

CSYE 6225: Network Structure & Cloud Computing Course

Tutorial and Practice: VMs and Containers

Intro to Docker

Objectives:

This tutorial covers basic containerization commands with Docker.

Prerequisites:

This tutorial will install docker on Mac. Then deploy a mysql container.

Docker: Docker is an open-source platform that automates the deployment, scaling, and management of applications within lightweight, portable containers. Containers are isolated environments that package an application and its dependencies, allowing it to run consistently across various computing environments.

Key Concepts

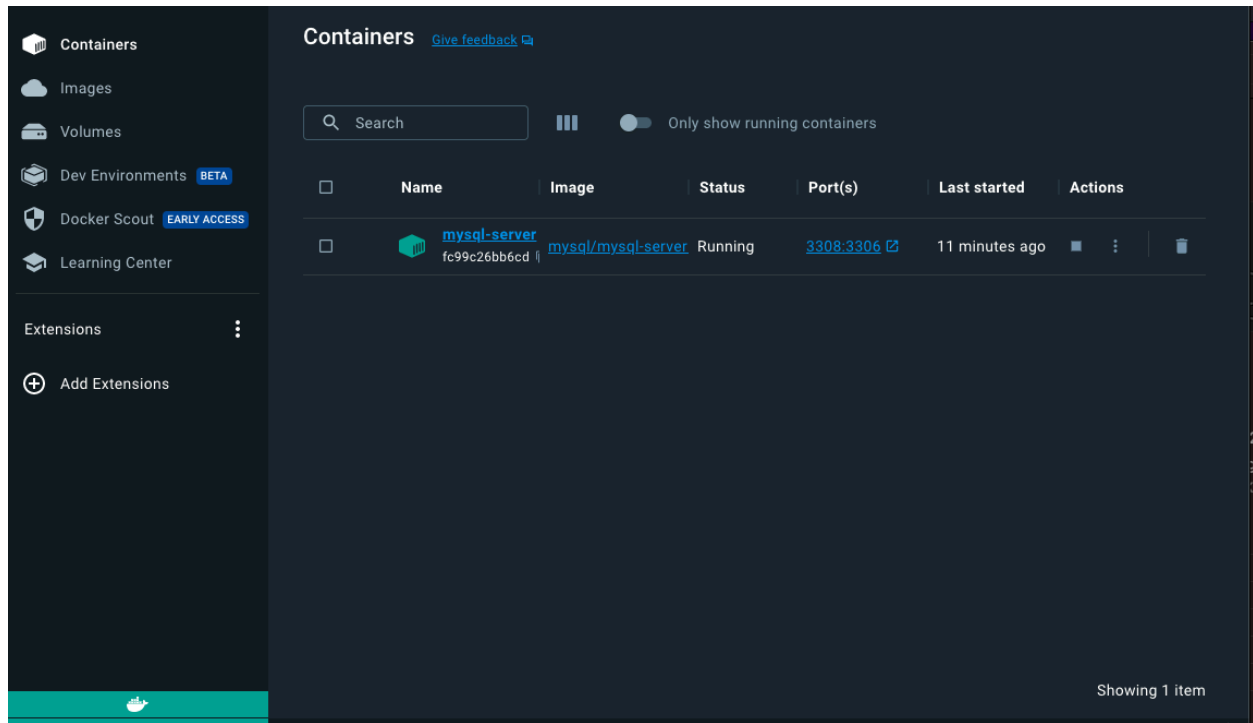
- **Containers:** Lightweight and portable units that include everything needed to run an application (code, libraries, dependencies). Share the host OS kernel, making them more efficient than virtual machines.
- **Images:** Read-only templates used to create containers. They contain the application code and dependencies. Docker images can be versioned, allowing developers to track changes and roll back if needed.
- **Dockerfile:** A text file that contains instructions for building a Docker image. It specifies the base image, dependencies, environment variables, and commands to run.
- **Docker Hub:** A cloud-based registry for storing and sharing Docker images. It allows users to find and download images created by others or upload their own.
- **Docker Compose:** A tool for defining and running multi-container Docker applications. You can specify all services, networks, and volumes in a docker-compose.yml file.

Hand-on Creating my-sql with Docker.

- Install Docker on laptop. There is a visual desktop version.

Link: <https://www.docker.com/products/docker-desktop/>

After installation, this is the Docker Desktop:



Note: You will not have any containers initially (the image shows one container running).

To create a container with mysql:

From command prompt, check docker is installed:

```
$ docker --version
```

```
(base) raja@LENOVO ~ % docker --version
Docker version 23.0.5, build bc4487a
```

Check deployed containers (you should see none. Screenshot shows one container):

`$docker ps`

```
(base) raja@LENOVO ~ % docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS              PORTS                               NAMES
fc99c26bb6cd   mysql/mysql-server:latest          "/entrypoint.sh mysql-" 14 minutes ago Up 14 minutes (healthy) 33060-33061/tcp, 0.0.0.0:3308->3306/tcp  mysql-server
```

Let's deploy a container with mysql:

`$ docker pull mysql/mysql-server`

```
(base) raja@LENOVO ~ % docker pull mysql/mysql-server
Using default tag: latest
latest: Pulling from mysql/mysql-server
Digest: sha256:d6c8301b7834c5b9c2b733b10b7e630f441af7bc917c74dba379f24eeeb6a313
Status: Image is up to date for mysql/mysql-server:latest
docker.io/mysql/mysql-server:latest
(base) raja@LENOVO ~ %
```

Run the “mysql-server” docker container with proper parameters:

`$ docker run --name=mysql-server -d -p 3308:3306 mysql/mysql-server:latest`

Get the current one-time root password via docker logs:

`$ docker logs mysql-server`

```
[Entrypoint] GENERATED ROOT PASSWORD: 12I+1T*=8pC1.+ts1lIj#8_Us7Znoa4:
```

Now you can access the “mysql-server” container using the given root password:

```
$ docker exec -it mysql-server mysql -u root -p
```

Enter password:

Change the root password from within mysql:

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'password'
```

Now container is ready with mysql installed:

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'pass'
-> ;
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
ERROR 1046 (3D000): No database selected
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.01 sec)
```

End Tutorial