Camunda 7 with MySQL Integration Using Docker

This document provides a step-by-step guide to configure and deploy Camunda 7 with MySQL using Docker and Docker Compose. By the end of this setup, Camunda will use MySQL as its database for process and task management.

Prerequisites

- Docker and Docker Compose installed on your system.
- Basic knowledge of YAML and Docker commands.

Overview

You have two options for the Camunda Docker image:

1. **Custom Camunda Image**: Use your own customized Camunda image that includes pre-configured plugins, libraries, or workflows.
2. **Official Camunda Image**: Use the official Camunda 7 image provided by Camunda on Docker Hub. This image is pre-configured for general use and easy to set up.

Choose the image type based on your requirements.

Steps

Step 1: Prepare the Docker Compose File

Create a `docker-compose.yml` file in your project directory.

Replace `custom-camunda-image` in the `image` section with either:

- **Your custom image name** (e.g., `your-dockerhub-username/camunda-custom-app:latest`), OR
- **The official Camunda image** (e.g., `camunda/camunda-bpm-platform:7.19.0`).

Example docker-compose.yml:

```
version: '3.8'
services:
 mysql:
   image: mysql:8.0  # update docker image or version
   container_name: mysql
   ports:
     - "3306:3306"
   environment:
     MYSQL_ROOT_PASSWORD: rootpass
     MYSQL_DATABASE: camunda
     MYSQL_USER: camunda_user
     MYSQL_PASSWORD: camunda_pass
   volumes:
     - camunda-mysql-data:/var/lib/mysql
   healthcheck:
     test: [ "CMD-SHELL", "mysqladmin ping -h localhost -u camunda_user -p'camunda_pass' || exit 1" ]
     interval: 10s
     retries: 5
   networks:
     - camunda-network

 camunda:
   image: camunda/camunda-bpm-platform:7.22.0 # update image and version
   container_name: camunda
   ports:
     - "8080:8080"
   environment:
     - DB_DRIVER=com.mysql.cj.jdbc.Driver
     - DB_URL=jdbc:mysql://mysql:3306/camunda?useSSL=false&serverTimezone=UTC
     - DB_USERNAME=camunda_user
     - DB_PASSWORD=camunda_pass
   depends_on:
     mysql:
       condition: service_healthy
   networks:
     - camunda-network

volumes:
 camunda-mysql-data:

networks:
 camunda-network:
```

Step 2: Run Docker Compose

1.

In the directory where docker-compose.yml is located, run the following command:
Bash

```
docker-compose up
```

```
PS C:\Users\Rutusoft\Downloads\spring-camunda-docker-mysql\spring-camunda-docker-mysql> docker compose up
```

```
time="2024-12-06T23:55:00+05:30" level=warning
msg="C:\\Users\\Rutusoft\\Downloads\\spring-camunda-docker-mysql\\sp
ring-camunda-docker-mysql\\docker-compose.yaml: the attribute
`version` is obsolete, it will be ignored, please remove it to avoid
potential confusion"
[+] Running 4/4
 ✔ Network spring-camunda-docker-mysql_camunda-network      Created
0.1s
 ✔ Volume "spring-camunda-docker-mysql_camunda-mysql-data"  Created
0.0s
 ✔ Container mysql                                          Created
0.1s
 ✔ Container camunda                                        Created
0.1s
Attaching to camunda, mysql
mysql    | 2024-12-06 18:25:00+00:00 [Note] [Entrypoint]: Entrypoint
script for MySQL Server 8.0.39-1.el9 started.
mysql    | 2024-12-06 18:25:01+00:00 [Note] [Entrypoint]: Switching
to dedicated user 'mysql'
mysql    | 2024-12-06 18:25:01+00:00 [Note] [Entrypoint]: Entrypoint
script for MySQL Server 8.0.39-1.el9 started.
mysql    | 2024-12-06 18:25:01+00:00 [Note] [Entrypoint]:
Initializing database files
mysql    | 2024-12-06T18:25:01.454502Z 0 [Warning] [MY-011068]
[Server] The syntax '--skip-host-cache' is deprecated and will be
removed in a future release. Please use SET GLOBAL host_cache_size=0
instead.
mysql    | 2024-12-06T18:25:01.455177Z 0 [System] [MY-013169]
[Server] /usr/sbin/mysqld (mysqld 8.0.39) initializing of server in
progress as process 80
mysql    | 2024-12-06T18:25:01.473114Z 1 [System] [MY-013576]
[InnoDB] InnoDB initialization has started.
mysql    | 2024-12-06T18:25:02.487121Z 1 [System] [MY-013577]
[InnoDB] InnoDB initialization has ended.
mysql    | 2024-12-06T18:25:05.449362Z 6 [Warning] [MY-010453]
[Server] root@localhost is created with an empty password ! Please
consider switching off the --initialize-insecure option.
mysql    | 2024-12-06 18:25:10+00:00 [Note] [Entrypoint]: Database
files initialized
mysql    | 2024-12-06 18:25:10+00:00 [Note] [Entrypoint]: Starting
temporary server
```

```
mysql    | 2024-12-06T18:25:10.454801Z 0 [Warning] [MY-011068]
[Server] The syntax '--skip-host-cache' is deprecated and will be
removed in a future release. Please use SET GLOBAL host_cache_size=0
instead.
mysql    | 2024-12-06T18:25:10.456511Z 0 [System] [MY-010116]
[Server] /usr/sbin/mysqld (mysqld 8.0.39) starting as process 124
mysql    | 2024-12-06T18:25:10.470279Z 1 [System] [
```

```
camunda  | Configure database
camunda  | 06-Dec-2024 18:25:23.043 INFO [main]
org.apache.catalina.startup.VersionLoggerListener.log Server version
name:   Apache Tomcat/10.1.30
camunda  | 06-Dec-2024 18:25:23.050 INFO [main]
org.apache.catalina.startup.VersionLoggerListener.log Server built:
Sep 13 2024 20:26:16 UTC
camunda  | 06-Dec-2024 18:25:23.051 INFO [main]
org.apache.catalina.startup.VersionLoggerListener.log Server version
number: 10.1.30.0
camunda  | 06-Dec-2024 18:25:23.051 INFO [main]
org.apache.catalina.startup.VersionLoggerListener.log OS Name:
Linux
camunda  | 06-Dec-2024 18:25:23.051 INFO [main]
org.apache.catalina.startup.VersionLoggerListener.log OS Version:
5.15.167.4-microsoft-standard-WSL2
camunda  | 06-Dec-2024 18:25:23.051 INFO [main]
org.apache.catalina.startup.VersionLoggerListener.log Architecture:
amd64
camunda  | 06-Dec-2024 18:25:23.051 INFO [main]
org.apache.catalina.startup.VersionLoggerListener.log Java Home:
/usr/lib/jvm/java-17-openjdk
camunda  | 06-Dec-2024 18:25:23.051 INFO [main]
org.apache.catalina.startup.VersionLoggerListener.log JVM Version:
17.0.12+7-alpine-r0
camunda  | 06-Dec-2024 18:25:23.052 INFO [main]
org.apache.catalina.startup.VersionLoggerListener.log JVM Vendor:
Alpine
camunda  | 06-Dec-2024 18:25:23.052 INFO [main]
org.apache.catalina.startup.VersionLoggerListener.log CATALINA_BASE:
/camunda
```

```
camunda  | 06-Dec-2024 18:25:23.052 INFO [main]
org.apache.catalina.startup.VersionLoggerListener.log CATALINA_HOME:
/camunda
camunda  | 06-Dec-2024 18:25:23.083 INFO [main]
org.apache.catalina.startup.VersionLoggerListener.log Command line
argument:
-Djava.util.logging.config.file=/camunda/conf/logging.properties
camunda  | 06-Dec-2024 18:25:23.083 INFO [main]
org.apache.catalina.startup.VersionLoggerListener.log Command line
argument:
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager
camunda  | 06-Dec-2024 18:25:23.083 INFO [main]
org.apache.catalina.startup.VersionLoggerListener.log Command line
argument: -Djdk.tls.ephemeralDHKeySize=2048
```

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

`-d` will run the container in the background. It's the same as adding `--detach`. What is being printed when you use `-d` is the container's ID.

Docker will:

- Start the MySQL container with the specified username, password, and database.
- Start the Camunda container (either custom or official), configured to connect to the MySQL database

Step 3: Verify the Deployment

1. Open a browser and navigate to the Camunda Web App:
    - URL: `http://localhost:8080/camunda`
    - Default credentials:
        - Username: `demo`
        - Password: `demo`
2. Confirm that Camunda is running and connected to the MySQL database.

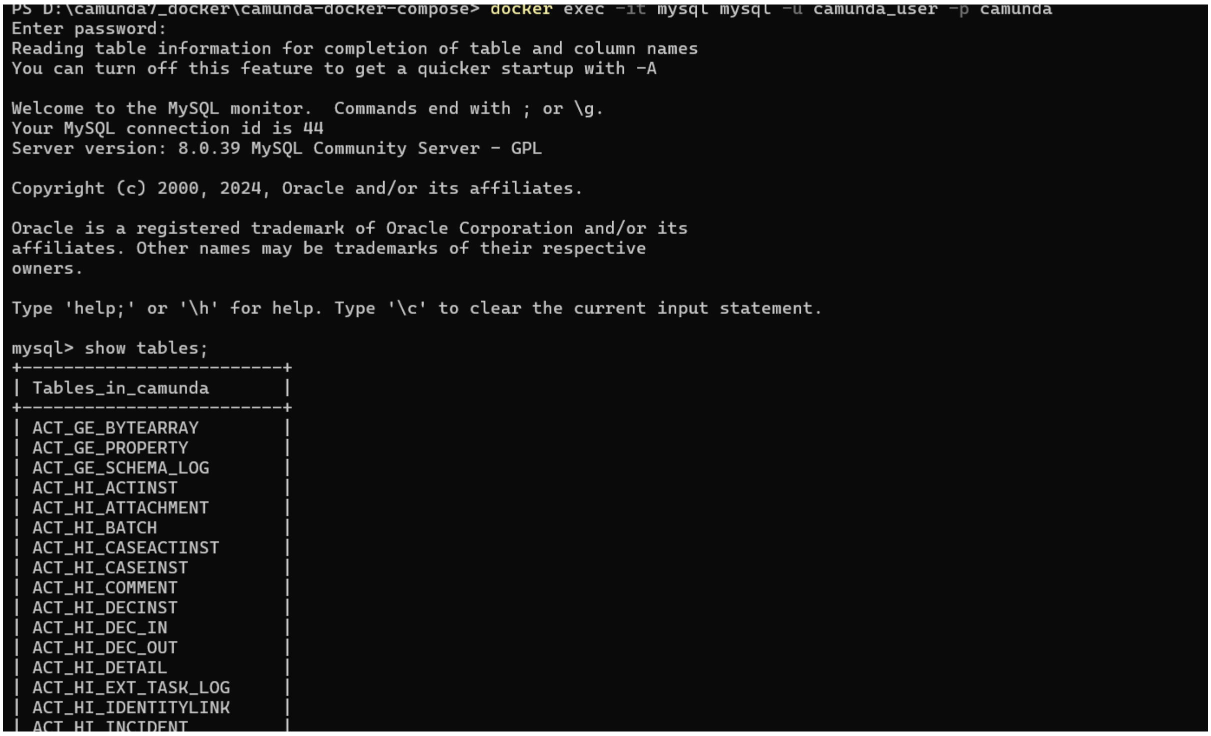
Step 4: Access the MySQL Database

1. To connect to the MySQL database, use a MySQL client or CLI:

```
docker exec -it mysql mysql -u camunda_user -p camunda
```

2. Enter the password: `camunda_pass`.
3. You can inspect the Camunda tables and data in the database:
sql

```
SHOW TABLES;
```



```
PS D:\camunda7_docker\camunda-docker-compose> docker exec -it mysql mysql -u camunda_user -p camunda
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 44
Server version: 8.0.39 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

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

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

mysql> show tables;
+--------------------------+
| Tables_in_camunda        |
+--------------------------+
| ACT_GE_BYTEARRAY         |
| ACT_GE_PROPERTY          |
| ACT_GE_SCHEMA_LOG        |
| ACT_HI_ACTINST           |
| ACT_HI_ATTACHMENT        |
| ACT_HI_BATCH             |
| ACT_HI_CASEACTINST       |
| ACT_HI_CASEINST          |
| ACT_HI_COMMENT           |
| ACT_HI_DECINST           |
| ACT_HI_DEC_IN            |
| ACT_HI_DEC_OUT           |
| ACT_HI_DETAIL            |
| ACT_HI_EXT_TASK_LOG      |
| ACT_HI_IDENTITYLINK      |
| ACT_HI_INCIDENT          |
```

As you can above table are created

Step 5: Stopping and Cleaning Up

To stop and remove the containers:

bash

Copy code

```
docker-compose down
```

```
docker-compose down --volumes
```

To remove volumes as well (erases all data):

bash

Copy code

Custom vs. Official Camunda Image

Here’s the updated documentation with the note about using custom Camunda images or the official Camunda 7 image.

Camunda 7 with MySQL Integration Using Docker

This document provides a step-by-step guide to configure and deploy Camunda 7 with MySQL using Docker and Docker Compose. By the end of this setup, Camunda will use MySQL as its database for process and task management.

Overview

You have two options for the Camunda Docker image:

1. **Custom Camunda Image**: Use your own customized Camunda image that includes pre-configured plugins, libraries, or workflows.
2. **Official Camunda Image**: Use the official Camunda 7 image provided by Camunda on Docker Hub. This image is pre-configured for general use and easy to set up.

Choose the image type based on your requirements.

Steps

Step 1: Prepare the Docker Compose File

Create a `docker-compose.yml` file in your project directory. Replace `custom-camunda-image` in the `image` section with either:

- **Your custom image name** (e.g., `your-dockerhub-username/camunda-custom-app:latest`), OR
- **The official Camunda image** (e.g., `camunda/camunda-bpm-platform:7.19.0`).

Example `docker-compose.yml`:

yaml

Copy code

```
version: '3.8'
services:
  camunda:
    image: camunda/camunda-bpm-platform:7.19.0 # Replace with your
custom image if needed
    container_name: camunda
    ports:
      - "8080:8080"
    environment:
      - DB_DRIVER=com.mysql.cj.jdbc.Driver
      -
DB_URL=jdbc:mysql://mysql:3306/camunda?useSSL=false&serverTimezone=U
TC
      - DB_USERNAME=camunda_user
      - DB_PASSWORD=camunda_pass
    depends_on:
      - mysql
    networks:
```

```yaml
      - camunda-network

  mysql:
    image: mysql:8.0
    container_name: mysql
    ports:
      - "3306:3306"
    environment:
      MYSQL_ROOT_PASSWORD: rootpass
      MYSQL_DATABASE: camunda
      MYSQL_USER: camunda_user
      MYSQL_PASSWORD: camunda_pass
    volumes:
      - camunda-mysql-data:/var/lib/mysql
    networks:
      - camunda-network

volumes:
  camunda-mysql-data:

networks:
  camunda-network:
```

Step 2: Run Docker Compose

In the directory where `docker-compose.yml` is located, run the following command:
bash
Copy code

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

1.
2. Docker will:
   - Start the MySQL container with the specified username, password, and database.
   - Start the Camunda container (either custom or official), configured to connect to the MySQL database.

Step 3: Verify the Deployment

1. Open a browser and navigate to the Camunda Web App:
   - URL: `http://localhost:8080/camunda`
   - Default credentials:
     - Username: `demo`
     - Password: `demo`
2. Confirm that Camunda is running and connected to the MySQL database.

Step 4: Access the MySQL Database

To connect to the MySQL database, use a MySQL client or CLI:
bash
Copy code

```
docker exec -it mysql mysql -u camunda_user -p camunda
```

1.
2. Enter the password: `camunda_pass`.

You can inspect the Camunda tables and data in the database:
sql
Copy code

```
SHOW TABLES;
```

3.

Step 5: Stopping and Cleaning Up

To stop and remove the containers:

bash

Copy code

```
docker-compose down
```

To remove volumes as well (erases all data):

bash

Copy code

```
docker-compose down --volumes
```

Custom vs. Official Camunda Image

Aspect	Custom Image	Official Image
Use Case	Pre-configured workflows, plugins, or libraries.	General purpose, quick setup for most applications.
Flexibility	Requires building and maintaining the image manually.	Ready-to-use with minimal configuration.

Examples	`your-dockerhub-username/camunda-custom-app:latest`	`camunda/camunda-bpm-platform:7.22.0`

To build a custom image, create a `Dockerfile` that extends the official image and includes your customizations.