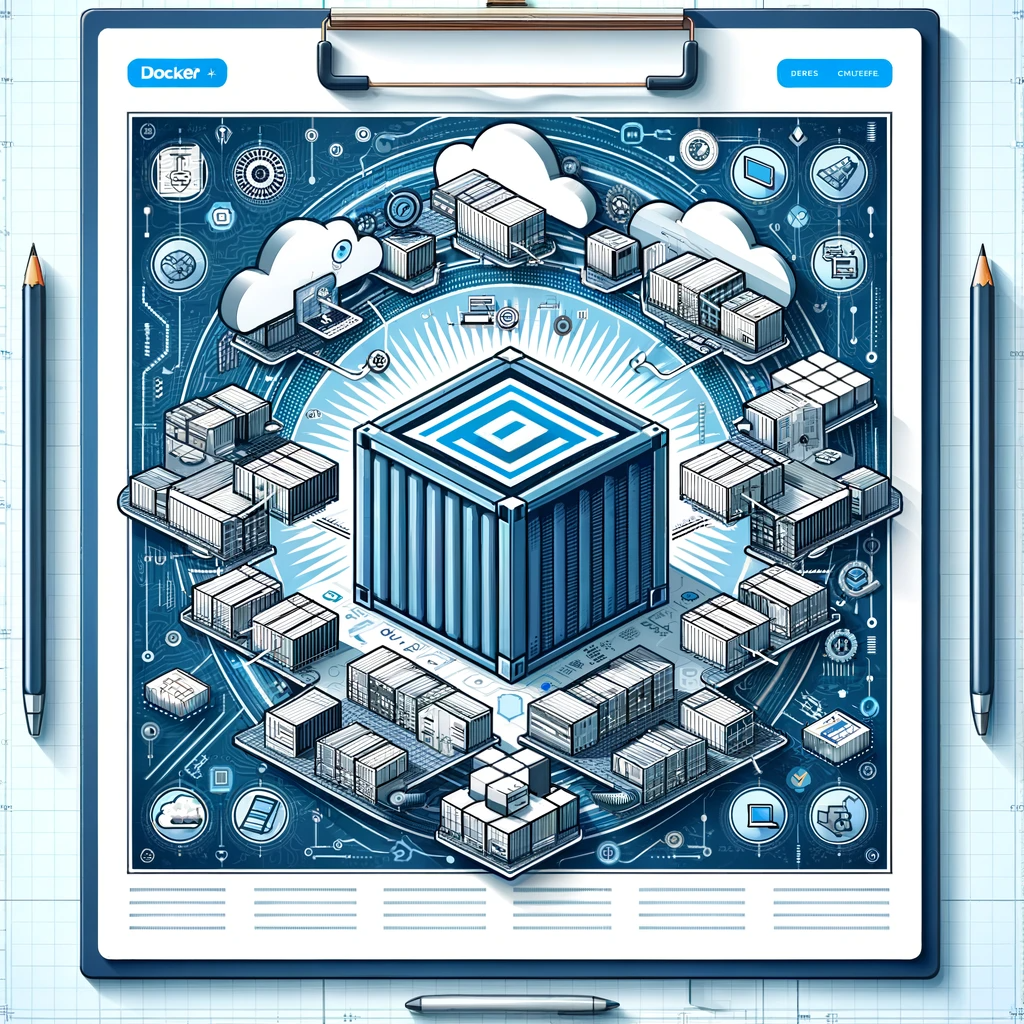
# Week 02 - Docker Worksheet

Louis Botha, louis.botha@tuni.fi



When seeing this emoji , take a screenshot and paste on the worksheet.

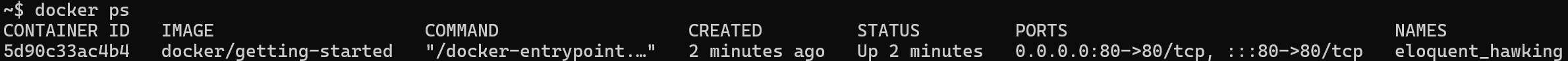
Return the worksheet to Moodle as a **PDF**.

1.  *Do the exercise outside of your git repository directory*

Start the exercise by running the following command:

docker run -d -p 80:80 docker/getting-started

1. Use a docker command to check if the container is running .



1. Run the following command:

docker run -d -p 80:80 docker/getting-started

* Did it work? If not why not?
* It did not work because port 80 is already allocated

Run the following command:

docker run -d -p 8080:80 docker/getting-started

* Did it work? Yes
* What is the differrence? Port 8080 wasn’t allocated yet.

Use a docker command to check how many containers are running 

A screenshot of a computer screen

Description automatically generated

1. User a docker command to check how many docker/getting-started images are on the machine .



1. Use the docker inspect <id> and check what is the NetworkSettings/Ports of the container .

A screen shot of a computer program

Description automatically generated

1. Use the necessary docker commands to stop all the running containers .

A black screen with white text

Description automatically generated

1. Copy the following content into a file named Dockerfile (the file does not have a type).

FROM nginx

Build the image

docker build -t nginx-web-server .

Use a docker command to check that the image were build and on the local machine .

A screenshot of a computer

Description automatically generated

Run the container with docker run nginx-web-server.

* Noticed what is happening? The container is running in the foreground.
* Press CTRL + C to exit the container and stop it.

Run the container with docker run -d nginx-web-server.

* Noticed what happened, what does the -d option do? It runs the container in the background

Try to access the container in a web browser by opening localhost.

* Did it work? If not, why not? Did not work and don’t exactly know why.
* Stop the running *nginx-web-server* container.

Run the container with docker run -d -p 80:80 nginx-web-server.

* Did it work?
* What is the difference?
* What does the -p 80:80 do?

Try to access the container in a web browser by opening localhost .

A black screen with white text

Description automatically generated

Use a docker command to stop the running containers.

1. Create a directory called webserver.

Inside the webserver directory, create a directory called html.

Inside the html directory, create a file called index.html with the following content:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Hello Nginx!</title>

</head>

<body>

  <h1>Hello Nginx!</h1>

</body>

</html>

The directory structure should look like this:

❯ tree

.

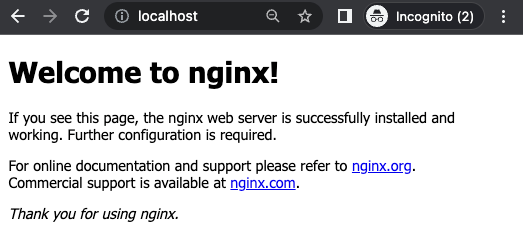
└── webserver

    └── html

        └── index.html

Run the container with docker run -d -p 80:80 nginx inside the websever directory.

Try to access the container in a web browser by opening localhost. You should see the **nginx** welcome text.



Notice that we don’t see the index.html content we created above.

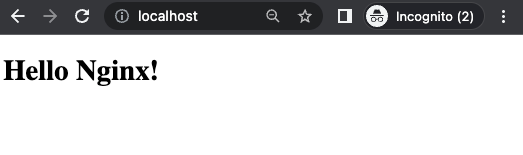
Stop the running container.

Run the container with:

docker run -d -p 80:80 -v <absolute path>/webserver/html/:/usr/share/nginx/html nginx

 Use pwd to get the absolute path

Try to access the container in a web browser by opening localhost. You should now see the index.html content we created above.



Make a change to the index.html file and check it in the browser .

 A hint, start by changing the file and refreshing the page int he browser and see if it works.

A close up of a sign

Description automatically generated

Noticed what the -v <source>:<destination> option in the docker command allows us to do?

Use a docker command to stop the running containers.

1. Inside the webserver directory create a Dockerfile with this content:

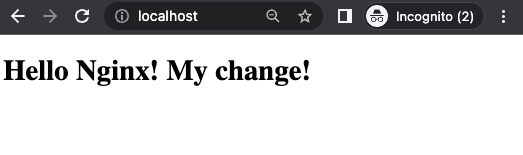
FROM nginx

COPY /html /usr/share/nginx/html

Run the command docker build -t webserver . to build the container.

Run the command docker run -p 80:80 webserver to run the container.

Try to access the container in a web browser by opening localhost. You should now see the latest index.html content that was created above.



Make some changes to the index.html file and refresh the web browser.

Notice any changes? If not, why not? Because the image was build on the index.html file before it was changed.

Run the command docker build -t webserver . to build the container.

Run the command docker run -p 80:80 webserver to run the container.

Try to access the container in a web browser by opening localhost. You should now see changes made to the index.html content above.

Use a docker command to stop the running containers.

1. Copy the following content into a file named docker-compose.yml.

services:

  web:

    image: nginx

    ports:

    - "8080:80"

Use a docker compose command to run the services on the local machine 

A screen shot of a computer screen

Description automatically generated

Try to access the container in a web browser by opening localhost

* Did it work? If not, why not? It worked when opened localhost:8080

Try to access the container in a web browser by opening localhost:8080

* Did it work?
* What is the difference?
* What does the line - "8080:80" do?

 10. Use a docker compose command to stop the services on the local machine 



1. Change the following content in the docker-compose.yml file.

services:

  web:

    image: nginx

    ports:

    - "8080:80"

    volumes:

    - ./html:/usr/share/nginx/html

Use a docker compose command to run the services on the local machine.

Try to access the container in a web browser by opening localhost:8080.

You should now see the latest index.html content that was created above .

A screen shot of a computer

Description automatically generated

Make a change to the index.html file and check it in the browser .

 A hint, start by changing the file and refreshing the page int he browser and see if it works.

A close up of a message

Description automatically generated

 Use a docker compose command to stop the services on the local machine.