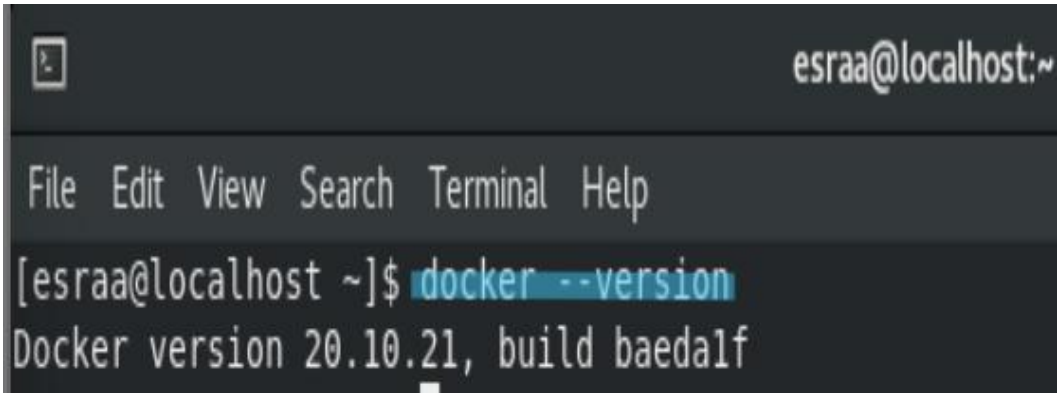
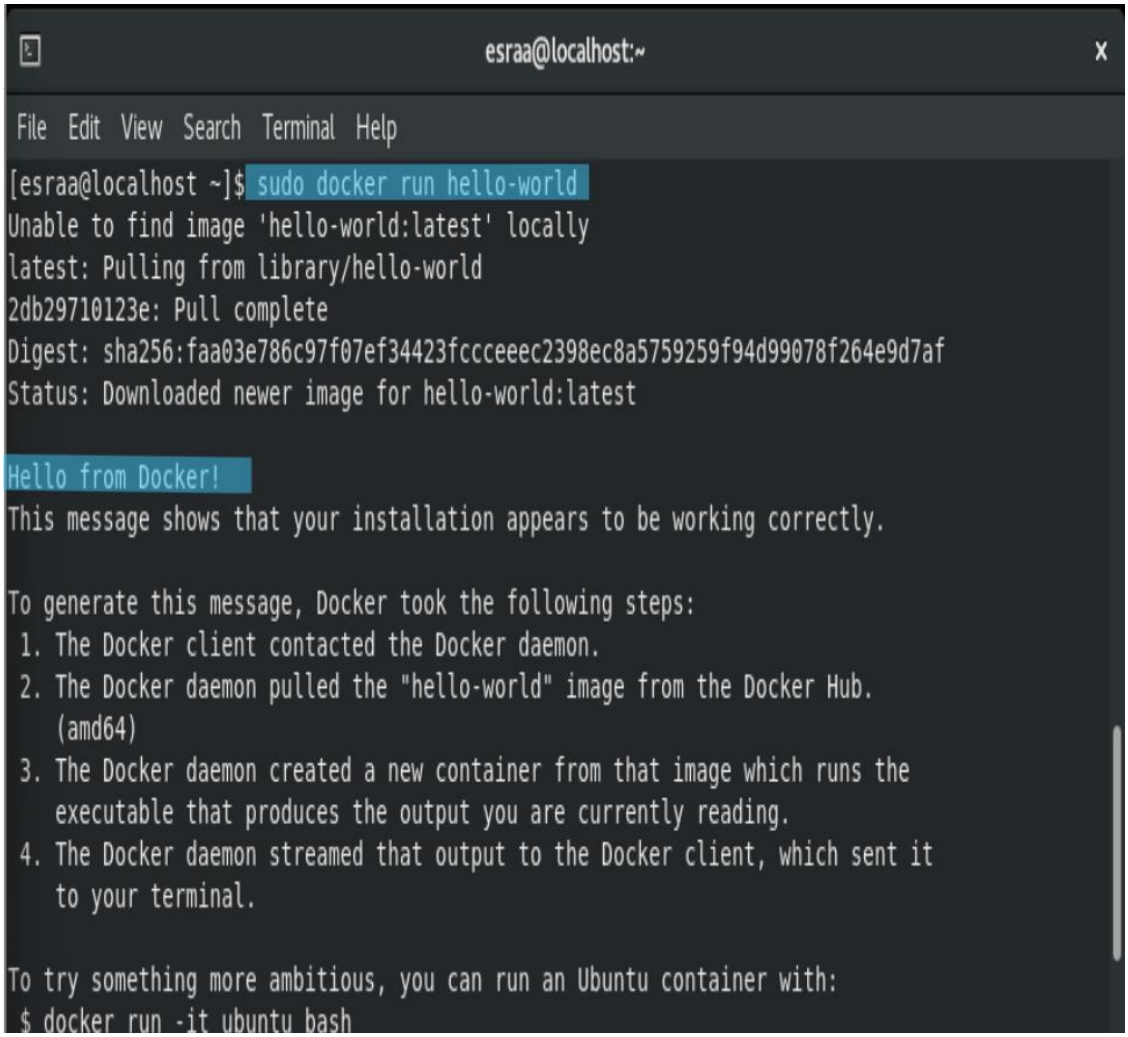
Lab 1

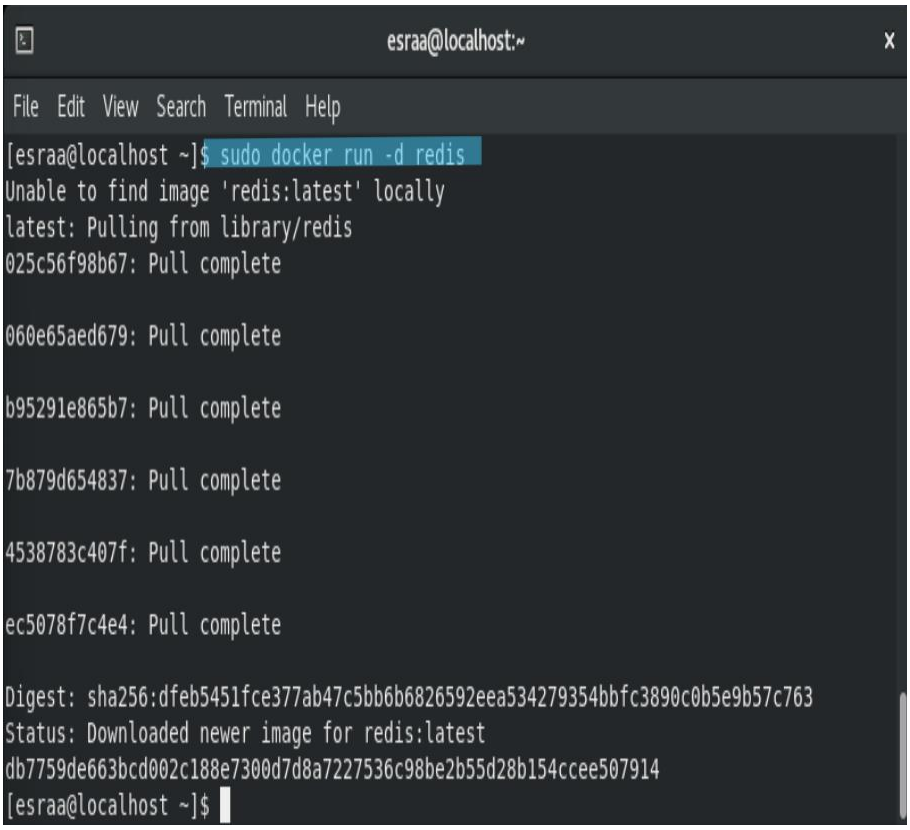
1- check the docker version installed



2- run docker container for hello-world

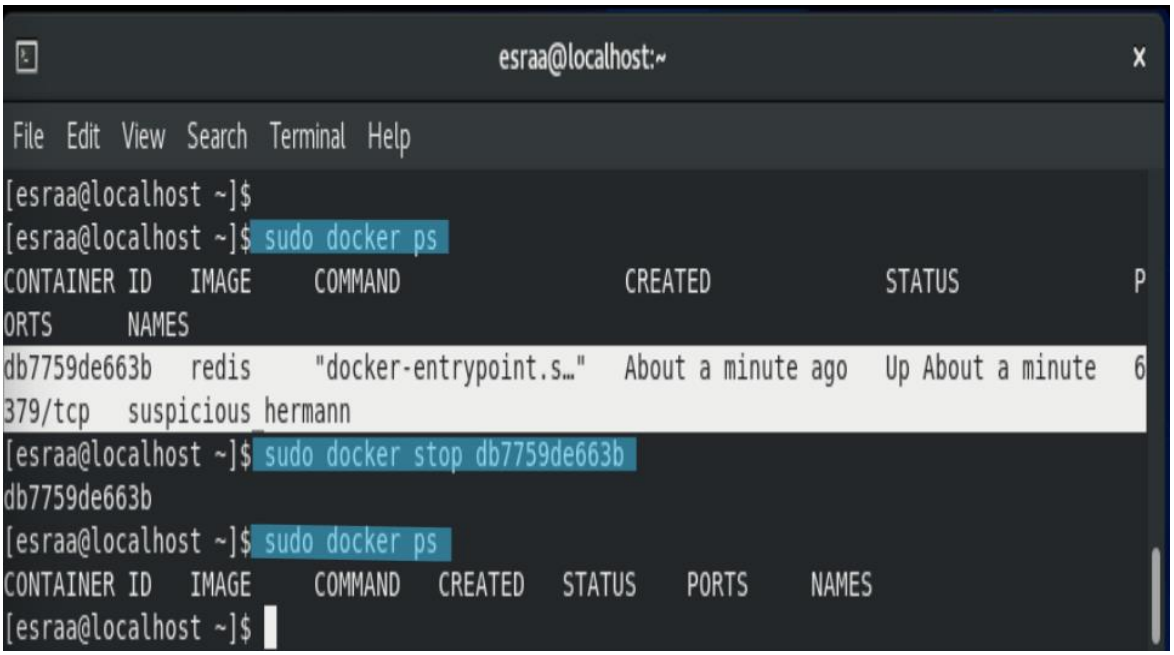


3- run docker container redis in detached mode

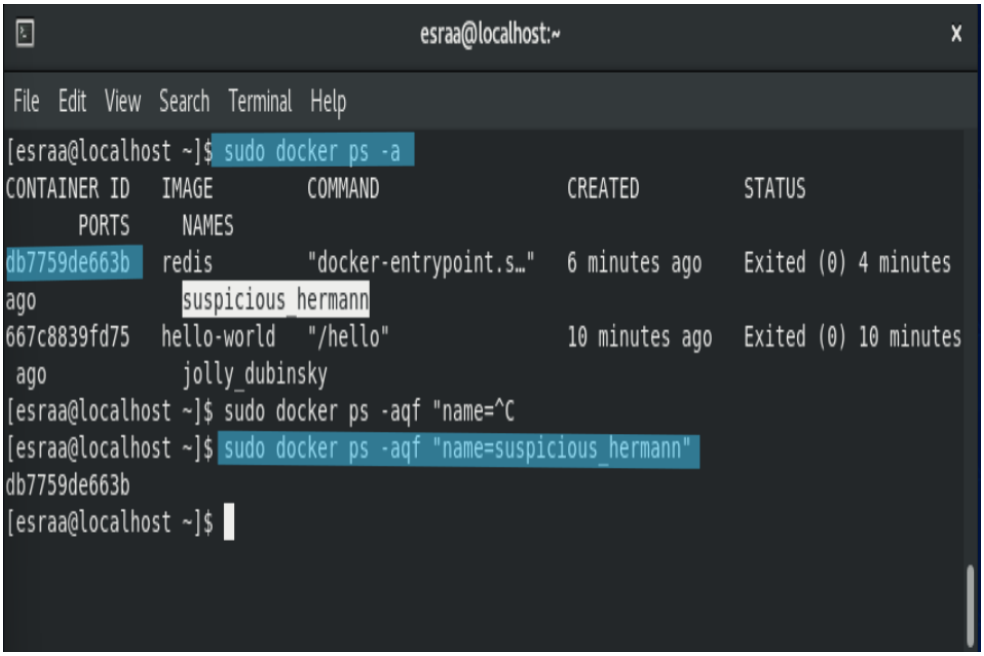


4- try to stop the running redis container

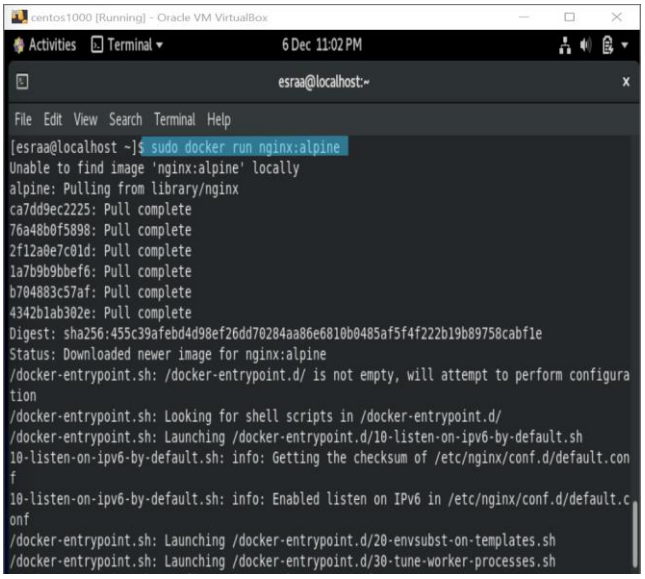
4- check the present container o the host

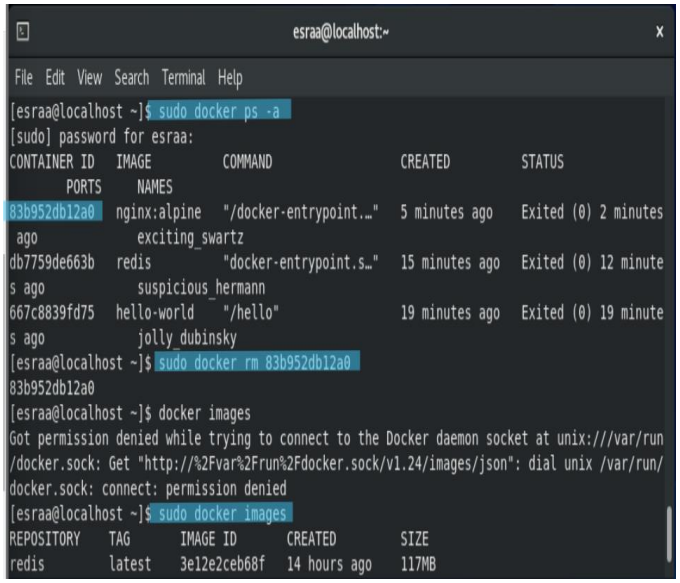


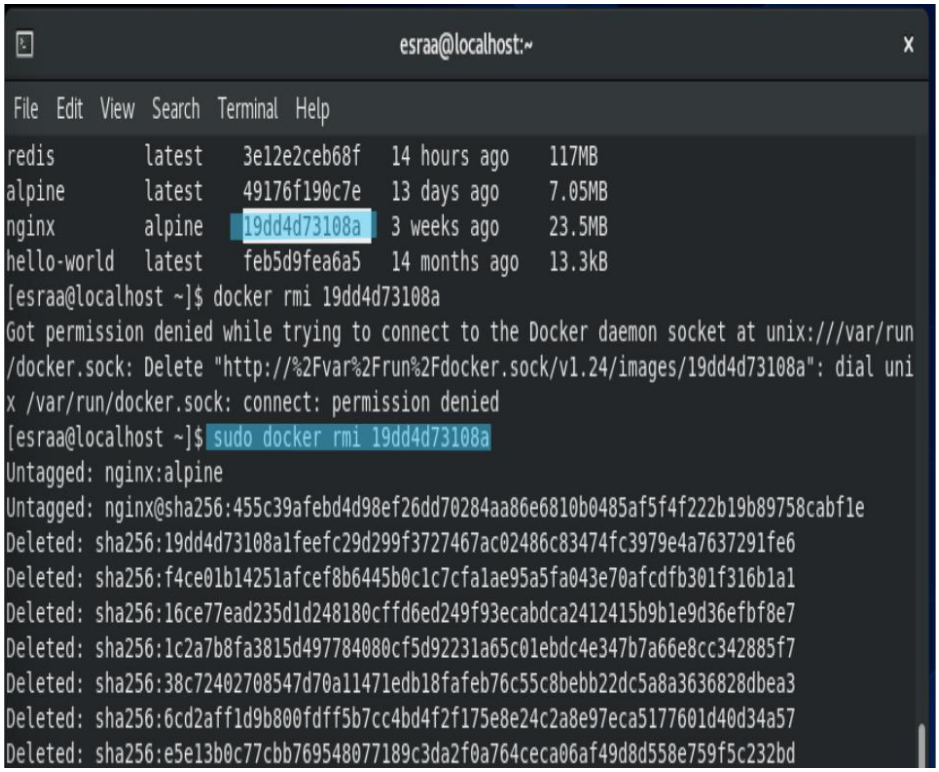
5- check the ID of the redis container



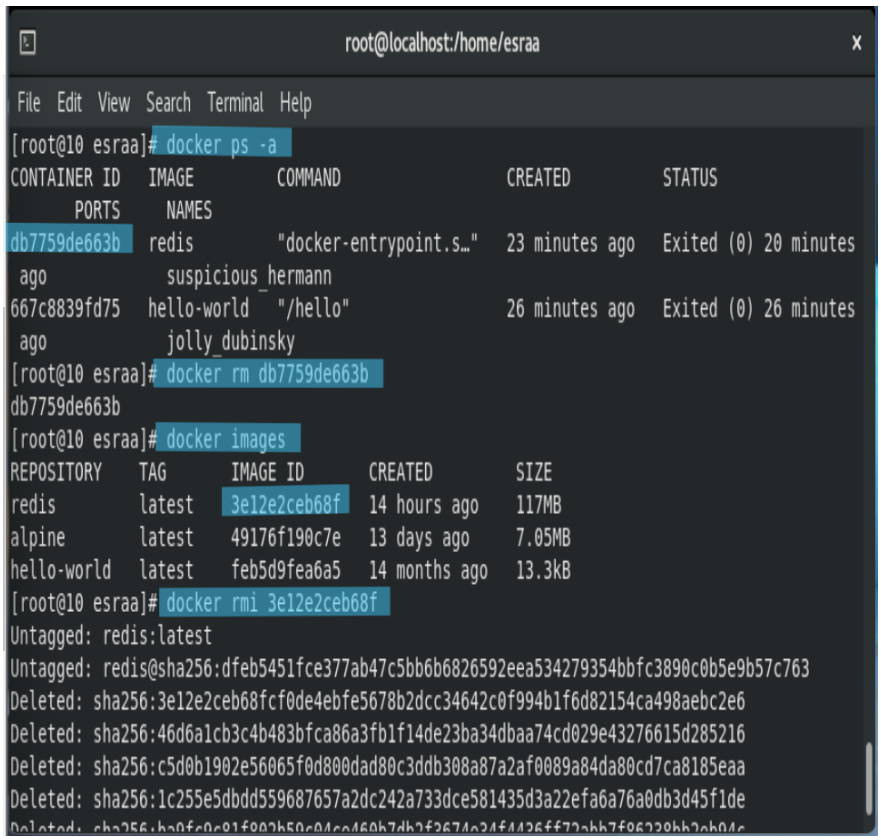
6- try to run a container from nginx:alpine and delete image





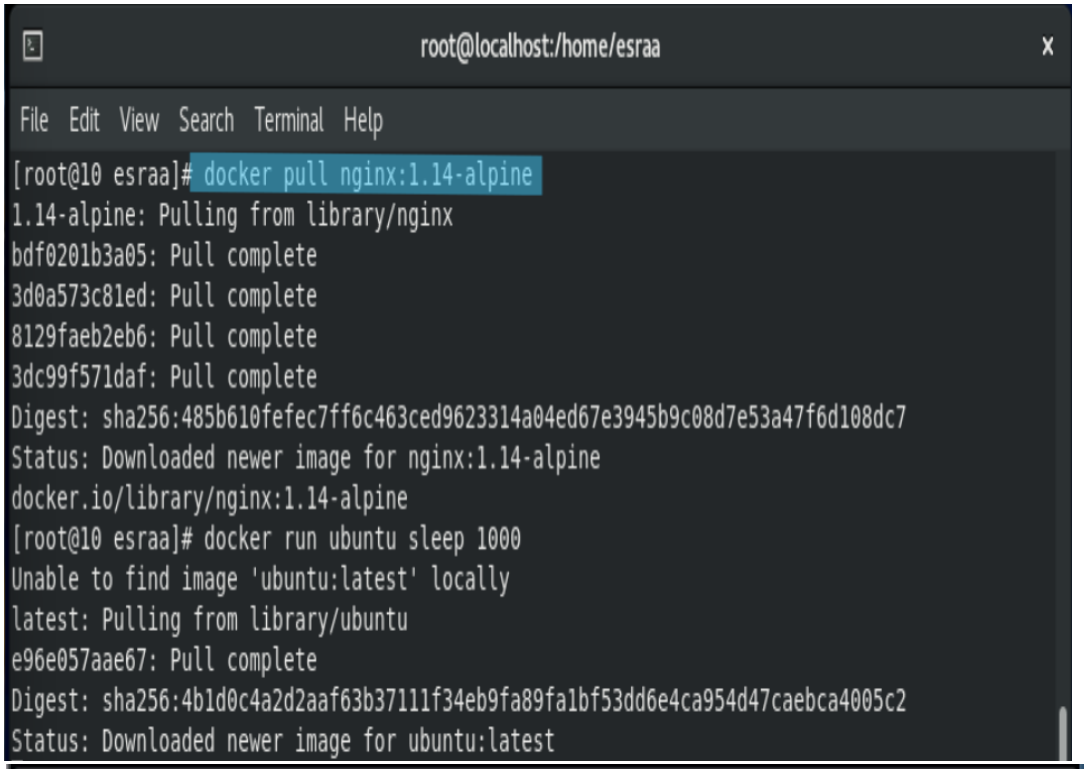


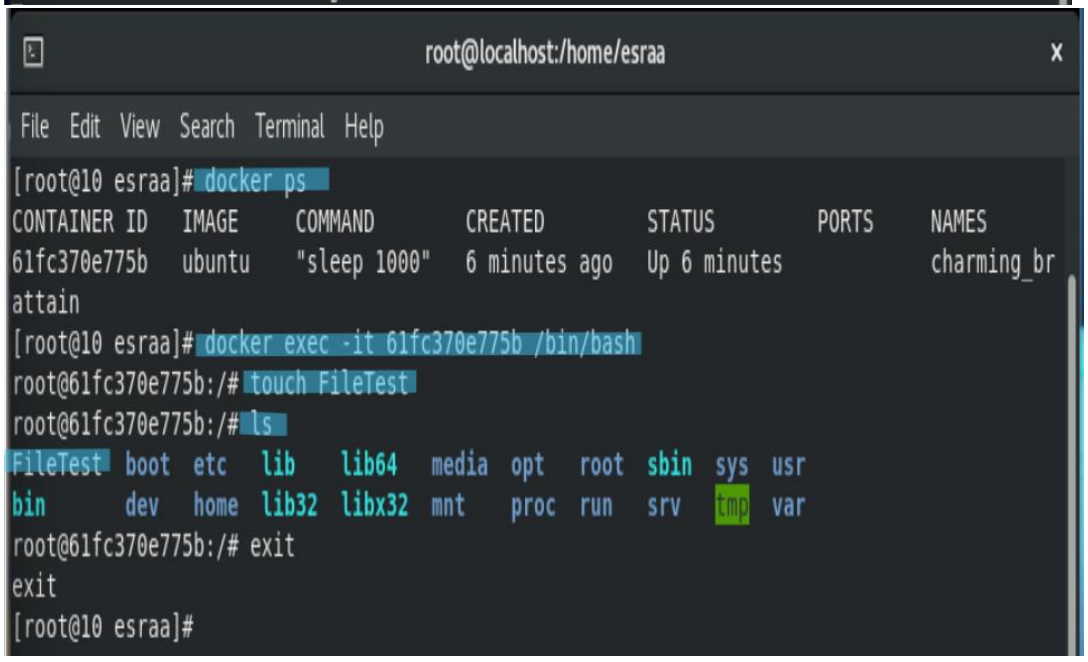
7- delete the image redis



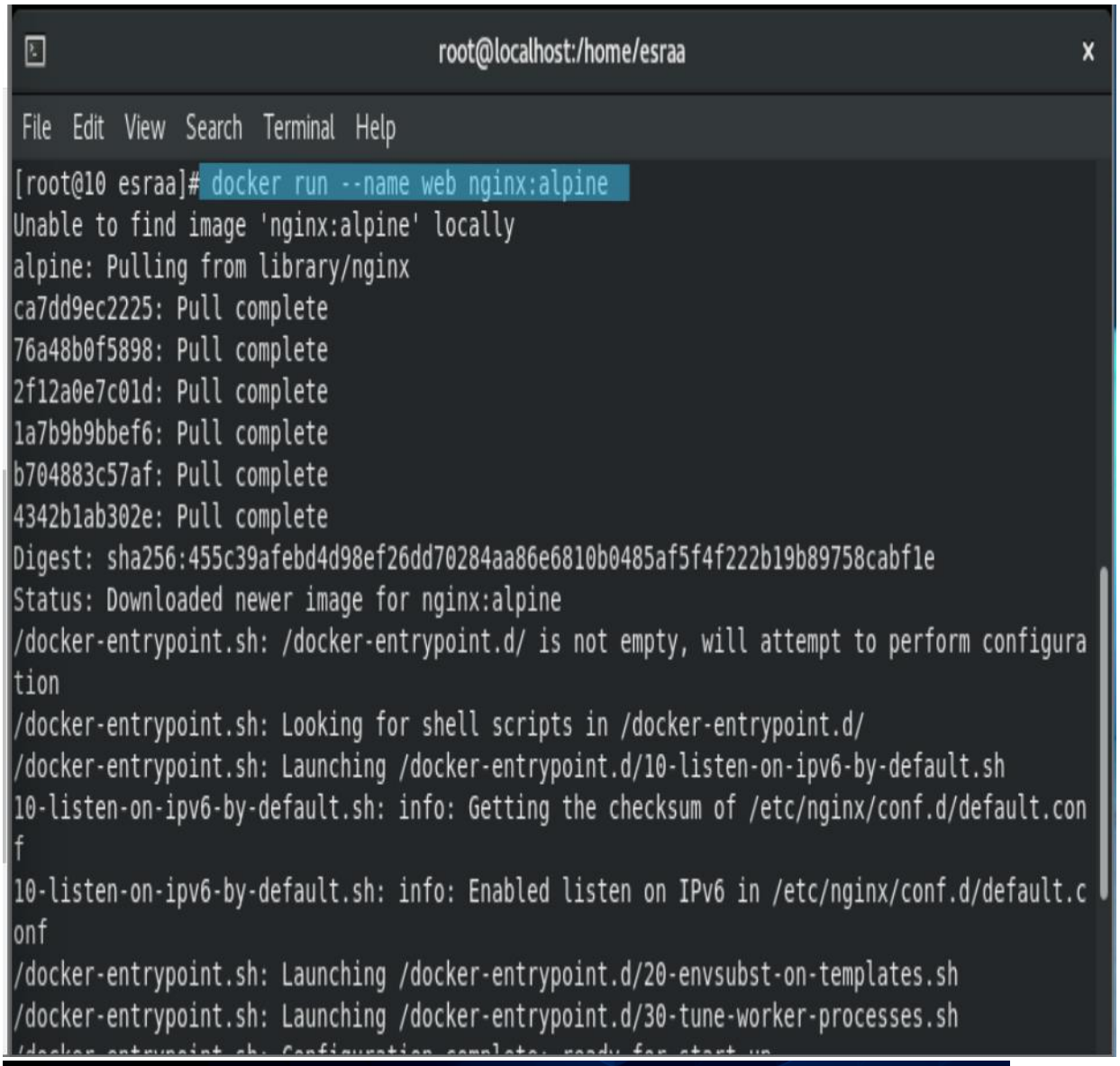
8- pull image from nginx:1.14-alpine

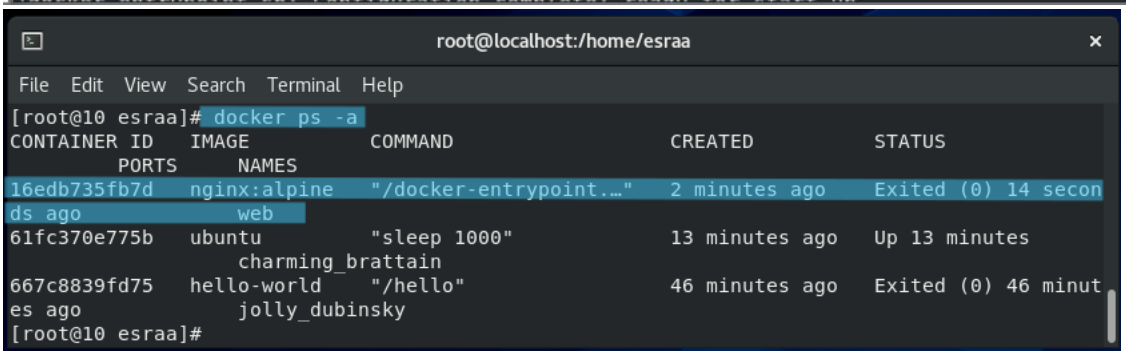
Run an instance of the ubuntu image to run the sleep 1000 command at startup.Exec into the container and touch a file called test-file.



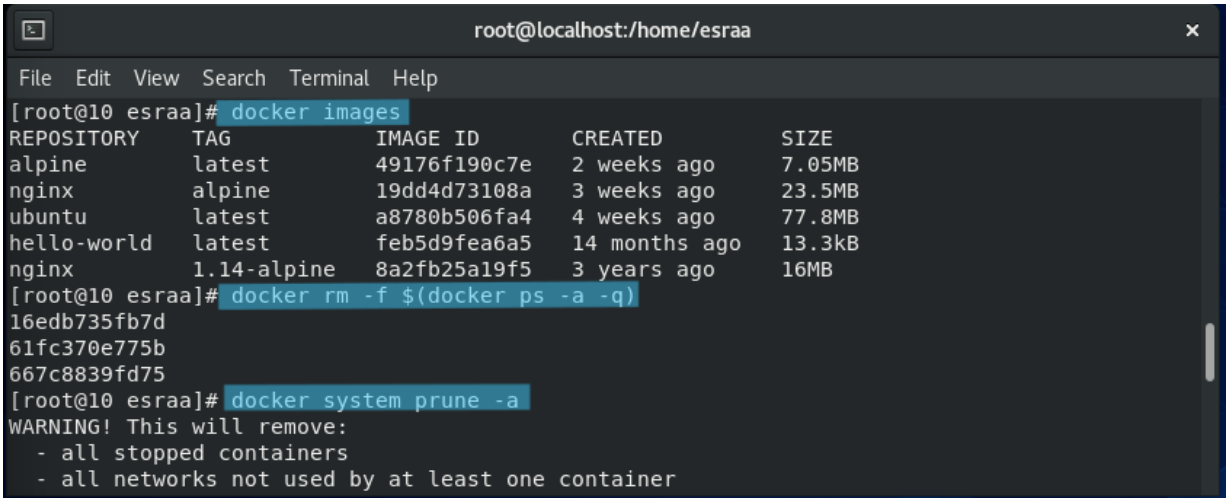


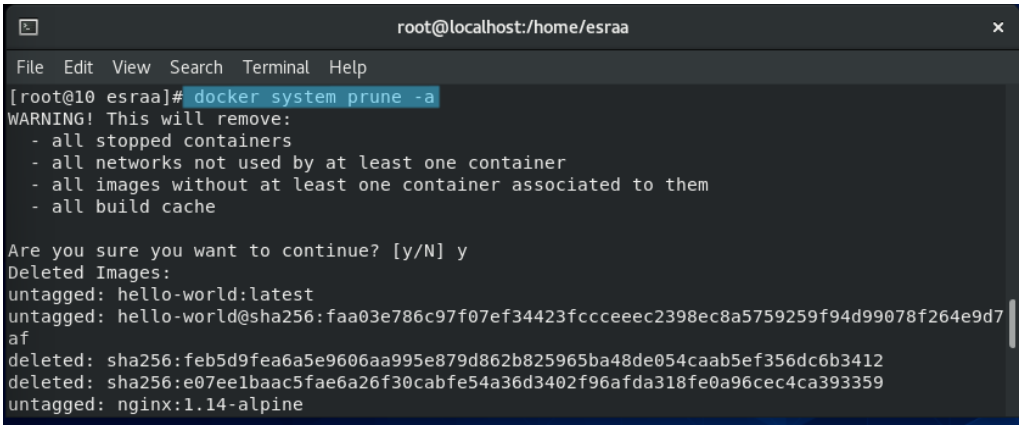
9- Run a container with the nginx:alpine image and name it web



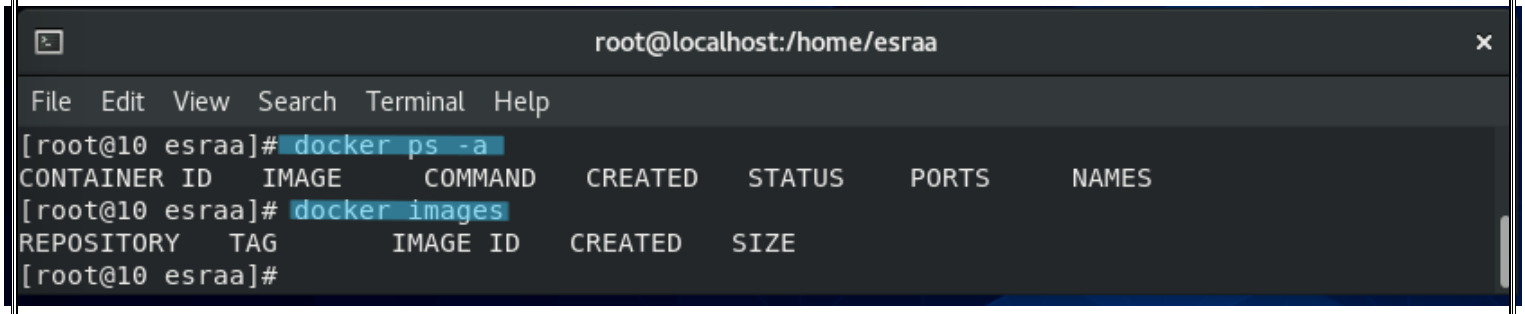


10- delete all the images from the host

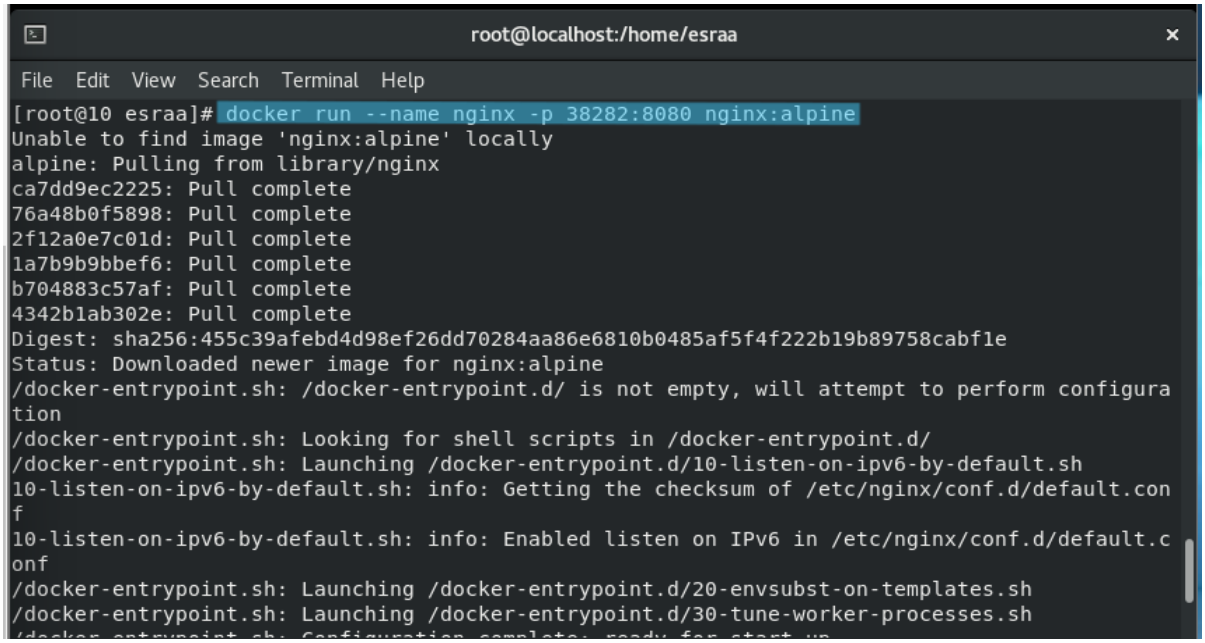




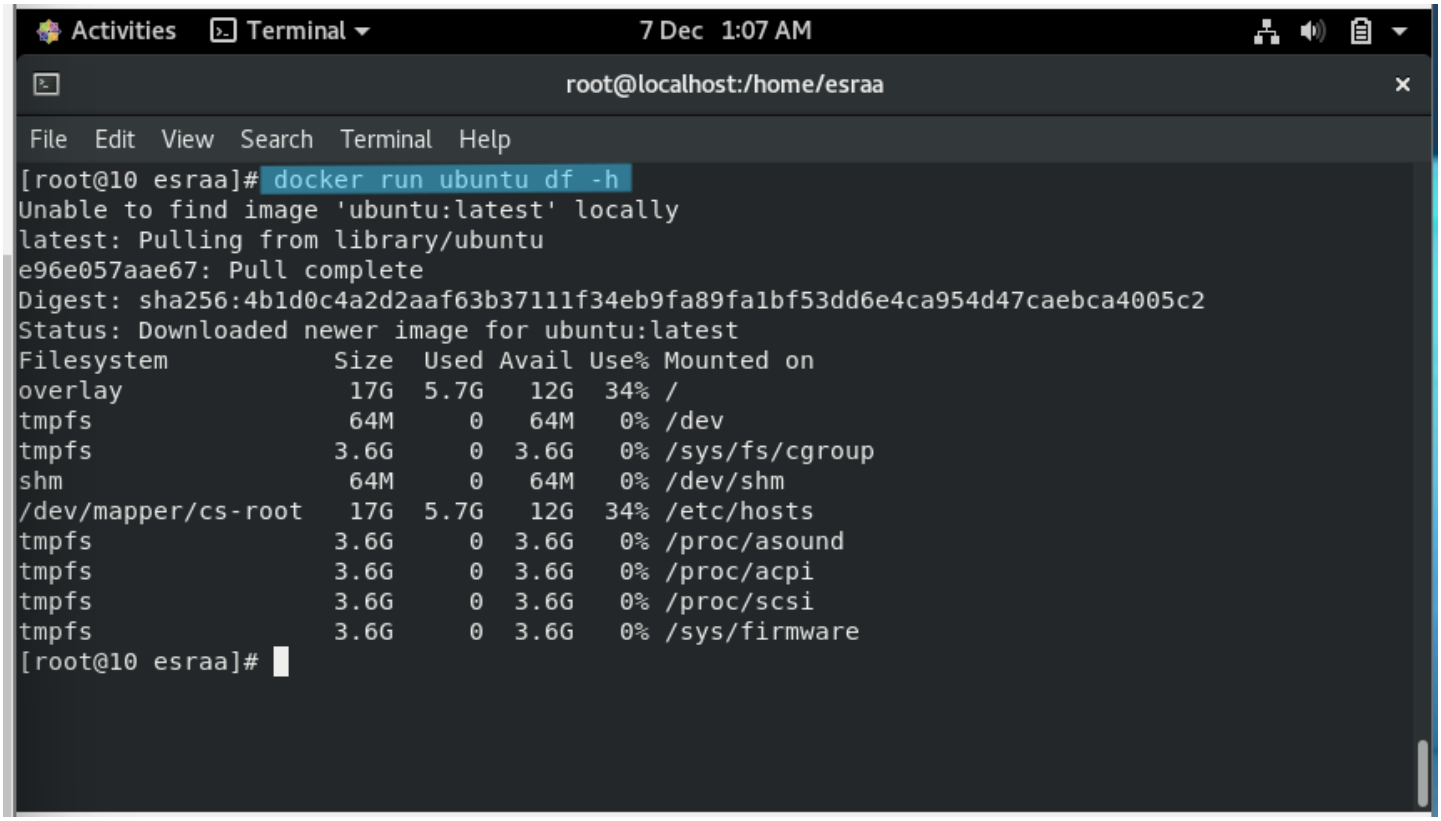
11- Run an instance of nginx:alpine with a name nginx and map port 8080 on the container to 38282



on the host.

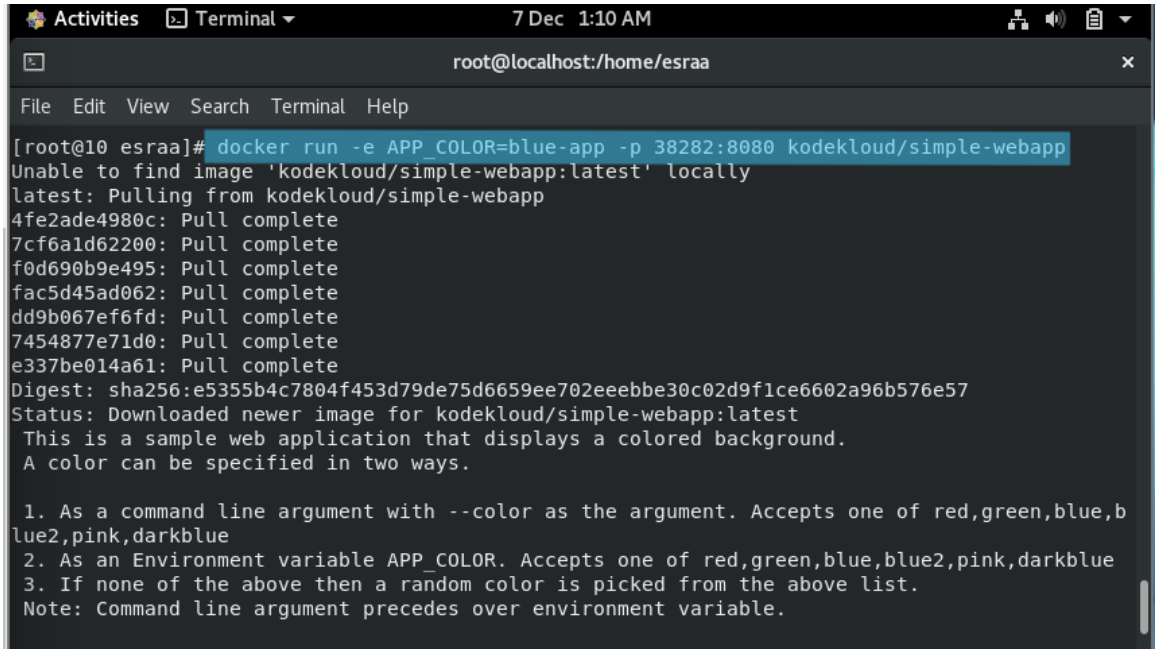


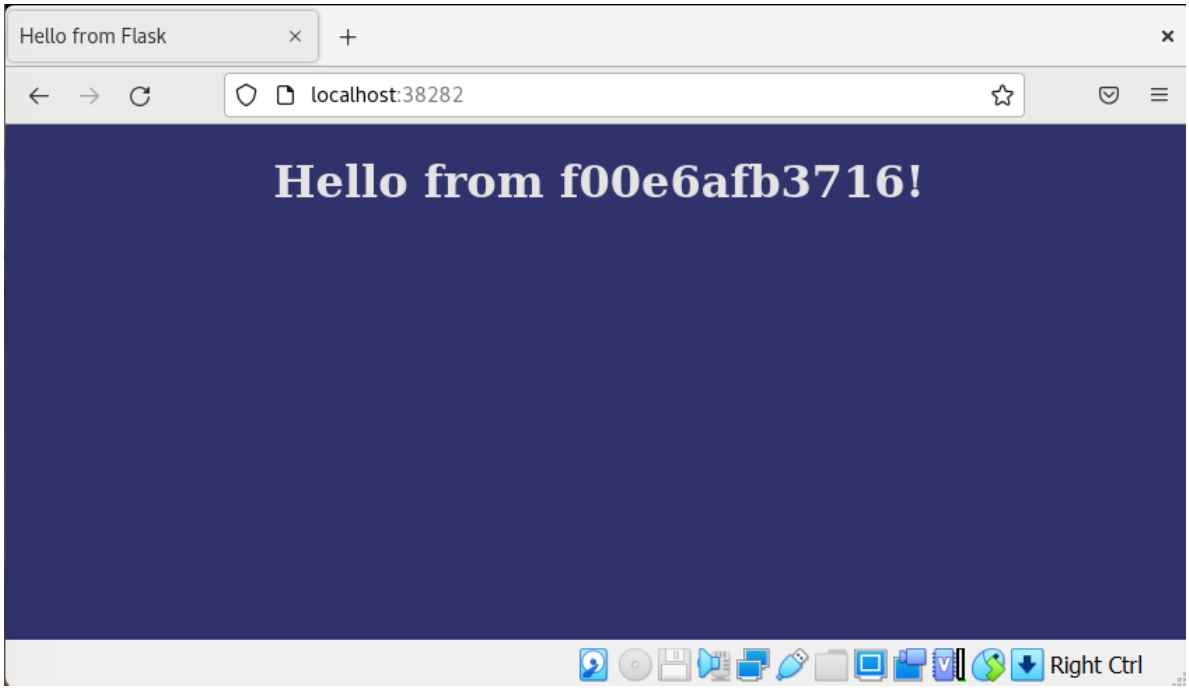
12- create ubuntu image and check the size of it



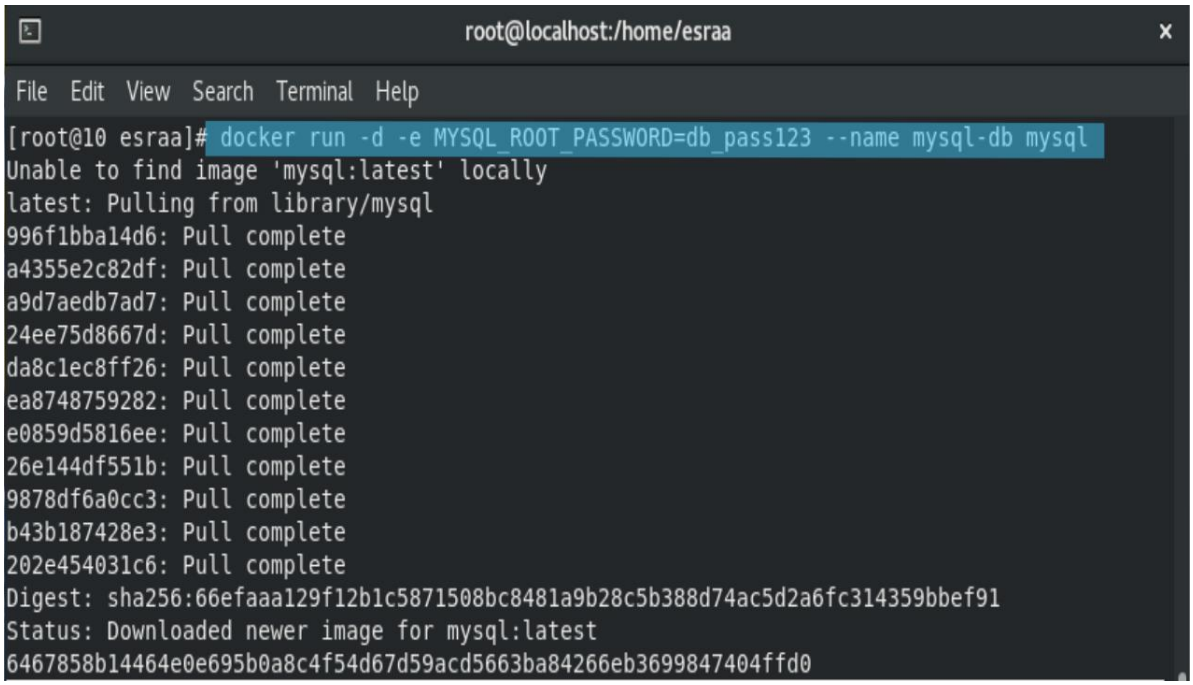
13- Run a container named blue-app using image kodekloud/simple_webapp and set the environment variable APP_COLOR to blue. Make the

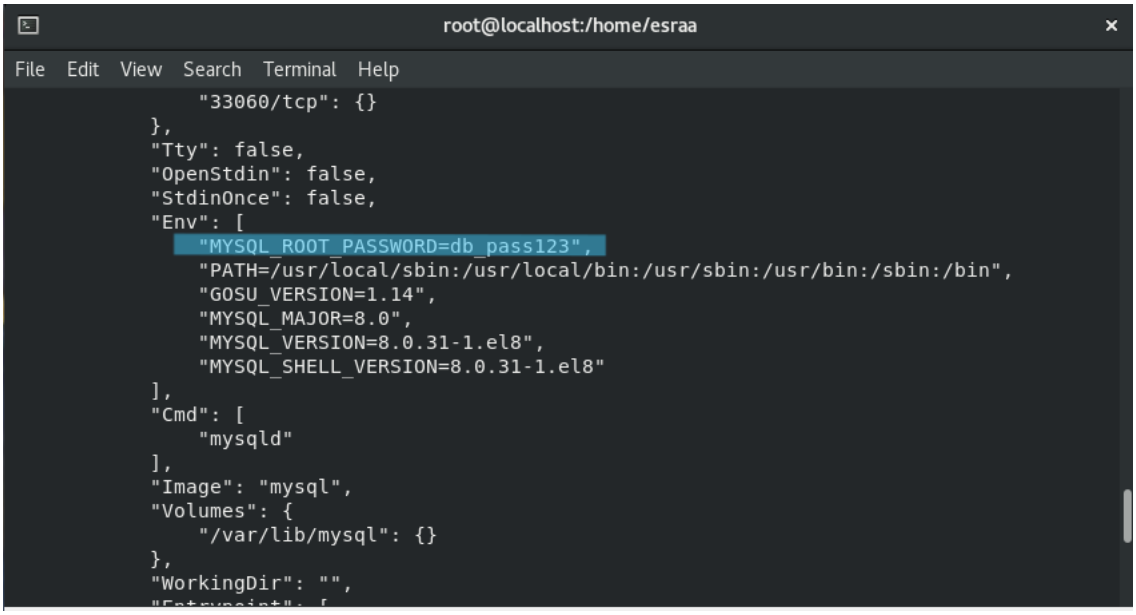
application available on port 38282 on the host. The application listens on port 8080



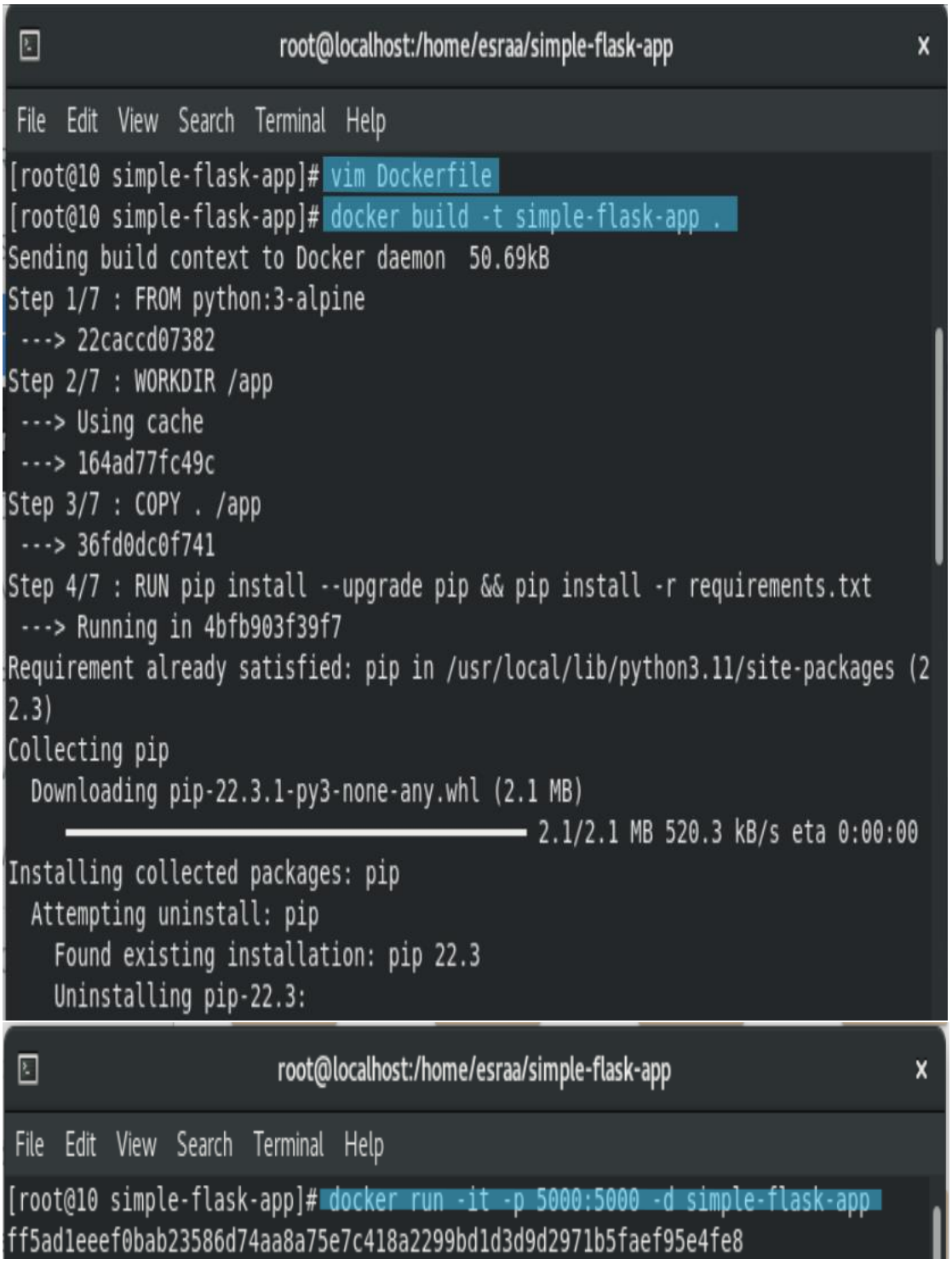


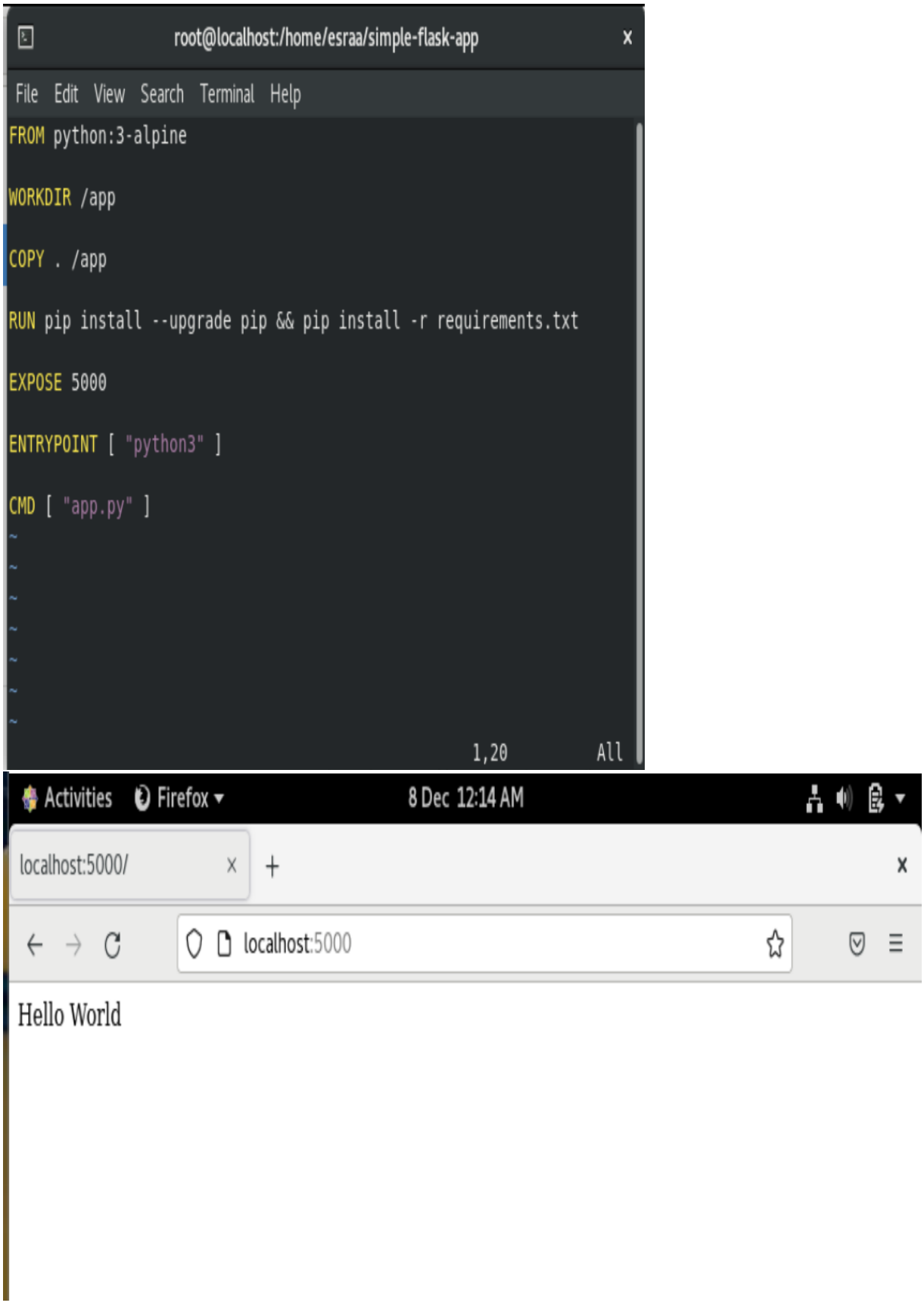
14- Deploy a mysql database using the mysql image and name it mysql_db Set the database password to use db_pass123 then inspect it to check the value





15- pull the code from https://github.com/sabreensalama/simple_flask-app/tree/main and create a docker file for this flask app

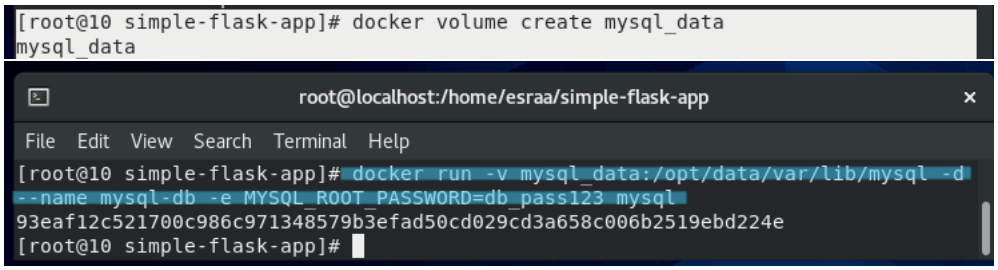



root@localhost:/home/esraa/simple-flask-app
File Edit View Search Terminal Help
FROM python:3-alpine
WORKDIR /app
COPY . /app
RUN pip install --upgrade pip && pip install -r requirements.txt
EXPOSE 5000
ENTRYPOINT ["python3"]
CMD ["app.py"]
1,20 All
Activities
Firefox
8 Dec 12:14 AM
localhost:5000/
localhost:5000
Hello World

16- Create a volume called mysql_data, Run a mysql container again,

but this time map a volume to the container so that the data stored

by the container is stored at /opt/data on the host.

Use the same name : mysql-db and same password: db_pass123 as

before. Mysql stores data at /var/lib/mysql inside the container.