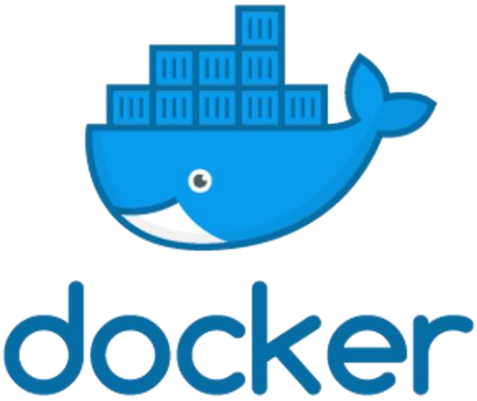
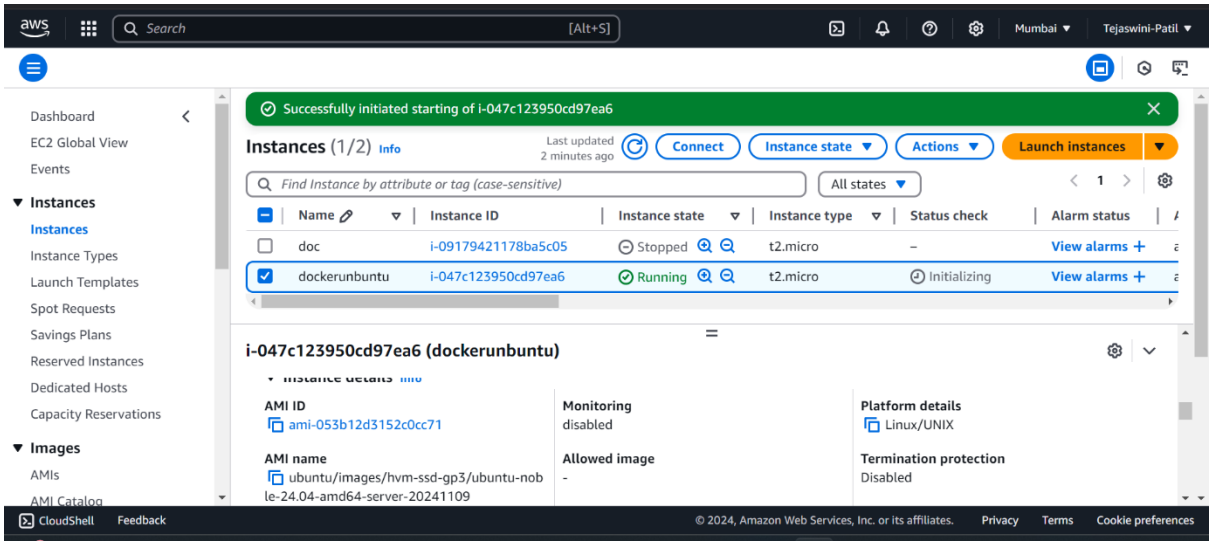
Using Docker CLI

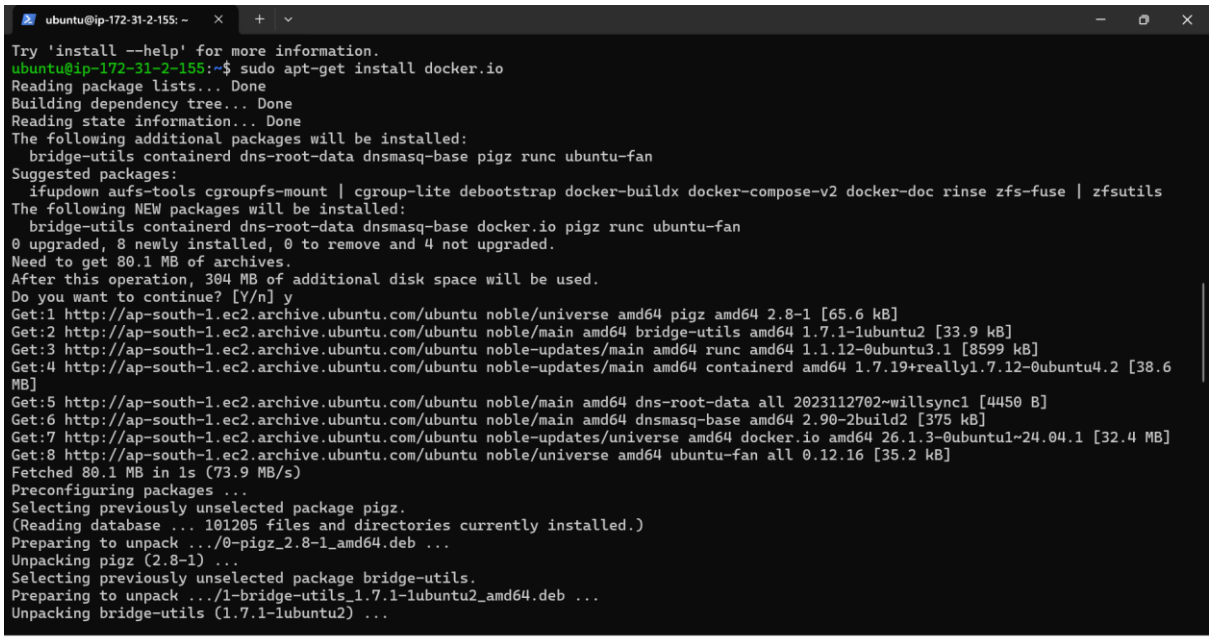


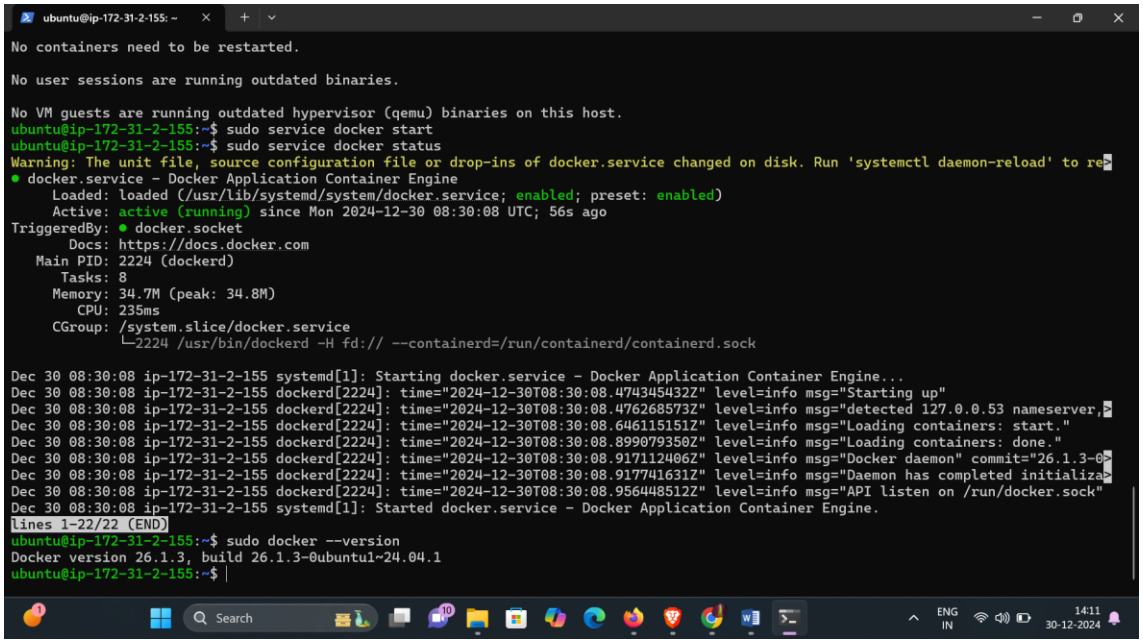
#Create Ec2:



Install and start Docker :

- sudo apt-get update
- sudo apt-get install docker-io
- sudo service docker start
- sudo service docker status





Root user and pull nginx :

- sudo su
- docker pull nginx

Show containers : (old command)

- docker ps
- docker ps -a

Show containers : (new command)

- docker container ls
- docker container ls -a

Show list of images :

- docker images
- docker image ls

```
ubuntu@ip-172-31-2-155:~$ sudo su
root@ip-172-31-2-155:/home/ubuntu# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
fd674058ff8f: Pull complete
566e42bcee1c: Pull complete
2b99b9c5d9e5: Pull complete
bd98674871f5: Pull complete
1e109dd2a0d7: Pull complete
da8cc133ff82: Pull complete
c44f27309ea1: Pull complete
Digest: sha256:42e917aaa1b5bb40dd0f6f7f4f857490ac7747d7ef73b391c774a41a8b994f15
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
root@ip-172-31-2-155:/home/ubuntu# docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
nginx        latest    f876bfc1cc63   4 weeks ago   192MB
root@ip-172-31-2-155:/home/ubuntu# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@ip-172-31-2-155:/home/ubuntu# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

Run command :

- docker run command is combination of docker pull , docker create , docker start .

- docker run nginx (runs nginx)
- docker run -d -p80:80 nginx (to run nginx in background)
- docker run -d -p80:80 --name mywebsite nginx (to run docker in background on port 80 of ec2 and port 80 of docker and give name to container)

Create command:

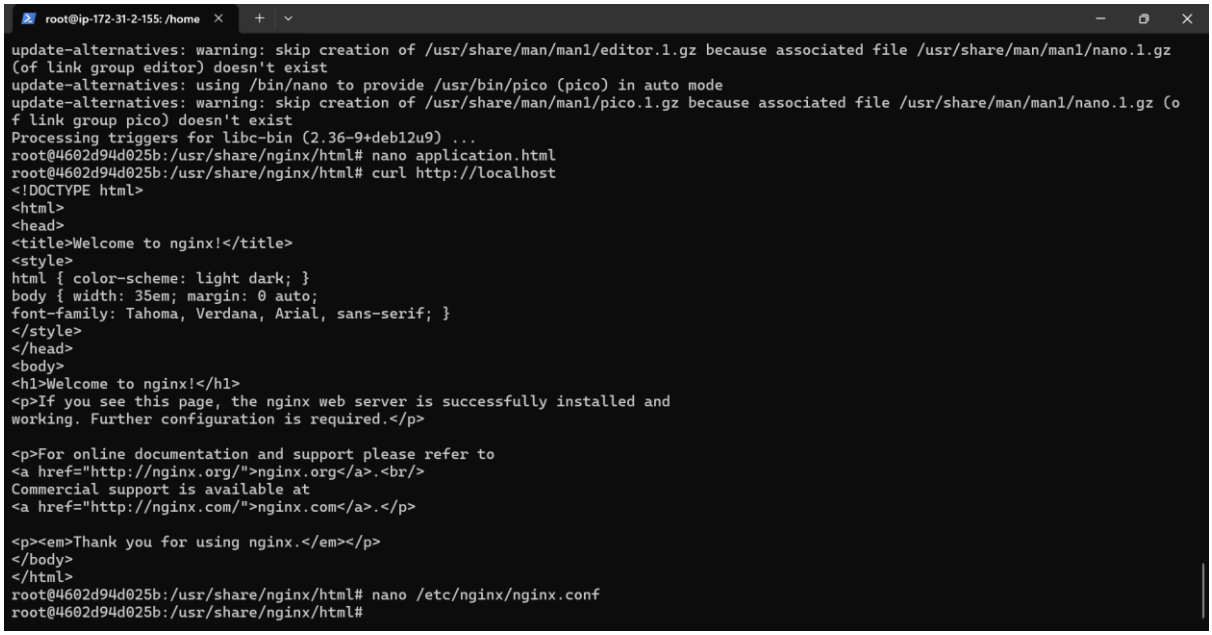
- docker create only creates container from image but the container is in stoped condition.

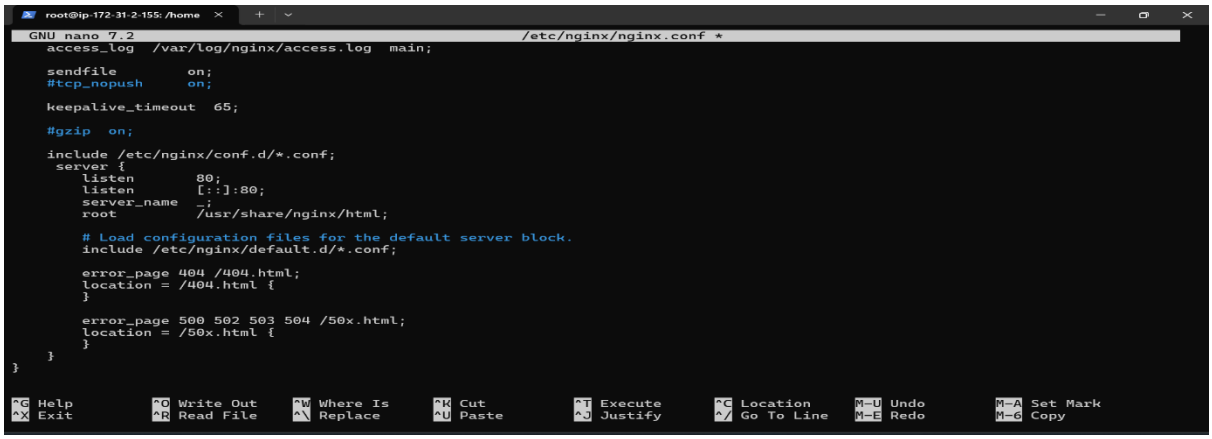
- docker run -d httpd (to get apache image) or
- docker create httpd
- docker start httpd (to start httpd)
- docker run -d -p80:80 --name containername imagename (to create new container)

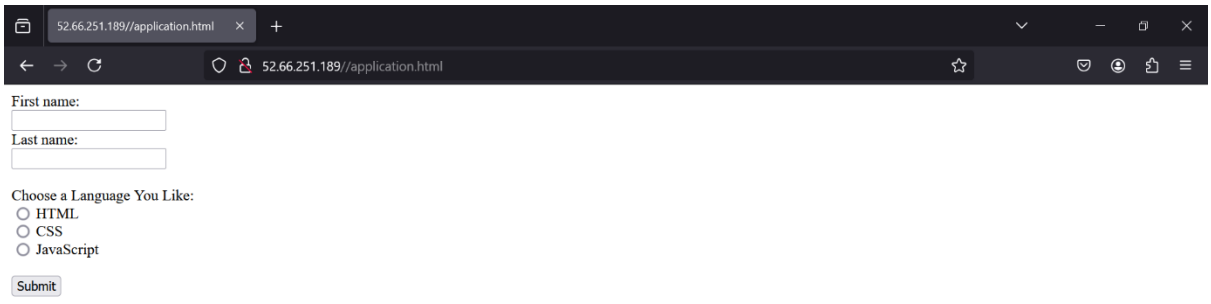
To execute Container :

- docker exec -it containerid/name /bin/bash

```
root@ip-172-31-2-155:/home/ubuntu# docker run -d -p 80:80 --name myweb nginx
4602d94d025b586beb15d4862b529a3f10504fae2a43f3cb34d9af313d725500
root@ip-172-31-2-155:/home/ubuntu# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                                   NAMES
4602d94d025b   nginx     "/docker-entrypoint.…"   27 seconds ago   Up 26 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp   myweb
root@ip-172-31-2-155:/home/ubuntu# docker exec -it myweb /bin/bash
root@4602d94d025b:/# ls
bin   dev                  docker-entrypoint.sh  home  lib64  mnt  proc  run   srv  tmp  var
boot  docker-entrypoint.d  etc                   lib   media  opt  root  sbin  sys  usr
root@4602d94d025b:/# cd /usr/share/nginx/html
root@4602d94d025b:/usr/share/nginx/html# apt install nano -y
```


root@ip-172-31-2-155: /home
update-alternatives: warning: skip creation of /usr/share/man/man1/editor.1.gz because associated file /usr/share/man/man1/nano.1.gz
(of link group editor) doesn't exist
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/pico.1.gz because associated file /usr/share/man/man1/nano.1.gz (o
f link group pico) doesn't exist
Processing triggers for libc-bin (2.36-9+deb12u9) ...
root@4602d94d025b:/usr/share/nginx/html# nano application.html
root@4602d94d025b:/usr/share/nginx/html# curl http://localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.
For online documentation and support please refer to
nginx.org.
Commercial support is available at
nginx.com.
Thank you for using nginx.
</body>
</html>
root@4602d94d025b:/usr/share/nginx/html# nano /etc/nginx/nginx.conf
root@4602d94d025b:/usr/share/nginx/html#


root@ip-172-31-2-155: /home
GNU nano 7.2 /etc/nginx/nginx.conf *
access_log /var/log/nginx/access.log main;
sendfile on;
#tcp_nopush on;
keepalive_timeout 65;
#gzip on;
include /etc/nginx/conf.d/*.conf;
server {
listen 80;
listen [::]:80;
server_name _;
root /usr/share/nginx/html;
Load configuration files for the default server block.
include /etc/nginx/default.d/*.conf;
error_page 404 /404.html;
location = /404.html {
}
error_page 500 502 503 504 /50x.html;
location = /50x.html {
}
}
}
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy


52.66.251.189//application.html
52.66.251.189//application.html
First name:
Last name:
Choose a Language You Like:
HTML
CSS
JavaScript
Submit

Stop container :

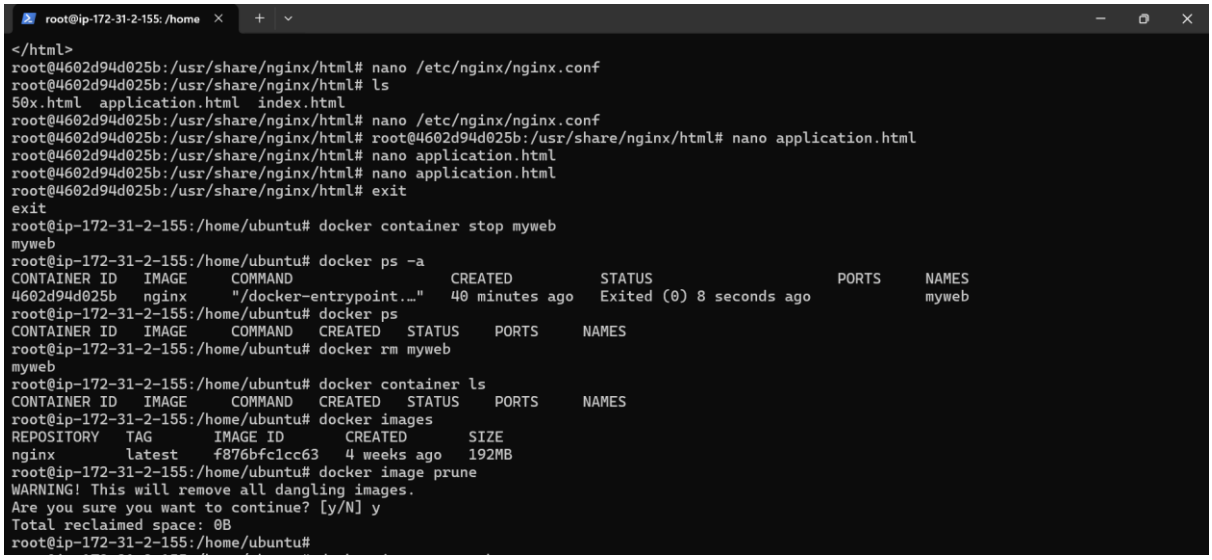
- docker container stop containerid/name
- docker stop $(docker ps -a)

Remove container:

- docker rm containerid/name (to remove one container)
- docker rm $(docker ps -aq) (to remove all stoped container which are stoped at once)

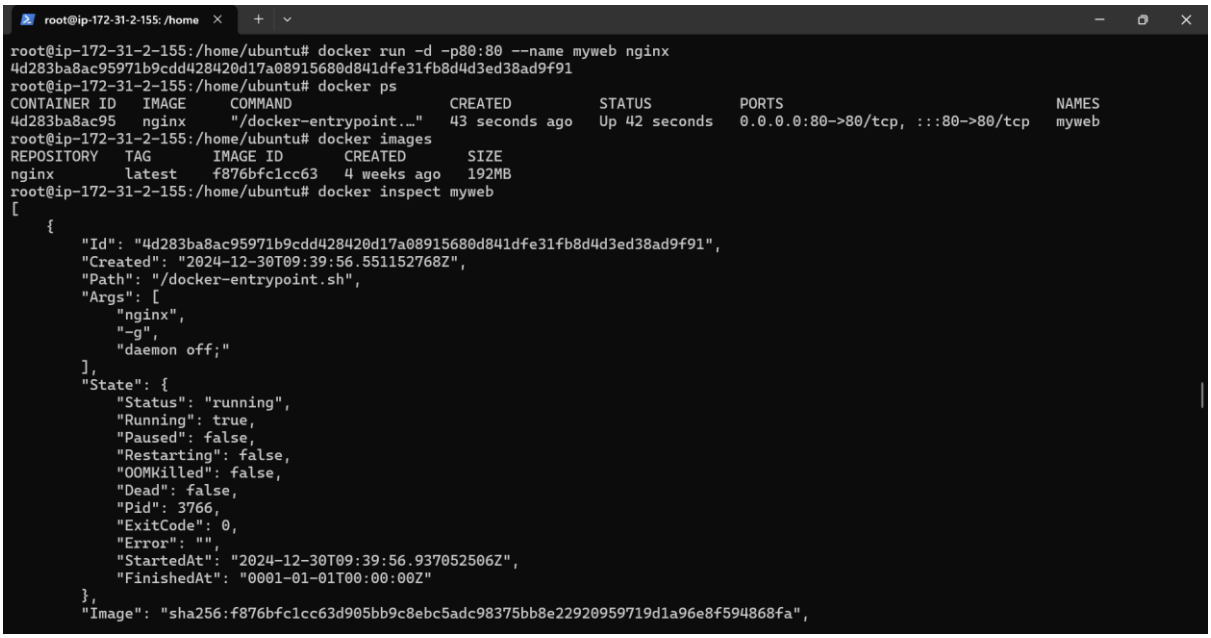
Remove image :

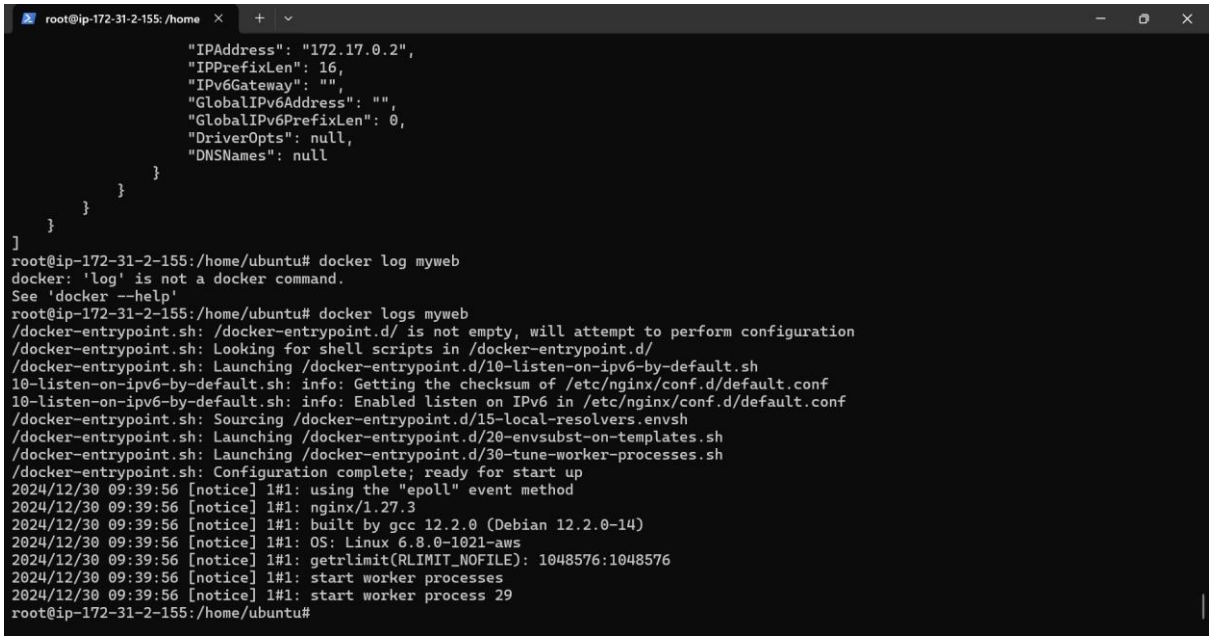
- docker rmi imageid (remove one image)
- docker image rm imageid (remove one image)
- docker image prune (remove all unused images)



Logs :

- docker logs containerid (see to any problem and error)
- docker inspect containerid (setting used to create container)





Types of Volume for Container :

- one volume can be assigned to multiple container.

1) Host path (Bind Mount):
2) Named Volume :
3) Anonymous Volume :

Allocate CPU and memory to container :

docker run --CPU 0.5 --memory 512mb -d