



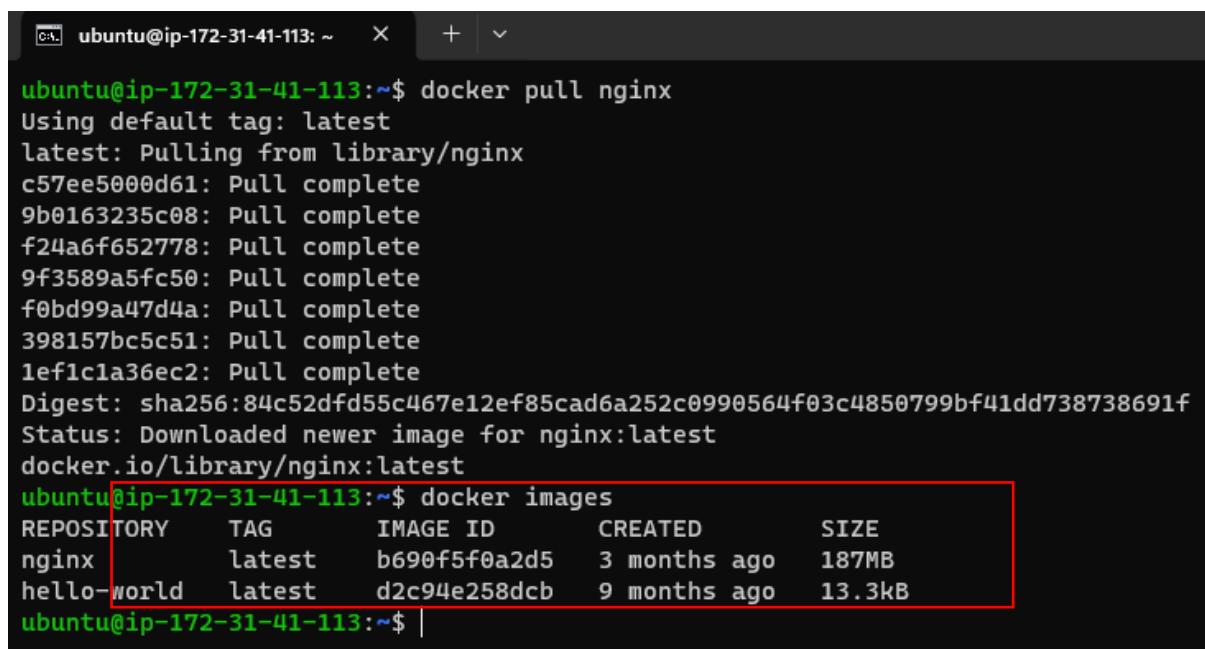
Docker Commands and Concepts

1. In this lab you are going to learn some docker commands.
2. So, first you need to visit this website for docker. This website contains all the images that you can use on docker.

<https://hub.docker.com/>

3. After that you need pull the image for Nginx using this command and run it. You will see again it is pulling the image from docker hub downloading then it is showing you that.
4. Now you'll run the command for docker images you will see that you have two images one is for nginx and other is for hello world which you created in the last lab.

docker pull nginx
docker images



```
ubuntu@ip-172-31-41-113:~$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
c57ee5000d61: Pull complete
9b0163235c08: Pull complete
f24a6f652778: Pull complete
9f3589a5fc50: Pull complete
f0bd99a47d4a: Pull complete
398157bc5c51: Pull complete
1ef1c1a36ec2: Pull complete
Digest: sha256:84c52dfd55c467e12ef85cad6a252c0990564f03c4850799bf41dd738738691f
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
ubuntu@ip-172-31-41-113:~$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
nginx           latest   b690f5f0a2d5   3 months ago  187MB
hello-world     latest   d2c94e258dcb   9 months ago  13.3kB
ubuntu@ip-172-31-41-113:~$ |
```

5. Now on the docker hub website you can find many things like you pulled nginx image from here. If you will go to nginx section on the website you will see its overview and tags.
6. So, you can use different tags by writing command in your Ubuntu session on docker. For example, if you want to change the current tag of nginx you can use this command below.
7. You can also find this command on docker hub website.

TAG					
mainline-bookworm			VULNERABILITIES		COMRESSED SIZE
Last pushed 2 days ago by d0ijanky			0	0	65.51 MB
DIGEST	OS/ARCH		38	2	
0b0c1d0dcce3c	linux/386		0	0	67.26 MB
d02f9b9db4d7	linux/amd64		0	0	60.18 MB
3884690c4c19	linux/arm/v5		38	2	
+5 more...					

docker pull nginx:mainline-bookworm
docker images

- Now if you run this command in your session. You can see that now you have two images for nginx one with the latest tag and the other with the tag that you used just now which is mainline-bookworm.

```
ubuntu@ip-172-31-41-113:~$ docker pull nginx:mainline-bookworm
mainline-bookworm: Pulling from library/nginx
Digest: sha256:84c52dfd55c467e12ef85cad6a252c0990564f03c4850799bf41dd738738691f
Status: Downloaded newer image for nginx:mainline-bookworm
docker.io/library/nginx:mainline-bookworm
ubuntu@ip-172-31-41-113:~$ docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
nginx           latest        b690f5f0a2d5   3 months ago  187MB
nginx           mainline-bookworm  b690f5f0a2d5   3 months ago  187MB
hello-world     latest        d2c94e258dcb   9 months ago  13.3kB
ubuntu@ip-172-31-41-113:~$ |
```

- Now you need to run this command. This command will return an ID.

docker run --name demo-docker -p 7090:80 -d nginx
docker ps
docker ps -a

```
ubuntu@ip-172-31-41-113:~$ docker run --name demo-docker -p 7090:80 -d nginx
9d1900e750eb82b6dc4a57cf15503f1a93b02594b12318768309de5f85bdc8ea
ubuntu@ip-172-31-41-113:~$ docker ps
CONTAINER ID      IMAGE      COMMAND                  CREATED             STATUS              PORTS
 NAMES
ubuntu@ip-172-31-41-113:~$ |
ubuntu@ip-172-31-41-113:~$ |
```

```
ubuntu@ip-172-31-41-113:~$ docker ps -a
CONTAINER ID      IMAGE      COMMAND                  CREATED             STATUS              PORTS
9d1900e750eb      nginx      "/docker-entrypoint..."  2 minutes ago    Up 2 minutes      0.0.0.0:7090->80/tcp, :::7090->80/tcp
fddeea1a345      hello-world  "/hello"                22 minutes ago   Exited (0) 22 minutes ago
ubuntu@ip-172-31-41-113:~$ |
```

- So, if you want to access your nginx server you need to add All traffic from everywhere in your security group on AWS Console.

11. Now if you'll copy the public IP address of your instance and add :7090 with it you can see your nginx webserver page up and running perfectly.

12. If you want to stop the nginx you can run the docker stop command with the container ID. And when you will run docker ps command you will see that it is in the exited state.

```
docker stop
docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
9d1900e750eb	nginx	"/docker-entrypoint..."	2 minutes ago	Up 2 minutes	0.0.0.0:7090->80/tcp, :::7090->80/tcp	demo-docker
f6d4eeala345	hello-world	"/hello"	22 minutes ago	Exited (0) 22 minutes ago		infallible_stonebraker

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ubuntu@ip-172-31-41-113:	nginx	"/docker-entrypoint..."	6 minutes ago	Exited (137) 17 seconds ago		demo-docker
f6d4eeala345	hello-world	"/hello"	26 minutes ago	Exited (0) 26 minutes ago		infallible_stonebraker

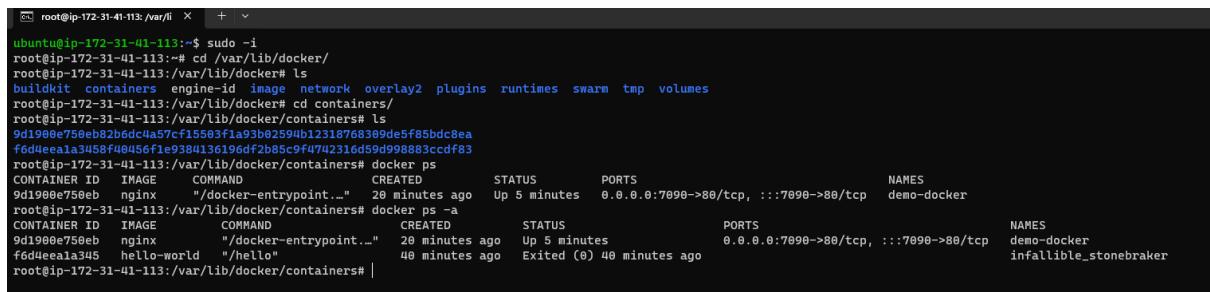
13. To start it again you can run the docker start command again with the container ID or name. Here you can see that the status changed to up and it is up for 5 seconds.

```
docker start
docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
9d1900e750eb	nginx	"/docker-entrypoint..."	15 minutes ago	Up 5 seconds	0.0.0.0:7090->80/tcp, :::7090->80/tcp	demo-docker
f6d4eeala345	hello-world	"/hello"	35 minutes ago	Exited (0) 35 minutes ago		infallible_stonebraker

14. Now a container run from directory, so switch to root user and go this location. After that if you do a listing, you can see the different folders there.
15. Now if you to the containers folder and do a listing there you will see two IDs because you have two images one for nginx and other is hello world.

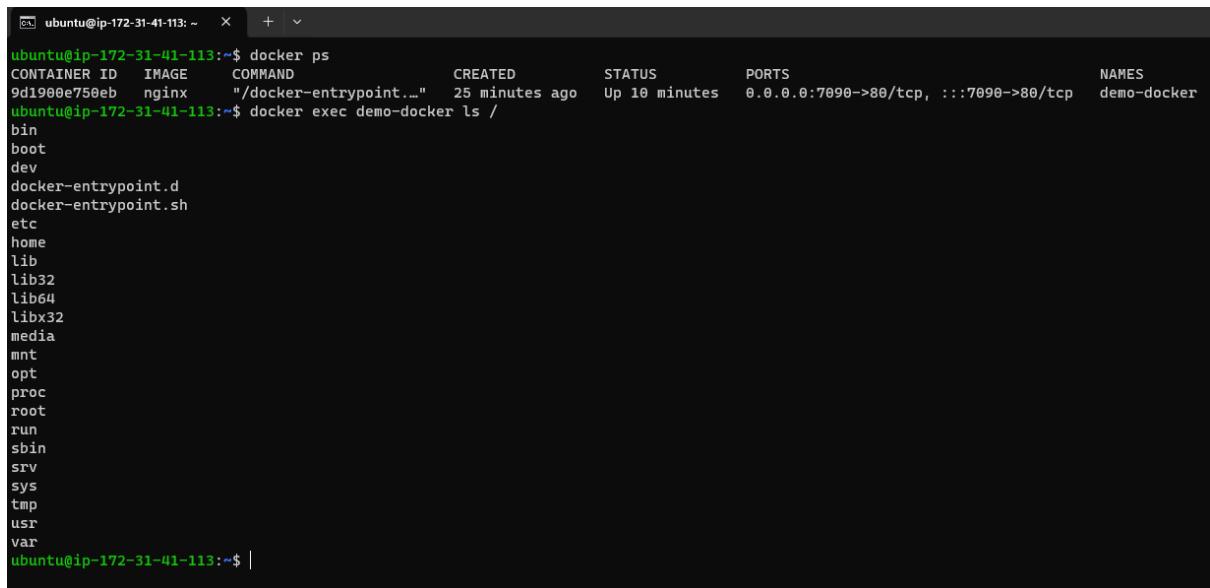
```
sudo -i
cd /var/lib/docker/
ls
cd containers/
ls
docker ps -a
```



```
root@ip-172-31-41-113:/var/lib/docker# sudo -i
root@ip-172-31-41-113:# cd /var/lib/docker/
root@ip-172-31-41-113:/var/lib/docker# ls
buildkit containers engine-id image network overlay2 plugins runtimes swarm tmp volumes
root@ip-172-31-41-113:/var/lib/docker# cd containers/
root@ip-172-31-41-113:/var/lib/docker/containers# ls
9d1980e750eb82b6dc4a57cf15503f1a93b02594012318768389de5f85bdc8ea
f6deea1a3458ff40456fe1e9384136196df2b85c9f4742316d59d998883ccdf83
root@ip-172-31-41-113:/var/lib/docker/containers# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
9d1980e750eb nginx "/docker-entrypoint..." 20 minutes ago Up 5 minutes 0.0.0.0:7090->80/tcp, :::7090->80/tcp demo-docker
root@ip-172-31-41-113:/var/lib/docker/containers# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
9d1980e750eb nginx "/docker-entrypoint..." 20 minutes ago Up 5 minutes 0.0.0.0:7090->80/tcp, :::7090->80/tcp demo-docker
f6deea1a345 hello-world "/hello" 40 minutes ago Exited (0) 40 minutes ago infallible_stonebraker
root@ip-172-31-41-113:/var/lib/docker/containers# |
```

16. Now run this command.

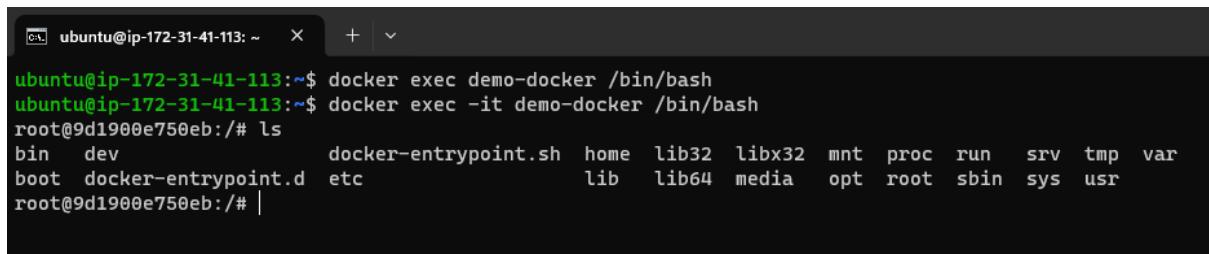
docker exec demo-docker ls /



```
ubuntu@ip-172-31-41-113:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
9d1980e750eb nginx "/docker-entrypoint..." 25 minutes ago Up 10 minutes 0.0.0.0:7090->80/tcp, :::7090->80/tcp demo-docker
ubuntu@ip-172-31-41-113:~$ docker exec demo-docker ls /
bin
boot
dev
docker-entrypoint.d
docker-entrypoint.sh
etc
home
lib
lib32
lib64
libx32
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
ubuntu@ip-172-31-41-113:~$ |
```

17. So, that command got executed in this container and you see that you should see similar directory structure like you see in any operating system or Linux operating system because it's a Linux container.
18. Now if you these commands below you will be inside the container.

```
docker exec demo-docker /bin/bash
docker exec -it demo-docker /bin/bash
```

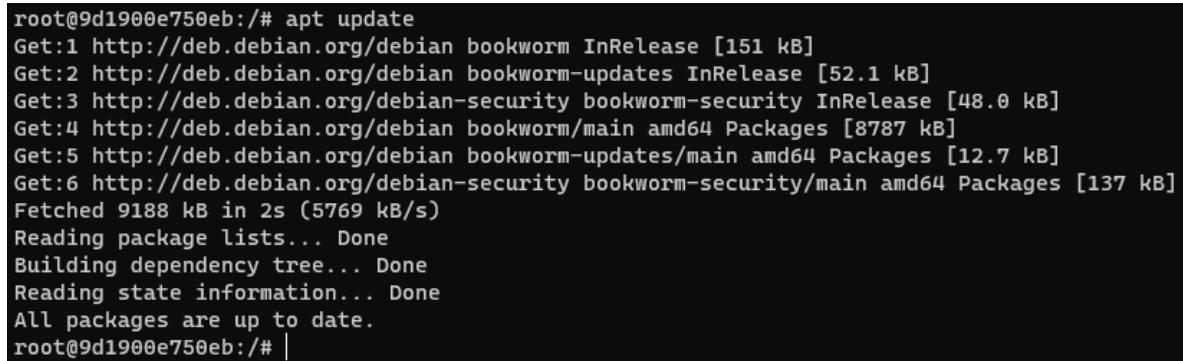


A terminal window titled 'ubuntu@ip-172-31-41-113: ~'. It shows the command 'ls' being run inside a Docker container named 'demo-docker'. The output lists standard Linux directories like bin, dev, boot, docker-entrypoint.d, etc, along with symbolic links for lib32, libx32, lib64, media, opt, root, sbin, sys, and usr.

```
ubuntu@ip-172-31-41-113:~$ docker exec demo-docker /bin/bash
ubuntu@ip-172-31-41-113:~$ docker exec -it demo-docker /bin/bash
root@9d1900e750eb:/# ls
bin  dev          docker-entrypoint.sh  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  docker-entrypoint.d  etc          lib    lib64  media  opt  root  sbin  sys  usr
root@9d1900e750eb:/# |
```

19. So, to run any command inside the container you need to first update it. Afterwards only you will be able to run commands in it.

```
apt update
apt install procps -y
ps -ef
```



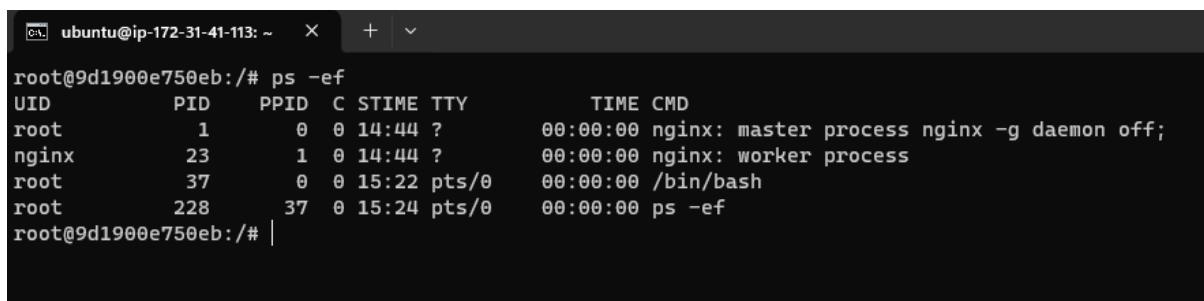
A terminal window titled 'root@9d1900e750eb:~'. It shows the 'apt update' command being run, which fetches packages from the 'bookworm' and 'bookworm-security' repositories. After the update, the 'ps -ef' command is run to show the process list, which includes the root shell and several system daemons.

```
root@9d1900e750eb:/# apt update
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [52.1 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8787 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [12.7 kB]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [137 kB]
Fetched 9188 kB in 2s (5769 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@9d1900e750eb:/# |
```

```

root@9d1900e750eb:/# apt install procps -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgpm2 libncursesw6 libproc2-0 psmisc
Suggested packages:
  gpm
The following NEW packages will be installed:
  libgpm2 libncursesw6 libproc2-0 procps psmisc
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 1178 kB of archives.
After this operation, 3778 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 libncursesw6 amd64 6.4-4 [134 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 libproc2-0 amd64 2:4.0.2-3 [62.8 kB]
Get:3 http://deb.debian.org/debian bookworm/main amd64 procps amd64 2:4.0.2-3 [709 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 libgpm2 amd64 1.20.7-10+b1 [14.2 kB]
Get:5 http://deb.debian.org/debian bookworm/main amd64 psmisc amd64 23.6-1 [259 kB]
Fetched 1178 kB in 0s (21.7 MB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libncursesw6:amd64.
(Reading database ... 7590 files and directories currently installed.)
Preparing to unpack .../libncursesw6_6.4-4_amd64.deb ...
Unpacking libncursesw6:amd64 (6.4-4) ...
Selecting previously unselected package libproc2-0:amd64.
Preparing to unpack .../libproc2-0_2%3a4.0.2-3_amd64.deb ...
Unpacking libproc2-0:amd64 (2:4.0.2-3) ...
Selecting previously unselected package procps.
Preparing to unpack .../procps_2%3a4.0.2-3_amd64.deb ...
Unpacking procps (2:4.0.2-3) ...
Selecting previously unselected package libgpm2:amd64.
Preparing to unpack .../libgpm2_1.20.7-10+b1_amd64.deb ...
Unpacking libgpm2:amd64 (1.20.7-10+b1) ...
Selecting previously unselected package psmisc.
Preparing to unpack .../psmisc_23.6-1_amd64.deb ...
Unpacking psmisc (23.6-1) ...
Setting up libgpm2:amd64 (1.20.7-10+b1) ...
Setting up psmisc (23.6-1) ...
Setting up libproc2-0:amd64 (2:4.0.2-3) ...
Setting up libncursesw6:amd64 (6.4-4) ...
Setting up procps (2:4.0.2-3) ...
Processing triggers for libc-bin (2.36-9+deb12u4) ...
root@9d1900e750eb:/# |

```



```

ubuntu@ip-172-31-41-113: ~ + ~
root@9d1900e750eb:/# ps -ef
UID      PID  PPID   C STIME TTY          TIME CMD
root        1      0  0 14:44 ?        00:00:00 nginx: master process nginx -g daemon off;
nginx      23      1  0 14:44 ?        00:00:00 nginx: worker process
root        37      0  0 15:22 pts/0    00:00:00 /bin/bash
root      228      37  0 15:24 pts/0    00:00:00 ps -ef
root@9d1900e750eb:/# |

```

20. Now you are going to remove the nginx image. For that you need to follow these commands below.

```

docker ps -a
docker images
docker rmi nginx:mainline-bookworm

```

```

ubuntu@ip-172-31-41-113:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
9d1900e750eb nginx "/docker-entrypoint...." 56 minutes ago Up 41 minutes 0.0.0.0:7090->8
0/tcp, :::7090->80/tcp demo-docker
f6d4eea1a345 hello-world "/hello"
infallible_stonebraker
ubuntu@ip-172-31-41-113:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest b690f5f0a2d5 3 months ago 187MB
nginx mainline-bookworm b690f5f0a2d5 3 months ago 187MB
hello-world latest d2c94e258dcb 9 months ago 13.3kB
ubuntu@ip-172-31-41-113:~$ docker rmi nginx:mainline-bookworm
Untagged: nginx:mainline-bookworm
ubuntu@ip-172-31-41-113:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest b690f5f0a2d5 3 months ago 187MB
hello-world latest d2c94e258dcb 9 months ago 13.3kB
ubuntu@ip-172-31-41-113:~$ |

```

21. But removing nginx directly is not easy because it running in the background so it needs to be stopped first. Then you need to remove the container ID then nginx itself.

docker ps

docker stop 'container ID'

docker rm 'container ID'

docker rmi nginx

```

ubuntu@ip-172-31-41-113:~$ docker rmi nginx
Error response from daemon: conflict: unable to remove repository reference "nginx" (must force) - container 9d1900e750eb is using its referenced image b690f5f0a2d5
ubuntu@ip-172-31-41-113:~$ |

```

```

ubuntu@ip-172-31-41-113:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
9d1900e750eb nginx "/docker-entrypoint...." About an hour ago Up 46 minutes 0.0.0.0:7090->80/tcp, :::7090->80/tcp demo-docker
ubuntu@ip-172-31-41-113:~$ docker rmi 9d1900e750eb
Error response from daemon: No such image: 9d1900e750eb:latest
ubuntu@ip-172-31-41-113:~$ docker stop 9d1900e750eb
9d1900e750eb
ubuntu@ip-172-31-41-113:~$ docker rmi 9d1900e750eb
Error response from daemon: No such image: 9d1900e750eb:latest
ubuntu@ip-172-31-41-113:~$ docker rm 9d1900e750eb
9d1900e750eb
ubuntu@ip-172-31-41-113:~$ docker rmi nginx
Untagged: nginx:latest
Untagged: nginx@sha256:84c52df55c467e12ef85cad6a252c0990564f03c4850799bf41dd738738691f
Deleted: sha256:b690f5f0a2d53ce5e08631aa508fef339c43bb91d5b1f7d77a1a05cea021a8
Deleted: sha256:2599673318db032df10bca9b9167be668b9579d72c3cedd1436a0ddcbc4686f
Deleted: sha256:3dfa00af383371dcbb76086fde405df32b75247bdf6db81110d992284140c5a3
Deleted: sha256:22bd60744d5a77166622ec69cc6528f63c9df0dd65b9c96934658c3684ae1f4
Deleted: sha256:d1a2f0b2f9a1e0eaf4488084f492dfef868dc0a64ca3e0e30f3b9be6ded452f0
Deleted: sha256:9cf4de78149512efb3285ea0da170ebcd38cda48d5eb900304006636db6facfb
Deleted: sha256:61901066ba33b727b13c970b9d7b7ed9a3056e30de96e835c9b01f4e73c4659a
Deleted: sha256:fb1bd2fc52827db4ce719cc1aafd4a035d68bc71183b3bc39014f23e9e5fa256
ubuntu@ip-172-31-41-113:~$ |

```

22. Now you are going to test one more image that is ubuntu itself.

```

ubuntu@ip-172-31-41-113:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
57c139bbda7e: Pull complete
Digest: sha256:e9569c25505f33ff72e88b2990887c9dcf230f23259da296eb814fc2b41af999
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
ubuntu@ip-172-31-41-113:~$ |

```

23. After getting the image if you want to interact with the image, you will see you are unable to do that because it is in exited status. It started and then it is dead.
24. That's because it is running bash shell and you need to interact with the bash shell. Otherwise, it cannot be running like that.

```
docker run ubuntu
docker ps -a
```

```
ubuntu@ip-172-31-41-113:~$ docker run ubuntu
ubuntu@ip-172-31-41-113:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
ubuntu@ip-172-31-41-113:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e739adc8afb8 ubuntu "/bin/bash" 7 seconds ago Exited (0) 6 seconds ago
f6d4eea1a345 hello-world "/hello" 2 hours ago Exited (0) 2 hours ago
ubuntu@ip-172-31-41-113:~$ |
```

25. By using the below commands, you will be in the container again. And you will see that the image is in the up state.

```
docker run -it ubuntu /bin/bash
ps -ef
```

```
ubuntu@ip-172-31-41-113:~$ docker run -it ubuntu /bin/bash
root@fa35cd34f8c9:/# ps -ef
UID          PID      PPID    C STIME TTY          TIME CMD
root            1          0   0 15:45 pts/0    00:00:00 /bin/bash
root            8          1   0 15:45 pts/0    00:00:00 ps -ef
root@fa35cd34f8c9:/# |
```

26. Now if you exit from the container, it will again be dead.
27. See it is in exited state again.

```
root@fa35cd34f8c9:/# exit
exit
ubuntu@ip-172-31-41-113:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
fa35cd34f8c9 ubuntu "/bin/bash" 3 minutes ago Exited (0) 8 seconds ago
e739adc8afb8 ubuntu "/bin/bash" 6 minutes ago Exited (0) 6 minutes ago
f6d4eea1a345 hello-world "/hello" 2 hours ago Exited (0) 2 hours ago
ubuntu@ip-172-31-41-113:~$ |
```

28. Now for the cleanup remove all the images.

```
ubuntu@ip-172-31-41-113:~$ docker rm elegant_feistel pedantic_newton infallible_stonebraker
elegant_feistel
pedantic_newton
infallible_stonebraker
```

```
ubuntu@ip-172-31-41-113:~$ docker rmi ubuntu hello-world
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:e9569c25505f33ff72e88b2990887c9dcf230f23259da296eb814fc2b41af999
Deleted: sha256:fd1d8f58e8aedc22ec0a3a7ce1a33de544a596eaa6cdb842f1af7c5e081d453f
Deleted: sha256:1a102d1cac2bdae8a0160ac4365d4f8653e9d6da56c793a665d556ae07fb7f82
Untagged: hello-world:latest
Untagged: hello-world@sha256:4bd78111b6914a99dbc560e6a20eab57ff6655aea4a80c50b0c5491968cbc2e6
Deleted: sha256:d2c94e258dcb3c5ac2798d32e1249e42ef01cba4841c2234249495f87264ac5a
Deleted: sha256:ac28800ec8bb38d5c35b49d45a6ac4777544941199075dff8c4eb63e093aa81e
ubuntu@ip-172-31-41-113:~$ |
```