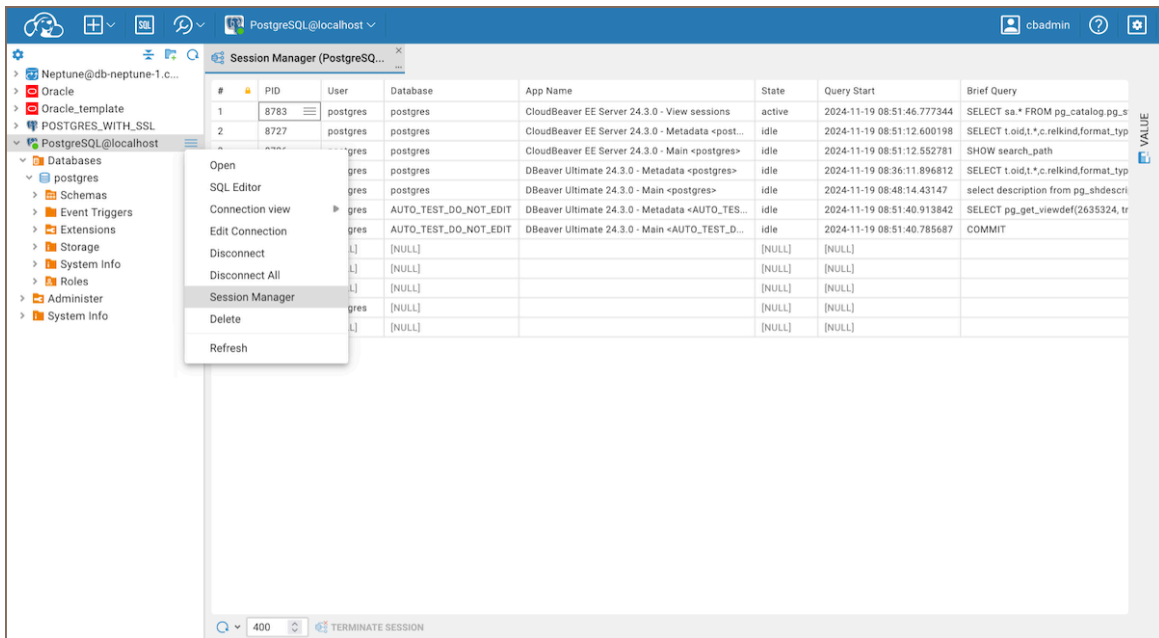
Supported databases

- Altibase
- Amazon Redshift
- Db2 for LUW
- Exasol
- Greenplum
- MariaDB
- Microsoft SQL Server
- MongoDB
- MySQL
- Oracle Database
- PostgreSQL

How to use

To open the **Session Manager**:

1. In the **Database Navigator**, navigate to your desired database.
2. Right-click and select **Session Manager**.



Once you have opened the **Session Manager**, you will have an ability to view all active database sessions.

To terminate a session, click the **Terminate Session** button  at the bottom of the **Session Manager** window.

Session details

For more detailed information, click the context menu button on a session and select **Show in value panel** from the context menu:

Session Manager (PostgreSQL)

#		PID	User	Database	App Name	State	Query
1		8783	postgres	postgres	CloudBeaver EE Server 24.3.0 - View sessions	active	2024-11-19 08:51:46.777344
2		8727	postgres	postgres	CloudBeaver EE Server 24.3.0 - Metadata <postgres>	idle	2024-11-19 08:51:46.777344
3		8726	postgres	postgres	CloudBeaver EE Server 24.3.0 - Main <postgres>	idle	2024-11-19 08:51:46.777344
4		7125	postgres	postgres	DBeaver Ultimate 24.3.0 - Metadata <postgres>	idle	2024-11-19 08:51:46.777344
5		7124	postgres	postgres	DBeaver Ultimate 24.3.0 - Main <postgres>	idle	2024-11-19 08:51:46.777344
6		7127	postgres	AUTO_TEST_DO_NOT_EDIT	DBeaver Ultimate 24.3.0 - Metadata <AUTO_TEST_DO_NOT_EDIT>	idle	2024-11-19 08:51:46.777344
7		7126	postgres	AUTO_TEST_DO_NOT_EDIT	DBeaver Ultimate 24.3.0 - Main <AUTO_TEST_DO_NOT_EDIT>	idle	2024-11-19 08:51:46.777344
8		25329	[NULL]	[NULL]		[NULL]	[NULL]
9		25327	[NULL]	[NULL]		[NULL]	[NULL]
10		25328	[NULL]	[NULL]		[NULL]	[NULL]
11		3279	postgres	[NULL]		[NULL]	[NULL]
12		3273	[NULL]	[NULL]		[NULL]	[NULL]

400

TERMINATE SESSION

DETAILS

QUERY

NAME	VALUE
Client	
User	postgres
Client Host	::1
Client Port	43124
App Name	CloudBeaver EE Server 24.3.0
Timings	
Backend Start	2024-11-19 08:51:46.737743
Xact Start	2024-11-19 08:51:46.775587
Query Start	2024-11-19 08:51:46.777344
State Change	2024-11-19 08:51:46.777345
General	
PID	8783
Database	postgres
State	active
Brief Query	SELECT sa.* FROM pg_catalog

Available tabs:

- **Details:** Detailed information about the session.
- **Query:** A summary of the SQL query associated with the session.

Deployment

Deployment options

CloudBeaver is a client-server application that requires server deployment. You can deploy it on a single host or in the cloud, depending on your needs.

Enterprise deployment options

CloudBeaver can be deployed in various environments. Choose the option that works best for your needs:

On-premises deployment

- [Docker Compose](#) - the simplest way to deploy and run the server on a single host.
- [Kubernetes](#) - ideal for those using Docker orchestration for managing clusters.

Cloud deployment

CloudBeaver supports deployment on major cloud platforms:

Amazon Web Services

- [AMI](#) - deploy using Amazon Machine Image.
- [ECS](#) - create and deploy a scalable ECS cluster for containerized applications.

Google Cloud

- [Google Cloud image](#)

Azure

- [Azure image](#)

AWS deployment options

- [AMI](#) - deploy using Amazon Machine Image.
- [AWS Marketplace](#) - use the AWS Marketplace.

Community deployment options

CloudBeaver Community Edition can be deployed on various platforms and environments:

- [Docker image](#)
- [AWS Marketplace](#)
- [Google Cloud](#)
- [Microsoft Azure](#)

CloudBeaver Enterprise

Docker compose

Note

This deployment method is available in the [Enterprise](#) edition only.

CloudBeaver Enterprise can be easily deployed using Docker Compose. This guide provides step-by-step instructions for setting up CloudBeaver Enterprise on Ubuntu, macOS, and Windows.

Note

CloudBeaver Enterprise's Docker Compose setup includes a pre-configured web service.

For advanced configurations, such as system requirements, ssl configuration, or podman requirements, refer to the [Advanced settings guide](#).

Required network access

To ensure that CloudBeaver can be deployed and updated successfully, make sure the server has outbound access to specific websites. These domains are required for downloading Docker images, validating licenses, updating components, and resolving database drivers.

Required domains

Add the following domains to your firewall or proxy allowlist:

Warning

If access to any of these sites is restricted, you may see errors when updating, downloading components, or validating the license.

Domain	Purpose
<code>hub.docker.com</code>	Pull Docker images from Docker Hub
Docker image mirrors	Content delivery domains used for image layers. See the full allowlist .

Domain	Purpose
<code>dbeaver.com</code>	License validation
<code>databases.team</code>	Let's Encrypt certificates for custom domains. Learn more on Domain Manager
Maven Central and repositories	Required to download some JDBC drivers we don't bundle with the product (see Maven Central)

Important

The `cloudbeaver` container **must** have outbound access to all required domains.

Check access from the command line

Before starting the containers, make sure your environment can access essential external services:

```
ping hub.docker.com
curl -I https://dbeaver.com
```

If you already have CloudBeaver running, you can also verify network access from inside a container.

If you're using **Docker**:

```
# List containers
docker ps

# Open a shell in the cloudbeaver container
docker exec -it <cloudbeaver_container_id> bash

# Test connectivity inside the container
ping hub.docker.com
curl -I https://dbeaver.com

# Check HTTPS certificate retrieval (useful if using a Domain Manager).
openssl s_client -connect <your-domain>:443 -servername <your-domain>
```

Tip

If you're having trouble resolving hostnames inside containers, review [Docker network settings](#). DNS issues can prevent access to required domains even if outbound connectivity is available.

If you're using **Kubernetes**:

```
# List pods
kubectl get pods

# Open a shell in the cloudbever pod
kubectl exec --stdin --tty <cloudbever_pod_name> -- /bin/bash

# Test connectivity inside the pod
ping hub.docker.com
curl -I https://dbeaver.com

# Check HTTPS certificate retrieval (useful if using a Domain Manager)
openssl s_client -connect <your-domain>:443 -servername <your-domain>
```

If DNS resolution fails inside a pod, check your **Kubernetes DNS configuration**. Misconfigured DNS can block access even when the network itself is available.

Important

Docker and Kubernetes environments require access to several domains for authentication, image pulls, and updates. Make sure all required domains are accessible from inside your containers or pods. You can test any domain from the **allowlist** using:

```
curl -I https://<domain>
```

Deploying with host networking

By default, CloudBeaver Enterprise runs using `docker-compose.yml`, which uses Docker's bridge network. If you need containers to run directly on the host network (for example, to bind CloudBeaver to a specific network interface), use `docker-compose-host.yml`.

Installation on Ubuntu

Required tools setup

- Ensure Docker and Docker Compose are installed by following the **Docker installation guide for Linux**.
- Install **Docker Compose** if it's not already installed.
- Install **Git** if it's not already installed.

Configuring and starting CloudBeaver Enterprise

1. Clone the deployment repository:

```
git clone https://github.com/dbeaver/cloudbeaver-deploy
```

2. Navigate to the CloudBeaver deploy directory:

```
cd cloudbeaver-deploy
```

3. Set up the `.env` file as described [here](#).

Important

You **must** change the default password in the `.env` file before starting the cluster.

4. Start the cluster:

```
docker compose up -d
```

5. Access CloudBeaver Enterprise at `http://<server-ip>`.

Tip

Replace `<server-ip>` with your server's actual IP address. You can configure the domain during the initial setup. For more details, see [Domain Manager](#).

Installation on macOS

Required tools setup

- Install Docker Desktop for macOS from [Docker website](#). Make sure Docker Desktop is running before proceeding.
- Install Docker Compose if it's not bundled with Docker Desktop:

```
brew install docker-compose
```

- Install Git if it's not already installed:


```
brew install git
```

- Verify the installation:

```
docker --version
docker-compose --version
git --version
```

Configuring and starting CloudBeaver Enterprise

1. Clone the deployment repository:

```
git clone https://github.com/dbeaver/cloudbeaver-deploy
```

2. Navigate to the CloudBeaver deploy directory:

```
cd cloudbeaver-deploy
```

3. Set up the `.env` file as described [here](#).

Important

You **must** change the default password in the `.env` file before starting the cluster.

4. Start the cluster:

```
docker-compose up -d
```

5. Access CloudBeaver Enterprise at `http://<server-ip>`.

Tip

Replace `<server-ip>` with your server's actual IP address. You can configure the domain during the initial setup. For more details, see [Domain Manager](#).

Installation on Windows

Required tools setup

- Install Docker Desktop.
- Install Git.

Tip

Supported on Windows 10 and 11. Docker Desktop doesn't work on Windows Server.

Configuring and starting CloudBeaver Enterprise

1. Clone the deployment repository:

```
git clone https://github.com/dbeaver/cloudbeaver-deploy
```

2. Navigate to the deployment directory:

```
cd cloudbeaver-deploy
```

3. Copy the example file to create your `.env` file:

```
bash
cp .env.example .env
```

4. Set up the `.env` file as described [here](#).

Important

You **must** change the default password in the `.env` file before starting the cluster.

5. Start the cluster:

```
docker-compose up -d
```

6. Access CloudBeaver Enterprise at `http://<server-ip>`.

Tip

Replace `<server-ip>` with your server's actual IP address. You can configure the domain during the initial setup. For more details, see [Domain Manager](#).

Update CloudBeaver Enterprise

For detailed update instructions, see [Update version](#).

Environment file configuration

The `.env` file provides configuration settings for CloudBeaver Enterprise. Below are key variables:

Docker Compose variables

Variable	Description	Default value
<code>IMAGE_SOURCE</code>	Defines the primary source of the Docker image.	<code>dbeaver</code>
<code>PODMAN_IMAGE_SOURCE</code>	Sets the Podman image source.	<code>docker.io/dbeaver</code>
<code>COMPOSE_PROJECT_NAME</code>	Defines the project name for Docker Compose.	<code>dbeaver</code>
<code>CLODBEAVER_VERSION_TAG</code>	Specifies the CloudBeaver version to use. See Docker tags .	<code>24.3.0</code>
<code>REPLICA_COUNT_EE</code>	Specifies the number of replicas for Enterprise Edition. See Docker Service Scale Documentation .	<code>1</code>
<code>PROXY_TYPE</code>	Specifies the proxy server to use. Supported values: <code>nginx</code> , <code>haproxy</code> . For more details on configuring proxy server, see the Proxy configuration .	<code>nginx</code>

CloudBeaver configuration variables

For the full list of CloudBeaver configuration variables and their descriptions, see [Server configuration](#).

AWS AMI

Note

This deployment method is available in the **Enterprise** edition only.

CloudBeaver Enterprise can be deployed on AWS using a preconfigured Amazon Machine Image (AMI).

Prerequisites

Before you start, ensure you have an **AWS account**. If you don't have one, you can create it on the [AWS signup page](#).

Access the AMI catalog

1. Log in to your AWS account and go to the **EC2 Dashboard**.
2. From the left-hand menu, select **AMI Catalog** under **Images**.
3. In the AMI catalog, click on **Community AMIs**.
4. In the **Search bar**, type `cloudbeaver-ee` and hit `Enter`.
5. You'll see a list of available AMIs. Select the version of CloudBeaver you'd like to deploy.

Tip

Choose the most up-to-date version to get the latest features.

Launch your instance

1. After selecting the AMI, click **Launch Instance with AMI**.
2. Choose an **Instance Type** that meets the minimum requirements:
 - 2 CPUs
 - 4 GB RAM
 - 100 GB SSD storage (recommended)

Security group setup

1. During instance configuration, set up your **security group**:
 - For security reasons, it's recommended **not to make the instance publicly accessible**.
 - Limit inbound traffic to specific IPs (e.g., your office IP address) and only open the necessary ports:
 - **Allow SSH traffic** (Optional): For administration and secure remote access.
 - **Allow HTTPS traffic from the internet**: To securely serve content over the web.
 - **Allow HTTP traffic from the internet**: For basic web traffic access.

Tip

Learn more about AWS security group best practices in the [AWS Security Groups documentation](#).

Review and launch

1. After configuring all the settings, review your instance details.
2. (Optional) Select or create a new **key pair** to access the instance later via **SSH**.
3. Click **Launch instance**.
4. Once the instance launches, monitor its status in the EC2 dashboard.
5. Begin the initial setup of the CloudBeaver.

Initial setup

1. In your browser, go to `http://<Instance_IP>/`, where `<Instance_IP>` is the public IP address of your instance.
2. You'll be redirected to CloudBeaver's initial configuration wizard to complete the setup.

Domain setup

After the initial setup, you can configure the domain.

1. To automatically generate a domain, navigate to **Settings -> Administration -> AWS Settings** and enter your **Agreement ID**.

Info

If you're setting up the domain manually, the Agreement ID is not required.

2. Navigate to the **Domain Manager** tab and follow the **instructions** to set up your domain.

Google Cloud

Note

This deployment method is available in the [Enterprise](#) edition only.

This guide shows how to deploy CloudBeaver Enterprise on Google Cloud Platform (GCP) using both the Google Cloud Console and the Google Cloud CLI.

Prerequisites

Before you begin, make sure you have:

- A **Google Cloud account** for accessing GCP services.
- **Compute Engine** set up in your project.

Info

If you don't have a GCP account, sign up on the [Google Cloud signup page](#).

Permissions

To deploy CloudBeaver in Google Cloud, you need certain permissions to create and manage resources like virtual machines, disks, and networks. Ensure you have the necessary permissions for your project.

Tip

For a full list of permissions and more information on how to manage them, see the official [Google Cloud Permissions documentation](#). If you don't have the required permissions, contact your system administrator.

Deployment in Google Cloud Console

Import the CloudBeaver custom image

1. Log in to the [Google Cloud Console](#), then go to **Compute Engine -> Images**. Click **[+] Create image**.
2. Complete the **Create an image** form:
 - Enter the image name in the format `cloudbeaver-ee-server-ubuntu-%version%` or

`cloudbeaver-ee-server-rhel-%version%`, depending on the operating system.

- Select **Virtual disk (VMDK, VHD)**, and in the **Virtual disk file** field, enter this URI: `cloudbeaver-ee-server/`. Click **BROWSE** and choose your version.

Tip

Replace `%version%` with the desired version number, such as `24-2-0`.

3. Click **Create**. The process may take up to 15 minutes.

Create a Compute Engine instance from the imported image

1. After the image is imported, navigate to the **Images** tab, find your imported image, and click **[+] Create instance**.
2. Name your instance and set up the following:
 - Under **Machine configuration**, choose a machine type with at least 1 CPU and 4GB RAM for optimal CloudBeaver performance.
 - Click **Change**, select your custom image, and set the disk size to at least 100GB.
3. In the **Firewall** section, check **Allow HTTP traffic** and **Allow HTTPS traffic** to make the instance accessible online.
4. Click **Create**. Your instance will be ready in a few minutes. Verify by pasting its **External IP** into your browser.
5. Begin the initial configuration of the CloudBeaver server in your browser.

Deployment with Google Cloud CLI

Activate Cloud Shell

1. In the [Google Cloud Console](#), go to **Compute Engine** and click **Activate Cloud Shell** to launch the CLI within GCP.
2. Authorize any prompts if needed.

Run the deployment command

Run this `gcloud` command to create a new instance:

```
gcloud beta compute instances create cloudbeaver-ee-server \
```

```
--zone=us-central1-a \  
--machine-type=e2-medium \  
--tags=http-server,https-server \  
--image=https://www.googleapis.com/compute/v1/projects/dbeaver-public/global/  
images/cloudbeaver-ee-ubuntu-24-2-0 \  
--create-disk=auto-delete=yes \  
--boot-disk-size=100GB --boot-disk-device-name=cloudbeaver-ee-server
```

You can customize the command as needed. For more details on using the Google Cloud CLI, see the [official documentation](#).

Tip

Replace the `--image` URL with a different **version** if needed.

Start with CloudBeaver server setup

1. After the instance is created, retrieve its external IP (needed for accessing the CloudBeaver server):

```
gcloud compute instances list --filter="name:cloudbeaver-ee-server" --  
format="get(networkInterfaces[0].accessConfigs[0].natIP)"
```

This command will return the `external-ip`, which you'll need to begin the initial setup.

2. Use the `external-ip` to start configuring and accessing the CloudBeaver interface.
3. In your browser, go to `http://<external-ip>/`, where `<external-ip>` is the public IP address of your GCP instance.
4. You'll be redirected to CloudBeaver's initial configuration wizard to complete the setup.

Tip

For guidance on locating your instance's public IP address, see [GCP: Viewing IP Addresses](#).

CloudBeaver public image URIs

- `https://www.googleapis.com/compute/v1/projects/dbeaver-public/global/images/cloudbeaver-ee-ubuntu-%version%`
- `https://www.googleapis.com/compute/v1/projects/dbeaver-public/global/images/cloudbeaver-ee-rhel-%version%`

Tip

Replace `%version%` with the desired version number, such as `24-2-0`. For more information on tags, see [CloudBeaver release cycles](#).

Microsoft Azure

Note

This deployment method is available in the **Enterprise** edition only.

This guide explains how to deploy CloudBeaver Enterprise on Microsoft Azure using the Azure Portal.

Prerequisites

Before you begin, ensure you have a **Microsoft Azure account** for accessing Azure services.

Tip

If you don't have an Azure account, sign up on the [Azure signup page](#).

Deployment

Select the CloudBeaver Enterprise image

1. Log in to the [Azure Portal](#) and search for **Community images** in the search bar.
2. In the search bar on the left side under **Community images**, enter `cloudbeaver-ee`. Select the image for your region from the list and click **Create VM**.

Configure the virtual machine

1. **Project details:**
2. **Subscription:** Choose your Azure subscription.
3. **Resource group:** Create a new group or select an existing one.
4. **Instance details:**
5. **Virtual machine name:** Set a unique name for the VM.
6. **Size:** Choose a VM with at least 2 vCPUs and 4 GiB RAM.
7. **Administrator account:**
8. **Username:** Set to `ubuntu`.
9. **SSH public key source:** Generate a new key pair or use an existing one.

10. **Inbound port rules:**
11. **Select inbound ports:** Enable ports **SSH (22)**, **HTTP (80)**, and **HTTPS (443)**.
12. Double-check your configuration and click **Review + Create** to finalize.
13. Begin the initial setup of the CloudBeaver server in your browser.

Initial setup

1. In your browser, go to `http://<Instance_IP>/`, where `<Instance_IP>` is the public IP address of your instance.
2. You'll be redirected to CloudBeaver's initial configuration wizard to complete the setup.

CloudBeaver AWS

AWS Marketplace

Note

This deployment method is available in the **AWS** edition only.

This guide provides step-by-step instructions for deploying CloudBeaver AWS on Amazon Web Services Marketplace.

Deployment steps

1. Log in to **Amazon Marketplace**. Search for `cloudbeaver aws`, then select **CloudBeaver AWS**.
2. Click **Launch new instance**.
3. Configure your software contract, then click **Continue to launch through EC2**.
4. Select the required CloudBeaver version in the **Software version** field, choose your region, and click **Continue to Launch**.
5. In **Instance Type**, select `t2.medium` or higher.
6. (Optional) Select or create a new **key pair** to access the instance later via **SSH**.
7. Set up your **Network settings**:
 - For security reasons, it's recommended **not to make the instance publicly accessible**.
 - Limit inbound traffic to specific traffic from IPs (e.g., your office IP address) or subnets and only open the necessary ports:
 - **Allow SSH traffic** (Optional): For administration and secure remote access.
 - **Allow HTTPS traffic from the internet**: To securely serve content over the web.
 - **Allow HTTP traffic from the internet**: For basic web traffic access.
8. Click **Launch instance**.

Initial setup

1. In your browser, go to `http://<Instance_IP>/`, where `<Instance_IP>` is the public IP address of your instance.
2. You'll be redirected to CloudBeaver's initial configuration wizard to complete the setup.

AWS AMI

Note

This deployment method is available in the **AWS** edition only.

CloudBeaver AWS can be deployed on AWS using a preconfigured Amazon Machine Image (AMI).

Prerequisites

Before you start, ensure you have an **AWS account**. If you don't have one, you can create it on the [AWS signup page](#).

Access the AMI catalog

1. Log in to your AWS account and go to the **EC2 Dashboard**.
2. From the left-hand menu, select **AMI Catalog** under **Images**.
3. In the AMI catalog, click on **Community AMIs**.
4. In the **Search bar**, type `cloudbeaver-aws` and hit `Enter`.
5. You'll see a list of available AMIs. Select the version of CloudBeaver you'd like to deploy.

Tip

Choose the most up-to-date version to get the latest features.

Launch your instance

1. After selecting the AMI, click **Launch Instance with AMI**.
2. Choose an **Instance Type** that meets the minimum requirements:
 - 2 CPUs
 - 4 GB RAM
 - 100 GB SSD storage (recommended)

Security group setup

1. During instance configuration, set up your **security group**:

- For security reasons, it's recommended **not to make the instance publicly accessible**.
- Limit inbound traffic to specific traffic from IPs (e.g., your office IP address) or subnets and only open the necessary ports:
 - **Allow SSH traffic** (Optional): For administration and secure remote access.
 - **Allow HTTPS traffic from the internet**: To securely serve content over the web.
 - **Allow HTTP traffic from the internet**: For basic web traffic access.

Tip

Learn more about AWS security group best practices in the [AWS Security Groups documentation](#).

Review and launch

1. After configuring all the settings, review your instance details.
2. (Optional) Select or create a new **key pair** to access the instance later via **SSH**.
3. Click **Launch instance**.
4. Once the instance launches, monitor its status in the EC2 dashboard.
5. Begin the initial setup of the CloudBeaver.

Initial setup

1. In your browser, go to `http://<Instance_IP>/`, where `<Instance_IP>` is the public IP address of your instance.
2. You'll be redirected to CloudBeaver's initial configuration wizard to complete the setup.

Domain setup

After the initial setup, you can configure the domain.

1. To automatically generate a domain, navigate to **Settings -> Administration -> AWS Settings** and enter your **Agreement ID**.

Info

If you're setting up the domain manually, the Agreement ID is not required.

2. Navigate to the **Domain Manager** tab and follow the **instructions** to set up your domain.

Additional setup and management

Version update

Update CloudBeaver Enterprise

Keep your CloudBeaver Enterprise server up to date to access the latest features, improvements, and security patches.

Tip

For detailed instructions, see the [Docker CLI Documentation](#) and [Docker Compose Documentation](#).

Update steps

Warning

Always **backup your workspace** before updating. Downgrades are not supported.

Update Docker Compose

To update a cluster, managed with Docker Compose:

1. Update the value of `CLOUDBEAVER_VERSION_TAG` in your `.env` file to the desired version. If using the `latest` tag, skip this step.
2. Pull the latest changes for `cloudbeaver-deploy` (Optional):

```
git pull
```

3. Pull new Docker images:

```
docker-compose pull
```

4. Restart the cluster:

```
docker-compose up -d
```

Update Docker image

To update using a standalone Docker image:

1. Stop and remove the existing container:

```
docker stop cloudbeeper-ee  
docker rm cloudbeeper-ee
```

2. Pull the new image:

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

3. Run the updated container:

```
docker run --name cloudbeeper-ee -d --restart unless-stopped -p 8978:8978  
-v /var/cloudbeeper-ee/workspace:/opt/cloudbeeper/workspace dbeaver/  
cloudbeeper-ee:latest
```

Workspace backup

Backup and restore data

CloudBeaver stores all data on disk in the `/opt/cloudbeaver/workspace` directory, which is typically mounted to a folder on the host machine (`/var/cloudbeaver/workspace` or `/var/cloudbeaver-ee/workspace`).

For additional information on these commands, see the official [Docker CLI Documentation](#).

To back up this data, follow these Docker commands for common tasks:

Access a running container

List services:

```
docker-compose config --services
```

- This command lists all services defined in your `docker-compose.yml` file.
- Find the name of the CloudBeaver service from the output.

Access the CloudBeaver container's shell:

```
docker-compose exec -it <service_name> /bin/bash
```

- Replace `<service_name>` with the exact name of your CloudBeaver service from the previous command.
- If your container uses a different shell (like `sh` instead of `bash`), adjust the command accordingly:

```
docker-compose exec -it <service_name> sh
```

Backup the database

If CloudBeaver uses a PostgreSQL database, back up the database to a dump file before stopping the container.

Info

For more details on an internal database, see [Server Database](#).

To create a backup of all PostgreSQL databases:

```
docker-compose exec <service_name> pg_dump -C -d cloudbever -U postgres > postgres_backup.sql
```

Copy workspace files from the container

First, create a backup of the workspace inside the running container:

```
docker-compose exec cloudbever tar -czvf /var/backups/backup.tgz -C /opt/cloudbever/workspace .
```

- This command compresses the `workspace` directory into `backup.tgz`.

To copy the backup file (`backup.tgz`) from the container to your local machine:

```
docker cp $(docker-compose ps -q <service_name>):/opt/cloudbever/backup.tgz /path/to/backup/location/
```

- Replace `/path/to/backup/location/` with the desired local destination.

Restore the database

Important

Ensure you have a PostgreSQL cluster running. If PostgreSQL isn't running, start it with:

```
docker-compose up -d postgres
```

If this is a new installation, clone the deployment repository and navigate to the directory:

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

To restore the PostgreSQL database:

1. Copy the database dump file back to the PostgreSQL container:

```
docker cp ./postgres_backup.sql $(docker-compose ps -q postgres):/postgres_backup.sql
```

2. Restore the database inside the PostgreSQL container:

```
docker-compose exec postgres psql -U postgres -f /postgres_backup.sql
```

3. Optionally, delete the dump file from the PostgreSQL container to free up space:

```
docker-compose exec postgres rm /postgres_backup.sql
```

Restore the workspace

To restore the workspace data:

1. Copy the workspace backup file back to the container:

```
docker cp /path/to/backup/location/backup.tgz $(docker-compose ps -q <service_name>)":/opt/cloudbeaver/backup.tgz"
```

- Replace `/path/to/backup/location/` with the desired local destination.

2. Extract the backup in the container:

```
docker-compose exec <service_name> tar -xvf /var/backups/backup.tgz -C /opt/cloudbeaver/workspace/
```

SSL certificate configuration

Proxy and SSL configuration

The instance contains an Nginx proxy server, the configuration of which is located at path `/etc/nginx/conf.d/cloudbeaver.conf`

To set up a connection via HTTPS with domain:

- You need to **create** or buy a valid TLS certificate for your domain endpoint.
- After you get SSL certificate for your domain you must put it to `/etc/nginx/ssl/fullchain.pem` as certificate and `/etc/nginx/ssl/privkey.pem` as a private key.
- Change `server_name _;` in configuration `/etc/nginx/conf.d/cloudbeaver.conf` to `server_name <your-domain>;`
- Enter in terminal `sudo systemctl reload nginx.service` to reload Nginx proxy
- Now you can open your `CloudBeaver Server` from the browser using your domain address.

Tip

If you need to change Java security properties, for example, to enable TLS 1.0, see [how to change Java security properties](#) for details.

Create self-signed certificate

Self-signed certificates are considered insecure for the Internet. Firefox will treat the site as having an invalid certificate, while Chrome will act as if the connection was plain HTTP

You can create self-signed certificate for `<your-domain>` by running the following script in the terminal:

```
SECRET_CERT_CSR="/C=US/ST=NY/L=NYC/O=CloudBeaver /OU=IT Department/CN=<your-domain>"
cd /etc/nginx/
mkdir ssl
cd ssl
sudo openssl req -x509 -sha256 -nodes -days 36500 -subj "$SECRET_CERT_CSR"
-newkey rsa:2048 -keyout privkey.pem -out fullchain.pem
```

How to change Java security properties

Java security settings can be customized by overriding the default security properties. This is useful when enabling legacy protocols (e.g., TLS 1.0), modifying cryptographic algorithms, or adjusting security policies.

Warning

Re-enabling deprecated security features (e.g., TLS 1.0, weak ciphers) exposes your system to known vulnerabilities. Only do this if required for compatibility with legacy systems, and consider upgrading your security infrastructure instead.

Steps to override Java security properties

1. Create a new file, e.g., `java_custom.secure.properties`, to override default security settings.
2. Modify the required security properties. For example, to enable TLS 1.0 (which is disabled by default), remove `TLSv1` from `jdk.tls.disabledAlgorithms` value:

- Default setting (TLS 1.0 is disabled):

```
jdk.tls.disabledAlgorithms=SSLv3, TLSv1, TLSv1.1, DTLSv1.0, RC4, DES, \
MD5withRSA, DH keySize < 1024, EC keySize < 224, 3DES_EDE_CBC, anon, NULL, \
ECDH
```

- Updated setting (TLS 1.0 is enabled):

```
jdk.tls.disabledAlgorithms=SSLv3, TLSv1.1, DTLSv1.0, RC4, DES, \
MD5withRSA, DH keySize < 1024, EC keySize < 224, 3DES_EDE_CBC, anon, NULL, \
ECDH
```

You can modify other properties in the same way.

3. Map the file in the container.
 - Ensure the custom file is accessible inside the container where Java runs. In a Dockerized environment, map it to a known location.
 - Update your `docker-compose.yml` file with the following lines to place the file inside the `cloudbeaver` container:

```
yaml
volumes:
  - ./java_custom.secure.properties:/location//java.secure.properties
```

environment:

- JAVA_OPTS="-Djava.security.properties=/location/with/java.secure.properties"

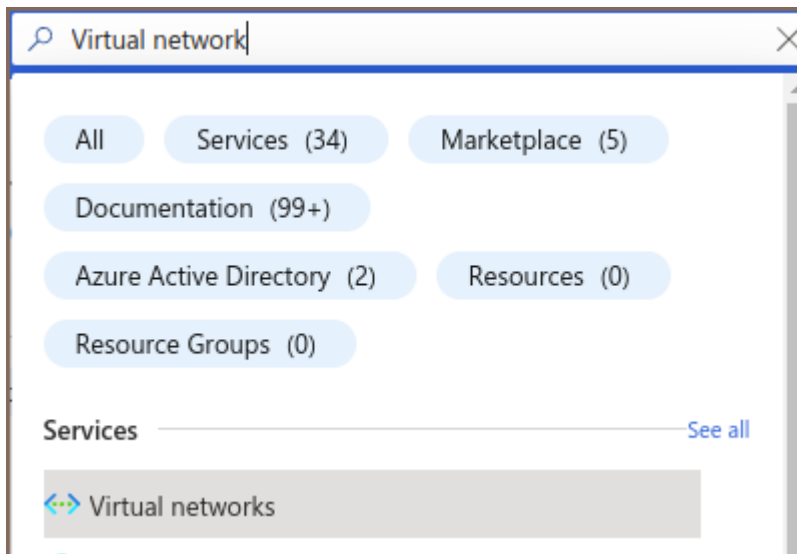
4. Restart the CloudBeaver to apply the changes.

How to connect CloudBeaver to a database on a separate machine in Azure

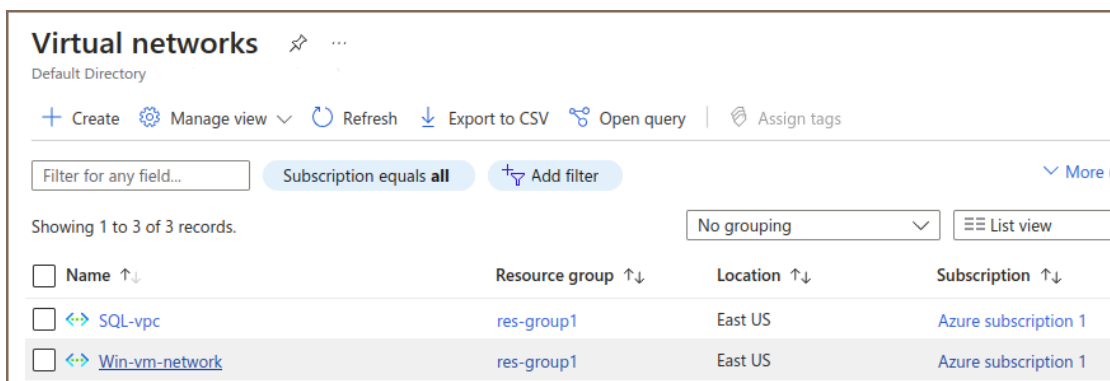
Connect your VM with CloudBeaver to PostgreSQL

Configure a peering for your networks

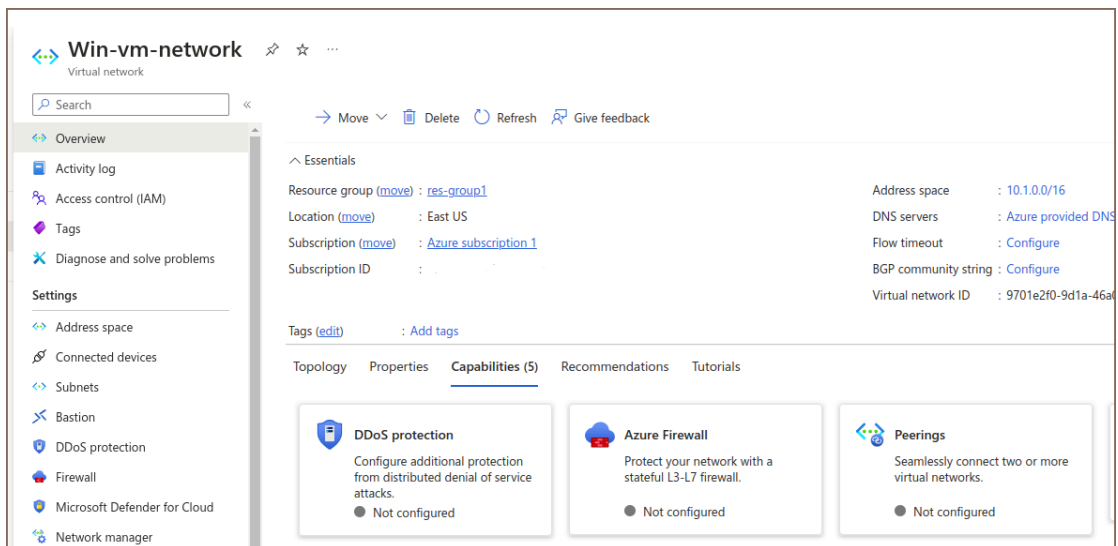
1. Log in to your account in **Microsoft Azure** and go to **Virtual networks**.



2. Select the network that you created during the VM startup process.



3. Select **Overview** -> **Capabilities** -> **Peerings** and click **Add peering**.



4. The Add peering window will be opened.
5. In the Peering link name field add the name for the new peering.
6. In the Virtual network field select the network where your database is located (SQL-vpc in this case).

Home > Virtual networks > Win-vm-network | Peerings >

Add peering

Win-vm-network

This virtual network

Peering link name *

win-vm-to-sql-peering ✓

☒ Allow 'Win-vm-network' to access the peered virtual network ⓘ

☐ Allow 'Win-vm-network' to receive forwarded traffic from the peered virtual network ⓘ

☐ Allow gateway in 'Win-vm-network' to forward traffic to the peered virtual network ⓘ

☐ Enable 'Win-vm-network' to use the peered virtual networks' remote gateway ⓘ

Remote virtual network

Peering link name *

win-vm-to-sql-peering ✓

Virtual network deployment model ⓘ

☒ Resource manager

☐ Classic

☐ I know my resource ID ⓘ

Subscription * ⓘ

Azure subscription 1 ▾

Virtual network * ⓘ

workers-vnet

SQL-vpc

Win-vm-network

☐ Enable the peered virtual network to use 'win-vm-network's' remote gateway ⓘ

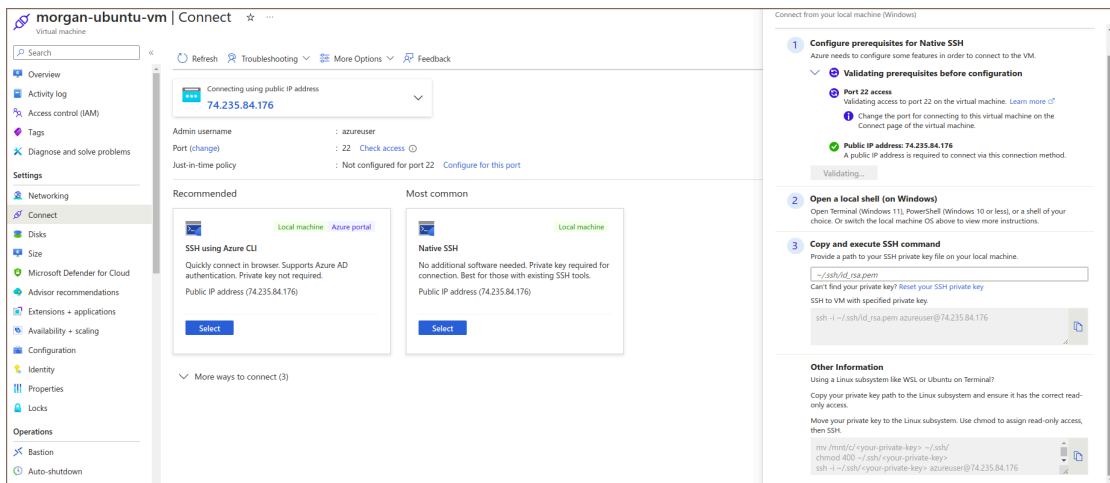
7. Click **Create**. The peering is configured.

Run CloudBeaver on the Linux VM and connect it to PostgreSQL

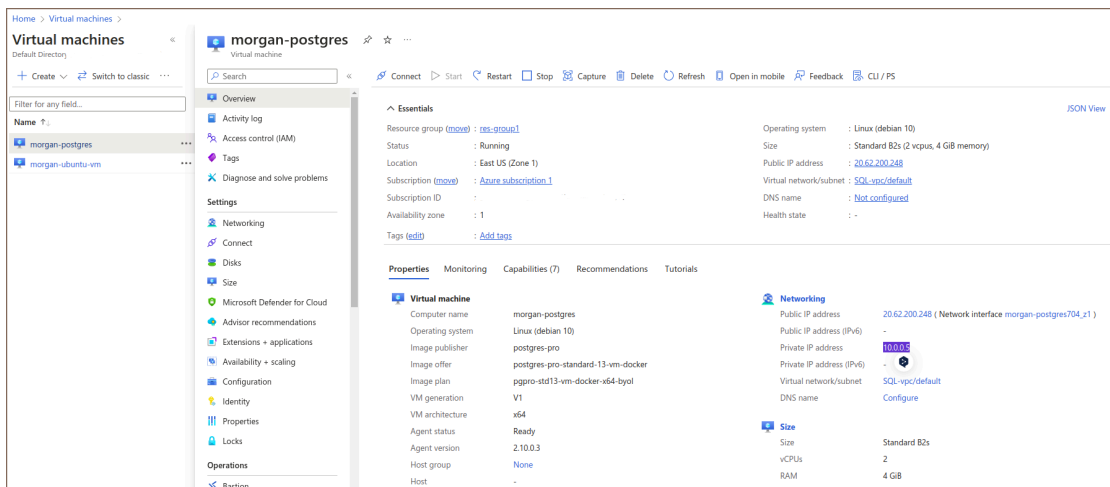
Step 1. Running CloudBeaver

1. Connect to your Linux VM via SSH:

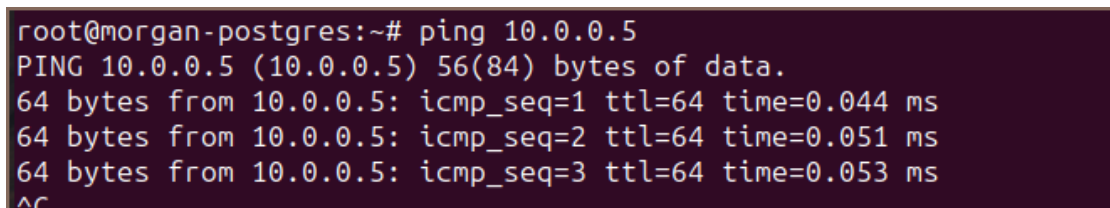
```
ssh -i ~/.ssh/<your_pub_key> <username>@<your_ip>
```



2. Check that your Linux VM has access to your SQL DB private IP.
3. Copy the private IP address of your database in **Virtual machines** -> **Your SQL DB** -> **Properties** -> **Networking**.



- Ping this IP in the terminal:



If everything was done correctly in the previous steps, you will get the result as in the screenshot above. All that remains is to connect our CloudBeaver server to your database.

4. Install Docker if it is not installed. Check this [documentation](#) for details.

5. Run CloudBeaver with this command:

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

```
root@morgan-ubuntu-vm:~# docker run -d --name cloudbeaer-ee --rm -ti -p 8978:8978 -v /var/cloudbeaver/workspace:/opt/cloudbeaver/workspace dbeaver/cloudbeaver-ee:latest
e5227a8cf91a4fd70b62959a964d8eb7d46e6e2243dcd9a823b217d064a3d83f
root@morgan-ubuntu-vm:~# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
e5227a8cf91a   dbeaver/cloudbeaver-ee:latest       "/run-server.sh"        19 seconds ago Up 19 seconds  0.0.0.0:8080->8978/tcp, :::8080->8978/tcp   cloudbeaer-ee
root@morgan-ubuntu-vm:~#
```

You can learn more about CloudBeaver deployment in our [documentation](#).

Step 2. Configure your network to access CloudBeaver

You need to make small changes to the network configuration so that you can access CloudBeaver UI.

1. Go to Virtual machines -> Networking -> Inbound port rules -> Add inbound port rule

The screenshot shows the 'Networking' page for a virtual machine named 'morgan-ubuntu-vm461_x1'. The 'Inbound port rules' tab is selected, displaying a table of existing rules. A 'Add inbound port rule' button is visible in the top right corner of the rules table.

Priority	Name	Port	Protocol	Source	Destination	Action
300	SSH	22	TCP	Any	Any	Allow
65000	AllowVnetInbound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInbound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInbound	Any	Any	Any	Any	Deny

2. Add access to port 8080. For security reasons, we recommend specifying your IP or the IP of your proxy in the resource to avoid third-party access.



Add inbound security rule



morgan-ubuntu-vm-nsg

Source ⓘ

My IP address



Source IP addresses/CIDR ranges ⓘ

134.209.249.195

Source port ranges * ⓘ

*

Destination ⓘ

Any



Service ⓘ

Custom



Destination port ranges * ⓘ

8080

Protocol

☒ Any

☐ TCP

☐ UDP

☐ ICMP

Action

☒ Allow

☐ Deny

Priority * ⓘ

310

Name *

AllowUIAccess



Description

Add

Cancel

Give feedback

3. Open in browser `http://<your_vm_ip>:8080`

That's all done! You are in the CloudBeaver interface.

Step 3. Configure CloudBeaver and connect to PostgreSQL

1. Enter your license key and configure your server.

The screenshot shows the 'INITIAL SERVER CONFIGURATION' window with a sidebar on the left containing 'Welcome', 'License', 'Server configuration', and 'Confirmation'. The main area is titled 'Main server configuration' and contains several sections for configuring the server:

- SERVER INFORMATION:** Includes fields for 'Server Name *' (pre-filled with 'CloudBeaver EE Web Server'), 'Server URL *' (pre-filled with 'http://74.235.84.176:8080'), and 'Session lifetime, min *' (pre-filled with '30').
- CONFIGURATION:** Includes checkboxes for 'Enable private connections', 'Navigator simple view', and 'Enable Resource Manager'.
- AUTHENTICATION SETTINGS:** Includes a checkbox for 'Allow anonymous access' and a radio button for 'Local' authentication.
- ADMINISTRATOR CREDENTIALS:** Includes a 'Login *' field pre-filled with 'cbeadmin'.
- SECURITY:** Includes a checkbox for 'Save credentials'.
- DISABLED DRIVERS:** Includes a search bar and a list of drivers: 'SQLite', 'H2 Embedded', and 'H2 Embedded V2'.

2. Go to `Connection Templates`, click `Add`, and type your database name (PostgreSQL for example).
3. In the `Host` field enter the IP address of your database you received earlier and indicate the authorization data for your database.

OPTIONS DRIVER PROPERTIES SSH TUNNEL SSL ACCESS CANCEL TEST CREATE

Driver PostgreSQL Configuration Manual ☒ URL ☐

Host * 10.0.0.5 **Port** 5432

Database postgres

Authentication Database Native

User name postgres **User password**

☐ Save credentials

Connection name * PostgreSQL@10.0.0.5

Description

Advanced Settings

▼ MISC

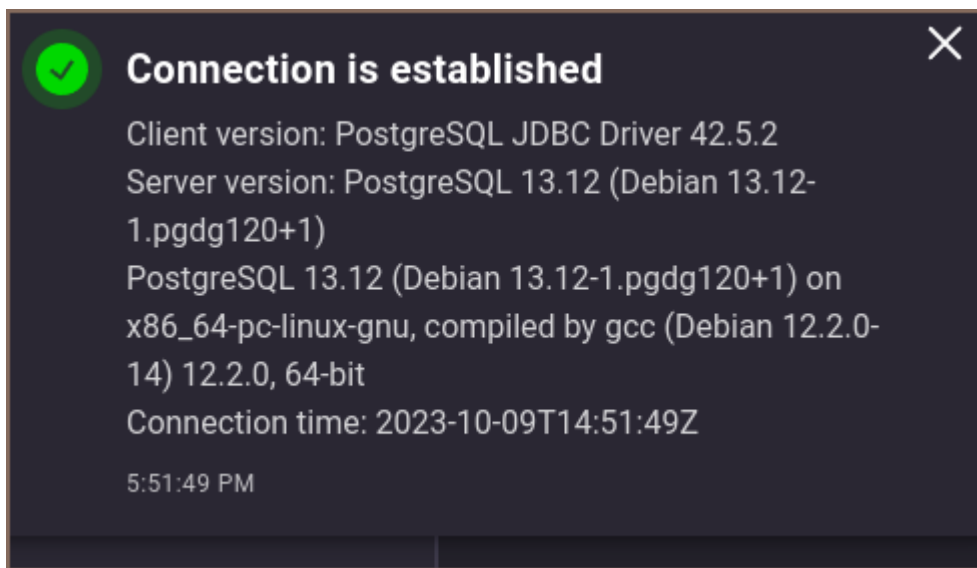
User role

> DATABASE LIST

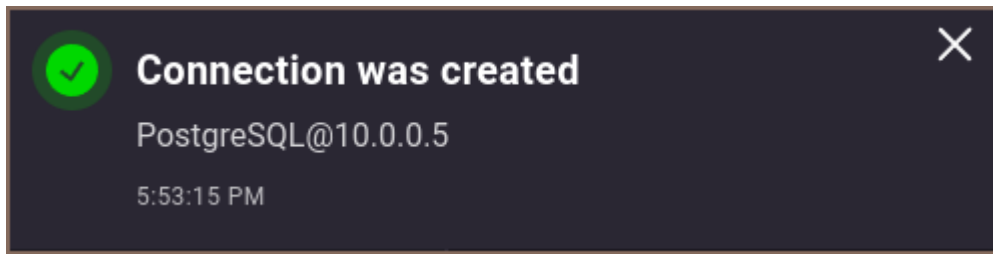
> SQL

☐ CONNECTION NAME ADDRESS

4. Click **TEST** to check that your configuration works.

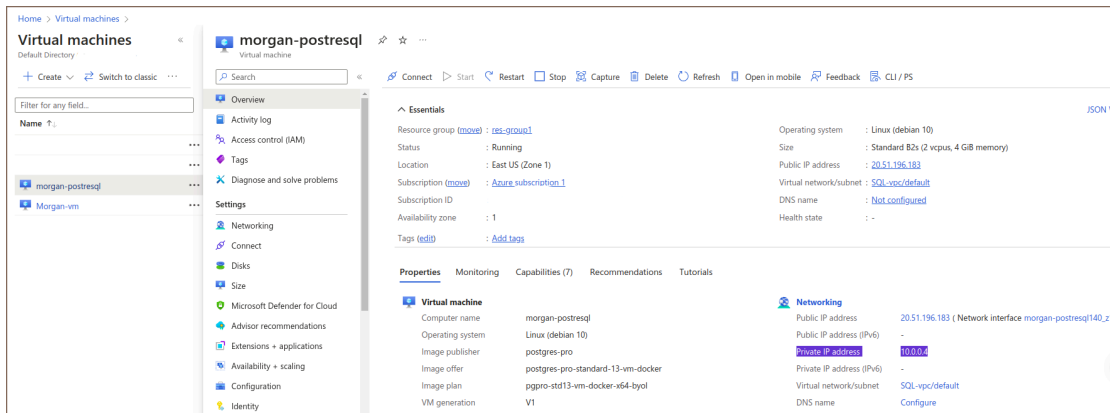


5. Perfect! Now click **CREATE** in the top right. Connection to your database is added to CloudBeaver.

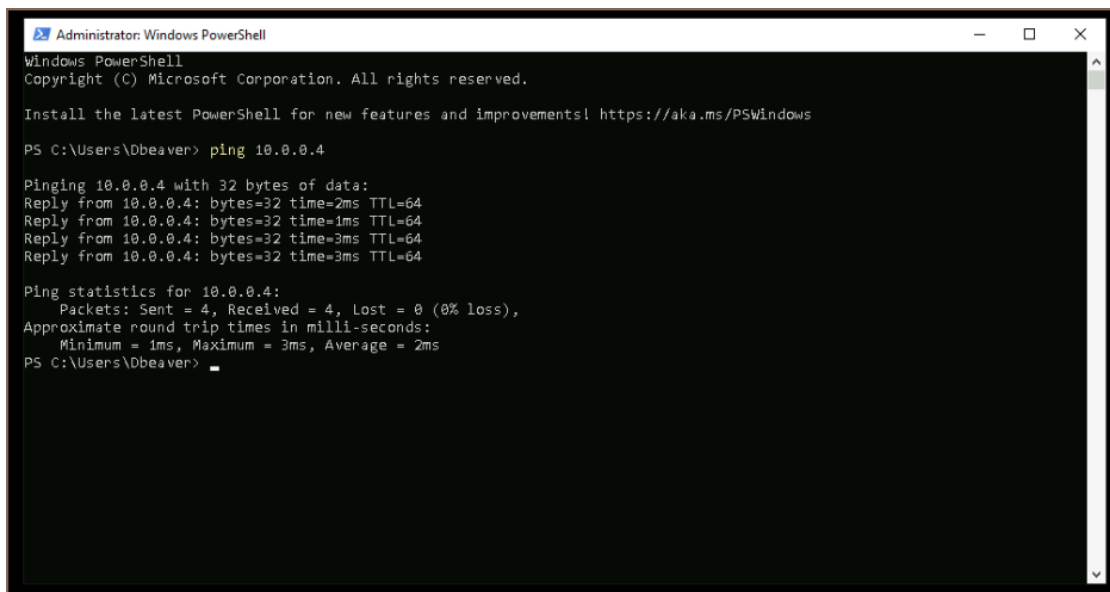


Connect DBeaver from Win VM to PostgreSQL

1. Connect to your Windows VM by using RDP.
2. Check that your Linux VM has access to your SQL DB private IP.
 - Copy the private IP address of your database in **Virtual machines** -> **Your SQL DB** -> **Properties** -> **Networking**.



- Ping this IP in the terminal:

A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The window has a black background with white text. It displays the standard PowerShell startup messages, including the copyright notice for Microsoft Corporation and a link to update PowerShell. The user, identified as "Dbeaver", has entered the command "ping 10.0.0.4". The output shows four successful replies from 10.0.0.4 with varying round-trip times (2ms, 1ms, 3ms, 3ms) and a TTL of 64. Ping statistics are also shown, indicating 4 packets sent, 4 received, and 0% loss, with an average round-trip time of 2ms.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

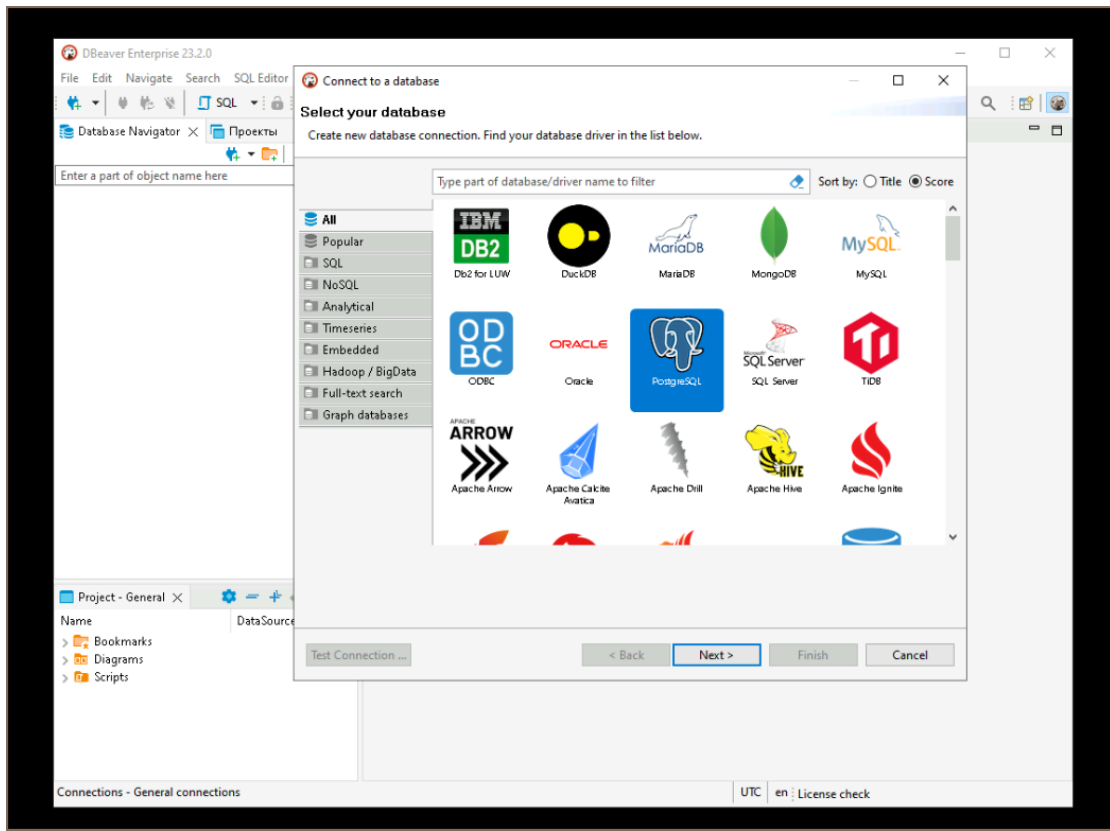
PS C:\Users\Dbeaver> ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=2ms TTL=64
Reply from 10.0.0.4: bytes=32 time=1ms TTL=64
Reply from 10.0.0.4: bytes=32 time=3ms TTL=64
Reply from 10.0.0.4: bytes=32 time=3ms TTL=64

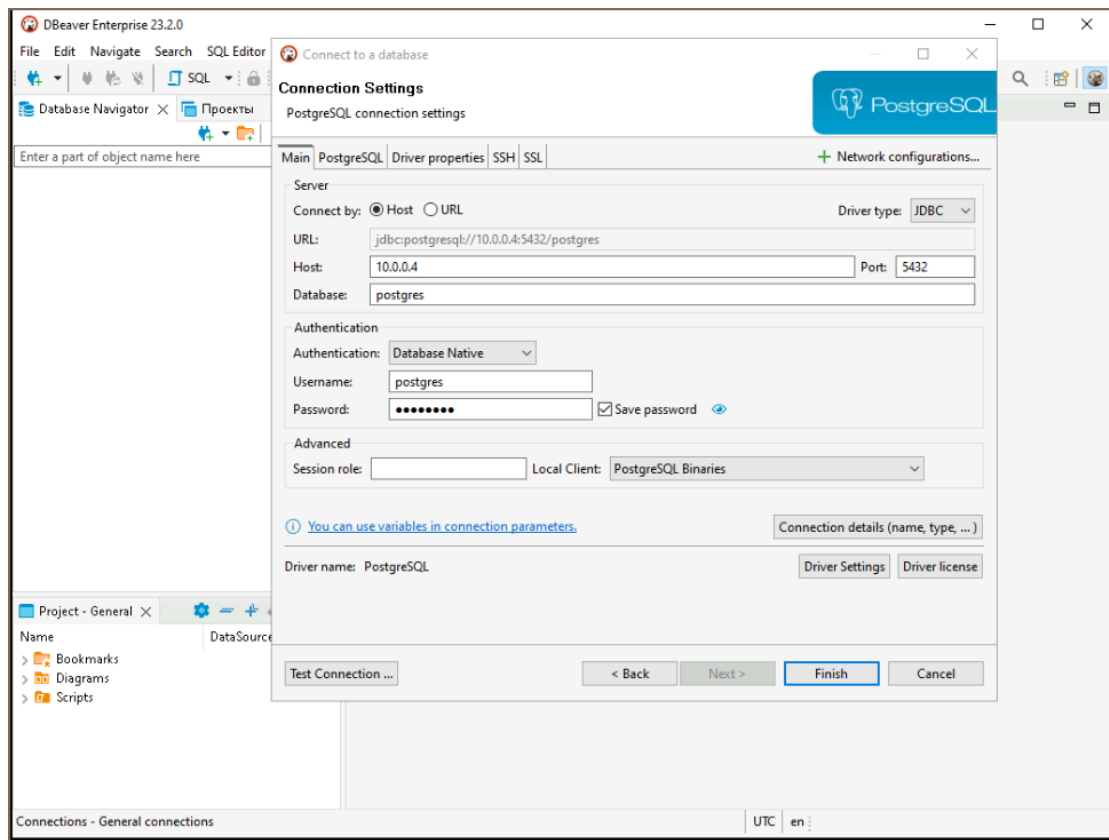
Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 3ms, Average = 2ms
PS C:\Users\Dbeaver> _
```

If everything was done correctly in the previous steps, you will get the result as in the screenshot above. All that remains is to connect our DBeaver to your database.

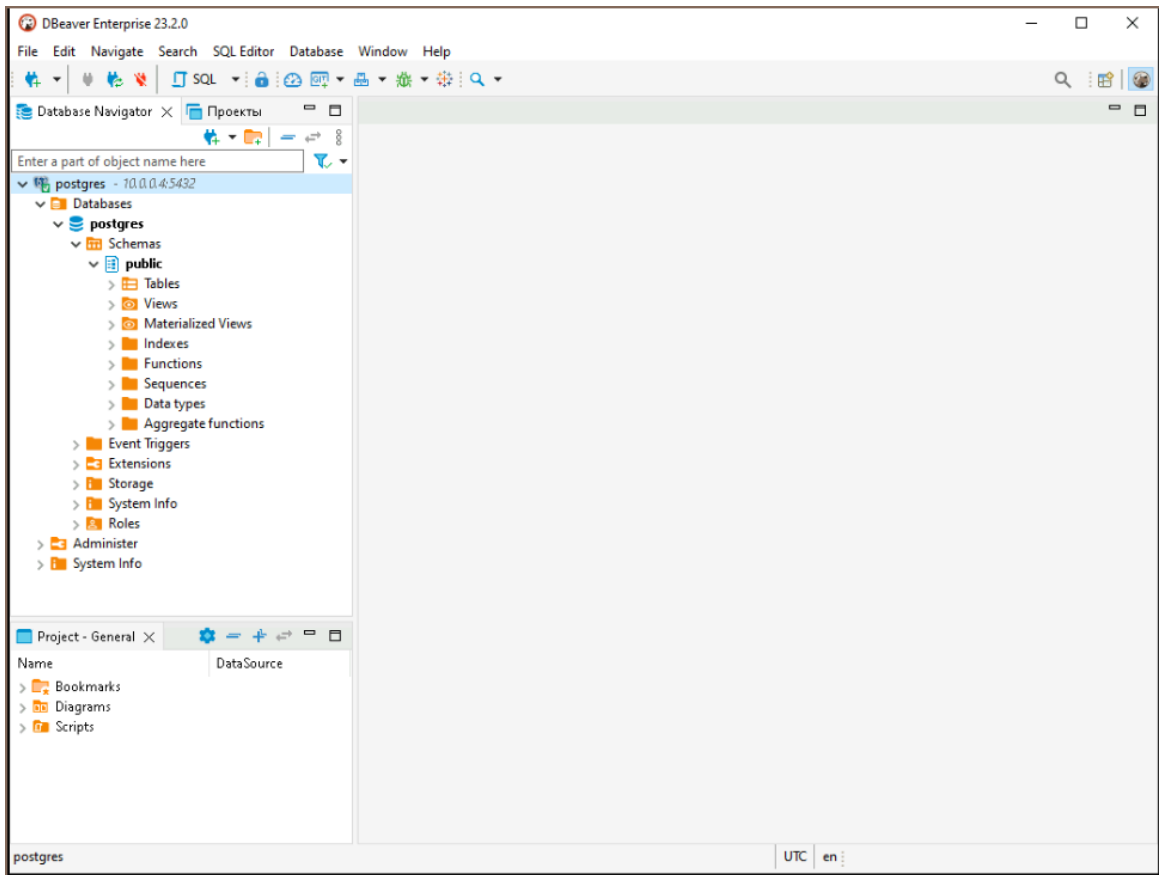
3. Add your license when starting the application.
4. Create a new connection and choose your database (for example, PostgreSQL).



5. In the **Host** field, specify the Private IP address of your database and fill in other fields.



Perfect! Your connection is ready.



Configuring HTTPS for Jetty server

This guide walks you through creating and configuring SSL certificates, setting up Jetty server configuration, and running your application with SSL enabled.

Important

You must replace `{ . . }` blocks with your own values.

Configuration steps

1. Open the terminal and navigate to the workspace directory
2. Type the following commands:
 - a. `openssl genrsa -des3 -passout pass:1 -out {your domain}.pass.key 2048`
 - b. `openssl rsa -passin pass:1 -in {your domain}.pass.key -out {your domain}.key`
 - c. `rm {your domain}.pass.key` (or `del {your domain}.pass.key` on Windows)
 - d. `openssl req -key {your domain}.key -sha256 -new -out {your domain}.csr`
 - e. `openssl x509 -req -days 3650 -in {your domain}.csr -signkey {your domain}.key -out {your domain}.crt`
 - f. `openssl pkcs12 -export -in {your domain}.crt -inkey {your domain}.key -out {your domain}.p12 -name {your domain} -passout pass:{your password}`
 - g. `keytool -importkeystore -deststorepass {your password} -destkeypass {your password} -destkeystore {your domain}.keystore -srckeystore {your domain}.p12 -srcstoretype PKCS12 -srcstorepass {your password} -alias {your domain}`
3. Create a new file called `ssl-config.xml` in the `.data` directory inside the workspace with the following content:

```
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN"
"http://www.eclipse.org/jetty/configure_10_0.dtd">
<Configure id="Server" class="org.eclipse.jetty.server.Server">
  <New id="httpConfig"
class="org.eclipse.jetty.server.HttpConfiguration">
    <Set name="sendServerVersion">false</Set>
    <Set name="sendDateHeader">false</Set>
  </New>

  <Call name="addBean">
    <Arg>
```

```

        <New id="sslContextFactory"
class="org.eclipse.jetty.util.ssl.SslContextFactory$Server">
        <Set name="keyStorePath">
            {Full path to your keystore. Example: /opt/cloudbeaver/
workspace/cb_keys/domain.test.keystore}
        </Set>
        <Set name="keyStorePassword">
            {The password you specified when creating certificates}
        </Set>
        <Set name="trustStorePath">
            {Full path to your keystore example: /opt/cloudbeaver/
workspace/cb_keys/domain.test.keystore}
        </Set>
        <Set name="trustStorePassword">
            {The password you specified when creating certificates}
        </Set>
        <Set name="IncludeProtocols">
            <Array type="String">
                <Item>TLSv1.2</Item>
            </Array>
        </Set>
        <Set name="IncludeCipherSuites">
            <Array type="String">
                <Item>TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384</Item>
                <Item>TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384</Item>
            </Array>
        </Set>
        <New id="tlsHttpConfig"
class="org.eclipse.jetty.server.HttpConfiguration">
            <Arg>
                <Ref refid="httpConfig" />
            </Arg>
            <Call name="addCustomizer">
                <Arg>
                    <New
class="org.eclipse.jetty.server.SecureRequestCustomizer">
                        <Set name="sniHostCheck">false</Set>
                    </New>
                </Arg>
            </Call>
        </New>
    </New>
</Arg>
</Call>

<Call id="sslConnector" name="addConnector">

```

```

        <Arg>
            <New class="org.eclipse.jetty.server.ServerConnector">
                <Arg name="server">
                    <Ref refid="Server"/>
                </Arg>
                <Arg name="factories">
                    <Array
type="org.eclipse.jetty.server.ConnectionFactory">
                        <Item>
                            <New
class="org.eclipse.jetty.server.SslConnectionFactory">
                                <Arg name="next">http/1.1</Arg>
                                <Arg name="sslContextFactory">
                                    <Ref refid="sslContextFactory"/>
                                </Arg>
                            </New>
                        </Item>
                        <Item>
                            <New
class="org.eclipse.jetty.server.HttpConnectionFactory">
                                <Arg name="config">
                                    <Ref refid="tlsHttpConfig"/>
                                </Arg>
                            </New>
                        </Item>
                    </Array>
                </Arg>
                <Set name="port">
                    8978
                </Set>
                <Set name="idleTimeout">
                    <Property name="jetty.idleTimeout" default="30000"/>
                </Set>
            </New>
        </Arg>
    </Call>
</Configure>

```

4. Start the application using the following command:

- In docker: `docker run --name={container name} -p 8978:8978 -ti -v {absolute path to workspace}:/opt/cloudbeaver/workspace dbeaver/cloudbeaver-ee:{container name}`
- From sources: `./run-cloudbeaver-server.sh`

Managing cloud deployed services

After launching your instance:

Access your instance

1. In your browser, go to `http://<Instance_IP>/` , where `<Instance_IP>` is the public IP address of your instance.

Tip

If you're unsure how to find your instance's public IP, refer to the relevant cloud provider's documentation.

2. You'll be redirected to CloudBeaver's initial configuration wizard to complete the setup.

API and Integration

API and Integration overview

Integration options

CloudBeaver supports several ways to integrate with external tools and automate workflows.

GraphQL API

Use GraphQL to interact with the server using flexible queries and mutations. Supports operations like user and session management, SQL execution, and more.

Learn more about [GraphQL API](#)

JDBC driver

Use the DBeaver Proxy Driver to let third-party tools connect to databases managed in CloudBeaver.

Learn more about the [JDBC driver](#)

GraphQL API

GraphQL API overview

CloudBeaver offers a powerful web API through the GraphQL protocol, enabling advanced server interactions and data manipulation.

Authentication

GraphQL API uses tokens for authentication.

Info

See [how to get and use a token](#).

API documentation

For a full list of available queries, mutations, and types, see the [GraphQL API documentation](#).

Console

The built-in GraphQL console lets you explore the schema, run queries, and test interactions directly in the browser.

Info

Read more about [GraphQL console](#).

CLI

Use the built-in server CLI to run a single GraphQL operation from a file - perfect for automation, CI/CD, or headless environments.

Info

Read more about [running GraphQL with the CLI](#).

Examples

See usage examples in the [cloudbeaver-graphql-examples](#) repository.

You'll find queries for signing in, connecting to databases, running SQL scripts, and managing sessions.

Generate API token

Note

This feature is available in **Enterprise**, **AWS**, and **Team** editions only.

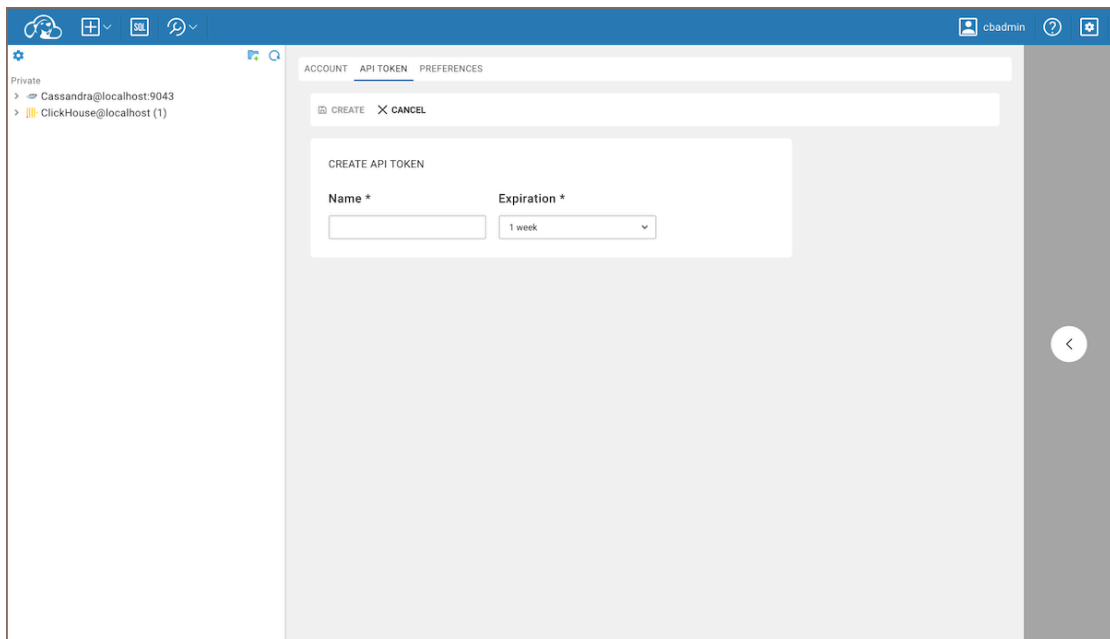
The CloudBeaver allows you to generate API tokens for automating tasks and interacting with the **GraphQL API**.

Note

Each user must create their own tokens.

How to generate API token

1. Enable the **API token** switch in **Settings -> Administration -> Server configuration -> Authentication settings**.
2. Navigate to your **Profile** page in the CloudBeaver.
3. Open the **API Token** tab.



4. Click **+ Add**.

Note

Users can create up to 5 tokens.

5. Choose **Name** and **Expiration** period. You can choose from options like 1 day, 1 week, 1 moth, 1 year, or no expiration (not recommended for security reasons).
6. Copy the token after it's generated.

Important

Tokens are displayed only once during creation.

How to delete API token

1. On the **API Token** tab.
2. Find the token you want to delete and click the **Delete** icon.
3. Confirm the deletion in the confirmation dialog.

Note

Deleting a token doesn't immediately end an active session. Users must log in again for the deletion to take effect.

Using the API token with GraphQL

Authenticate with the **GraphQL API console** using your token by including it in the query.

- **Example query:**

```
query authLogin($credentials: Object) {  
  authLogin(  
    provider: "token"  
    credentials: $credentials  
  ) {  
    authId  
    userTokens {  
      userId  
    }  
    authStatus  
  }  
}
```

- Variables:

```
{
  "credentials": {
    "token": "your_api_token_here"
  }
}
```

- Example response:

The screenshot shows the GraphQL IDE interface. At the top, there are tabs for 'authLogin' (selected) and 'userConnections'. The main editor on the left contains a query:

```
1 query authLogin($credentials: Object) {
2   authLogin(
3     provider: "token"
4     credentials: $credentials
5   ) {
6     authId
7     userTokens {
8       userId
9     }
10    authStatus
11  }
12 }
```

Below the query editor, there are tabs for 'Variables' and 'Headers'. The 'Variables' tab is active, showing the following JSON:

```
1 {
2   "credentials": {
3     "token": "your_api_token_here"
4   }
5 }
```

On the right side of the IDE, the JSON response is displayed:

```
{
  "data": {
    "authLogin": {
      "authId": null,
      "userTokens": [
        {
          "userId": "cbadmin"
        }
      ],
      "authStatus": "SUCCESS"
    }
  }
}
```

GraphQL console

The GraphQL console in CloudBeaver lets you test requests directly from your browser. To open the console, go to the following URL:

Use GraphQL console

```
https://${serverAddress}/api/gql/console
```

Replace `${serverAddress}` with the actual server host name.

Important

GraphQL console is available only when the server is running in **Development mode**. Make sure `develMode` is set to `true` in your configuration to enable access. For information on configuring these variables, see the [Server configuration](#) guide.

You can explore public GraphQL API console on our [demo server](#).

Tip

Run GraphQL requests without the browser using the built-in server CLI. It executes a single operation from a JSON file - useful for automation, CI/CD, or headless use. Read more about [running GraphQL with the CLI](#).

Run GraphQL requests from CLI

Run a single GraphQL script from a file using the built-in CLI in CloudBeaver

Prerequisites

- make sure the server container is running
- have an [API access token](#)
- prepare a file with exactly one GraphQL operation, save it in any format (such as `.json` or `.txt`), and make sure the content is valid `JSON`

Tip

with Docker Compose, either mount the script file or a folder of scripts into the service container at startup, or copy the file into the running container.

Write a GraphQL script file

Create a JSON file with `query`, `variables`, and `operationName`. For schema details and more operations, see [GraphQL API overview](#).

Example

```
{
  "query": "mutation createConnection($projectId: ID!, $config:
ConnectionConfig!) { connection: createConnection(projectId: $projectId,
config: $config) { ...DatabaseConnection } } fragment DatabaseConnection on
ConnectionInfo { id projectId name }",
  "variables": {
    "projectId": "g_GlobalConfiguration",
    "config": {
      "configurationType": "MANUAL",
      "credentials": { "userPassword": "qwe", "userName": "qwe" },
      "mainPropertyValues": { "host": "localhost", "port": "3306" },
      "networkHandlersConfig": [],
      "providerProperties": {},
      "driverId": "mysql-ee:mysql8",
      "name": "MySQL@localhost",
      "host": "localhost",
      "port": "3306",
      "databaseName": null,
      "authModelId": "native",
      "properties": {}
    }
  },
  "operationName": "createConnection"
}
```

For more ready-to-run examples, see the [CloudBeaver graphql examples](#) repository.

Enter the running container

Open a shell in the Compose service container:

```
docker compose exec <service_name> bash
```

List services, and get the container id for a service:

```
docker compose ps
docker compose ps -q <service_name>
```

Copy your script from the host into the service container:

```
docker cp /path/on/host/create-connection.json $(docker compose ps -q
<service_name>):/<your_custom_directory>/create-connection.json
```

Tip

if you use the older v1 CLI, replace `docker compose` with `docker-compose`.

Execute the script

Run the command:

```
docker compose exec <service_name> ./cbvr.sh -gql \
-input-file /<your_custom_directory>/create-connection.json \
--access-token <api_token>
```

The command prints the GraphQL response to stdout in `JSON`, including `data`, and, if present, `errors`. If errors occur, the command returns a non-zero `exit code`.

Save output to a file

To save the GraphQL response to a file:

```
docker compose exec <service_name> ./cbvr.sh -gql \
-input-file /<your_custom_directory>/create-connection.json \
-output-file /opt/cloudbeaver/workspace/out.txt \
--access-token <api_token>
```

Command line parameters

Option / Command	Applies to	Description
<code>-gql</code>	GraphQL	Runs a GraphQL request from a file
<code>import-license</code>	License import	Imports a license file into CloudBeaver. Learn more

Option / Command	Applies to	Description
<code>--access-token <token></code> <code>/ -token <token></code>	GraphQL	API access token for authentication
<code>-input-file <path></code>	Both	Path to the input file inside the container (<code>.json</code> , <code>.txt</code> , or license file)
<code>-output-file <path></code>	GraphQL	Path to the file where the response is saved (optional). A safe writable path inside the container is <code>/opt/cloudbeaver/workspace/</code> .

Use in AWS

If you run CloudBeaver on AWS, do the same steps **inside the instance**: place the `JSON` file on the instance, then run the CLI there.

Info

For connection and file transfer basics, see AWS documentation:

- [connect to your Linux instance using SSH](#)
- [connect using EC2 Instance Connect](#)
- [transfer files with scp](#)

Exit codes

When CloudBeaver finishes executing a command, it returns an **exit code** to indicate the result.

Code	Name	Description
<code>-1</code>	<code>EXIT_CODE_CONTINUE</code>	Indicates that the process continues running. You won't normally see this code when running commands.
<code>0</code>	<code>EXIT_CODE_OK</code>	Command executed successful

Code	Name	Description
		ly.
1	EXIT_CODE_ERROR	A general error occurred during execution.
2	EXIT_CODE_ILLEGAL_ARGUMENTS	Invalid command line arguments were provided.

DBeaver Proxy driver

Proxy Driver

Note

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

DBeaver Proxy Driver is a JDBC driver that allows third-party applications to connect to databases configured in CloudBeaver. Instead of setting up multiple database drivers and handling complex authentication mechanisms, you can connect to databases through a single proxy driver.

It is useful for:

- Users of **BI tools** like Tableau, JasperReports, Metabase, or BIRT who need to access databases managed in CloudBeaver.
- Developers who want to connect external applications to CloudBeaver-hosted databases without direct access to database credentials.

Info

See the [Tableau connection example](#).

Download

The DBeaver Proxy Driver repository is available on [GitHub](#). You can download the latest version on the [releases page](#).

Set up DBeaver Proxy Driver

1. [Download](#) the driver.
2. Add the driver to your tool. You can configure the Proxy Driver in any JDBC-compatible tool by adding the `jar` files to the tool's driver settings.
3. As an administrator, enable the Proxy Driver in **Settings -> Administration -> Server Configuration -> DBeaver Proxy Driver**.
4. Generate an API access token. See [Token generation](#) for details.

Connecting to the database

To establish a connection, specify the following parameters:

JDBC URL

JDBC URLs for the DBeaver Proxy Driver follow this format:

```
jdbc:dbeaver-upd:<server-host>:<server-port>:<project>:<connection>
```

Where:

Placeholder	Description
<code><server-host></code>	The server hostname or IP address.
<code><server-port></code>	The server port.
<code><project></code>	The project name. In CloudBeaver, the only available project is <code>Shared</code> .
<code><connection></code>	The connection ID or the name of the database connection in CloudBeaver.

Info

To find the connection ID or name, right-click your database in **Database Navigator**, select **Open**, and view the **ID** and **Name** in the opened window.

Properties

API token authentication::

- `dbeaver.token` : The Proxy Driver requires an **Access token** for authentication.

User/password authentication::

- `dbeaver.user` : Specifies the username for authentication.
- `dbeaver.password` : Specifies the password for authentication.

Note

Database-specific driver properties appear only after establishing a connection.

Driver class name

Use the following driver class name - `com.dbeaver.jdbc.upd.driver.UPDDriver`.

Limitations

Currently, the driver supports only basic queries and database operations. It does not yet support:

- Opening more than five connections.
- Editing **date/time values**.
- Performing **dump/restore operations**.
- Switching **schemas**.
- Handling **special data types** (e.g., `timestamp`, `arrays`).
- Executing **complex queries** (e.g., `JOIN` operations).

Connect Tableau to a database using DBeaver Proxy Driver

Note

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

To connect [Tableau](#) to a database via the [DBeaver Proxy Driver](#), follow these steps:

Prerequisites

Before you start, ensure you have:

- A Tableau account.
- Tableau Desktop installed.
- DBeaver Proxy Driver enabled on your CloudBeaver server.
- An API access token generated. See [Token generation](#) for details.

Steps

1. In Tableau's main menu, go to **To a Server -> More**.
2. In the window that opens, select **Other Databases (JDBC)**.
3. Enter the [JDBC URL](#), choose the database **dialect**, and enter your database **username and password**. **Example JDBC URL:**

```
jdbc:dbeaver-upd:your-domain.com:443:Shared:postgres-  
jdbc-194f450b17f-63a1de2f34b44deb
```

4. To use the API access token, create a `.properties` file and add the following line:

```
dbeaver.token=value
```

5. Place the `.properties` file and all Proxy Driver `.jar` files in the `Tableau/Drivers` folder.

Note

For more details on using JDBC drivers in Tableau, see [Tableau JDBC documentation](#).

6. Click **Sign In** to establish the connection.

Licenses

Import license

Important

Licensing for **CloudBeaver AWS** is managed entirely through the AWS Marketplace.

To start using commercial versions of CloudBeaver you can:

- **buy** a subscription
- request a **Trial license** for 2 weeks
- request an **Academic license** if you are a student or a teacher

Info

You will receive a license text by email, and it will also be available in your personal account on our **website**. The license text contains your License ID (for example, DB-841MRZHY-ZH54), start date, and the license owner's name and company.

Import from email

1. Copy-paste the license key

Hello,

Thank you for your interest to DBeaver.

We are glad to see you among DBeaver users. Please, find your license below:

```
-- DBeaver EE LICENSE - DB-841MRZHY-ZH53
-- Starts at Tue Oct 11 10:14:33 UTC 2022 to Louis Anderson //
PK2uj9eCXU4Zmgd1wvMufQyC2Jk+cSTtePOXewaUoS2/jYvuaX1opFJ0G/H81sLTzWqSa81EsEN
BG3Z6TJ0VFPko5bvXkm2r0hRXjRusa18gM47snPK/NaN/7fg/CEzHhy3+NrstRtRqXwiBucs5b/c
lRgIZm3Z485qg8G1lV8K2gG99p8K0e690gTB6vWY1hECRfs3vdrGz4x5QHnQrNuIr1cVXhI8qbM5
l3L9/7e3QgBMQya+XF9yVeGiVRQGdQDP8GNwY1WMoHBzpGKuhmltwBj+jF7FyUAFw7Cw6jzAk0es
gb3Fg2REPkLwvIq2yxangUpnNGDUE1bH7VXoSw==
```

Enjoy your database research with DBeaver and good luck!

Regards,

DBeaver team

Note

You need to copy-paste the full license text (not just the license ID). The license text starts with “=” and ends with “==” characters.

2. **Insert the copied license key**

Import from the personal account

1. **Sign in**
2. Open the **Licenses** tab, where you can find all your licenses

Ben Johnson

ORDERS **LICENSES** ACCOUNT TICKETS

Licenses | Assignment

Search licenses...

Rows Per Page: 10 | 20 | 50 | 100

☐	License ID	Users	Product	Type	End Support Date	
			All	All	All	
☐	DB-9Q1MSZGF-ZDAM	5	DBeaver Enterprise Edition	Yearly subscription	Dec 6, 2024	≡
☐	DB-9Q1MSZGC-ZDAL	1	DBeaver Enterprise Edition	Trial	Dec 21, 2023	≡
☐	DB-9Q1MSZGB-ZDAK	1	DBeaver Enterprise Edition	Monthly subscription	Jan 7, 2024	≡

Bulk actions: **EXTEND** **GET LICENSE KEYS** **EXPORT INFO**

Export all licenses info as csv

3. Click the license ID link. Here you can find your license status, type, maintenance period, and end support date

Ben Johnson

[ORDERS](#)[LICENSES](#)[ACCOUNT](#)[TICKETS](#)

License ID:DB-9Q1MSZGF-ZDAM

License Type:Yearly subscription

Maintenance Period:365 days

Maximum users:5

Issue Time:Dec 7, 2023

Expiration Time:Dec 6, 2024

License Status:Valid

License customer:Ben Johnson, Test

License Keys:

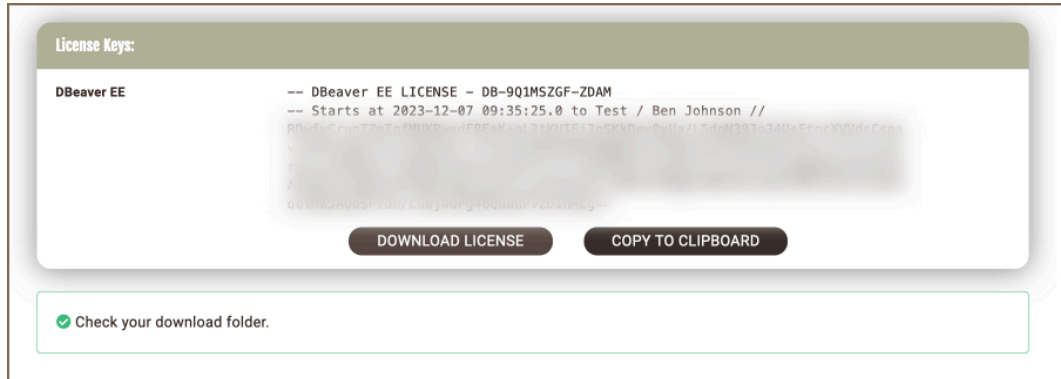
DBeaver EE-- DBeaver EE LICENSE - DB-9Q1MSZGF-ZDAM-- Starts at 2023-12-07 09:35:25.0 to Test / Ben Johnson //

DOWNLOAD LICENSECOPY TO CLIPBOARD

4. At the bottom of the page you can find the license key required to start using CloudBeaver. There are two options how to copy your license key from the personal account:
- press the **COPY TO CLIPBOARD** button, then press **OK**. The license text will be copied to the clipboard



- press the **DOWNLOAD LICENSE** button, then press **OK**. The `.txt` file with your license key will be downloaded to your download folder. The file name is License ID, e. g. `DB-841MRZHY-ZH54` .



5. Insert the copied license key

Activate license

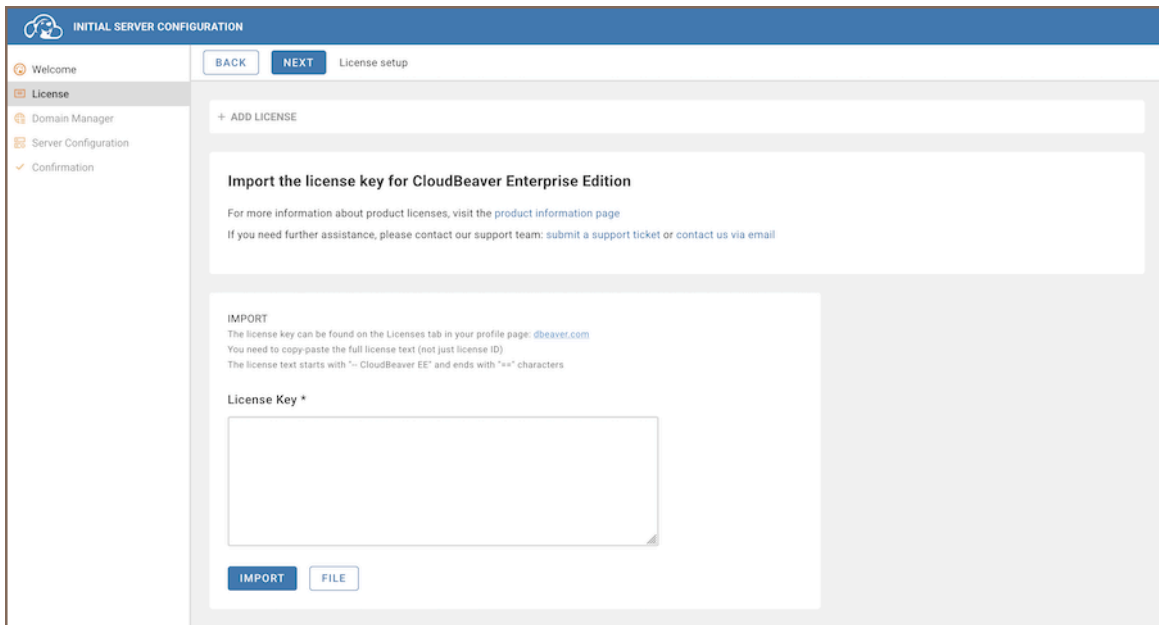
Install license during initial server configuration

The license is added during **initial server configuration**.

To install the license:

1. Open the license dialog.
2. Choose one of the import options:
 - Copy and paste the full license text (not just the ID). It starts with `--` and ends with `==` .
 - Use the file upload button to import the license from disk.

3. Click **Import** to apply the license.



The screenshot shows the 'INITIAL SERVER CONFIGURATION' page with a sidebar on the left containing 'Welcome', 'License', 'Domain Manager', 'Server Configuration', and 'Confirmation'. The 'License' section is active. The main content area has a 'License setup' header with 'BACK' and 'NEXT' buttons. Below is a '+ ADD LICENSE' button. A section titled 'Import the license key for CloudBeaver Enterprise Edition' provides instructions and a link to the product information page. An 'IMPORT' section follows, explaining that the license key is found in the user's profile page and must be copied in full. It includes a text area for the 'License Key *' and 'IMPORT' and 'FILE' buttons at the bottom.

Install license using environment variable

You can install the license without using the UI.

Warning

When you install the license using this method, it can't be changed from the UI. Any changes made in the UI will be ignored and overwritten by the mounted file after restart.

Set the `DBEAVER_LICENSE_CUSTOM_PATH` environment variable to the full path of the license file (including the file name). The license file must be available **inside the container** (for example, mounted with a volume).

For more information on setting environment variables, see [CloudBeaver configuration variables](#).

Install license using CLI

You can also install the license from the command line using the built-in CLI tool (`cbvr.sh`).

Tip

use this method when the UI isn't available, for example, in automated setups, AWS instances, or headless servers.

Run the following command inside the service container:

```
docker compose exec <service_name> ./cbvr.sh import-license \
-input-file /opt/cloudbeaver/<your_custom_directory>/CB-471MUZET-ZKLX.txt
```

This command installs the license file into the current workspace. The path must point to a file that exists **inside the container**.

Note

the `cbvr.sh import-license` command doesn't require an API token and can be run before the server is fully initialized.

Update license

Licenses expire after a set period. You may need to update yours if it has expired, or if you've received a new or upgraded license.

Use one of the methods below depending on your situation:

Renew license

- if the server has internet access, the license updates automatically after you upgrade it on dbeaver.com.
- if the server doesn't have internet access, open the [Administration panel](#) and re-install the updated license manually.
- if the license is installed via [environment variable](#), replace the license file at the specified path. The new license will be applied automatically after the next server restart.

Change license

If your license was updated:

1. Go to **Settings -> Administration -> Product Information**
2. Click + **Add license**
3. Paste or upload your new license

The previous license stays in the system but becomes inactive if the new one has a later issue date.

Tip

Use the same steps to replace a trial license with commercial license.

Reassign license

After purchasing a set of CloudBeaver licenses, you have to assign each license to an end user. If an employee leaves the company or the team that is using CloudBeaver, the license admin may need to reassign the license to another employee. You can reassign the license to another user in your personal account.

Steps to reassign

1. [Sign in](#).
2. Open the **Licenses** tab to view all your licenses.

Ben Johnson

ORDERS

LICENSES

ACCOUNT

TICKETS

Licenses

Assignment

Search licenses...

Rows Per Page: 10 | 20 | 50 | 100

☐	License ID	Users	Product	Type	End Support Date	
☐	DB-9Q1MSZGF-ZDAM	5	DBeaver Enterprise Edition	Yearly subscription	Dec 6, 2024	⋮
☐	DB-9Q1MSZGC-ZDAL	1	DBeaver Enterprise Edition	Trial	Dec 21, 2023	⋮
☐	DB-9Q1MSZGB-ZDAK	1	DBeaver Enterprise Edition	Monthly subscription	Jan 7, 2024	⋮

Bulk actions:

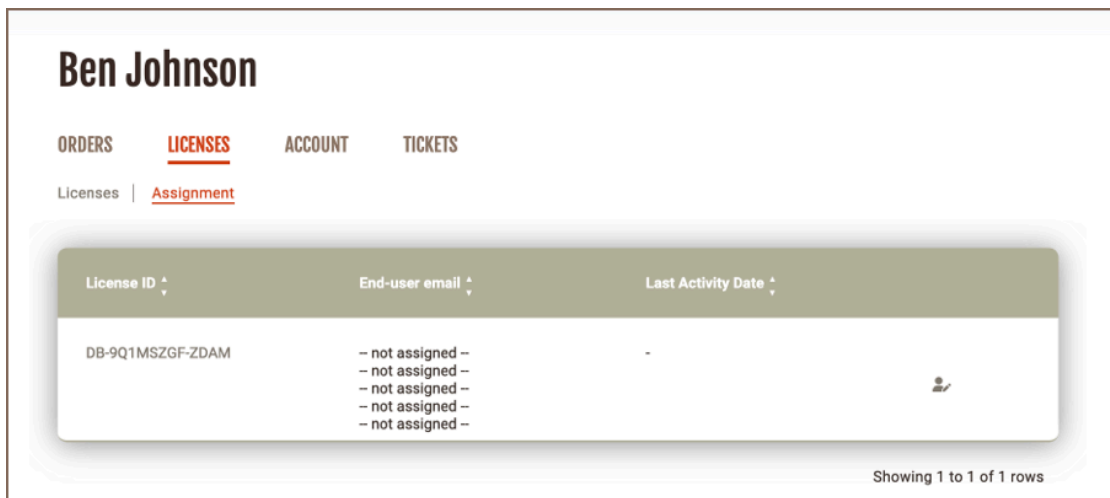
EXTEND

GET LICENSE KEYS

EXPORT INFO

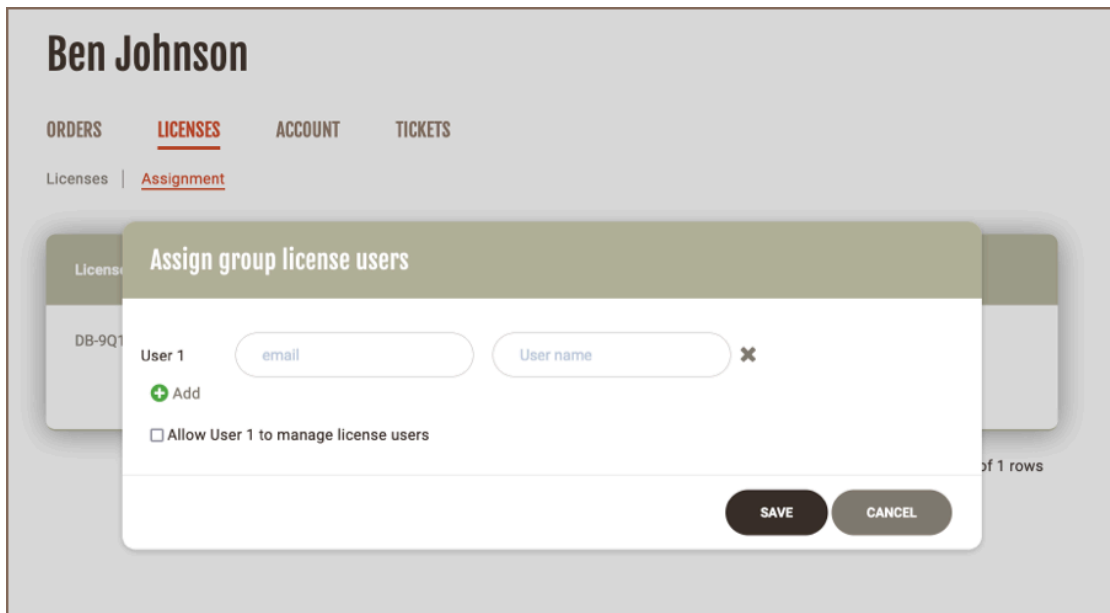
Export all licenses info as csv

3. Click the **Assignment** subtab to manage license end-users and track license usage.



4. Select the license you need to reassign and press the icon in the right column.
5. In the pop-up window, enter the new license end user's details:
 - Email
 - First and last name
 - Company

Then press **SAVE**. The license owner will be changed, and the license key (which contains the license end user's name and company) will be updated.



Note

When the license manager reassigns a license to another user, the license is completely recreated. The new user receives a new license, and the old license becomes invalid and is no longer visible in the customer's account.

Manage group licenses

If you have a group license, you can add users to it. To let another user manage the license, check **Allow User 1 to manage license users**. This user becomes a **license manager** and can add or remove users.

Info

You can assign multiple license managers for the same group license. Each of them will be able to manage the list of users independently.

The screenshot shows the 'Ben Johnson' account interface with tabs for ORDERS, **LICENSES**, ACCOUNT, and TICKETS. Under the 'LICENSES' tab, there is a sub-tab for 'Assignment'. A modal titled 'Assign group license users' is open, displaying a table with two rows for 'User 1' and 'User 2'. Each row has an 'email' input field and a 'User name' input field with a clear (X) button. Below the table is a '+ Add' button and a checkbox labeled 'Allow User 1 to manage license users'. At the bottom right of the modal are 'SAVE' and 'CANCEL' buttons. The background shows a list of licenses with one row visible, labeled 'DB-9Q1'.

The license stays in your account, but the assigned user's email appears in the **End user** column on the **Assignment** tab.

The new user will also see the license in their account and can **import it**. They can't change any license details.

If the new user doesn't have an account yet, it's created automatically. They'll get an email with a link to set a password, and the previous user will get an email that the license was removed.

Import users from CSV

You can also upload a list of end users for a group license using a CSV file. Use this method when you need to assign many users at once.

To do this:

1. Go to your **Assignment** tab
2. Click **Import from CSV**
3. Upload your `CSV` file containing the user details

Each line in the CSV represents a user to be assigned to the group license.

Example

```
alice@dbeaver.com;Alice Cooper  
user-bob;Bob Bundy
```

License types

License types

CloudBeaver offers different license types Use the table below to compare license types, support availability, and limitations.

License comparison

	Single-server	Multi-server	AWS Marketplace	Academic	Trial
Product availability	Enterprise	Enterprise	AWS	Enterprise	Enterprise
Version updates	Included during subscription	Included during subscription	Included (billed via AWS)	Included for 1 year	Full access to all features during the 14-day trial
Duration of access	1 year	1 year	Managed by AWS (hourly or annual billing)	1 year	14 days
Access blocked	After expiration or failed renewal	After expiration or failed renewal	Managed by AWS		Access is blocked after the trial period is finished
Renewal	Annual renewal	Annual renewal	Managed by AWS	Must request new license after expiration	Not renewable
Support	While the subscription is active	While the subscription is active	While the subscription is active	Included while the trial period is active	Not included

	Single-server	Multi-server	AWS Marketplace	Academic	Trial
Offline use	Yes	Partially supported Requires internet for user sync or manual reimport	Not supported	Yes	Yes
User management	Managed on-premises or in cloud	Managed via dbeaver.com plus UI/SSO if seats are available	Managed by AWS	Manual	Fixed: 5 users
Limitations	Per named user, per server	Per named user, across multiple servers	Per instance, AMI-based licensing	Academic use only	Limited to 5 users

Info

To check your current license, go to [dbeaver profile](#).

Multi server license

Multi-server licenses are designed for teams that need to run CloudBeaver on more than one server.

Use cases

- several independent environments (production, staging, test)
- distributed infrastructure
- high availability setups with standby servers

How it works

A multi-server license covers a fixed number of servers and named users under one subscription. All users must be listed in the license, even if only some of them connect to multiple servers.

Setup

To run a multi-server license, follow these steps:

1. Install the product on each server where it should run.
2. During installation, import the license key into each server. This key is the same for all servers in the subscription.
3. Users synchronization
 - when the license is imported, the initial user list is pulled from the License Manager on dbeaver.com.
 - all servers share the same named users from License Manager.
 - users can also be added in several ways:
 - via License Manager
 - via the CloudBeaver UI. [Learn more](#)
 - via SSO login (users are created automatically if seats are available). [Learn more](#)
 - via AWS IAM (users are created automatically if seats are available). [Learn more](#)
4. Changes in user management
 - updates from License Manager synchronize automatically once per hour.

Tip

To apply changes immediately, restart the server.

- if a user is deleted or blocked in the UI, they can be re-added immediately through License Manager. The same action in the UI is only available after two weeks.

Important

License Manager always has the priority over all other user management methods.

5. Offline work

- multi-server licenses require internet access for automatic synchronization.
- if a server has no internet connection, you must manually re-import the updated license file whenever the user list changes.

6. Server configuration

- multi-server subscriptions don't require special configuration parameters. Servers work the same way as with single-server licenses.
- each server can have its own domain, connections, and settings. The only shared element across servers is the licensed user list.

Multi-server license specifics

- minimum of 5 users per license.
- Can be deployed On-premises, in private or public clouds, or in hybrid environments.

Info

For details, see [Deployment options](#).

The number of users can be increased at any time during the subscription by sending a request by email or using the website. Additional users require an updated license. You can reduce the number of users, but only before the next renewal.

The number of servers can also be changed, but only by email request. As with users, servers can be reduced only before renewal.

Differences with single-server license

Aspect	Single-server license	Multi-server license
User management	Managed fully inside the server admin panel. The license key is imported and updated in the UI.	Main management through dbeaver.com . Users can also be added in the UI or via SSO if seats are available.
User list	All users are named. Every user accessing the server must be counted in the license.	All users must be named and counted in the license, even if not all connect to every server. One license key is shared across servers.
Synchronization	Internet is not required. The server works offline and license updates are handled manually in the UI.	Requires internet for automatic user synchronization, by default every hour. Without internet the license file must be reimported manually.
Offline mode	Works offline without restrictions.	Works offline only if the license file is manually reimported after user list changes. Without this, user updates will not apply.

Info

Learn more about [license types](#).

About CloudBeaver

Editions

Enterprise edition

CloudBeaver Enterprise is a lightweight web app for secure, all-around data management. It provides a single cloud solution to work with multiple data sources directly from your browser.

Built for teams and organizations, it offers features like an AI assistant for SQL queries, advanced user management, and seamless support for AWS, Microsoft Azure, and GCP.

Info

Explore CloudBeaver's features on our [Demo Server](#) to get a hands-on experience.

Enterprise Edition features

The Enterprise Edition provides powerful tools designed for professional database management, security, and integration:

- **Easy server setup**
 - **One-command deploy:** Quickly install using a single Docker command. [Learn more](#)
 - **Domain configuration:** Set up a custom domain and SSL certificate. [Learn more](#)
- **Multiple database support**
 - **Unified interface:** Work seamlessly with multiple databases from a single platform. [Learn more](#)
 - **Driver management:** Create and modify custom drivers. [Learn more](#)
- **Secure access**
 - **Enterprise authentication:** Choose from a variety of authentication types, including SSO, OpenID, and IAM, for secure and flexible application access. [Learn more](#)
 - **User provisioning:** Import users from systems like AzureAD, Okta, or Google to pre-configure access. [Learn more](#)
 - **Connection sharing:** Save and share credentials for all users with access. [Learn more](#)
 - **Multi-user collaboration:** Enable teams to work together securely with customized

permissions. [Learn more](#)

- **Data management**
 - **Interactive tools:** Apply flexible filters and view data interactively. [Learn more](#)
 - **SQL development:** Enhance SQL writing with autocomplete, syntax highlighting, and script execution. [Learn more](#)
- **Data import/export:** Import data from files or export to various formats. [Learn about data import](#) | [Learn about data export](#)
- **Database visualization:** View database structures, relationships, and metadata with entity diagrams. [Learn more](#)
- **SQL editor and advanced tools**
 - **SQL editor:** Write and execute SQL scripts, manage queries, and view results in a user-friendly interface. [Learn more](#)
 - **AI assistant:** Use AI for smart SQL completions and suggestions. [Learn more](#)
 - **Visual query builder:** Build complex queries with an intuitive graphical interface. [Learn more](#)
- **Integrated cloud tools**
 - **Cloud Explorer:** Access and manage databases hosted in cloud environments. [Learn more](#)
 - **Cloud file management:** Work with files stored in cloud storage systems. [Learn more](#)

Subscription model

CloudBeaver Enterprise Edition is offered as an annual subscription, which should be renewed to continue receiving updates and support.

[Discover more about CloudBeaver Enterprise Edition and subscribe.](#)

Supported databases

CloudBeaver Enterprise provides support for a wide range of databases. For a full list of supported databases, refer to [Supported databases](#).

This edition also includes the Cloud Explorer feature, which provides access to cloud databases such as those on Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure. Cloud Explorer enables centralized authentication and automatic configuration, simplifying the management of cloud database connections.

Cloud Explorer databases

For a list of supported databases, please refer to the [related section](#) in the Cloud Explorer guide.

AWS edition

CloudBeaver AWS is a multi-user web tool for managing databases, built to work with Amazon Web Services (AWS). It allows teams to collaborate easily and take advantage of AWS infrastructure. This edition is especially useful for teams and organizations using AWS cloud environments. It provides a shared, web-based platform for managing databases, making it easier to collaborate, access, and manage data while using AWS infrastructure.

CloudBeaver AWS key features

The CloudBeaver AWS Edition provides a powerful set of features tailored for database management within the AWS ecosystem. These features focus on seamless integration with AWS services, scalability, security, and collaboration.

- **AWS Services**
 - **Cloud Explorer:** A tool for managing and connecting cloud-hosted databases. [Learn more.](#)
 - **Amazon S3 support:** Manage cloud storage directly from CloudBeaver by integrating with Amazon S3. [Learn more.](#)
- **Secure access**
 - **Access management:** CloudBeaver AWS integrates IAM, SAML, OpenID, and various security identity providers to provide secure access. [Learn more.](#)
 - **User management:** Centralize access control with admin tools. Import users directly from identity providers like Google, Okta, and Azure. [Learn more.](#)
- **Collaboration**
 - **Multi-user collaboration:** Collaborate in real-time with multiple users, setting custom teams and permissions to maintain data security.
 - **Query history:** Track and review past queries. [Learn more.](#)
- **Data management and SQL development**
 - **Data filters and views:** Utilize flexible filters and interactive views for efficient data handling. [Learn more.](#)
 - **SQL development tools:** Enhance SQL development with autocomplete, syntax highlighting, and script execution capabilities. [Learn more.](#)
- **Data import/export**
 - **Data export:** Export data to a variety of formats, including CSV, Excel, and more. [Learn more.](#)
 - **Data import:** Import data from files directly into your database. [Learn more.](#)

- **Data Visualization**
 - **Entity relationship diagrams:** Visualize database schemas through entity-relationship diagrams. [Learn more.](#)
- **Query development**
 - **AI assistant in SQL Editor:** Integrates an AI assistant to offer guidance and automation within the SQL Editor. [Learn more.](#)
 - **Search in query history:** Enables searching through previous queries for easier management and reuse. [Learn more](#)
 - **Visual query builder:** Provides a graphical interface to construct SQL queries without writing code. [Learn more.](#)

Integrated database drivers

CloudBeaver AWS has a comprehensive set of JDBC drivers, offering connectivity to various databases without needing separate downloads and configurations.

Supported databases

The CloudBeaver AWS provides support for a wide range of databases. For a full list of supported databases, refer to [Supported databases](#).

Subscription and licensing

CloudBeaver AWS is available for subscription through the [AWS Marketplace](#). The subscription price depends on the instance size you choose. A bigger instance allows more users to work with the application at the same time.

You get a one-month free trial with access to all CloudBeaver features. After the trial, billing is managed based on the AWS pricing plan for the chosen instance size. You don't need an extra license, as everything is handled through the AWS Marketplace subscription.

Info

In rare cases, CloudBeaver AWS may request a license if access to the EC2 metadata service is restricted. This is often due to enforced IMDSv2 policies by AWS. Since CloudBeaver runs inside a Docker container with a different network interface, it might be unable to communicate with the EC2 instance metadata. Refer to the [Troubleshooting](#) section for guidance on resolving this issue.

Initial setup

After subscribing, navigate to `http://<EC2_IP>/` in your browser, where `<EC2_IP>` is the public IP of your EC2 instance. This will open the initial CloudBeaver configuration wizard, which guides you through the setup process.

Info

For more information on initial configuration, see [Server configuration wizard](#).

CloudBeaver AWS updates

CloudBeaver AWS runs as a Docker Compose cluster on your EC2 instance. To update it manually, connect to your EC2 instance via SSH or the instance console, and pull the desired Docker image.

Important

CloudBeaver AWS uses Docker Compose to manage containers and volumes. Use Docker Compose for all update operations.

Refer to the following documentation for additional guidance:

- [CloudBeaver AWS images and tags on Docker Hub](#)
- [AWS EC2: Connecting via SSH](#)
- [AWS key pairs](#)
- [Docker compose documentation](#)

Support

For technical support with CloudBeaver AWS, you will need to submit a ticket through the DBeaver customer portal on dbeaver.com. When creating a support request, ensure to include your AWS **Agreement ID**, which is necessary for verifying your subscription.

You can locate your **Agreement ID** by going to **AWS Marketplace > Manage Subscriptions** and selecting **CloudBeaver AWS**. This ID is essential for the CloudBeaver support team to handle your request properly.

Tip

If you need assistance on how to access the **Manage Subscriptions** section, refer to [the official AWS documentation](#).

Once you have your Agreement ID, log into your account on the [DBeaver website](#), navigate to **Profile -> Tickets**, and create a new support ticket. Be sure to include all relevant details about the issue to allow the support team to assist you effectively.

Troubleshooting

CloudBeaver AWS installations from the AWS Marketplace may prompt for a license due to issues with EC2 metadata service availability. This can happen when AWS policies enforce the use of IMDSv2 and restrict access to IMDSv1.

Since CloudBeaver runs inside a Docker container, it operates with a different IP address, which prevents it from connecting to the EC2 metadata service using IMDSv2. As a result, CloudBeaver cannot verify that it was launched through a Marketplace installation.

Solution: To resolve this, ensure your Docker container is run with the `--network host` option, which allows CloudBeaver to access the EC2 metadata service by sharing the host's IP address.

Release cycles

This article explores the release cycles and support periods, detailing how we organize software updates and support.

CloudBeaver release cycles

CloudBeaver releases updates on a structured schedule, providing users with regular enhancements, new features, and fixes.

Release schedule

- **Regular releases:** CloudBeaver releases a new stable version every three months, resulting in four major updates annually.
- **Early Access (EA) versions:** Additionally, CloudBeaver provides Early Access (EA) versions every two weeks. These versions allow users to engage with upcoming features and changes, offering feedback to help refine these enhancements before their official release.

Important

The EA version may not be as stable as the regular releases. Users are advised to use EA versions with caution.

Docker tags

The following tags determine the version you install on your machine:

- `ea` : The tag `ea` is associated with the latest Early Access (EA) version that has been released.
- `latest` : The `latest` tag points to the most recent stable version.
- Versions with the specific numbers:
- `xx.0.1` : Specifying a version in this format will download the exact version of the CloudBeaver.
- `xx.0` : Using a version number formatted as `xx.0` will download the latest version within the `xx.0` series.
- `xx` : Specifying just the major version number, like `24` will download the latest version within the major version `24` family.

Tip

For instruction on deployment, see [CloudBeaver Deployment](#).

Customer technical support

Accessing technical support

CloudBeaver ensures all users receive technical support for the duration of their paid license. Tickets for technical support can be submitted via the [website](#).

Tip

If you're an administrator, open **Settings - Administration** to find the **Tech Support** link in the sidebar. It takes you straight to the support request page.

End of support

Please be aware that technical support ends alongside the expiration of the license period. This occurs even though CloudBeaver remains usable for users with a Perpetual license type.

FAQ

How to check driver version

The easiest way to check your driver version:

1. In **Database Navigator**, right-click your connection
2. Click **Edit Connection**
3. In the window that opens, click **Test**
4. In the summary pop-up, check the **Client version**-that's your driver

How to update driver version

To update a driver version:

1. As an administrator, go to **Administration -> Driver Management**
2. Open the driver
3. Go to the **Libraries** tab
4. Click **Add file** and upload the new version. You need to download the JAR file manually. Sources vary depending on the driver:
 - **Official vendor site**-some drivers are only available on the database vendor's site
 - **GitHub releases**-many open-source drivers publish binaries in the **Releases** section
 - **Maven Central**-some drivers are available at search.maven.org
 - **Other public repositories**-some projects host drivers on project-specific sites or custom registries
5. Delete the old file, then click **Save**

Note

In the Community edition, you can update drivers only on the server. For details, see [Driver Management](#).

How to get logs

To check logs for errors or connection issues, use one of the methods below.

In the UI

Go to **Tools** -> **Log Viewer**

In Docker

```
docker compose ps # find service name
docker logs <container_name> > container.log 2>&1
```

Tip

If the server won't start after an update, run `docker compose ps -a` to list all containers, including stopped ones.

In Kubernetes

```
kubectl get pods # list all pods
kubectl logs <pod_name> > pod.log 2>&1
```

Tip

- add `-n <namespace>` if your pods run in a custom namespace
- use `kubectl logs <pod_name> -c <container_name>` if the pod has multiple containers

In CloudBeaver AWS

If you deployed CloudBeaver on AWS, you can access logs through the AWS Console:

1. In the **EC2 Console**, find your instance
2. Click **Connect**
3. Use **Session Manager** or **SSH**
4. Run:

```
docker compose ps # find service name
docker logs <container_name> > container.log 2>&1
```

Tip

When you run `docker logs >> logs.txt`, the file is saved inside the instance, in the directory where the command was run.

UI freeze or SSO stuck

If the UI looks normal but never updates, or SSO/login hangs, check if WebSockets are working:

1. In your browser, open **DevTools -> Network**
2. Filter by **ws**. You should see a `ws://<your-domain>/api/ws` or `wss://<your-domain>/api/ws` request with status **101 Switching Protocols**

Info

- if you see **101** and the connection stays open -> WebSockets are working
- if it's missing, shows errors like 404/502/503, or closes quickly -> the proxy or firewall may be blocking WebSockets

3. If the connection closes quickly, increase proxy or load balancer timeouts

For details, see [WebSockets](#).