SA-MIRI 2025

Practice Pb: Containers

Jakub Seliga (jakub.seliga@estudiantat.upc.edu)
Thomas Aubertier (thomas.aubertier@estudiantat.upc.edu)

Task 3.4 Install Docker (Linux)

- Source : https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository

```
$ sudo apt-get update

$ sudo apt-get install ca-certificates curl

$ sudo install -m 0755 -d /etc/apt/keyrings

$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc

$ sudo chmod a+r /etc/apt/keyrings/docker.asc

$ echo \ "deb [arch=$(dpkg --print-architecture)
signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \ sudo
tee /etc/apt/sources.list.d/docker.list > /dev/null

$ sudo apt-get update
```

Verify if Docker is correctly installed

```
$ docker --version
```

Task 3.4 Install Docker (Windows)

- User can have its permission denied from Docker. In this case, we have to add ourself to the Docker privilege group

```
$ sudo usermod -a -G docker $USER
```

```
$ ssh-keygen -t rsa
```

```
$ ssh-copy-id nct01XXX@glogin1.bsc.es
```

Enter current password to validate

```
$ ssh nct01XXX@glogin1.bsc.es
```

Task 3.5 Download Docker Image

- User can have its permission denied from Docker. In this case, we have to add ourself to the Docker privilege group

  ```
  $ sudo usermod -a -G docker $USER
  ```

  ```
  $ newgrp docker
  ```

  Verification

  ```
  $ grep docker /etc/group
  ```

- Now we can download the Docker image

  ```
  $ docker pull jorditorresbcn/dl
  ```

Task 3.6 Run Docker Image

- Launch the downloaded image

  ```
  $ docker run -it jorditorresbcn/dl:latest
  ```

  Verification (in second terminal with same privileges)

  ```
  $ docker ps
  ```

```
CONTAINER ID   IMAGE                      COMMAND       CREATED          STATUS          PORTS                NAMES
822e39ebce99   jorditorresbcn/dl:latest   "/bin/bash"   4 seconds ago    Up 3 seconds    8888/tcp, 8954/tcp   friendly_lehmann
```

Task 3.7 Stop Docker Image

- Using the “`CONTAINER ID`” gotten in the previous step, we can identify and kill the image

  ```
  $ docker stop 822e39ebce99
  ```

- We see that it is equivalent to type “exit” in the first terminal (in interactive mode)

  ```
  root@822e39ebce99:/app# exit
  ```

Task 3.8 Run Docker with Port Mapping

- Launch the downloaded image with a specified port

  ```
  $ docker run -it -p 8888:8888 jorditorresbcn/dl:latest
  ```

  Verification (in second terminal with same privileges)

  ```
  $ docker ps
  ```

```
CONTAINER ID   IMAGE                      COMMAND       CREATED          STATUS          PORTS                                                NAMES
4c1e276704c2   jorditorresbcn/dl:latest   "/bin/bash"   6 seconds ago    Up 6 seconds    0.0.0.0:8888->8888/tcp, [::]:8888->8888/tcp, 8954/tcp   stoic_wiles
```

Task 3.9 Start the Jupyter Notebook Server

- Launch Jupyter Notebook with a specified port from the Docker Image :

  ```
  $ jupyter notebook --ip=0.0.0.0 --port=8888 --no-browser --allow-root
  ```

  Port 8888 must not be used prior.

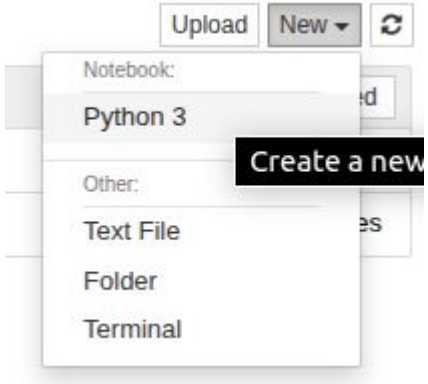
- `--no-browser` avoid to open the browser by default. Instead, it gives the following link :

  ```
  [I 09:34:43.929 NotebookApp] The Jupyter Notebook is running at:
  [I 09:34:43.929 NotebookApp] http://0.0.0.0:8888/
  ```

  Clicking on it successfully display Jupyter.

Task 3.10 Create and Run a Test Notebook

- Create new notebook



- Test line

```
In [1]: print("hello docker I am a human")
```

```
hello docker I am a human
```