Docker-Compose

Overview

Docker Compose allows you to define and run multi-container Docker applications. With Compose, you use a YAML (YAML Ain't markup language, a human readable serialisation languate, used for configuration files) file to configure your application's services. It is easy to create and start all the services from YAML configuration with a single command. For example, we can create and manage webservers using a simple YAML file.

- Part 1: Installing Docker and Docker Compose
- Part 2: Setting up Docker Compose
- Part 3: Running the Servers

Learning outcomes

After completing this lab, students will lear:

- the basic usage of Docker Compose
- how to create a simple YAML file
- how to build, run, and manage mutiple web servers

Part 1: Installing Docker and Docker Compose

Install Docker Compose:

```
sudo apt install docker-compose
```

Check if the Docker is running with:

```
sudo service docker status
```

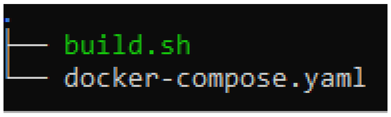
if it's not running, start Docker with:

```
sudo service docker start
```

Part 2: Setting up Docker Compose

First create two files:

1. docker-compose.yaml
2. build.sh



```
.
├── build.sh
└── docker-compose.yaml
```

Figure 1: screenshot: create docker-compose.yaml and build.sh files

You can download the Docker-compose.yaml file from here

and build.sh content is here:

```
#!/bin/bash
for i in 1 2 3
do
mkdir nginx$i
echo "FROM nginx:latest
COPY ./index.html /usr/share/nginx/html/index.html" > nginx$i/Dockerfile
```

```
echo "<html> <h1> Hello from container $i </h2> </html>" > nginx$i/index.html
done
docker-compose up -d
```

Part 3: Running the Servers

First, run the build.sh script in your terminal.

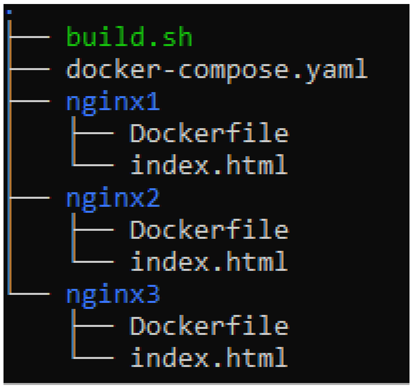
Make it executable first with:

```
chmod +x build.sh
```

and then:

```
./build.sh
```

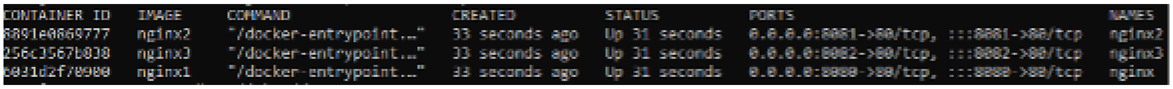
After running the the `build.sh` script, you will get a following file structure:



```
.
├── build.sh
├── docker-compose.yaml
├── nginx1
│   ├── Dockerfile
│   └── index.html
├── nginx2
│   ├── Dockerfile
│   └── index.html
└── nginx3
    ├── Dockerfile
    └── index.html
```

Figure 2: screenshot: file structure after running the build script

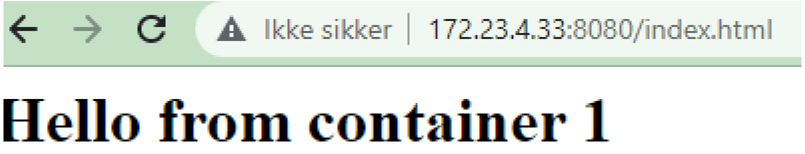
and run `docker ps` to see the container structure:



```
CONTAINER ID   IMAGE    COMMAND                  CREATED          STATUS          PORTS                                   NAMES
8891e0869777   nginx2   "/docker-entrypoint.…"   33 seconds ago   Up 31 seconds   0.0.0.0:8081->80/tcp, :::8081->80/tcp   nginx2
256c3567b038   nginx3   "/docker-entrypoint.…"   33 seconds ago   Up 31 seconds   0.0.0.0:8082->80/tcp, :::8082->80/tcp   nginx3
b031d2f70900   nginx1   "/docker-entrypoint.…"   33 seconds ago   Up 31 seconds   0.0.0.0:8080->80/tcp, :::8080->80/tcp   nginx
```

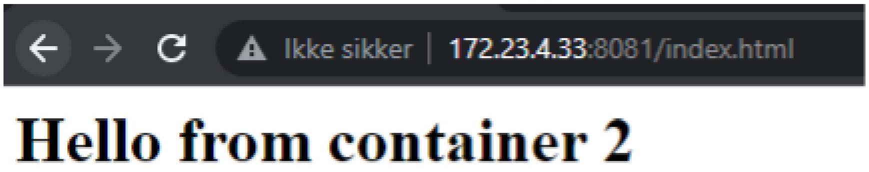
Figure 3: screenshot: container structure after running the build script

We now have 3 web servers running in three separate containers. To access them, go to `localhost:8080`, `localhost:8081` and `localhost:8082` in your favorite browser.



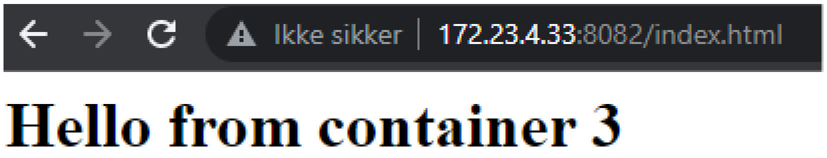
Hello from container 1

Figure 4: screenshot: container 1



Hello from container 2

Figure 5: screenshot: container 2



Hello from container 3

Figure 6: screenshot: container 3