**SSH to your AWS Workstation**

**ssh devops@<public-ip-addr**> of your Workstation  
Password is : **Dev0p$!!/**

1. SSH to your AWS Workstation.

|  |
| --- |
| $ sudo su |

2. Clone the Git Repo that contains the python code.

|  |
| --- |
| # cd /home/devops/ # git clone https://github.com/LovesCloud/python-docker-mercedes.git # cd python-docker-mercedes/ |

3. Create Dockerfile in the /python-docker directory.

|  |
| --- |
| # vim Dockerfile |

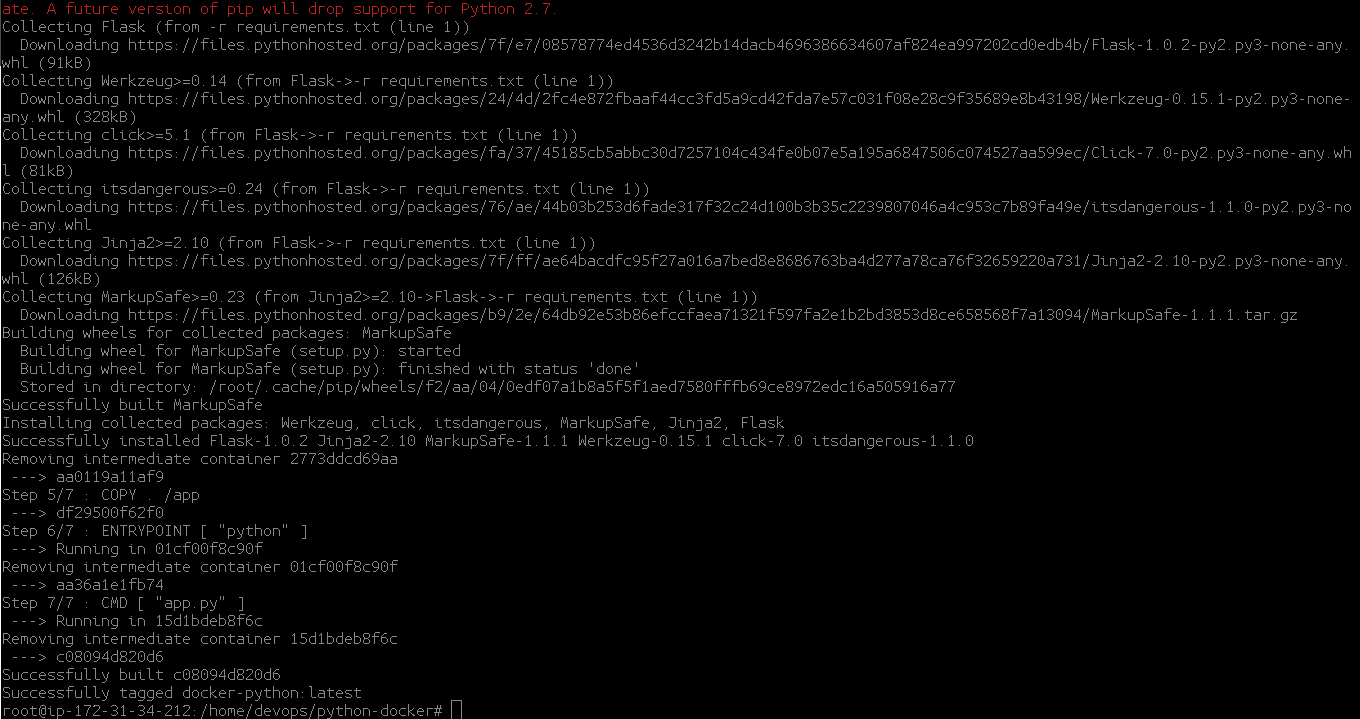
4. Paste the below script

|  |
| --- |
| FROM python:2.7-alpine COPY ./requirements.txt /app/requirements.txt WORKDIR /app RUN pip install -r requirements.txt COPY . /app  ENTRYPOINT [ "python" ] CMD [ "app.py" ] |

Save and exit by vim editor by pressing (:wq and press Enter)

5. Run the below command from python-docker/ directory

|  |
| --- |
| # docker build . -t docker-python-mercedes |



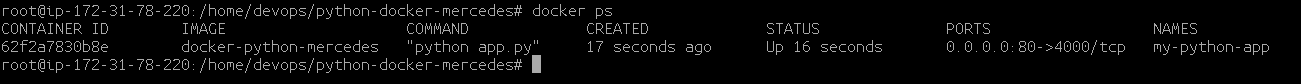
6. Once the build is successful run the below command to check the build details.

|  |
| --- |
| # docker images |

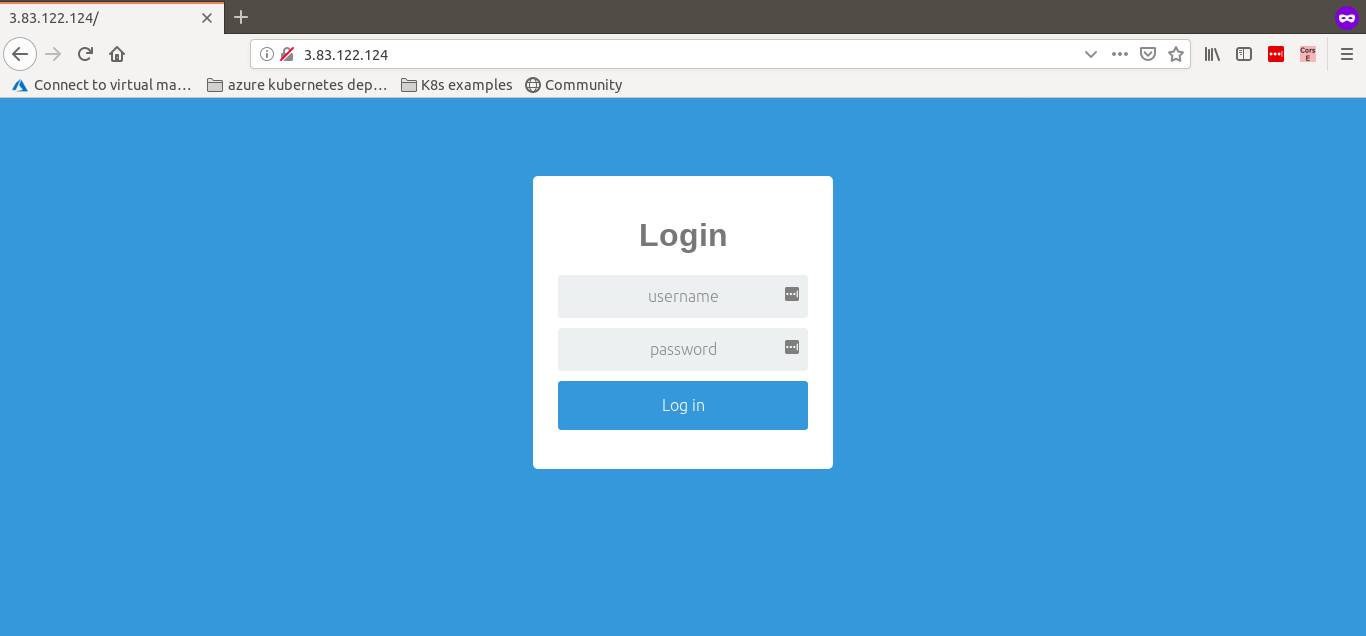


7. Run your container

|  |
| --- |
| # docker run -d -p 80:4000 --name my-python-app docker-python-mercedes  # docker ps |

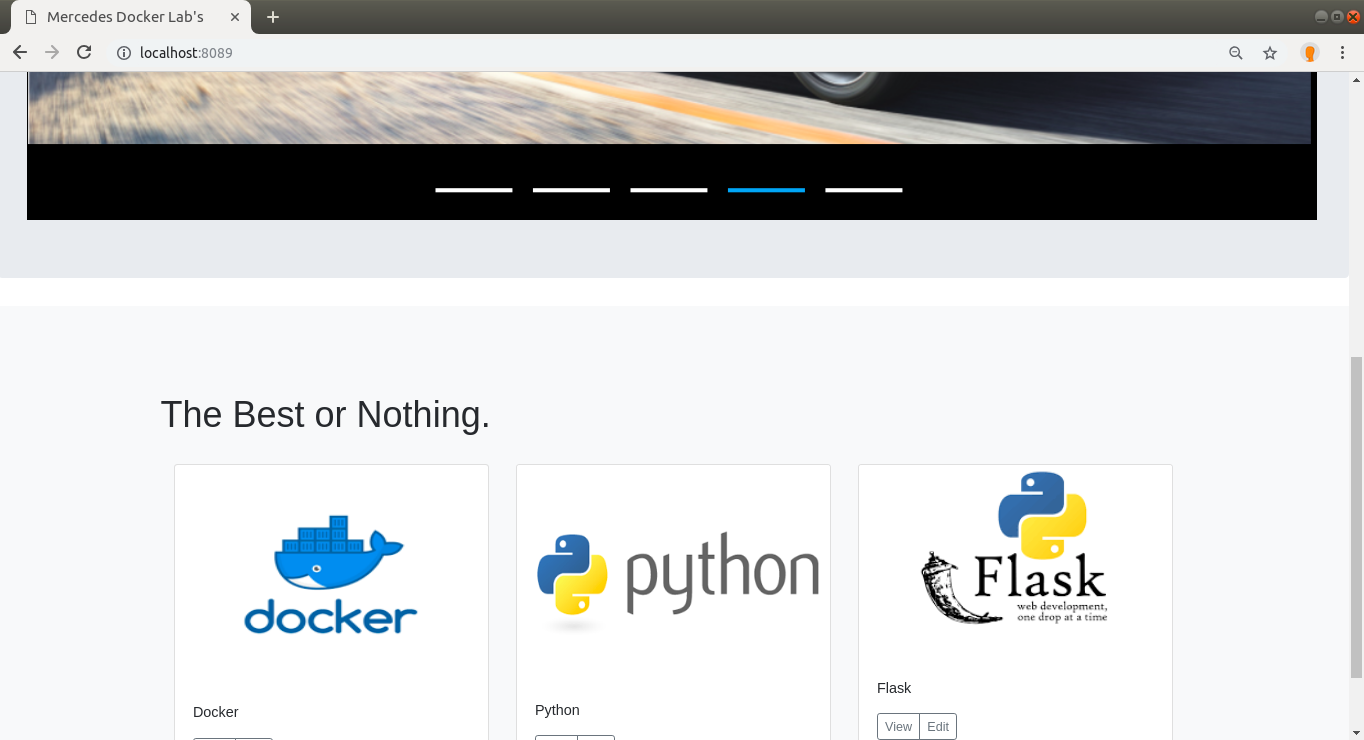


The Python Docker Container running on port 80, and can be accessed from the public IP of the AWS Workstation on default Port 80

http://<public-ip-address-of-aws-workstation>

8. Login to the Application using the below credentials.

|  |
| --- |
| Username : admin Password : password |



**STOP THE CONTAINER BEFORE PROCEEDING TO THE NEXT LAB**

9. Stop the container by running the below command.

|  |
| --- |
| # docker stop my-python-app |