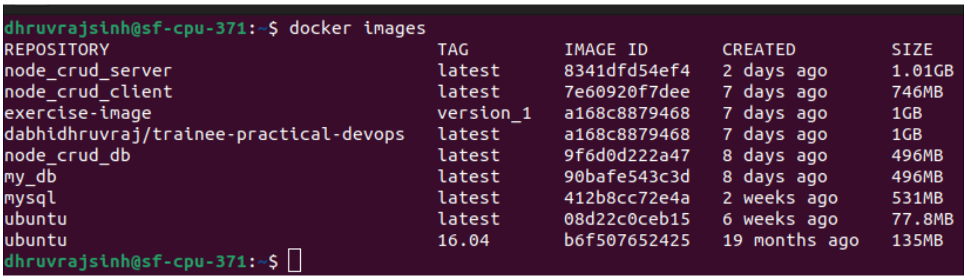
Exercise_1: running container

when we want to run a particular container, we must pull the image.

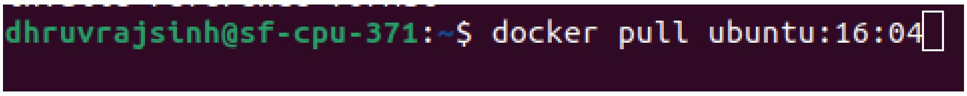
pulling an image:

If we want to check what images, we have currently on our machine by running **docker images.**



```
dhruvrajsinh@sf-cpu-371:~$ docker images
REPOSITORY                             TAG         IMAGE ID       CREATED         SIZE
node_crud_server                       latest      8341dfd54ef4   2 days ago      1.01GB
node_crud_client                       latest      7e60920f7dee   7 days ago      746MB
exercise-image                         version_1   a168c8879468   7 days ago      1GB
dabhidhruvraj/trainee-practical-devops latest      a168c8879468   7 days ago      1GB
node_crud_db                           latest      9f6d0d222a47   8 days ago      496MB
my_db                                  latest      90bafe543c3d   8 days ago      496MB
mysql                                  latest      412b8cc72e4a   2 weeks ago     531MB
ubuntu                                 latest      08d22c0ceb15   6 weeks ago     77.8MB
ubuntu                                 16.04       b6f507652425   19 months ago   135MB
dhruvrajsinh@sf-cpu-371:~$ 
```

let's pull the images from dockerhub by using the docker pull command. we want the image of ubuntu.



```
dhruvrajsinh@sf-cpu-371:~$ docker pull ubuntu:16:04
```

here 16:04 is a specific version of ubuntu images and we can also pull different version on the same image.

After this we can run **docker images** to see our images is pulling down or not

Removing images:

Ok now we want to delete or remove these images so we can do that by using **docker rmi** command then provide an id of that images or name of that images.

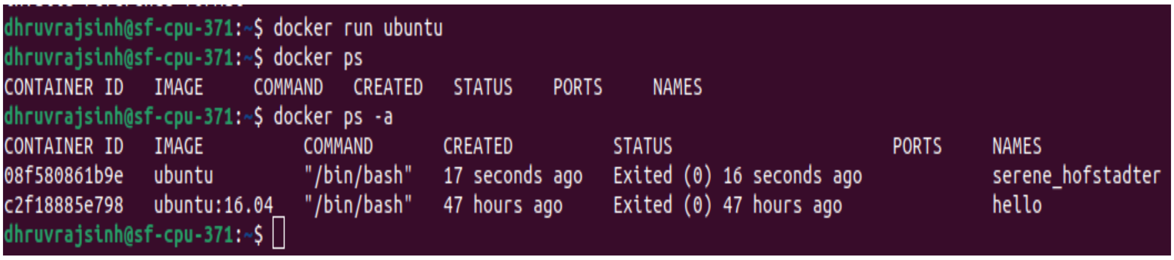
$ docker rmi 08d22

or

$ docker rmi ubuntu

running a container:

now we pull the images of ubuntu let's run our first container by using docker run command.

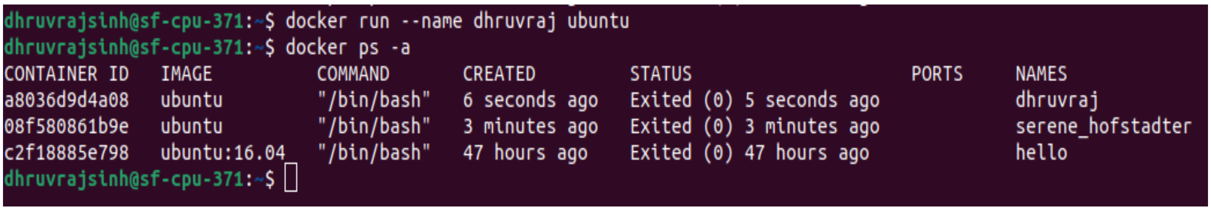


```
dhruvrajsinh@sf-cpu-371:~$ docker run ubuntu
dhruvrajsinh@sf-cpu-371:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
dhruvrajsinh@sf-cpu-371:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED          STATUS                      PORTS     NAMES
08f580861b9e   ubuntu         "/bin/bash"   17 seconds ago   Exited (0) 16 seconds ago             serene_hofstadter
c2f18885e798   ubuntu:16.04   "/bin/bash"   47 hours ago     Exited (0) 47 hours ago               hello
dhruvrajsinh@sf-cpu-371:~$ 
```

here we don't mention a name of container so docker create a random name for that container but if we want to give name by ourselves we can do that by **--name**

$ docker run --name dhruvraj ubuntu:16.04

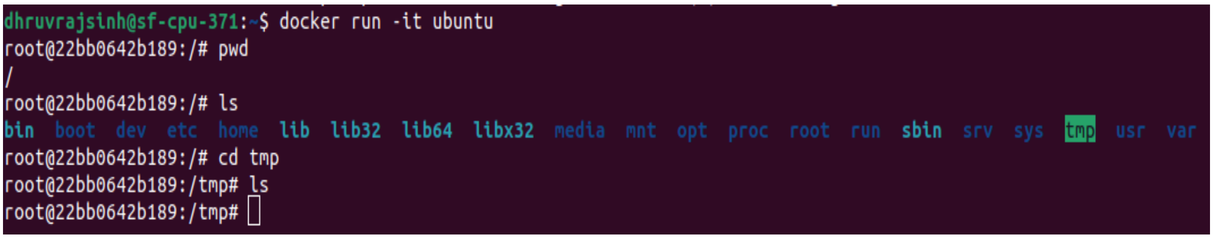
we can check the list of containers by typing docker ps command



```
dhruvrajsinh@sf-cpu-371:~$ docker run --name dhruvraj ubuntu
dhruvrajsinh@sf-cpu-371:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED          STATUS                      PORTS     NAMES
a8036d9d4a08   ubuntu         "/bin/bash"   6 seconds ago    Exited (0) 5 seconds ago              dhruvraj
08f580861b9e   ubuntu         "/bin/bash"   3 minutes ago    Exited (0) 3 minutes ago              serene_hofstadter
c2f18885e798   ubuntu:16.04   "/bin/bash"   47 hours ago     Exited (0) 47 hours ago               hello
dhruvrajsinh@sf-cpu-371:~$ 
```

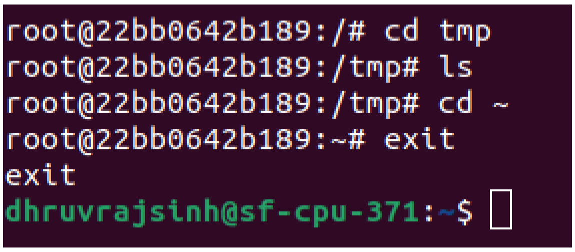
we can also see the previous container or stopped container by typing **docker ps -a** command.

also, we can run the container in interactively mode by providing -it.



```
dhruvrajsinh@sf-cpu-371:~$ docker run -it ubuntu
root@22bb0642b189:/# pwd
/
root@22bb0642b189:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@22bb0642b189:/# cd tmp
root@22bb0642b189:/tmp# ls
root@22bb0642b189:/tmp# 
```

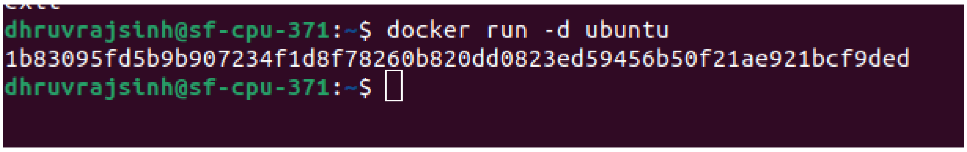
If we want to exit from this container provide exit.



```
root@22bb0642b189:/# cd tmp
root@22bb0642b189:/tmp# ls
root@22bb0642b189:/tmp# cd ~
root@22bb0642b189:~# exit
exit
dhruvrajsinh@sf-cpu-371:~$ 
```

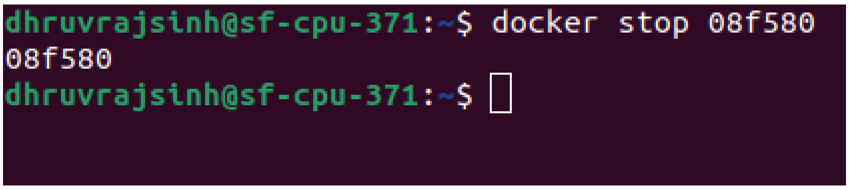
By default our terminal remains attached to the container when you run docker run. What if we don't want to remain attached?

By adding the -d flag, we can run in detached mode, meaning the container will continue to run as long as the command is, but it won't print the output.



```
exit
dhruvrajsinh@sf-cpu-371:~$ docker run -d ubuntu
1b83095fd5b9b907234f1d8f78260b820dd0823ed59456b50f21ae921bcf9ded
dhruvrajsinh@sf-cpu-371:~$ 
```

Let's use docker stop, passing it the first few characters of the container name or container id we want to stop.



```
dhruvrajsinh@sf-cpu-371:~$ docker stop 08f580
08f580
dhruvrajsinh@sf-cpu-371:~$ 
```