

## 3.2. Steps to install Soffid

Follow these steps to install Soffid IAM:

1. First of all, you must **create a folder to save the yaml files** you are going to create.

```
mkdir lab-soffid-iam
```

2. **Go inside the folder**

```
cd lab-soffid-iam
```

3. I recomend the very first time you install Soffid in an environment, to make **three different YAML files**, one to the Repository, one to the Console, and one to the Sync Server. Once you have Soffid running properly, you can merge the files and use only one to start and stop Soffid using only one step.

- 3.1. You need to create one folder for each component, for example using those commands.

```
mkdir 01repository  
mkdir 02console  
mkdir 03syncserver
```

4. **Create the Repository**

- 4.1. Go inside the folder 01repository

```
cd 01repository
```

- 4.2. Once you are inside the folder, you must create a docker-compose.yaml file with the Repository service definition. To create the YAML files you can use your usual text editor.

```
services:  
  mariadb:  
    image: mariadb:11.4  
    environment:  
      MYSQL_ROOT_PASSWORD: XXXXX  
      MYSQL_DATABASE: soffid01  
      MYSQL_USER: soffid
```

```
MYSQL_PASSWORD: XXXXX
command: --max_allowed_packet=128M --innodb_log_file_size=256M --character-set-server=utf8mb4 --
collation-server=utf8mb4_general_ci
networks:
  - network
volumes:
  - mariadb_data:/var/lib/mysql
networks:
  network:
    name: YOUR_NETWORK
    driver: bridge
volumes:
  mariadb_data:
    name: compose_mariadbdata
```

4.3 Execute this command to initialize the Repository container (thanks to the -d option, containers will continue to run in the background, even if you close the terminal)

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

4.4. Check the containers: to check the container you can use a docker or a docker-compose command, depend on what you want to check.

4.4.1. In the folder: you can use a docker-compose command

```
sudo docker-compose ps
```

4.4.2. All of them: you can use a docker command

```
sudo docker ps
```

4.5. Check the logs: docker logs are detailed records of the activities that occur within containers. They are like a diary that records everything that happens, from starting and stopping the container to error messages, application outputs, and any other interactions.

4.5.1. You can use a docker-compose command

```
sudo docker-compose logs <SERVICE_NAME>
```

4.5.2. Or you can use a docker command

```
sudo docker logs -f <CONTAINER_NAME/CONTAINER_ID>
```

#### 4.6. If you need to stop the container:

```
sudo docker-compose down
```

### 5. Create Soffid Console

#### 5.1. Go inside the folder 02console

```
cd 02console
```

#### 5.2. Once you are inside the folder, you must create a docker-compose.yaml file with the Console service definition

```
services:
  console:
    image: soffid/iam-console:3.6.38
    environment:
      DB_URL: jdbc:mariadb://mariadb/soffid01
      DB_USER: root
      DB_PASSWORD: <ROOT_PASSWORD>
      SOFFID_TRUSTED_SCRIPTS: true
    networks:
      - network
    ports:
      - "8080:8080"
    volumes:
      - console_trust:/opt/soffid/iam-console-3/trustedcerts
      - console_conf:/opt/soffid/iam-console-3/conf
      - console_index:/opt/soffid/iam-console-3/index
networks:
  network:
    name: YOUR_NETWORK
    driver: bridge
volumes:
  console_trust:
    name: compose-console-trustedcerts-lab
  console_conf:
    name: compose-console-conf-lab
  console_index:
    name: compose-console-index-lab
```

5.3 Execute this command to initialize the Console container (thanks to the -d option, containers will continue to run in the background, even if you close the terminal)

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

5.4. Check the containers: to check the container you can use a docker or a docker-compose command, depend on what you want to check.

5.4.1. In the folder: you can use a docker-compose command

```
sudo docker-compose ps
```

5.4.2. All of them: you can use a docker command

```
sudo docker ps
```

5.5. Check the logs: docker logs are detailed records of the activities that occur within containers. They are like a diary that records everything that happens, from starting and stopping the container to error messages, application outputs, and any other interactions.

5.5.1. You can use a docker-compose command

```
sudo docker-compose logs <SERVICE_NAME>
```

5.5.2. Or you can use a docker command

```
sudo docker logs -f <CONTAINER_NAME/CONTAINER_ID>
```

5.6. If you need to stop the container:

```
sudo docker-compose down
```

5.7. Connect to the Soffid Console <http://localhost:8080/soffid>

- User: admin
- Password: changeit

---

The first time you connect, Soffid will require you to change the password.

## 6. Create Soffid Sync Server

6.1. Go inside the folder 03syncserver

```
cd 03syncserver
```

6.2. Once you are inside the folder, you must create a docker-compose.yaml file with the Sync Server service definition

```
services:
  sync-server:
    image: soffid/iam-sync:3.6.34
    hostname: sync-server
    environment:
      SOFFID_PORT: 1760
      SOFFID_HOSTNAME: <sync-server.YOUR_NETWORK>
      SOFFID_MAIN: yes
      DB_URL: jdbc:mysql://mariadb/soffid01
      DB_USER: soffid
      DB_PASSWORD: XXXXX
    networks:
      - network
    volumes:
      - sync_conf:/opt/soffid/iam-sync/conf
    ports:
      #- "1443:1443"
      - "1760:1760"
networks:
  network:
    name: YOUR_NETWORK
    driver: bridge
volumes:
  sync_conf:
    name: compose_sync_conf
```

6.3 Execute this command to initialize the Sync Server container (thanks to the -d option, containers will continue to run in the background, even if you close the terminal)

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

6.4. Check the containers: to check the container you can use a docker or a docker-compose command, depend on what you want to check.

6.4.1. In the folder: you can use a docker-compose command

```
sudo docker-compose ps
```

6.4.2. All of them: you can use a docker command

```
sudo docker ps
```

6.5. Check the logs: you should bear in mind that the Sync Server log could be too big, and is constantly growing. So, I recommend you to use **less tool**

6.5.1. You can use a docker-compose command

```
sudo docker-compose logs <SERVICE_NAME> | less
```

6.5.2. Or you can use a docker command

```
sudo docker logs -f <CONTAINER_NAME/CONTAINER_ID> | less
```

6.5.3. In addition, you can check the log from inside the container.

```
sudo docker -it <CONTAINER_NAME/CONTAINER_ID> bash
```

Once you are inside the container, you can go to the directory `/var/log/soffid/`

```
cd /var/log/soffid/
```

And check the `syncserver.log`

```
tail -f syncserver.log
```

6.6. If you need to stop the container:

```
sudo docker-compose down
```

**7.** Once Soffid Console and Sync Server are working fine, you can **merge all the YAML** file. You can then run this YAML file to update any services or add any additional settings.

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

---

<https://bookstack.soffid.com/books/installation/chapter/installing-soffid-using-docker-composeb>