



+ New

🕒 Now



Share

please translate this page https://qiita.com/saitamanokusa/items/ffb8f05cbc8e75d435ce to english

🔗 Sources

ffb8f05cbc8e75d435ce

📄 ffb8f05cbc8e75d435ce



Answer

Starting JS7[®] JobScheduler with docker-compose (MySQL version)

Overview

This guide is for those who want to quickly set up and use JS7 JobScheduler using docker-compose.

Preparation

Prepare the necessary files and directories:

1. Create `docker-compose.yml` file
2. Create `hibernate.cfg.xml` file
3. Create `.env` file for environment variables
4. Create required directories

docker-compose.yml

text



Ask follow-up

🏠 Home

🔍 Discover

🌟 Spaces

📖 Library

```
version: '3'
```

```
services:
```

```
  db:
```

```
    image: mysql:8.0
```

```
    ports:
```

```
      - "3306:3306"
```

```
    command:
```

```
      - --default-authentication-plugin=caching_sha2_password
```

```
      - --character-set-server=utf8mb4
```

```
      - --collation-server=utf8mb4_bin
```

```
      - --skip-character-set-client-handshake
```

```
    volumes:
```

```
      - db_data:/var/lib/mysql
```

```
  networks:
```

```
    - js7network
```

```
  environment:
```

```
    MYSQL_ROOT_PASSWORD: iLoveMySQL8.0!
```

```
    MYSQL_DATABASE: js7db
```

```
    MYSQL_USER: js7user
```

```
    MYSQL_PASSWORD: js7password
```

```
  restart: "no"
```

```
js7-joc-primary:
```

```
  depends_on:
```

```
    - db
```

```
  image: sosberlin/js7:joc-${JS7VERSION}
```

```
  hostname: js7-joc-primary
```

```
  ports:
```

```
    - "4446:4446"
```

```
  volumes:
```

```
    - js7-joc-primary-config:/var/sos-berlin.com/js7/joc/resources/joc
```

```
    - js7-joc-primary-logs:/var/log/sos-berlin.com/js7/joc
```

```
  networks:
```

```
    - js7network
```

```
  environment:
```

```
    RUN_JS_JAVA_OPTIONS: -Xmx256m
```

```
    RUN_JS_USER_ID: "${JS7USERID}:${JS7GROUPID}"
```



```
image: sosberlin/js7:controller-${JS7VERSION}
hostname: js7-controller-primary
volumes:
  - js7-controller-primary:/var/sos-berlin.com/js7/controller
networks:
  - js7network
environment:
  RUN_JS_JAVA_OPTIONS: -Xmx256m
  RUN_JS_USER_ID: "${JS7USERID}:${JS7GROUPID}"
restart: "no"
```

```
js7-agent-primary:
  image: sosberlin/js7:agent-${JS7VERSION}
  hostname: js7-agent-primary
  volumes:
    - js7-agent-primary:/var/sos-berlin.com/js7/agent/
  networks:
    - js7network
  environment:
    RUN_JS_JAVA_OPTIONS: -Xmx256m
    RUN_JS_USER_ID: "${JS7USERID}:${JS7GROUPID}"
  restart: "no"
```

```
volumes:
  db_data:
    driver: local
    driver_opts:
      type: none
      device: ${PWD}/db_data
      o: bind
  js7-joc-primary-config:
    driver: local
    driver_opts:
      type: none
      device: ${PWD}/js7-joc-primary-config
      o: bind
  js7-joc-primary-logs:
    driver: local
```



```
    o: bind
js7-controller-primary:
  driver: local
  driver_opts:
    type: none
    device: ${PWD}/js7-controller-primary
    o: bind
js7-agent-primary:
  driver: local
  driver_opts:
    type: none
    device: ${PWD}/js7-agent-primary
    o: bind
```

```
networks:
  js7network:
```

hibernate.cfg.xml

xml



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<hibernate-configuration>
  <session-factory>
    <property
name="hibernate.connection.driver_class">org.mariadb.jdbc.Driver</property>
    <property name="hibernate.connection.password">js7password</property>
    <property name="hibernate.connection.url">jdbc:mariadb://db:3306/
js7db?allowPublicKeyRetrieval=true&useSSL=false</property>
    <property name="hibernate.connection.username">js7user</property>
    <property
name="hibernate.dialect">org.hibernate.dialect.MySQLInnoDBDialect</
property>
    <property name="hibernate.show_sql">>false</property>
    <property name="hibernate.connection.autocommit">>false</property>
    <property name="hibernate.format_sql">>true</property>
    <property name="hibernate.temp.use_jdbc_metadata_defaults">>false</
property>
    <property
name="hibernate.connection.provider_class">org.hibernate.hikaricp.internal.
HikariCPConnectionProvider</property>
    <property name="hibernate.hikari.maximumPoolSize">10</property>
  </session-factory>
</hibernate-configuration>
```

.env file

Create the .env file with the following content:

text



```
JS7USERID=1000
JS7GROUPEID=1000
JS7VERSION=2-5-7
```

Create directories



bash

Home

Discover

Spaces

Library

```
mkdir db_data
mkdir js7-agent-primary
mkdir js7-controller-primary
mkdir js7-joc-primary-config
mkdir js7-joc-primary-logs
```

Starting the containers

1. Start the containers:

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



2. Copy the `hibernate.cfg.xml` file to the volume mount destination:

```
bash
cp -f hibernate.cfg.xml js7-joc-primary-config/
```



3. Install the database tables:

```
bash
docker compose exec js7-joc-primary /bin/sh -c /opt/sos-berlin.com/
js7/joc/install/joc_install_tables.sh
```



4. Restart the js7-joc-primary container:

```
bash
docker compose restart js7-joc-primary
```



Checking the operation

1. Access the JS7 JobScheduler web interface at <http://localhost:4444>.




4. Register the Agent with the following details:

- Agent ID: `js7-agent-primary`
- Agent Name: `js7-agent-primary`
- Title: (optional)
- Alias Names: `primaryAgent` (for sample data usage)
- URL: `http://js7-agent-primary:4445`

5. Deploy the registered agent

Additional Information

- For PowerShell users, there's an agent-plus image available with PowerShell and JS7 modules pre-installed.
- To update JS7, stop the containers, update the .env file with the new version, and restart the containers.
- Refer to the original article for the docker-compose.yml file for versions 2.5.1 and earlier, as there were changes in volume path specifications from version 2.5.2 onwards.

 Share  Rewrite

   ...

5. Deploy the agent

