**1.Create a docker container using Ubuntu image & check Ubuntu Image is fetched/created locally.**

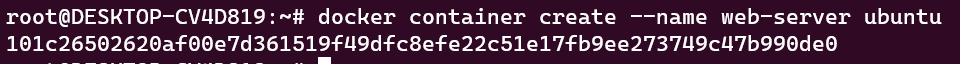
**2.Check the Ubuntu docker container to see it is only in the created state.**

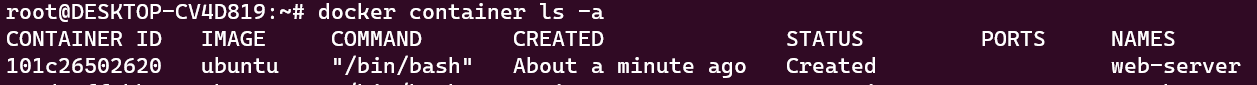
**3. Run the Ubuntu docker container and check the State of the container & it must be running.**

**Answer:**

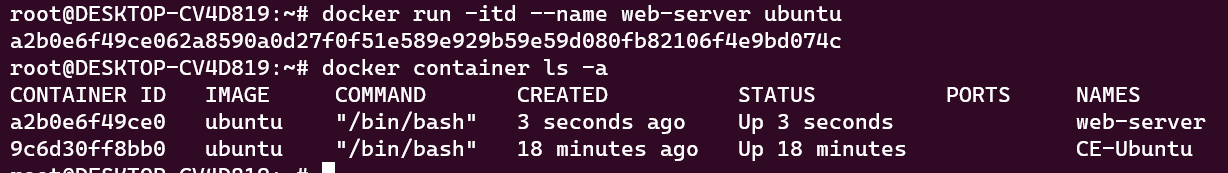
* We have the ubuntu image downloaded locally in our laptop, so we directly run the container.
* # docker container create --name CE-Ubuntu ubuntu
* #docker container ls -a
* This command will create a container with our ubuntu image. And we can chech the status by the command ls -a



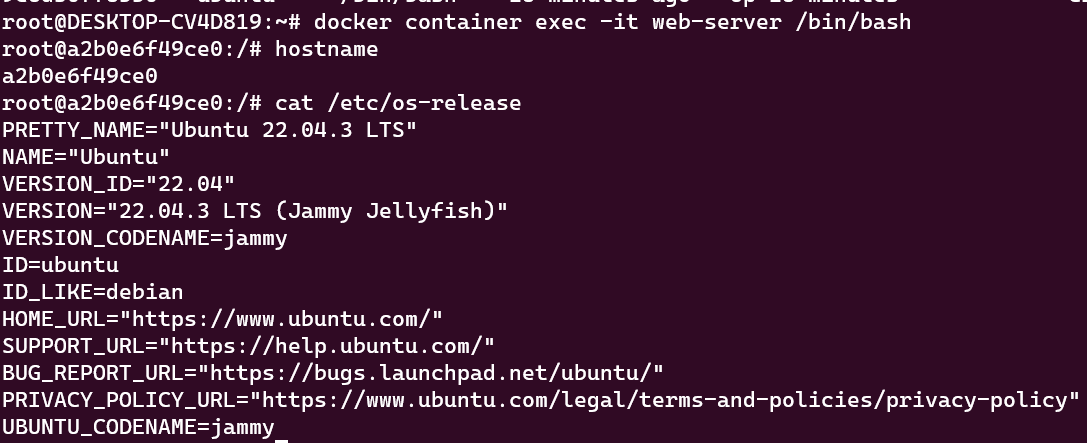




* # docker run -itd --name CE-Ubuntu ubuntu
* This command will run a docker container using ubuntu image, and it will have a interactive terminal which will be detached from our wsl terminal.



* Now, we will try to go inside of the container and exit without stopping the container.
* We run the command, # docker container exec -it CE-Ubuntu /bin/bash Using this command, we are able to log into our ubuntu conatiner with a interactive terminal working on a bash shell.
* To exit the shell without stopping it, we shall press the key ctrl+p+q. This will allow us to exit the container without stopping it

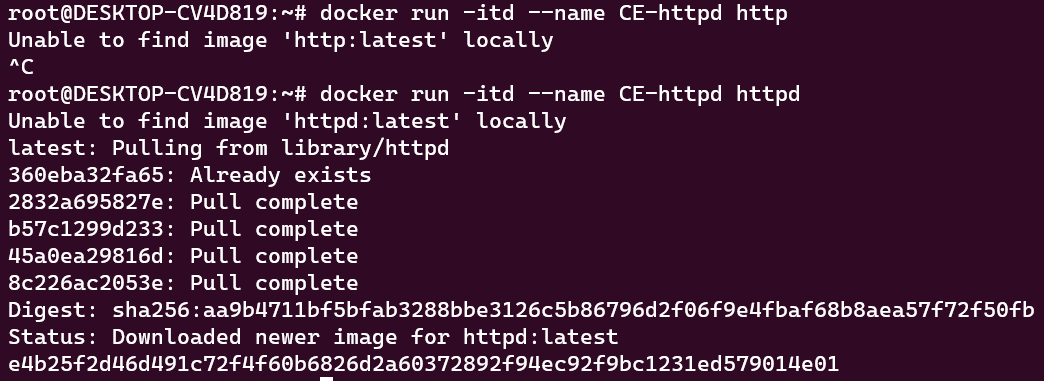


**4.Run an HTTP container and check the state of the container. Then delete the container and image from the local system.**

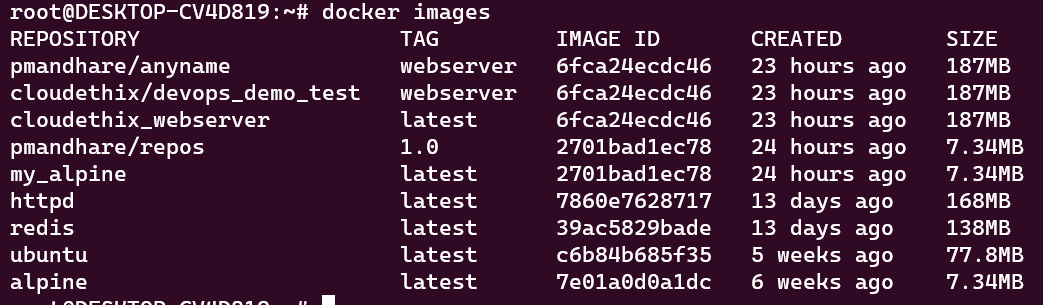
**Answer:**

* For httpd container we use the command

# docker run -itd



* Now, we have the httpd image in our local system, we will now delete the container and image from the local system.



* First we need to stop and remove all the containers associated with the httpd image.
* We have now deleted the httpd image from our local system and stopped the containers too.

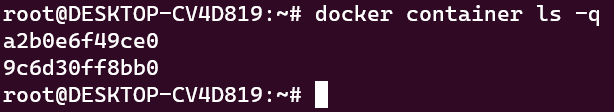


**5. Run a command to filter the containers with short container IDs for all the containers**.

**Answer:**

* For getting short IDs of the conatiners, we use the below command.

# docker container ls -q



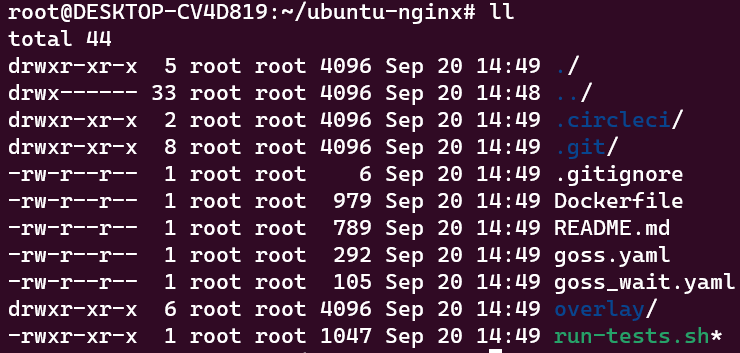
**6. Clone the lancachenet/ubuntu-nginx Git repo locally and build the docker image from the Dockerfile. Please add meaningful tags while building the docker image.**

**Answer:**

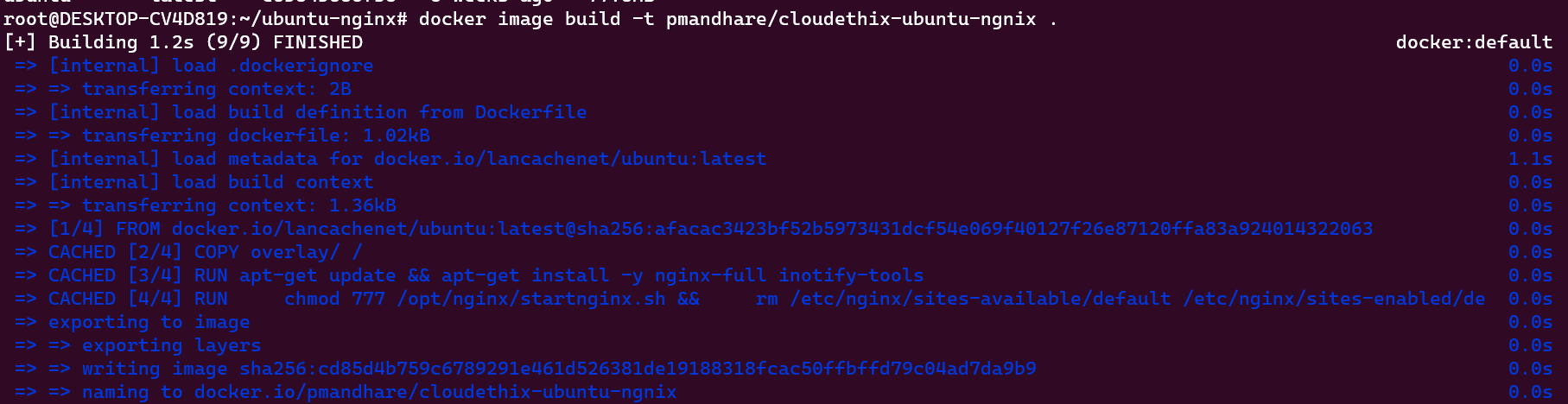
* First we clone the git repo in our local system

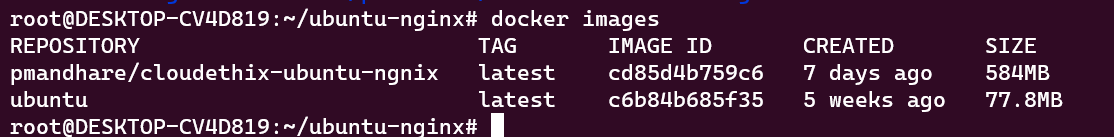
Repo- >> lancachenet/ubuntu-nginx <<





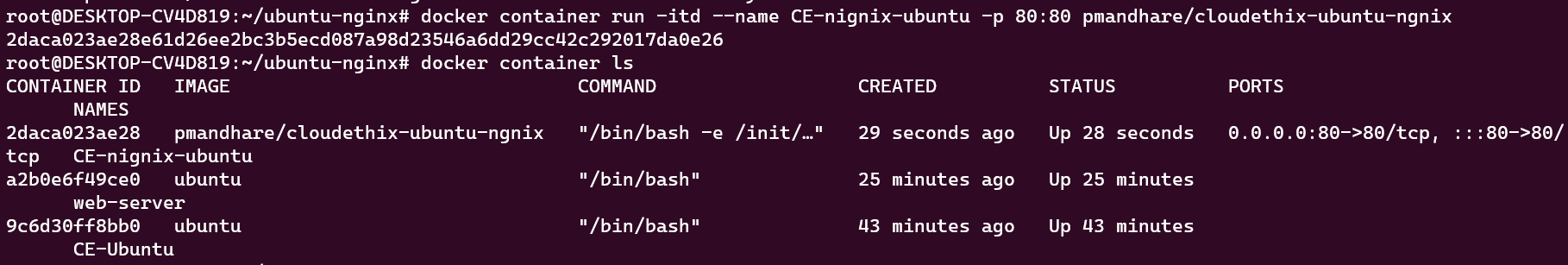
* Now we build the image using the Dockerfile
* # docker image build -t pmandhare/cloudethix-ubuntu-ngnix .

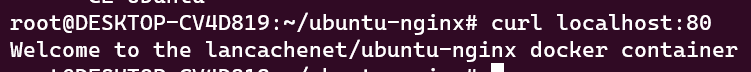




Now we run the container using the image that was built

# docker container run -itd --name CE-nignix-ubuntu -p 80:80 pmandhare/cloudethix-ubuntu-ngnix

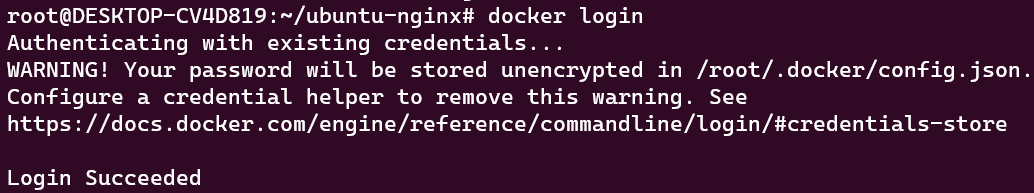




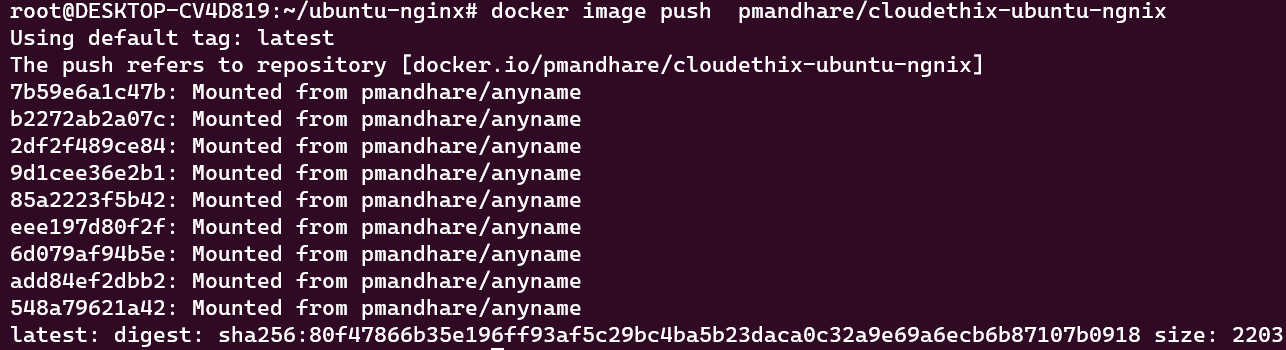
**8. Create a DockerHub account & create one repository in your account with name cloudethix\_devops\_yourname. Then tag the locally created docker image with using your\_repo\_name/application\_name:version\_number.**

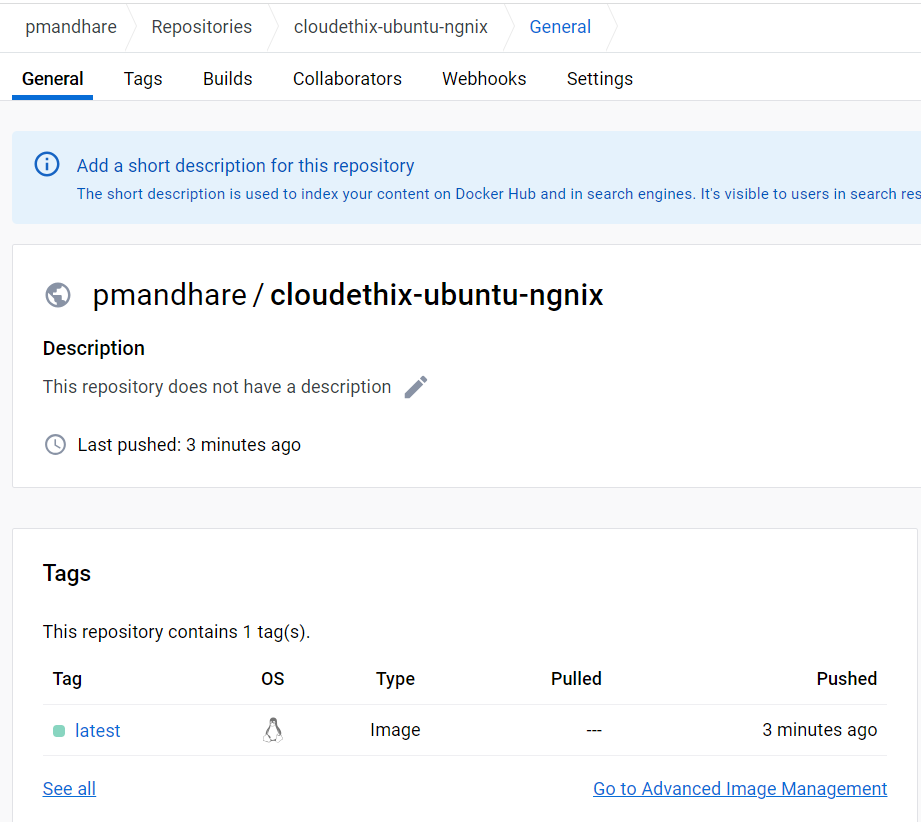
**Answer:**

* We have made an account on docker hub and we were able to login using our terminal



* As we have already tagged the image while building, we can directly push the image to docker hub and we are able to see the image pushed at the dockerhub too.

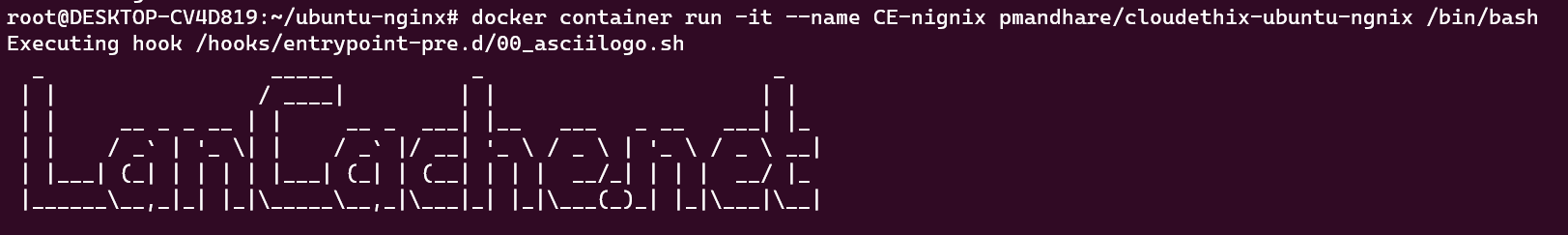




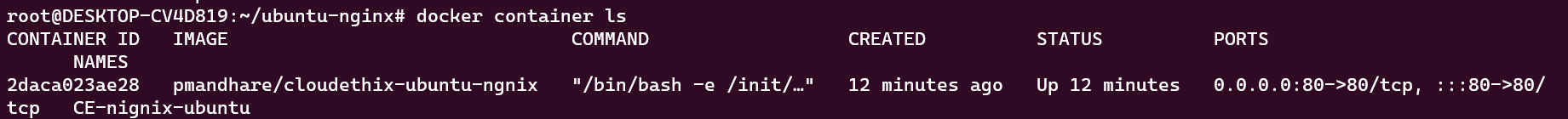
**9. Run the ubuntu container & prove that the escape sequence is working. Also once you detached the container with an escape sequence , re-attach the same and check the IP address of the container.**

**Answer:**

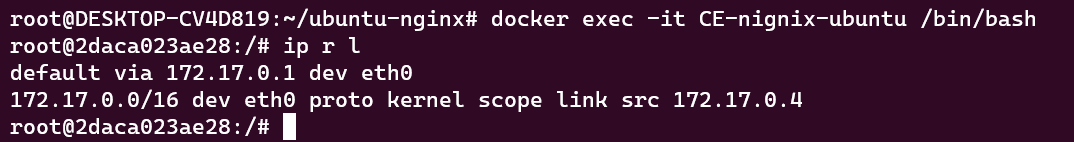
* We have run the container and we are able to use the /bin/bash shell



* We have exited the conatiner without stpooing it using ctrl+p+q



* We are able to reattach the container and check the ip

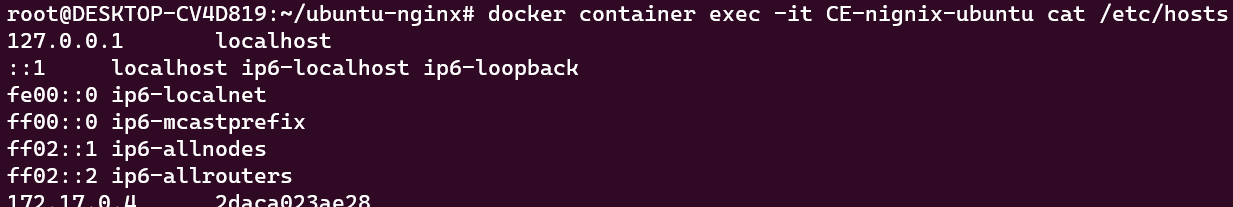


**10. Run a docker command to check the hostname & /etc/hosts file of the httpd container.**

**Answer:**

* Below are the outputs of the

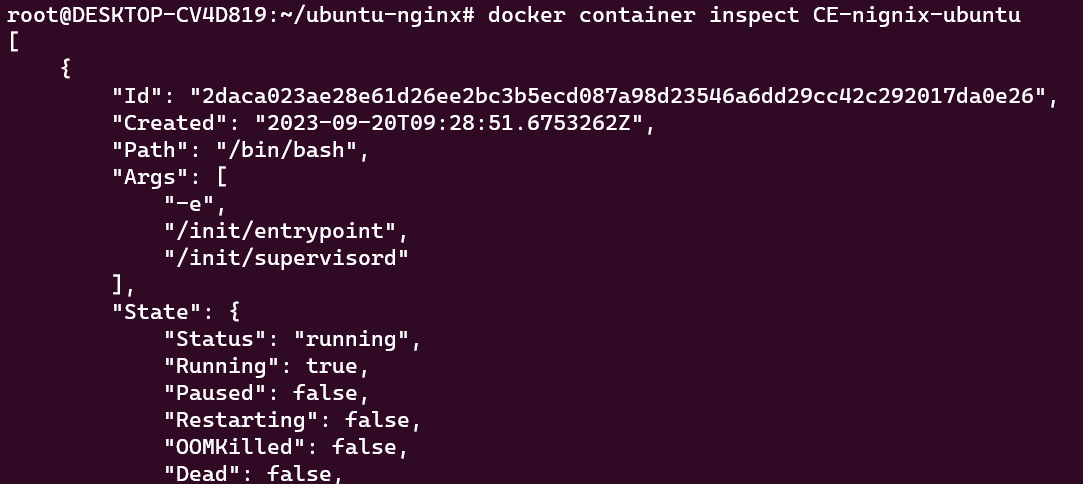


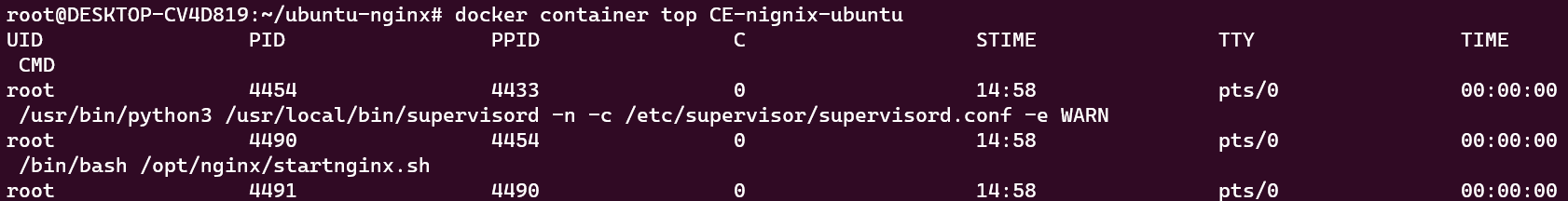


**11. Run a docker command to inspect a container created using lancachenet/ubuntu-nginx repo. Also check the stats and resources utilization using TOP of same container.**

**Answer:**

* Below are the outputs of the

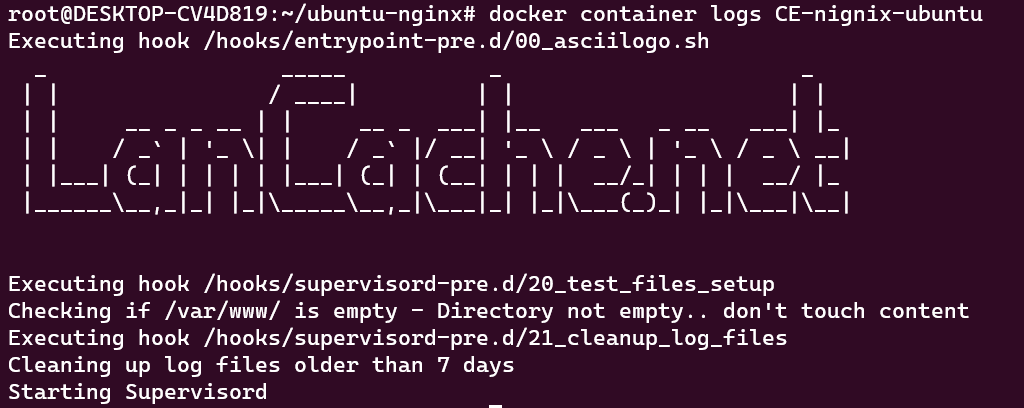




**12. Run a docker command to check the hostname & /etc/hosts file of the httpd container.**

**Answer:**

* Below are the outputs of the



**13. Check the System events and filter out the event by date & last 30 mins. Also apply the two filters at a time using name & event**.

**Answer:**

* Below are the outputs of the required commands
* # docker system events --since 30m --filter date=27-01-2023

