Cloud Computing Applications and Services
Containers

October 23, 2022

Docker

Docker is the most widely-known container technology. Containers are intended to be a loosely isolated (when compared to Virtual Machine's isolation levels) and lightweight environment to run applications and services. Along this exercise guide, we will go through the steps of configuring and deploying the Swap application, from previous guides, on top of the Docker platform. For this, we will be deploying two Docker containers. One will serve the database management system and the other will serve the remaining Swap services.

Getting started

Tasks

1. Setup an Ubuntu 20.04 virtual machine.
2. Install Docker (`https://docs.docker.com/engine/install/ubuntu/`)
3. (Optional) Add your user to the *Docker* group to avoid using *sudo* all the time (`https://docs.docker.com/engine/install/linux-postinstall/`)
4. Run the sample application (`https://docs.docker.com/get-started/02_our_app/`).
5. Understand the commands `docker {image, ps, exec, stop, start, kill, rm, logs}`. E.g., understand and explore the `docker exec -ti <container_id> /bin/sh` command.
6. Access the application from your browser (`192.168.56.101:3000`).

Network

Similarly with the network adapter created and configured for the Virtual Machines in previous guides, let us set up a network for our containers. In Docker, creating such a network will also allow us to use hostnames to refer to each container, thus simplifying our setup.

Tasks

1. Create a network using the following command: `docker network create <network_name>`
2. Use the `docker network list` and `docker network inspect` commands to check details about the new network.

Note: This network will be used to connect the two Docker containers needed to deploy the Swap application.

Database

Let us start by setting up a MySQL database engine in its own Docker container. For this step, we will use an already prepared image.

Tasks

1. Pull the official MySQL Docker image (`mysql:latest`).
   Explore the `docker image pull <image_name>` command to pull the image.
   Explore the image's documentation at `https://hub.docker.com/_/mysql`.

2. Create a Docker container with a MySQL installation. Run:

   ```
   docker run --name <db_container_name> --net <network_name> -p 3306:3306 -e
       MYSQL_USER=<user> -e MYSQL_PASSWORD=<password> -e MYSQL_DATABASE=<db_name>
       -e MYSQL_ALLOW_EMPTY_PASSWORD=true -d mysql:latest
   ```

3. Run the container and check that mysql is running. For this, connect to the container `docker exec -ti <db_container_name> /bin/sh` and run the command:

   ```
   mysqladmin --host=0.0.0.0 --user=<user> --password=<password> status
   ```

Swap components

Now, let us set up the actual Swap app. Please refer to Guide 1 and your previous Swap installation experience for the next tasks.

1. Explore the Dockerfile reference documentation at
   `https://docs.docker.com/engine/reference/builder/`

2. Explore the Dockerfile instructions (e.g., `FROM`, `RUN`, `WORKDIR`, `COPY`, `EXPOSE`, `CMD`) and set up a Dockerfile to build a Docker image for the Swap app.

   Such app should use the MySQL installation configured earlier as its database engine. Please note that you should configure the environment accordingly (`.env`).

   Also, use Ubuntu 20.04 as the base image for the Dockerfile (`FROM` instruction).

   Note: The task for seeding the database should be ignored in this phase.

3. Use the command `docker build` to build the Swap Docker image. For example:

   ```
   docker build . -t <swap_image_name>
   ```

4. Use the command `docker run` to run the app. For example:

   ```
   docker run --net <network_name> -p 8000:8000 --name <swap_container_name> -d
       <swap_image_name>
   ```

   Note: This is only an example, you should add any other aditional container flags that your image may require.

5. Test your setup by accessing the app through a web browser from your host machine.

6. Ensure that the variables `DB_HOST`, `DB_DATABASE`, `DB_USERNAME`, and `DB_PASSWORD`, defined at Swap's `.env` configuration file, are dynamically set when running the Swap container and are not hardcoded on the Dockerfile (for example, by using the instruction `ENV`).

Extras

1. Modify the Swap Docker image so that the database seed task can run optionally (i.e., only when the user asks).

2. Setup Redis in a separate container and add such component to the Swap deployment.

Learning outcomes Hands-on experience with software container technology (Docker). Deployment of an application on top of a container platform.