Introduction to Docker

Short description

Lab focus: Introduction to Docker containers and environment setup.
Submission deadline: This is not a graded assignment but finishing it will help you a lot with the next graded assignments.

Exercise One - Setup your environment and test Docker

Task 1: Visit the Docker webpage at (https://www.docker.com/), and create an account by choosing the "Sign up" button.

Task 2: Download and install Docker.

Task 3: Once Docker is installed, open the Docker app and sign in with your account. Please make sure that the Docker deamon is running whenever you want to use Docker (as shown in figure 1).

Task 4: Complete the tutorial from the Docker Desktop. You can find it by going to the Docker image shown in your task bar (see figure 1) and select the "Quick start guide", where you build your first container.

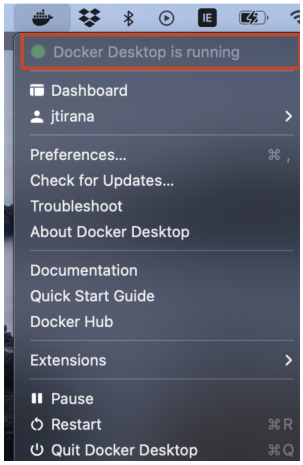

Figure 1:

Below you can see a small description of the steps, but you should use the commands as are shown inside the Docker's tutorial.

Step 1: Download the repository that contains all the important files to build your first image. This is done by executing a git clone command through Docker's environment.

Step 2: Go to the "getting-started" folder from the repository you just cloned and build the image using the command:

```
docker build -t docker101tutorial .
```

NOTE! With the `docker build [OPTIONS] PATH | URL | -` command you can build Docker images from a Dockerfile and a "context". The build's context is the set of files located in the specified PATH or URL. In the example's case we use the currently working PATH by inputing '.'

NOTE! With option `-t` we tag the resulting image with the name "docker101tutorial"

For more insights for the docker build command visit the documentation (external resource #3)

Step 3: You can start a container based on the image you have already created by using the command:

```
docker run -d -p 80:80 --name docker-tutorial docker10tutorial
```

Module Coordinator: Dimitris Chatzopoulos- School of Computer Science - UCD - dimitris.chatzopoulos@ucd.ie
Teaching Assistant: John Byabazaire- School of Computer Science - UCD - john.byabazaire@ucd.ie

Let's take a look at the arguments of this command.

1. -d: run on the background
2. -p 80:80: map port 80 of the host to port 80 in the container
3. -name docker-tutorial: we define a name for the container. This is one of the the three ways you can identify the container. The other two ways are by the UUID long and short identifier.
4. examine how the `docker run` command works in detail on the reference (external resource #4)

Step 4: You can save your image by using the command:

```
docker tag docker101tutorial yourdockerusername/docker101tutorial:tag
```

Step 5: You can upload your images on the Docker Hub* and allow other users to create containers with your image. To upload your image use the following command:

```
docker push yourdockerusername/docker101tutorial:tag
```

Step 6: After that you can see your uploaded images on: https://hub.docker.com/repositories

Exercise Two - Use an existing image

Task 0: Open your terminal and let' s clean our system. Clean up the containers using their IDs. First find the IDs using the following command:

```
docker ps
```

Stop the containers using their IDs using `docker stop <container-id>` and then remove them using `docker rm <container-id>`. Alternatively, you can stop and remove the containers through the Docker's Dashboard.

Task 1: Create a Busybox container using the following command:

```
docker run -it --rm busybox
```

BusyBox combines tiny versions of many common UNIX utilities into a single small executable. Use the produced container to browser via the command line and test various basic commands (e.g., ls, mkdir, top, whoami, ping, etc). You can stop the container by typing `exit` or by pressing `ctrl D`.

Task 2: Send an HTTP request through an interactive container using Docker internal network.

Step 1: start a nginx web server container using the following command:

```
docker run -d -p 80:80 --name web nginx
```

Check the server by visiting http://localhost/ with your browser.

Step 2: create the BusyBox container like previously, but this time link it with the port 80.

```
docker run -it --rm --link web:web --name web_test busybox
```

Step 3: Open the busybox shell and run the following command to send a HTTP request to port 80:80

```
wget -O - http://web:80
```

Step 4: Do not forget to clean your machine, by applying the steps from Task 0 .

External resources

(1) https://hub.docker.com/

(2) https://docs.docker.com/engine/reference/commandline/build/

(3) https://docs.docker.com/engine/reference/run/