Cloud Computing Applications and Services

Containers Part I

November 6, 2021

Docker

Docker is the most widely-know 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 virtual machine (or reuse an existing one).
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 usage of the commands *docker image, ps, exec, stop, kill, rm*.
   E.g., understand and explore the *docker exec -ti "container id" /bin/sh* command.
6. Access the application from your browser (10.0.0.3:3000).

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 simplifying our setup.

Tasks

1. Create a network using the following command:

   ```
   docker network create swap_net
   ```

2. Use the *docker network list* and *docker network inspect* commands to check details about the new network.

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. In the next step, we will be setting up the remaining Swap services from a Docker file defined by us.

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 swapdb --net swap_net -p 3306:3306 -e MYSQL_ROOT_PASSWORD=
       swap -e MYSQL_DATABASE=swap -d mysql:latest
   ```

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

   ```
   mysqladmin --host=0.0.0.0 --user=root --password=swap 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. Setup a Docker file to build a container with the Swap app (e.g. *docker build . -t swap*). Such app should use the MySQL installation configured earlier as its database engine. Please note that you should configure the environment accordingly (*.env*).

2. Use Docker to run the app (e.g. *docker run --net swap_net -p 8000:8000 --name swap swap*).

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

Extras

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

Questions

1. Why are containers labeled as "lightweight"?

2. What could be a potential disadvantage of containers when compared with VMs?

3. Let us imagine you are responsible for the development and maintenance of the Swap application in production. Could you think of one use case for containers in such context?

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