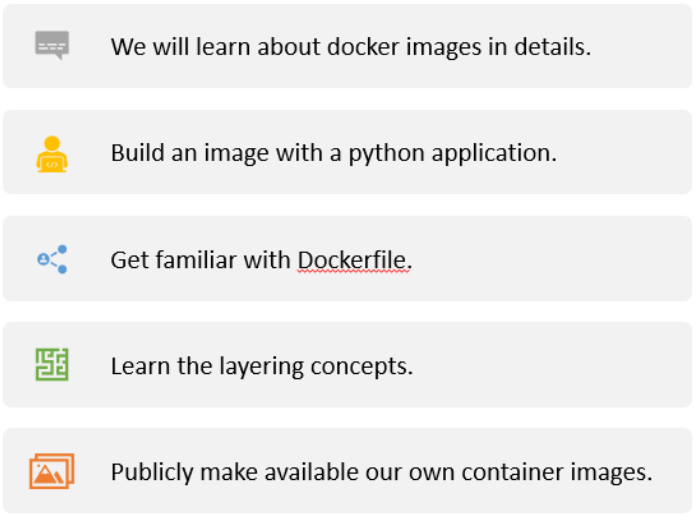
Lab Report 4 (Docker II) - Antonio Manuel Luque Molina



Types of Docker Images:

- Base Images – These images don't have any parent images, for example, ubuntu, Debian, busybox , etc.
- Child Images – These images are built on base images with some added functionality like hello-world.

There is one more way to distinguish images:

- Official – These images are officially maintained and supported by people at docker. Example includes ubuntu, hello-world, etc.
- User - These images are created by users like you. These are built on base images and some functionalities are added. The name format is user/imagename.

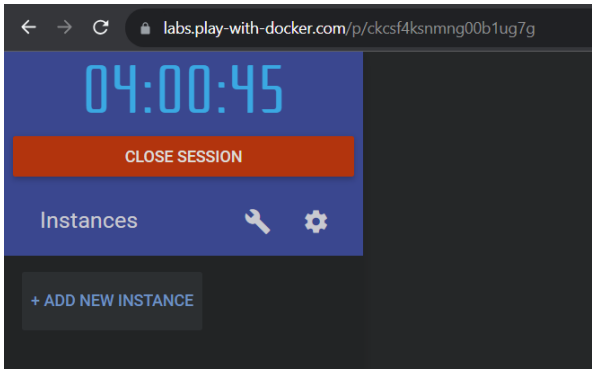
We need to make a Dockerfile to build an image with this application;

- A Dockerfile is a simple text file that contains a list of commands that the Docker client calls while creating an image.
- The commands are like linux commands, so no need to learn syntax to write a Dockerfile.

Using Docker Playground:

We go to the link: https://labs.play-with-docker.com/

We click on Start,

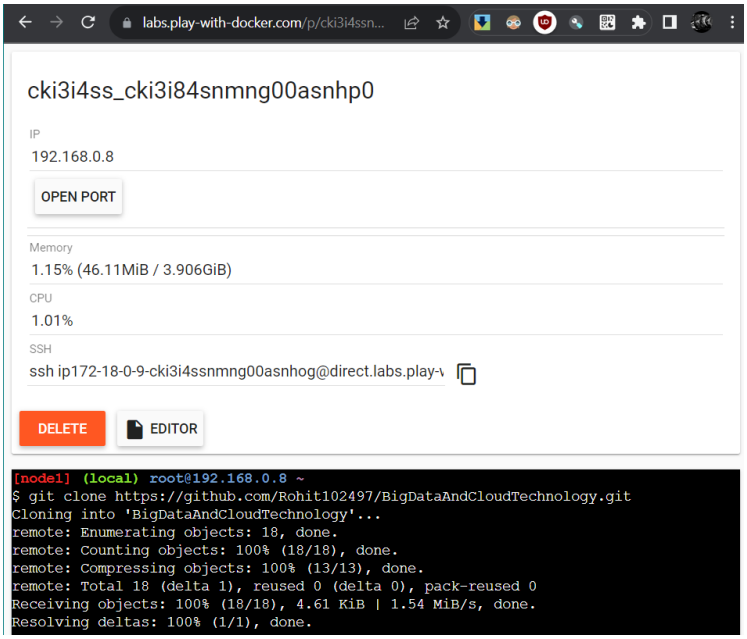


We click on Add New Instance,

This creates a VM instance for us to run docker commands.

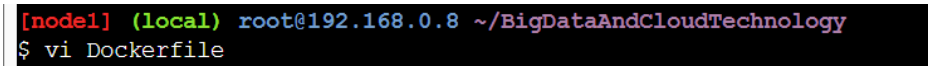
Building our first image:

We will clone this link: https://github.com/Rohit102497/BigDataAndCloudTechnology

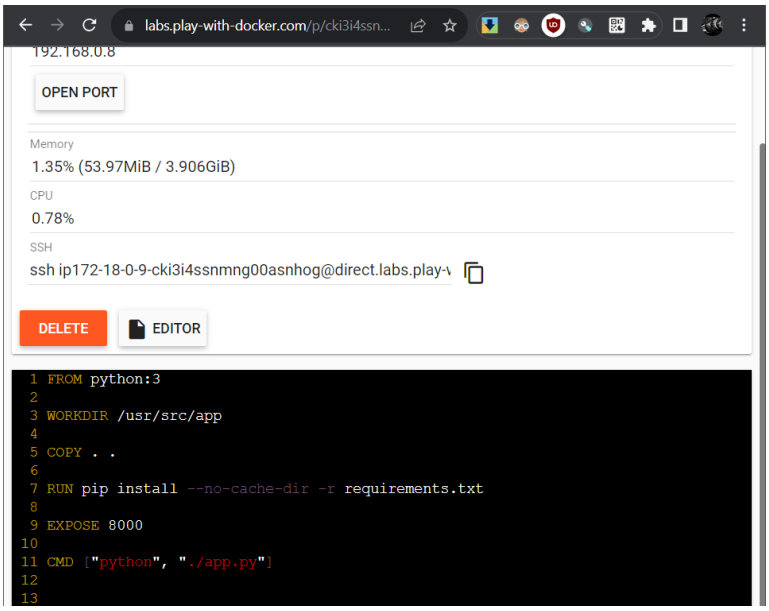


Now We have cloned it and We will access it:

We need to make a Dockerfile to build an image with this application, so:



We press “i” to insert text and We write the following:



After that We press esc and We write :wq to save and quit it.

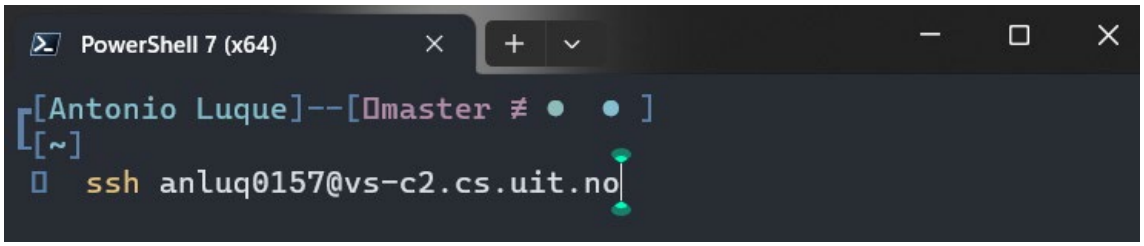
In the next step We will build the image:

```
[node1] (local) root@192.168.0.8 ~/BigDataAndCloudTechnology
$ docker build -t rag004/doggif .
[+] Building 44.4s (9/9) FINISHED
```

Once it is builded We will run it:

Now, clicking in the port We are able to see the webpage.

Using Server Access:



This command initiates an SSH (Secure Shell) connection to the server at vs-c2.cs.uit.no.

Building our first image:

We will clone this link: https://github.com/Rohit102497/BigDataAndCloudTechnology

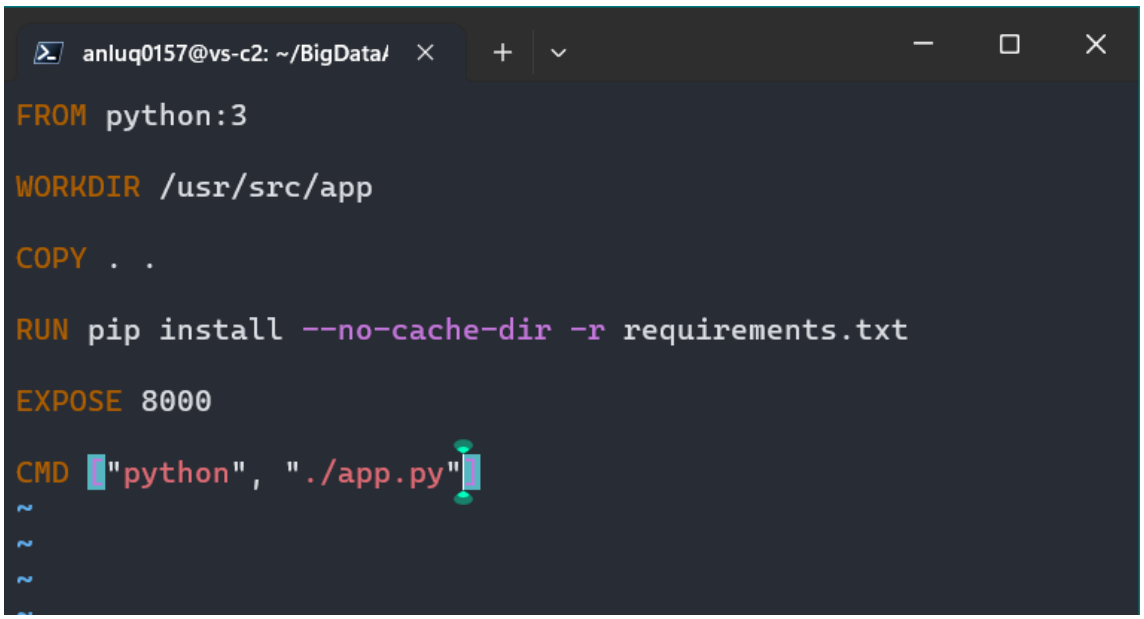
```
anluq0157@vs-c2:~$ git clone https://github.com/Rohit102497/
BigDataAndCloudTechnology.git
```

Now We have cloned it and We will access it:

```
anluq0157@vs-c2:~$ ls
BigDataAndCloudTechnology  bin
anluq0157@vs-c2:~$ cd BigDataAndCloudTechnology/
anluq0157@vs-c2:~/BigDataAndCloudTechnology$ ls
app.py  Dockerfile  requirements.txt  templates
anluq0157@vs-c2:~/BigDataAndCloudTechnology$
```

We need to make a Dockerfile to build an image with this application, so:

```
anluq0157@vs-c2:~/BigDataAndCloudTechnology$ vi Dockerfile
```



After write that we will type :wq in order to save and quit.

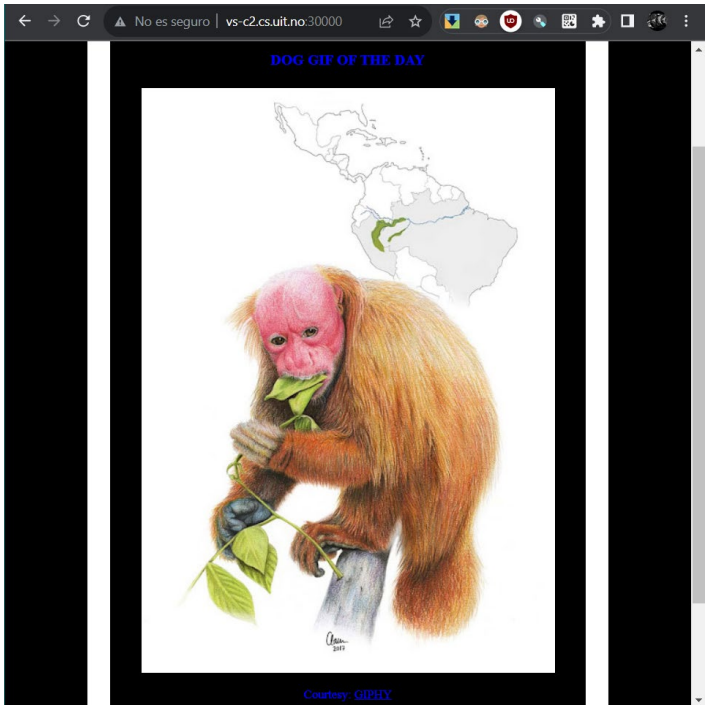
In the next step We will build the image;

```
anluq0157@vs-c2:~/BigDataAndCloudTechnology$ docker build -t rag004/doggif .

DEPRECATED: The legacy builder is deprecated and will be removed in a future
 release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon   72.7kB
Step 1/6 : FROM python:3
 ---> b94d01b49295
Step 2/6 : WORKDIR /usr/src/app
 ---> Using cache
 ---> 16583913156b
Step 3/6 : COPY . .
 ---> Using cache
 ---> c6dd0e11ef7a
Step 4/6 : RUN pip install --no-cache-dir -r requirements.txt
 ---> Using cache
 ---> 461441a13aee
Step 5/6 : EXPOSE 8000
 ---> Using cache
 ---> cd516ded899e
Step 6/6 : CMD ["python", "./app.py"]
 ---> Using cache
 ---> 459badd4dfb0
Successfully built 459badd4dfb0
Successfully tagged rag004/doggif:latest
```

Once it is builded We will run it:

```
anluq0157@vs-c2:~/BigDataAndCloudTechnology$ docker run -p 30000:8000 --name
 doggif rag004/doggif
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production depl
oyment.
   Use a production WSGI server instead.
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deploym
ent. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8000
 * Running on http://172.17.0.2:8000
Press CTRL+C to quit
```

Now, We can go to the webpage to see it in the web browser:

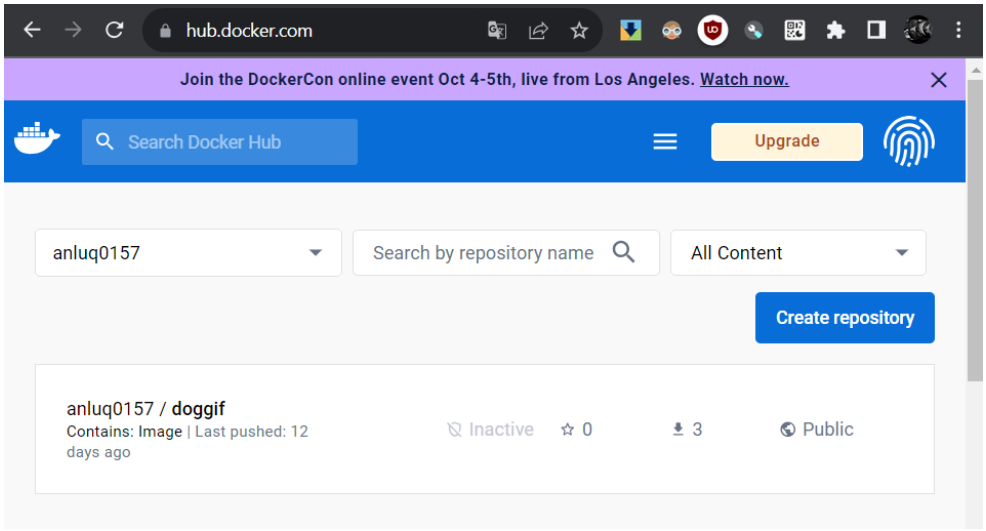


This view is with my update in the application app.py.

Now We will save our image to our docker hub.

Firs of all We will do **docker login** in the console to login in dockerhub, and after that We will write docker **push rag004/doggif** to uplading our image to our docker hub.

Finally, We can see our image there:



We can pull our image and run it in our system and We can pull our peers images too.

Thank you for reading!