ceedcv
CENTRE ESPECÍFIC
D´EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

UT 11.
DOCKER
Activities

Computer Systems
CFGS DAW

Álvaro Maceda
a.macedaarranz@edu.gva.es

2022/2023
Version:230227.0717

License

Nomenclature

Throughout this unit different symbols will be used to distinguish important elements within the content. These symbols are:

📚 Important

⚡ Attention

📢 Interesting

UT 11. Docker

Activities

You will need Docker installed in your machine in order to carry out these exercises. If you don't want to install Docker in your computer, remember that you can use a virtual machine and install Docker there.

Take a look often to your containers with `docker ps -a` until you can understand easily what is happening.

0. Exercise 0

Run a container from the `hello-world` image from your computer. This will help you to test your Docker installation.

Remember to run `docker ps -a` after that.

1. Exercise 1

Do the following using `docker create` and `docker start` commands. You will need to launch additional commands, too:

1. Create three files named `index.html` in three different directories. The content of the files will be like this (change **`1`** and **`FIRST`** for the corresponding ordinal):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Page 1</title>
</head>
<body>
  <h1>I am the FIRST web server</h1>
</body>
</html>
```

2. Launch three different HTTP web servers using the official image httpd:

   https://hub.docker.com/_/httpd

3. Open the three web pages at the same time in your browser
4. Stop and destroy the containers

Repeat the exercise, but this time using `docker run` instead of `docker create` + `docker start`.

2. Exercise 2

You need to solve a bug in an old Python script, written in Python version 2.7.18. The corrected version of the script is this:

```
import sys
```

```
param = sys.argv[1]
print param[::-1]
```

The script must be launched with `python reverse.py “This is a string”`.

Open a shell in a `python:2.7.18` container, write the script and test that it works (you will need to install a text editor like `vim` or `nano`) Try to open a shell for editing the file and another one for executing the script.

Destroy the container and create a new one to test it again. Is your script there? Do you need to reinstall the text editor?

3. Exercise 3

Part 1

In this exercise, you will run a database server and create a database using Docker.

1. First, create and run a MySQL server container with the official image:

   https://hub.docker.com/_/mysql

   Do NOT launch it in detached mode: a new MySQL server takes a while to initialize the first time it runs and you will need to wait until the initialization is complete (with a message like “ready for connections”)

Once the server is running, you will need to test it. For that, you will need a MySQL client.

2. The server already has a client installed. Open a shell to the server and connect to the database server executing this command in the server’s shell:

```
mysql --user root --password=secret
```

Once connected, you can show the databases and create one with these commands:

```
show databases;
create database my_database;
show databases;
```

You can create some tables and populate them if you want.

3. Now, test the server from your computer. Install `mysql-client` in your system. To connect to a MySQL server, you can use this command:

```
mysql --host=127.0.0.1 --port 3306 --user=root --password=secret
```

That command will connect to a database server in your computer, listening at port 3306. You will need to redirect that port from the server to your computer, or the client won’t be able to reach the server.

Create another database

4. Stop and destroy the server. Connect to the server again. Are your databases there?

Part 2

You want to keep your database with all your data when the container is destroyed:

1. Create a Docker volume for your database server
2. Launch the database server mounting the volume at `/var/lib/mysql`
3. Create a database on the server, using the method that you prefer
4. Destroy the container and launch a new one mounting the volume in the same directory:
   - Has the new container been initialised faster?
   - Is it your database there?

You can check the space that each image, container and volume uses in your system with `docker system df -v`. Run it and see what you have there.

Remember to destroy the volume once you have finished the exercise. Run `docker volume ls` to check it.

Part 3

This time, you will use bind mounts to store the database:

1. Create an empty directory in your machine
2. Launch the database server mounting your host's directory at `/var/lib/mysql`
3. Create a database on the server, using the method that you prefer
4. Destroy the container and launch a new one mounting the volume in the same directory. Is it your database there?
5. Who is the owner of the files created on your computer? Why?