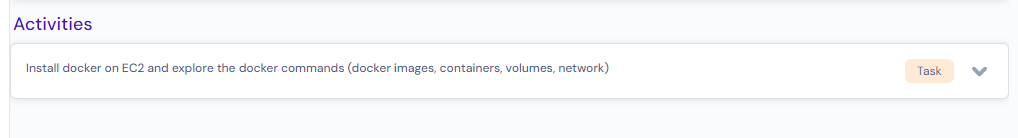
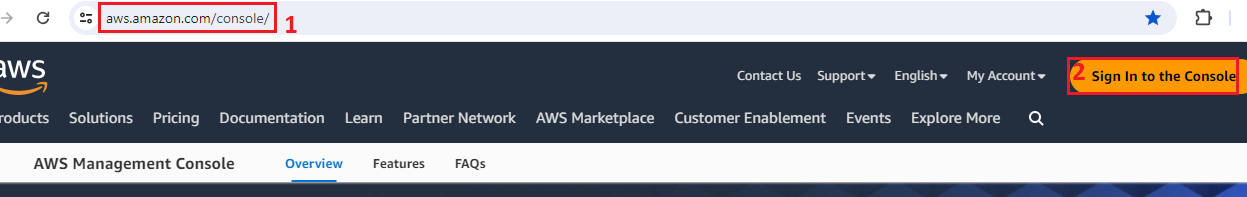
**Assignment: -**



# Steps to launch EC2(Elastic Compute cloud) Instance:

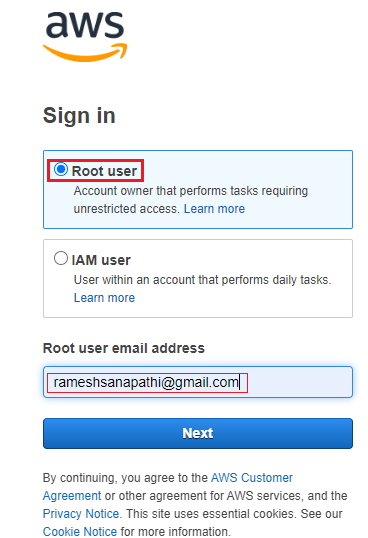
## Open URL <https://aws.amazon.com/console/> in the browser.



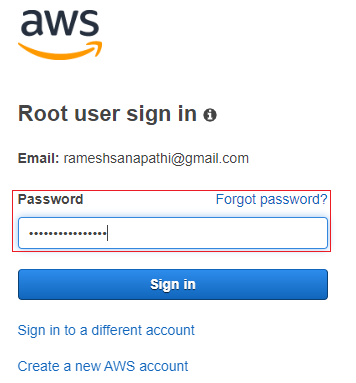
## Click on “**sign in to the console**”.

## Select “**Root user**”, enter **email-id** and click on **“Next”.**

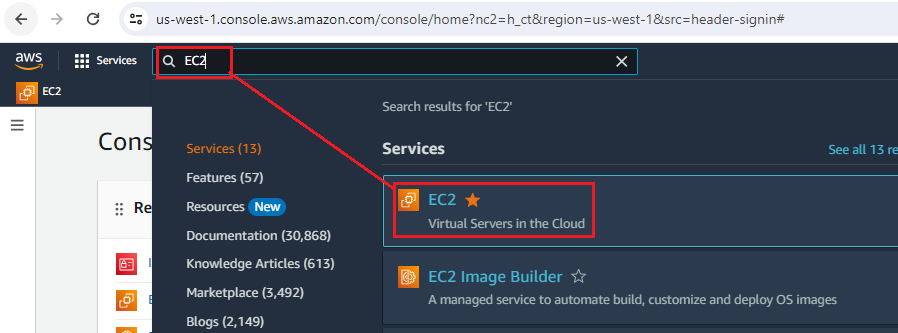
**[ Assume: user already registered and have valid account to login to aws]**



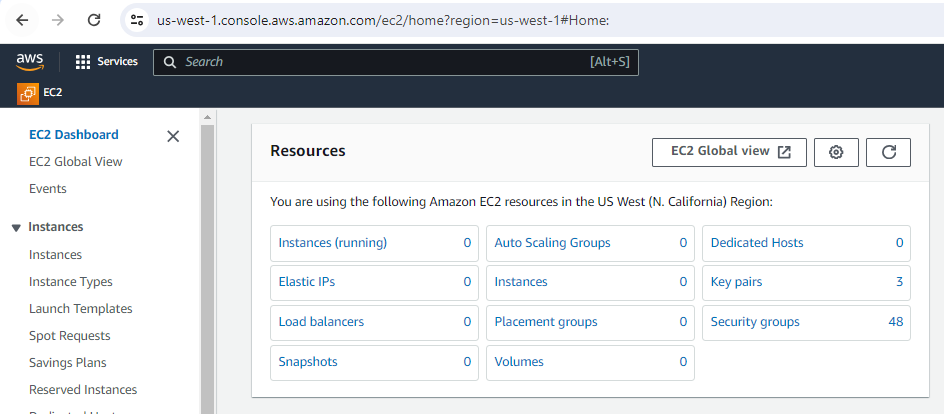
## Enter **password** and click on **“Sign in”.**



## Search “EC2” in search box and select “EC2” as shown below.

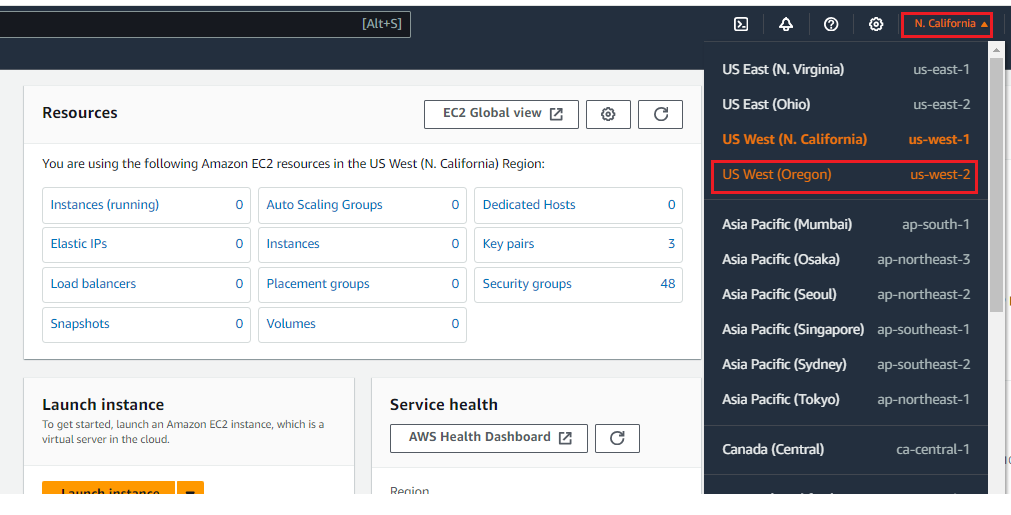


## EC2 dashboard gets displayed as shown below.

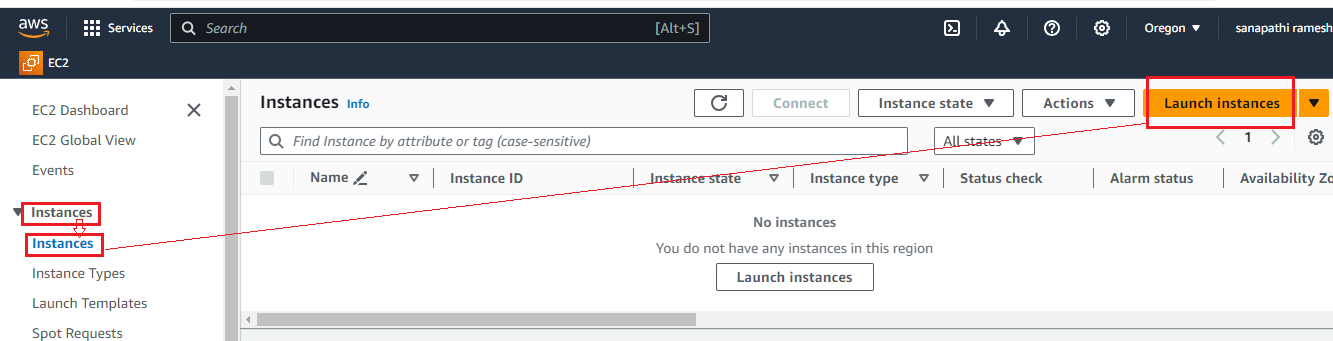


1. Select the desired “**Region**” where EC2 instance needs to be created. For my testing purpose, I have chosen “**US West (Oregon) us-west-2**” to create an EC2 instance.

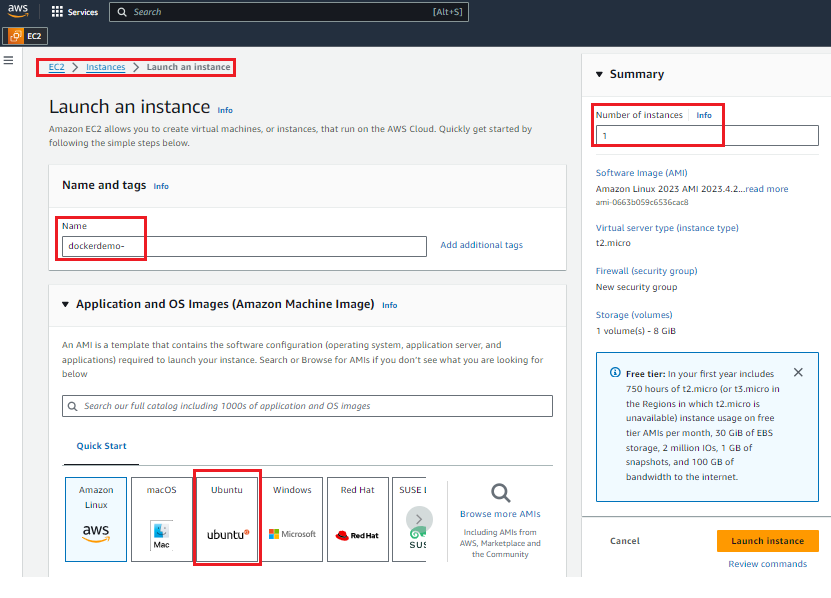
**Note: -** Here, each Region indicates where physical data center infrastructure is located, created and managed by AWS.

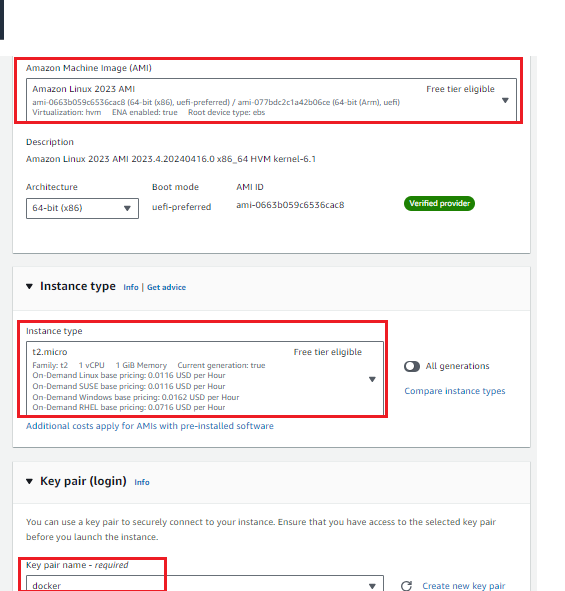


1. In the left side, click on “**instances**” under “**instances**” -> click on “**Launch instances**”.



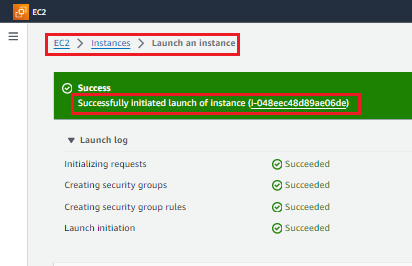
## 7. Enter all required details by following below picture.

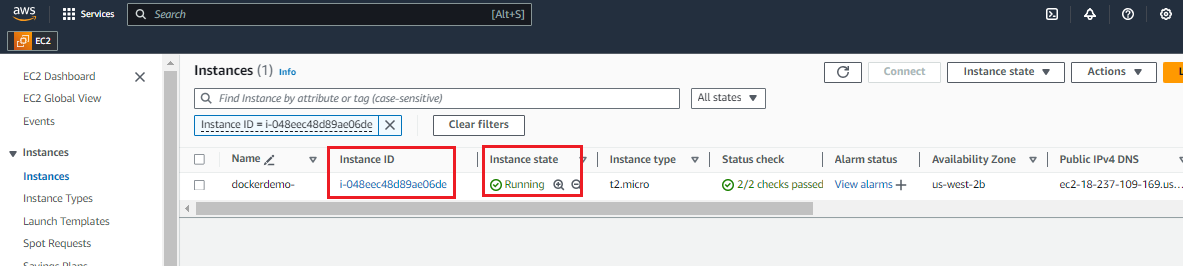




### **Note: - Keep other details as default and click on “Launch” to create an EC2 instance.**

## EC2 instance has been created successfully if below success got displayed. Then, click on “**instance-id**” to navigate to instance page.

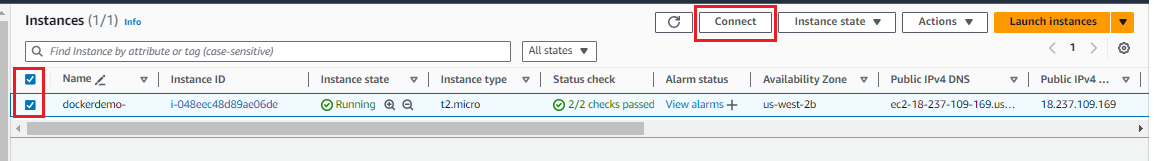




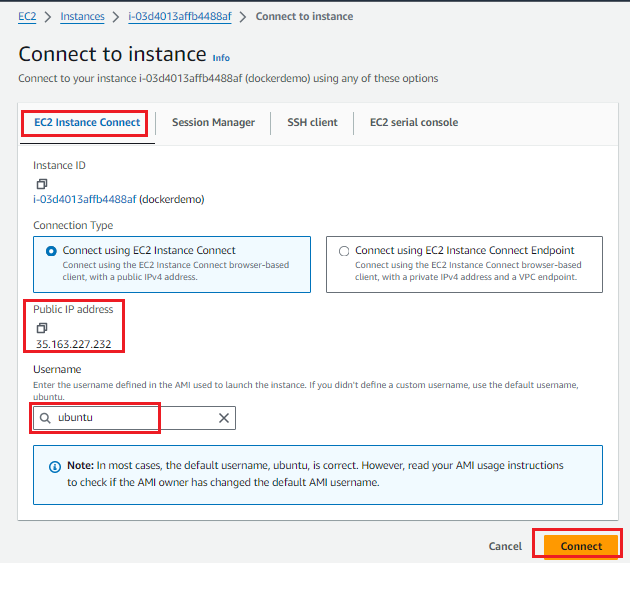
**Note: -** Make sure “**instance state should be Running**”.

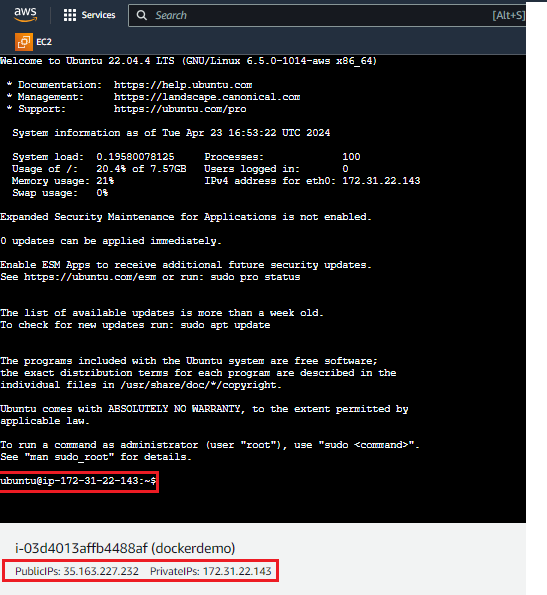
# Steps to login to EC2 instance.

## As shown below, select “**EC2 instance**” and click on “**Connect**”.



## As shown below, click on “**Connect**” to login to EC2 instance.



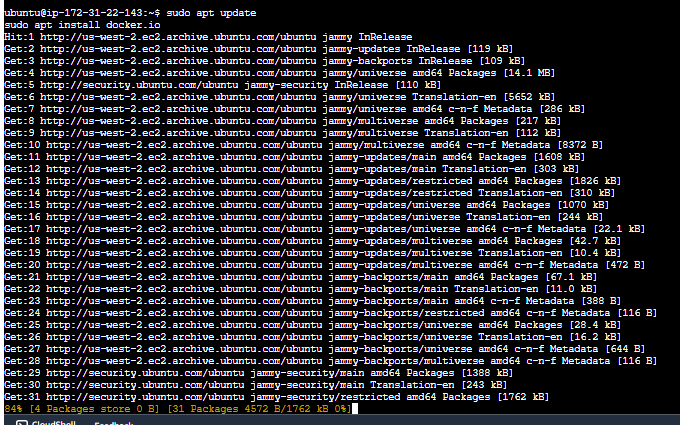


# Steps to Install Docker

## Run the below command to Install Docker.

sudo apt update

sudo apt install docker.io



## Add **“ubuntu”** user to docker group to get an access to docker commands.

Note: - It is recommended not to create or run any docker commands under root user.

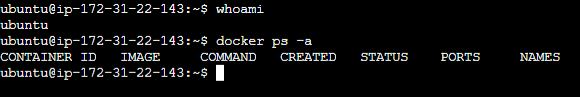
**Command:**

sudo su -

usermod -aG docker ubuntu

systemctl restart docker

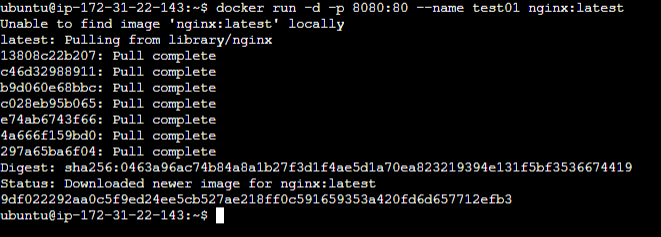
Note: make sure, ubuntu is able to execute docker commands.



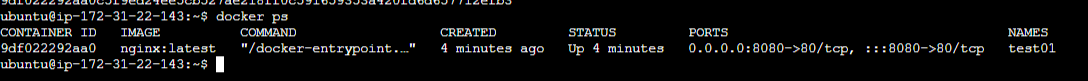
# Docker commands:

## To create & start a docker container.

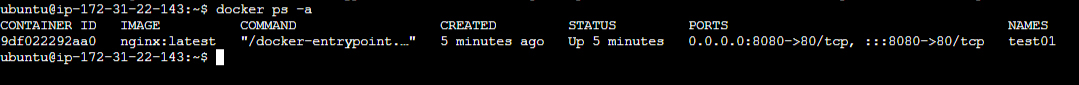
[command]: docker **run** -d -p 8080:80 --name test01 nginx:latest



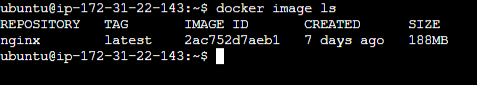
## To display only running container >> **docker ps**



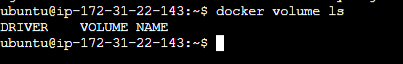
## To display all containers including both running and stopped containers. >> **docker ps -a**



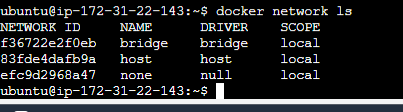
## To display docker images. >> **docker image ls**

s

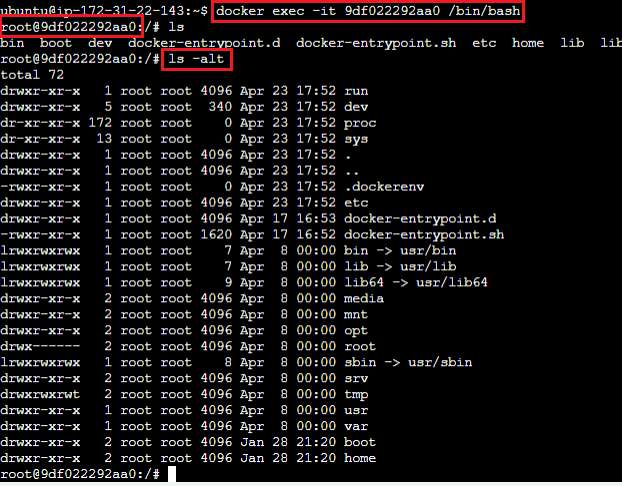
## To display volumes >> **docker volume ls**.



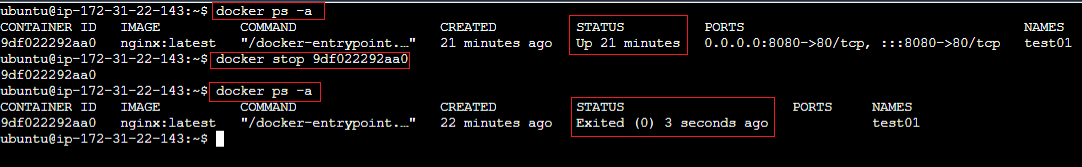
## To display docker networks >> **docker network ls**



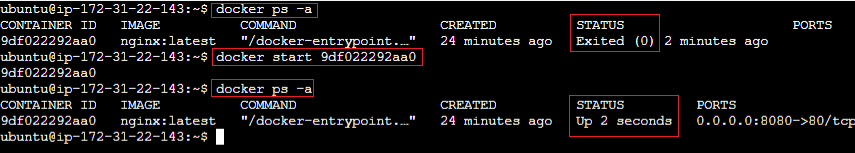
## To login to container >> **docker exec -it 9df022292aa0 /bin/bash**



## To stop a container.



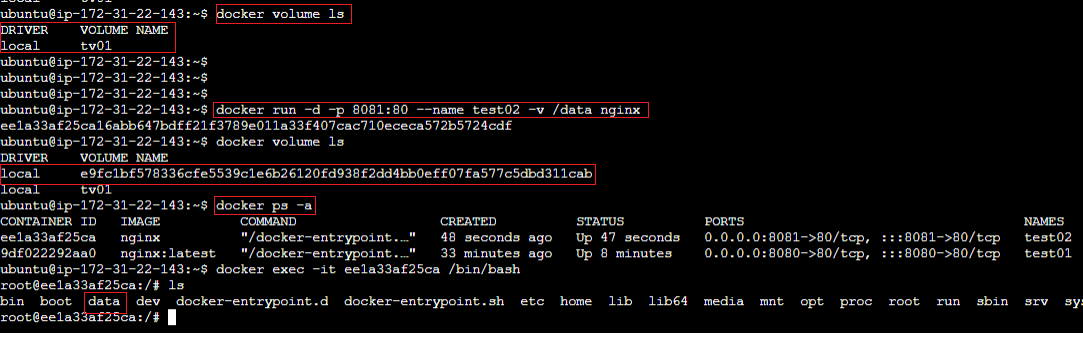
## To start a container.



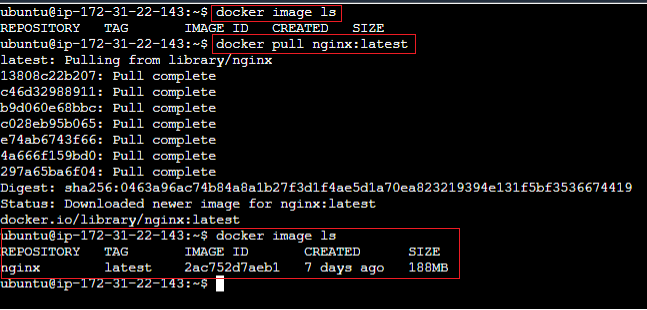
## To create a volume



## To mount a volume to docker container.

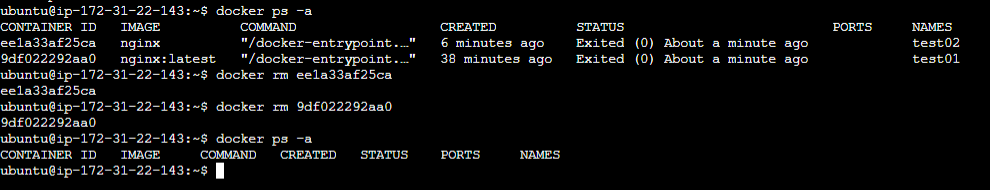


## Download a docker image from hub.docker.com

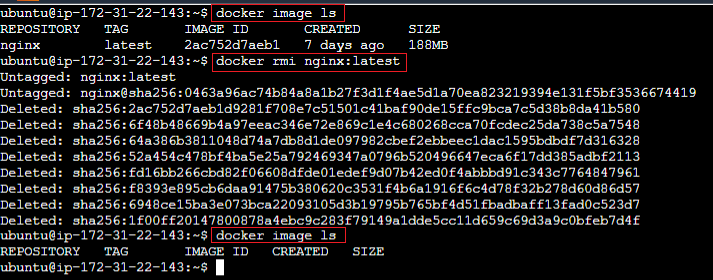


## To get only docker container IDs >> docker ps -a -q

## To delete container >> docker rm container\_id

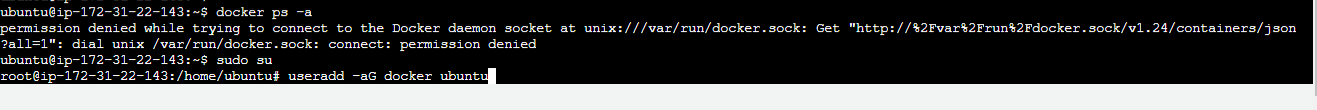


## To delete an image.



# FQA:

## Permission denied while trying to connect to the Docker daemon socket.



Solution: Add **“ubuntu”** user to docker group to get an access to docker commands.

**Command:**

sudo su -

usermod -aG docker ubuntu

systemctl restart docker

