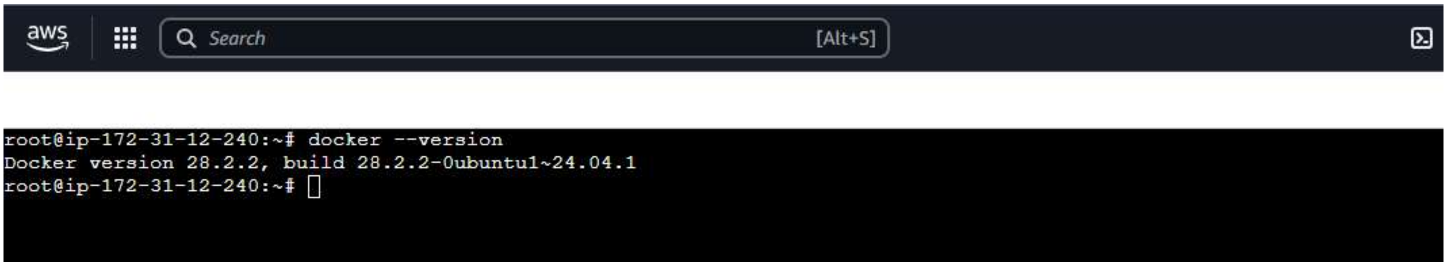
Docker Hands-On Practice & Revision Assignment.

1. Understanding of Docker.

1. Docker is an open-source containerization platform that allows developers to build, and run applications in isolated environments called containers. Also, Docker is a tool that helps create and run applications inside lightweight containers. It makes apps easy to build, test, and deploy across different environments without compatibility issues. I learned how Docker improves speed, consistency, and efficiency in software deployment.
2. **Docker Structure**:
   Docker Client: The command-line tool we use to talk to Docker (docker run, docker build, etc.).
   Docker Daemon: The background service that actually builds and runs containers.
   Docker Images: Ready-made blueprints used to create containers.
   Docker Containers: Running versions of images that hold our app and everything it needs.
   Docker Hub: Place to store and share Docker images online.
3. Docker makes the software delivery process easier by:
   Dockerfile: Package applications with all required files.
   Docker Hub: Through Store and share container images.
   Docker Compose. Using Run and manage multiple containers.
   Overall, Docker has helped me understand DevOps better by improving application portability, consistent deployments, and automation in the CI/CD process.

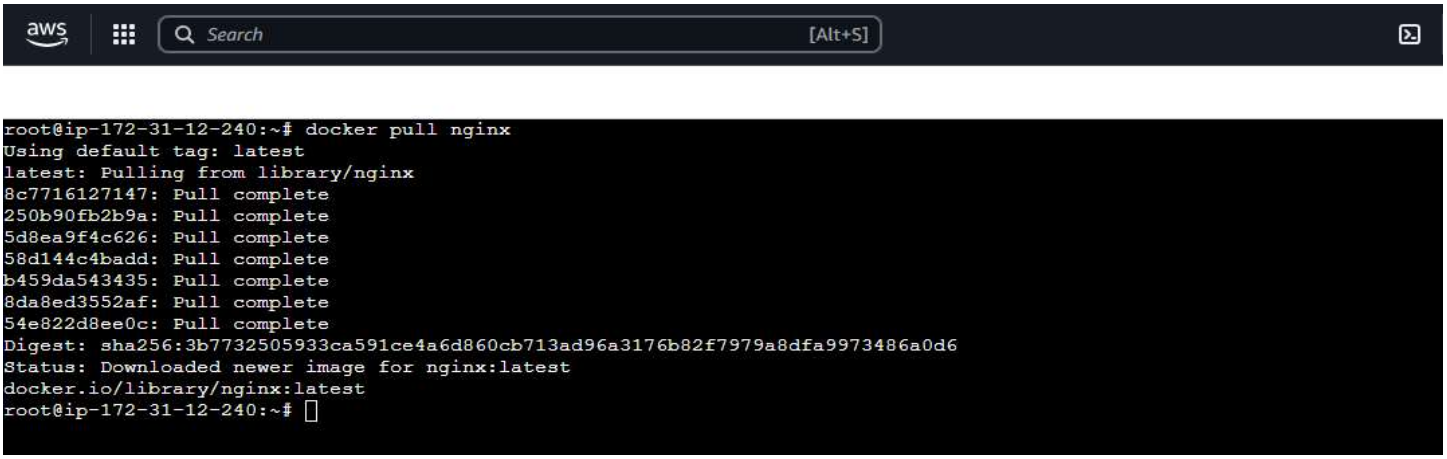
2. List of Docker Commands We Practiced.

CMD: docker –version: Check installed docker version.



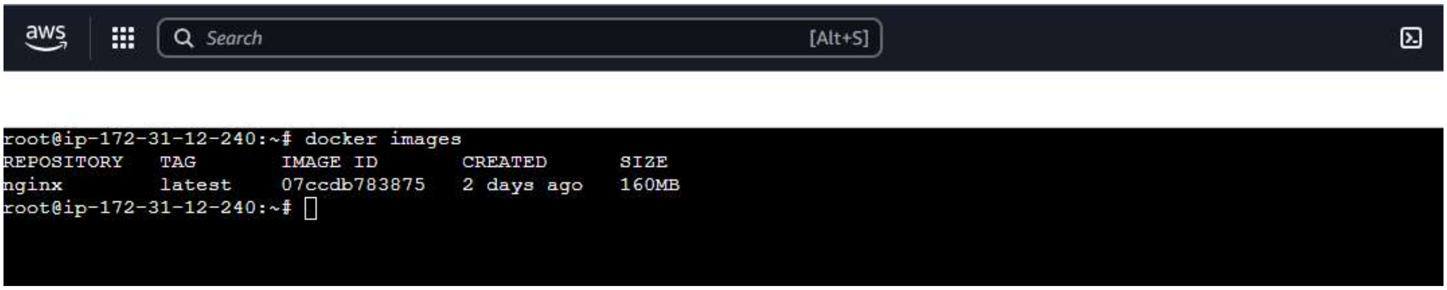
```
root@ip-172-31-12-240:~# docker --version
Docker version 28.2.2, build 28.2.2-0ubuntu1~24.04.1
root@ip-172-31-12-240:~# 
```

CMD: docker pull <Image-Name:Tags>: Download images from Docker Hub.



```
root@ip-172-31-12-240:~# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
8c7716127147: Pull complete
250b90fb2b9a: Pull complete
5d8ea9f4c626: Pull complete
58d144c4badd: Pull complete
b459da543435: Pull complete
8da8ed3552af: Pull complete
54e822d8ee0c: Pull complete
Digest: sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
root@ip-172-31-12-240:~# 
```

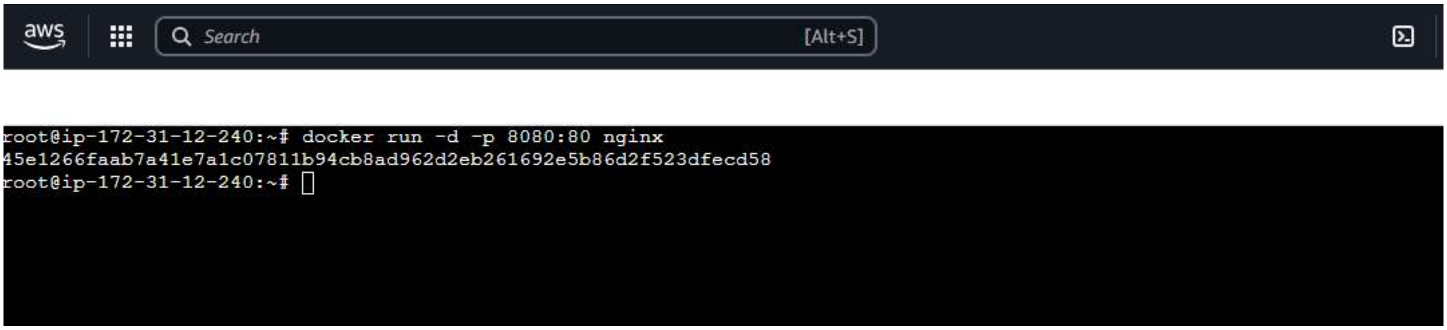
CMD: docker images: List all available docker images on server.



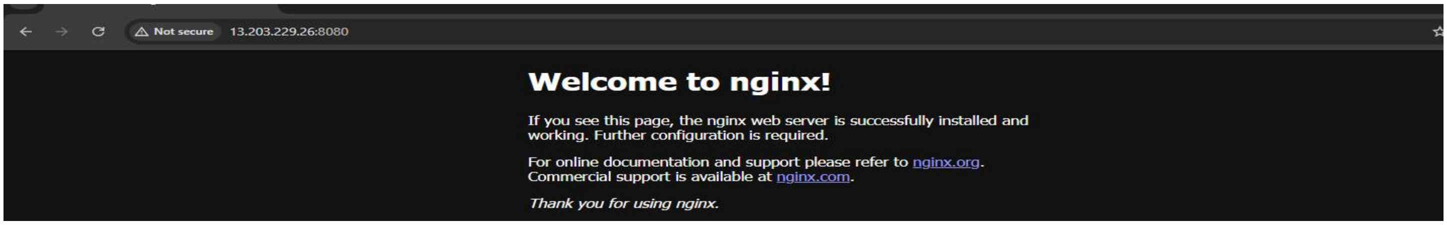
```
root@ip-172-31-12-240:~# docker images
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
nginx        latest    07ccdb783875   2 days ago   160MB
root@ip-172-31-12-240:~# 
```

CMD: docker run: Run a container from Images

docker run -d -p <Port-number> <Image-name>



```
root@ip-172-31-12-240:~# docker run -d -p 8080:80 nginx
45e1266faab7a41e7a1c07811b94cb8ad962d2eb261692e5b86d2f523dfecd58
root@ip-172-31-12-240:~# 
```



Welcome to nginx!

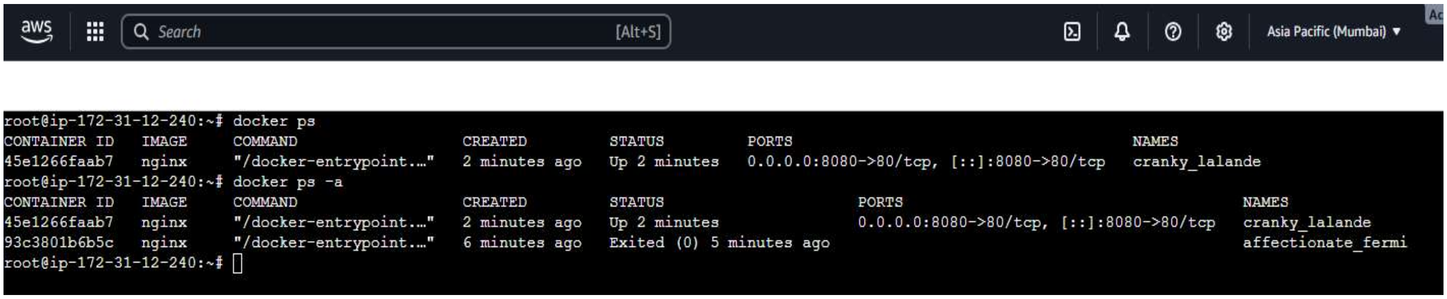
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

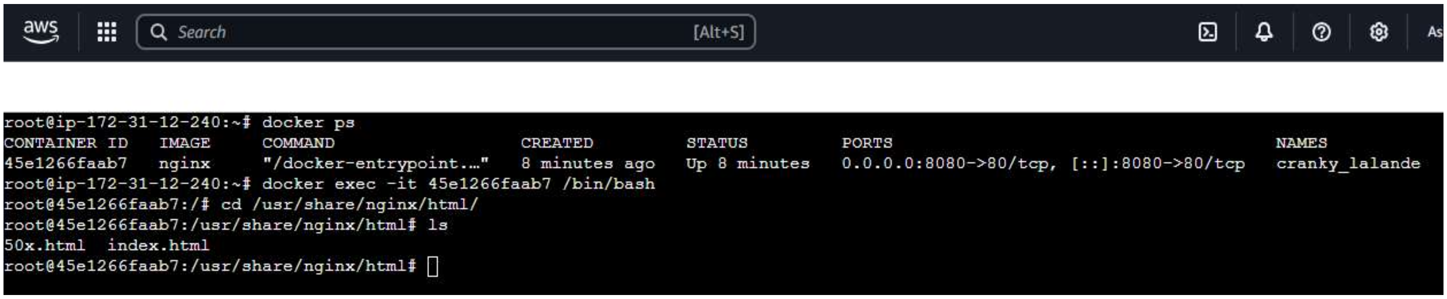
CMD: docker ps: List running container.

Docker ps -a: List all running and stopped container.



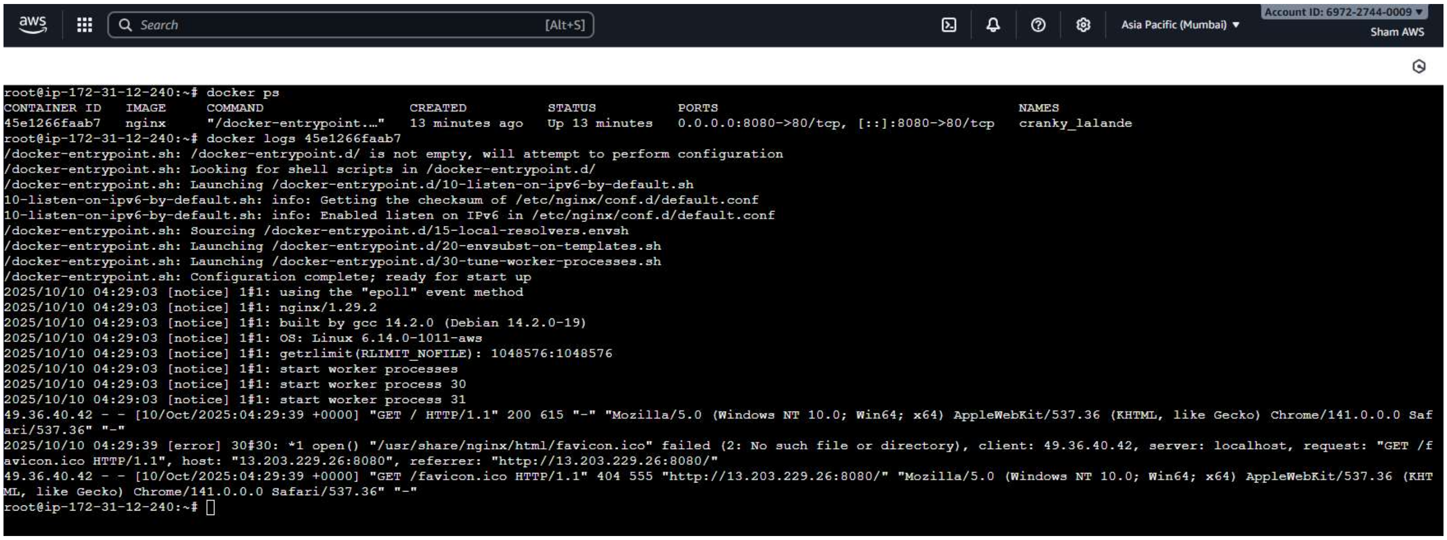
```
root@ip-172-31-12-240:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                                   NAMES
45e1266faab7   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   cranky_lalande
root@ip-172-31-12-240:~# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS                     PORTS                                   NAMES
45e1266faab7   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes               0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   cranky_lalande
93c3801b6b5c   nginx     "/docker-entrypoint.…"   6 minutes ago   Exited (0) 5 minutes ago                                           affectionate_fermi
root@ip-172-31-12-240:~# 
```

CMD: docker exec -it <Container-ID> /bin/bash: Access running container in shell.



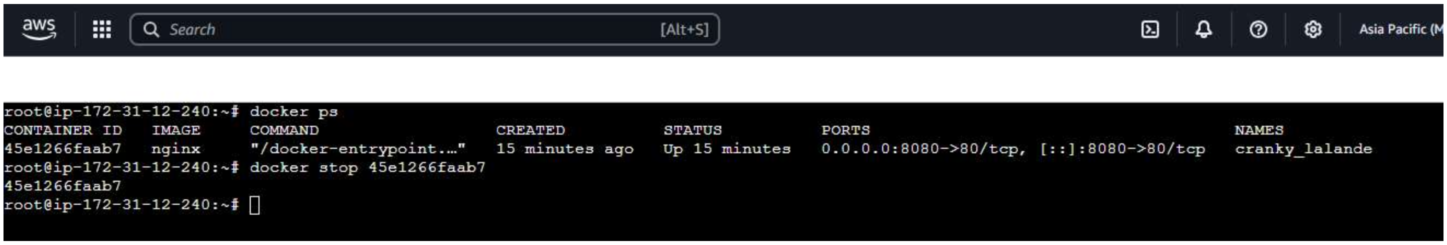
```
root@ip-172-31-12-240:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                                   NAMES
45e1266faab7   nginx     "/docker-entrypoint.…"   8 minutes ago   Up 8 minutes   0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   cranky_lalande
root@ip-172-31-12-240:~# docker exec -it 45e1266faab7 /bin/bash
root@45e1266faab7:/# cd /usr/share/nginx/html/
root@45e1266faab7:/usr/share/nginx/html# ls
50x.html  index.html
root@45e1266faab7:/usr/share/nginx/html# 
```

CMD: docker logs <Container-ID>: View logs of Container.



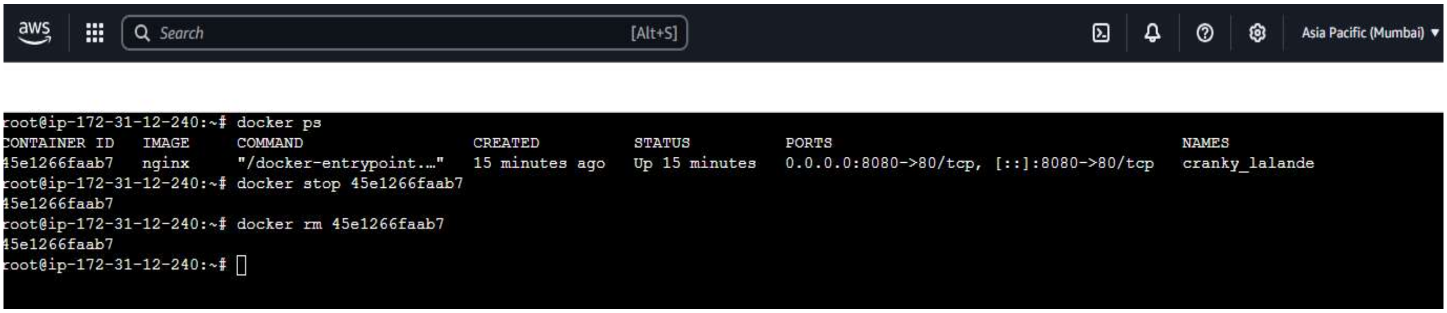
```
root@ip-172-31-12-240:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                                     NAMES
45e1266faab7   nginx     "/docker-entrypoint.…"   13 minutes ago   Up 13 minutes   0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   cranky_lalande
root@ip-172-31-12-240:~# docker logs 45e1266faab7
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/10/10 04:29:03 [notice] 1#1: using the "epoll" event method
2025/10/10 04:29:03 [notice] 1#1: nginx/1.29.2
2025/10/10 04:29:03 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2025/10/10 04:29:03 [notice] 1#1: OS: Linux 6.14.0-1011-aws
2025/10/10 04:29:03 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/10/10 04:29:03 [notice] 1#1: start worker processes
2025/10/10 04:29:03 [notice] 1#1: start worker process 30
2025/10/10 04:29:03 [notice] 1#1: start worker process 31
49.36.40.42 - - [10/Oct/2025:04:29:39 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Saf
ari/537.36" "-"
2025/10/10 04:29:39 [error] 30#30: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 49.36.40.42, server: localhost, request: "GET /f
avicon.ico HTTP/1.1", host: "13.203.229.26:8080", referrer: "http://13.203.229.26:8080/"
49.36.40.42 - - [10/Oct/2025:04:29:39 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://13.203.229.26:8080/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHT
ML, like Gecko) Chrome/141.0.0.0 Safari/537.36" "-"
root@ip-172-31-12-240:~# 
```

CMD: docker stop <Container-ID>: Stop Running Container.



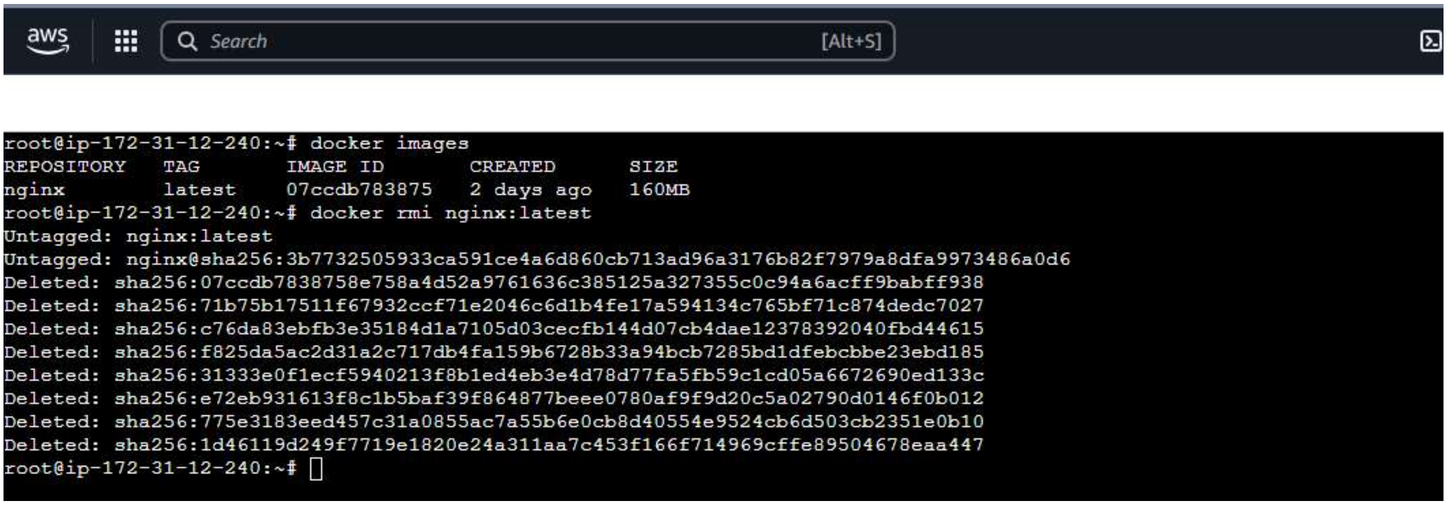
```
root@ip-172-31-12-240:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                                     NAMES
45e1266faab7   nginx     "/docker-entrypoint.…"   15 minutes ago   Up 15 minutes   0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   cranky_lalande
root@ip-172-31-12-240:~# docker stop 45e1266faab7
45e1266faab7
root@ip-172-31-12-240:~# 
```

CMD: docker rm <Container-ID> Remove stopped Container.



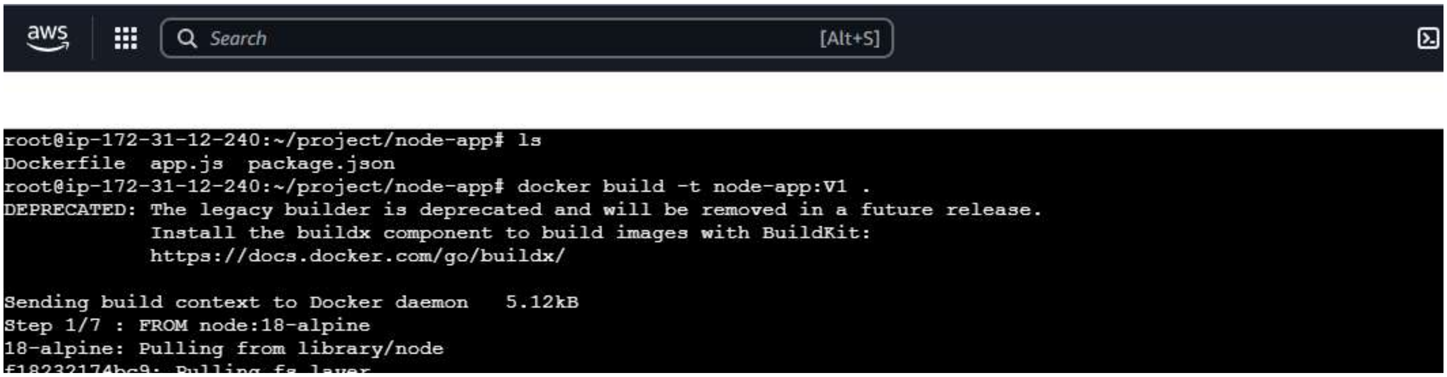
```
oot@ip-172-31-12-240:~# docker ps
ONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                                     NAMES
5e1266faab7   nginx     "/docker-entrypoint.…"   15 minutes ago   Up 15 minutes   0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   cranky_lalande
oot@ip-172-31-12-240:~# docker stop 45e1266faab7
5e1266faab7
oot@ip-172-31-12-240:~# docker rm 45e1266faab7
5e1266faab7
oot@ip-172-31-12-240:~# 
```

CMD: docker rmi <Image-name>: Remove an Image from local System.

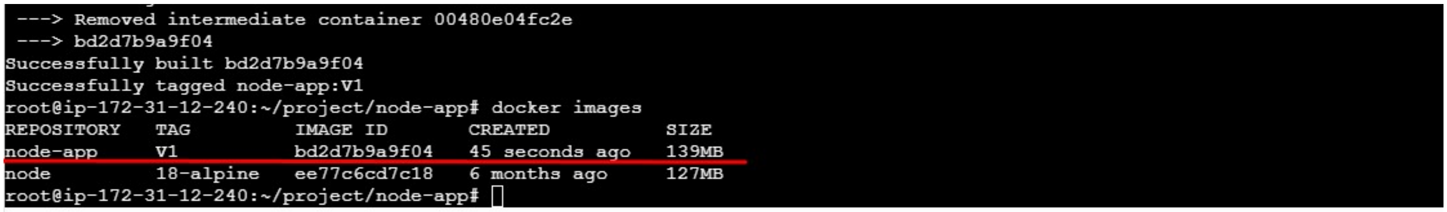


```
root@ip-172-31-12-240:~# docker images
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
nginx        latest    07ccdb783875   2 days ago   160MB
root@ip-172-31-12-240:~# docker rmi nginx:latest
Untagged: nginx:latest
Untagged: nginx@sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6
Deleted: sha256:07ccdb7838758e758a4d52a9761636c385125a327355c0c94a6acff9babff938
Deleted: sha256:71b75b17511f67932ccf71e2046c6d1b4fe17a594134c765bf71c874dedc7027
Deleted: sha256:c76da83ebfb3e35184d1a7105d03cecfb144d07cb4dae12378392040fbd44615
Deleted: sha256:f825da5ac2d31a2c717db4fa159b6728b33a94bcb7285bd1dfebcbbe23ebd185
Deleted: sha256:31333e0f1ecf5940213f8b1ed4eb3e4d78d77fa5fb59c1cd05a6672690ed133c
Deleted: sha256:e72eb931613f8c1b5baf39f864877beee0780af9f9d20c5a02790d0146f0b012
Deleted: sha256:775e3183eed457c31a0855ac7a55b6e0cb8d40554e9524cb6d503cb2351e0b10
Deleted: sha256:1d46119d249f7719e1820e24a311aa7c453f166f714969cffe89504678eaa447
root@ip-172-31-12-240:~# 
```

CMD: docker build -t <Image-Name>:<Tags> . : Build a Custom Image from Dockerfile.

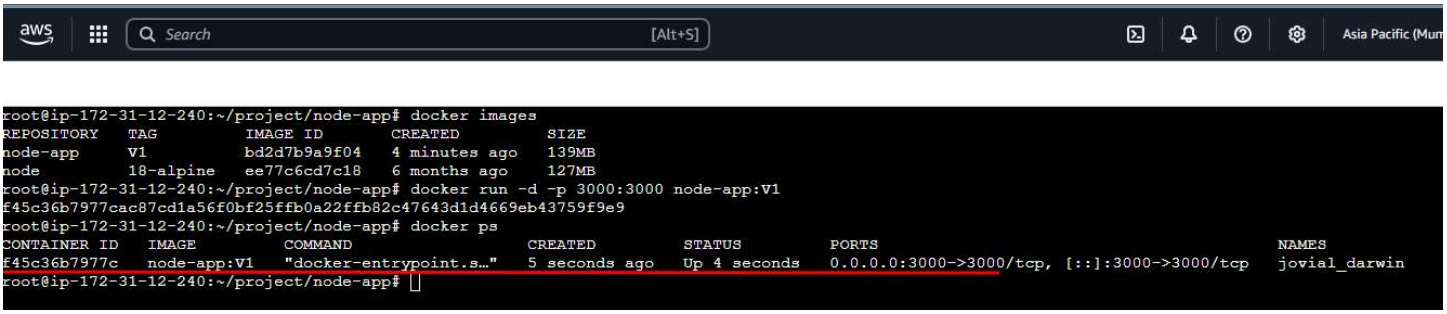


```
root@ip-172-31-12-240:~/project/node-app# ls
Dockerfile  app.js  package.json
root@ip-172-31-12-240:~/project/node-app# docker build -t node-app:V1 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon   5.12kB
Step 1/7 : FROM node:18-alpine
18-alpine: Pulling from library/node
f18232174bc9: Pulling fs layer
```

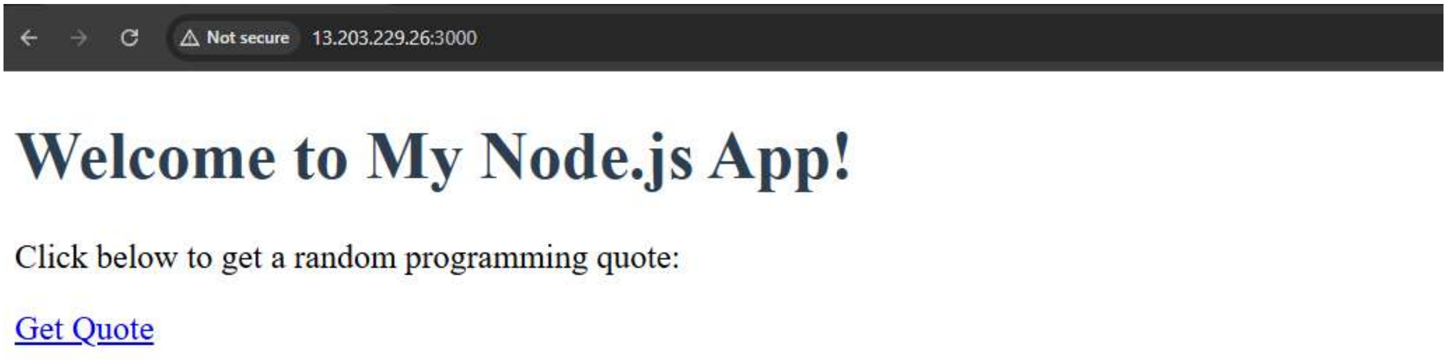


```
 ---> Removed intermediate container 00480e04fc2e
 ---> bd2d7b9a9f04
Successfully built bd2d7b9a9f04
Successfully tagged node-app:V1
root@ip-172-31-12-240:~/project/node-app# docker images
REPOSITORY   TAG         IMAGE ID       CREATED          SIZE
node-app     V1          bd2d7b9a9f04   45 seconds ago   139MB
node         18-alpine   ee77c6cd7c18   6 months ago     127MB
root@ip-172-31-12-240:~/project/node-app# 
```

CMD: docker run -d -p <Port-Number> <Image-Name> : Run Container from Custom build Image.



```
root@ip-172-31-12-240:~/project/node-app# docker images
REPOSITORY   TAG         IMAGE ID       CREATED         SIZE
node-app     V1          bd2d7b9a9f04   4 minutes ago   139MB
node         18-alpine   ee77c6cd7c18   6 months ago    127MB
root@ip-172-31-12-240:~/project/node-app# docker run -d -p 3000:3000 node-app:V1
f45c36b7977cac87cd1a56f0bf25ffb0a22ffb82c47643d1d4669eb43759f9e9
root@ip-172-31-12-240:~/project/node-app# docker ps
CONTAINER ID   IMAGE         COMMAND                  CREATED         STATUS         PORTS                                       NAMES
f45c36b7977c   node-app:V1   "docker-entrypoint.s…"   5 seconds ago   Up 4 seconds   0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp   jovial_darwin
root@ip-172-31-12-240:~/project/node-app# 
```

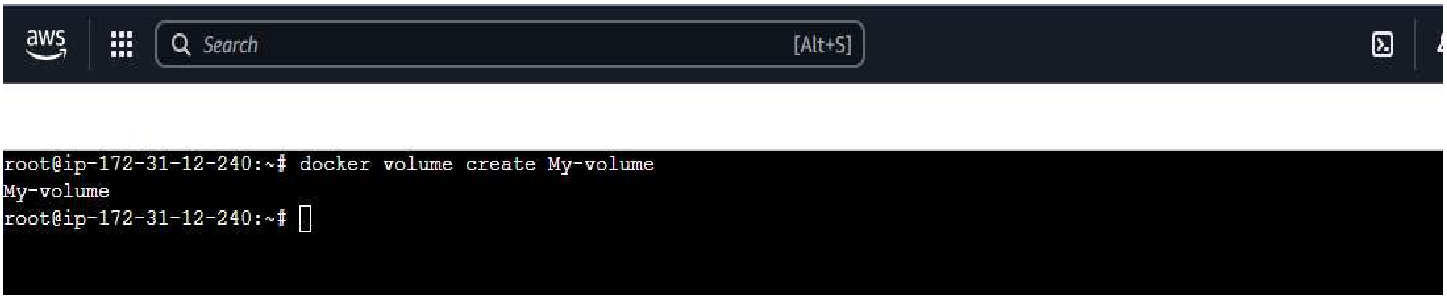


Welcome to My Node.js App!

Click below to get a random programming quote:

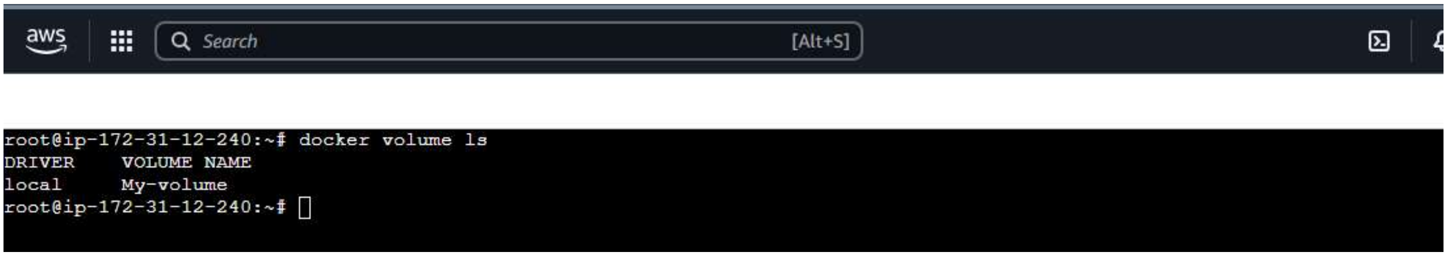
Get Quote

CMD: docker volume create <Volume-Name>: Create a Docker Persistence Volume.



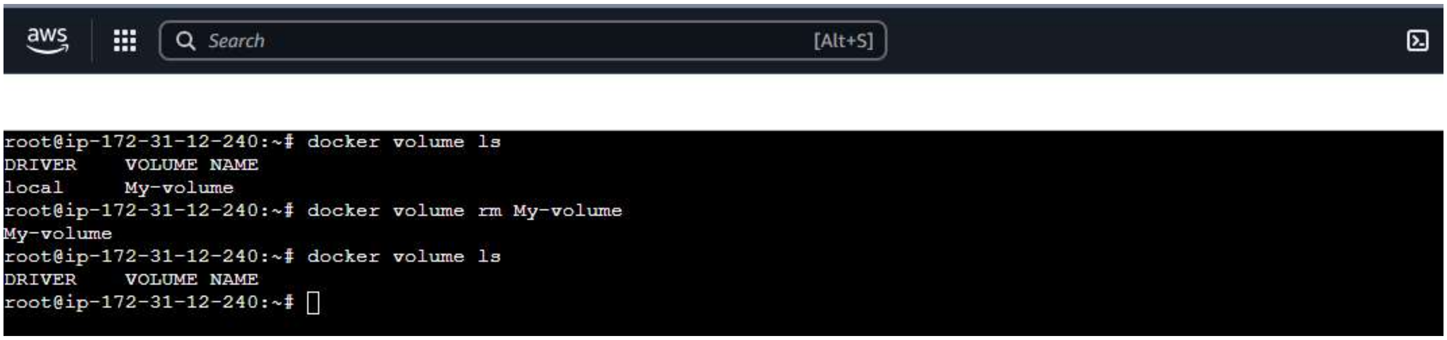
```
root@ip-172-31-12-240:~# docker volume create My-volume
My-volume
root@ip-172-31-12-240:~# 
```

CMD: docker volume ls: List all Docker Volume.



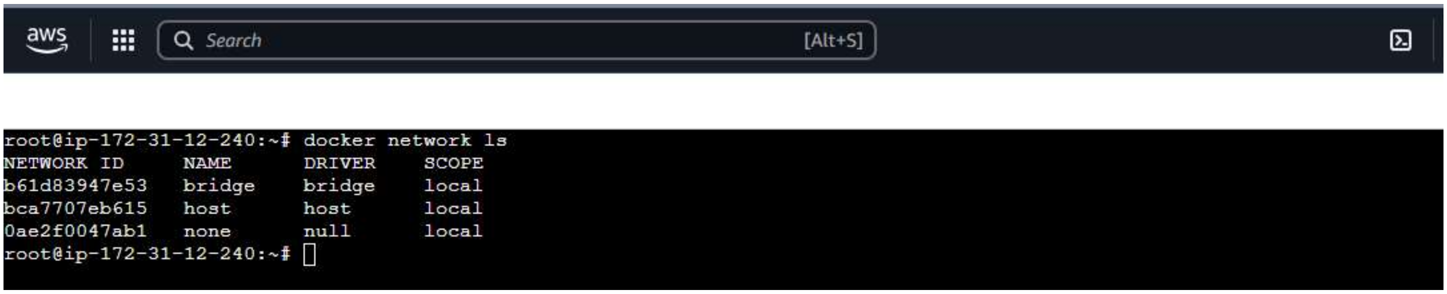
```
root@ip-172-31-12-240:~# docker volume ls
DRIVER    VOLUME NAME
local     My-volume
root@ip-172-31-12-240:~# 
```

CMD: docker volume rm <Volume-Name>: Remove a Docker Volume.



```
root@ip-172-31-12-240:~# docker volume ls
DRIVER    VOLUME NAME
local     My-volume
root@ip-172-31-12-240:~# docker volume rm My-volume
My-volume
root@ip-172-31-12-240:~# docker volume ls
DRIVER    VOLUME NAME
root@ip-172-31-12-240:~# 
```

CMD: docker network ls: List all Docker networks

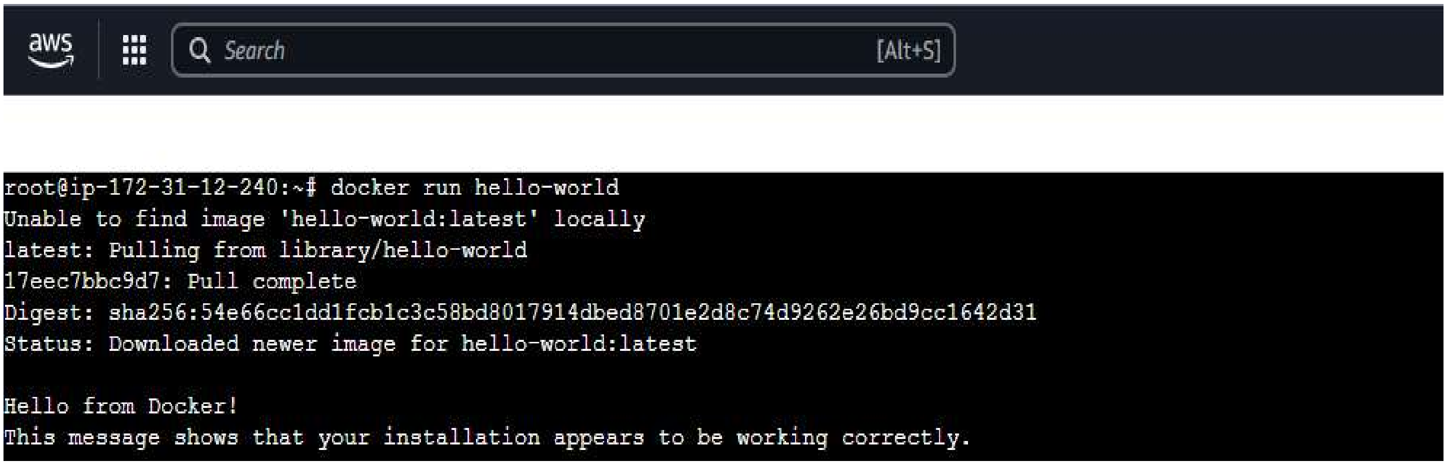


```
root@ip-172-31-12-240:~# docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
b61d83947e53   bridge    bridge    local
bca7707eb615   host      host      local
0ae2f0047ab1   none      null      local
root@ip-172-31-12-240:~# 
```

3. Docker run scenarios we have tried:

1. Running a Simple Container.

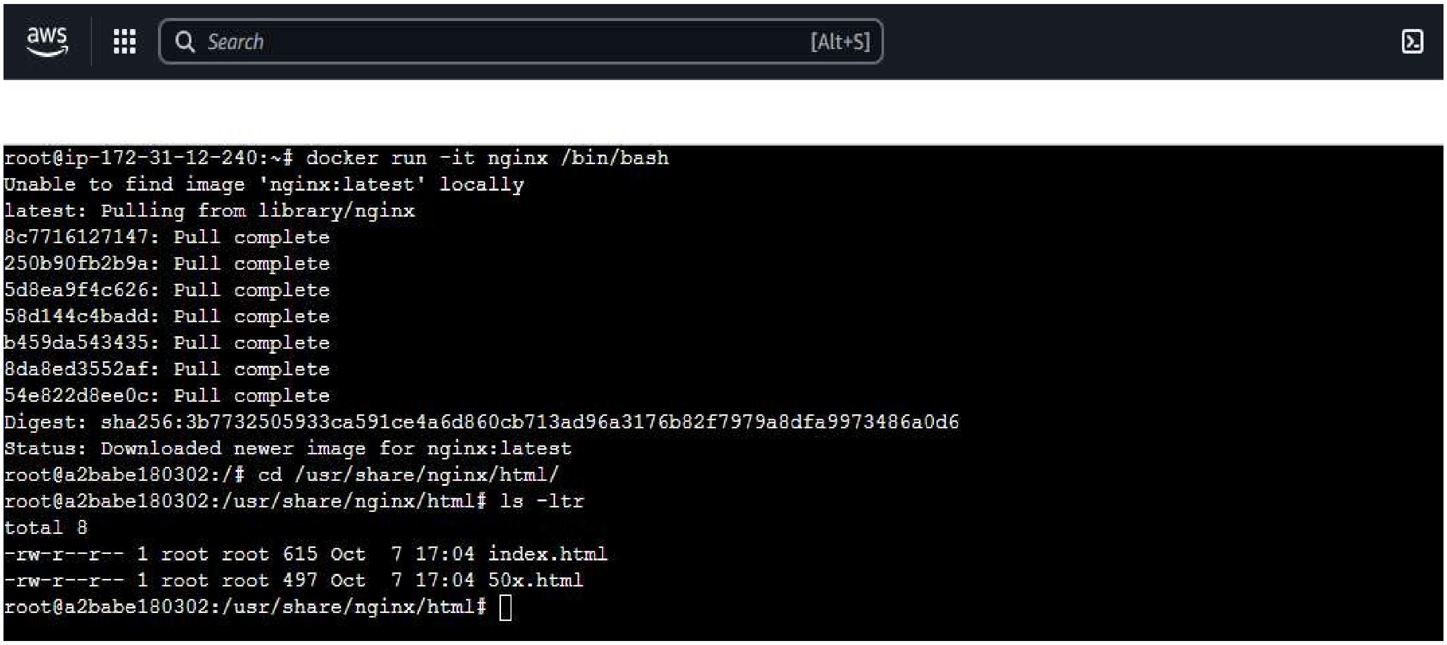
CMD: docker run hello-world -- Docker pulls the hello-world image and runs it. It prints a welcome message



```
root@ip-172-31-12-240:~# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:54e66cc1dd1fcb1c3c58bd8017914dbed8701e2d8c74d9262e26bd9cc1642d31
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

2. Running a Container in Interactive Mode.

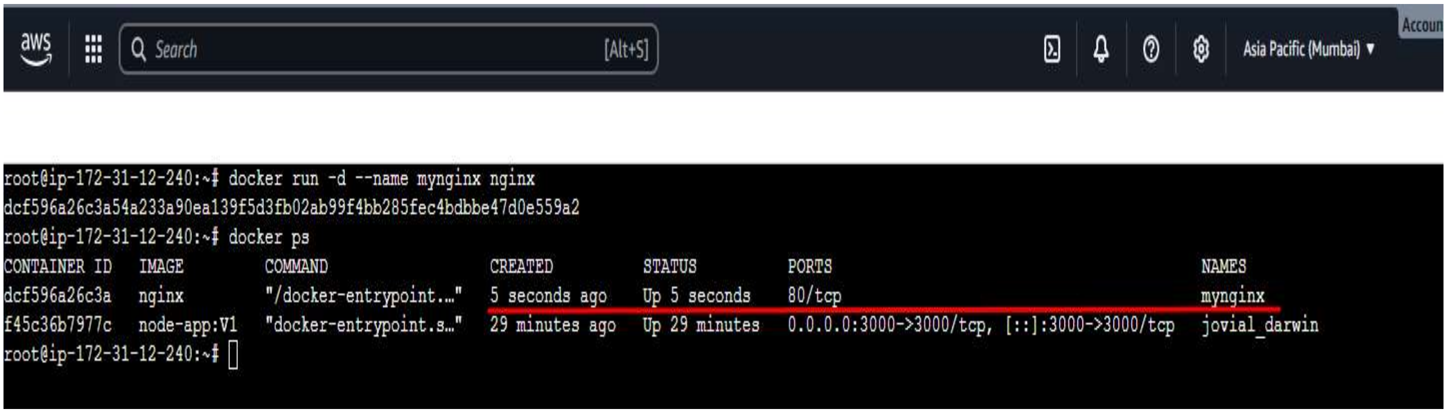
CMD: docker run -it <Image-Name> /bin/bash -- Starts container interactively with a bash shell and Useful for testing commands inside a container.



```
root@ip-172-31-12-240:~# docker run -it nginx /bin/bash
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
8c7716127147: Pull complete
250b90fb2b9a: Pull complete
5d8ea9f4c626: Pull complete
58d144c4badd: Pull complete
b459da543435: Pull complete
8da8ed3552af: Pull complete
54e822d8ee0c: Pull complete
Digest: sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6
Status: Downloaded newer image for nginx:latest
root@a2babe180302:/# cd /usr/share/nginx/html/
root@a2babe180302:/usr/share/nginx/html# ls -ltr
total 8
-rw-r--r-- 1 root root 615 Oct  7 17:04 index.html
-rw-r--r-- 1 root root 497 Oct  7 17:04 50x.html
root@a2babe180302:/usr/share/nginx/html# 
```

3. Running a Container in Detached Mode.

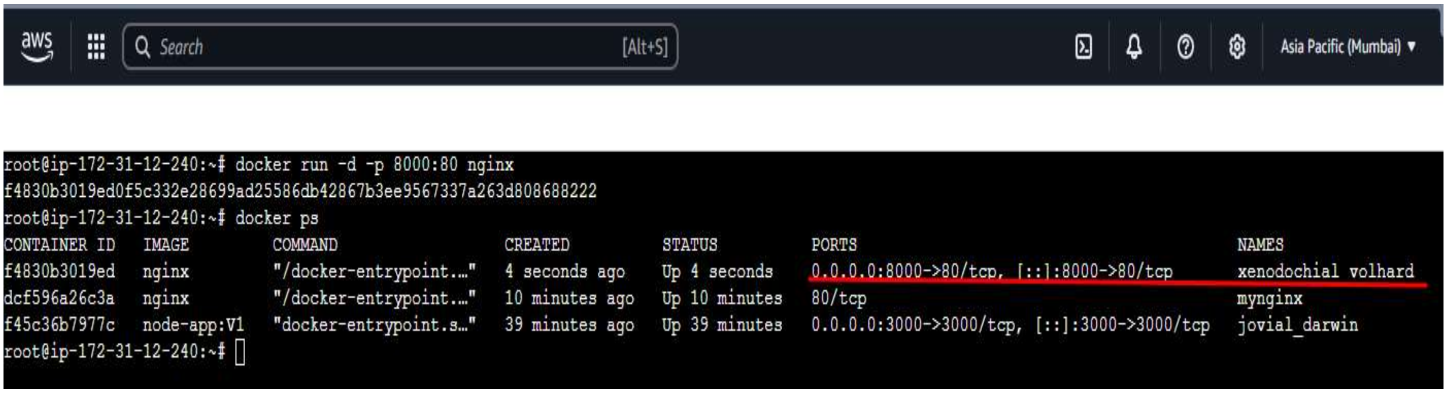
CMD: docker run -d --name <Image-name> <Image> -- Runs Container in the background.



```
root@ip-172-31-12-240:~# docker run -d --name mynginx nginx
dcf596a26c3a54a233a90ea139f5d3fb02ab99f4bb285fec4bdbbe47d0e559a2
root@ip-172-31-12-240:~# docker ps
CONTAINER ID   IMAGE         COMMAND                  CREATED          STATUS          PORTS                                         NAMES
dcf596a26c3a   nginx         "/docker-entrypoint.…"   5 seconds ago    Up 5 seconds    80/tcp                                        mynginx
f45c36b7977c   node-app:V1   "docker-entrypoint.s…"   29 minutes ago   Up 29 minutes   0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp   jovial_darwin
root@ip-172-31-12-240:~# 
```

4. Running a Container with Port Mapping.

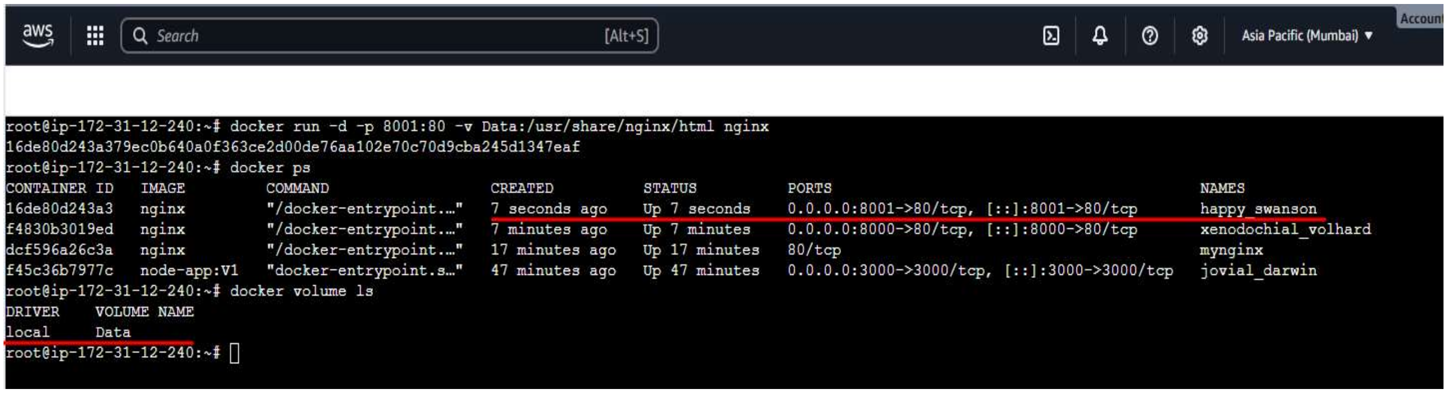
CMD: docker run -d -p <Port-Number> <Image> -- Maps host port 8080 to container port 80. Allows access to Nginx on *http://IP-address:Port-Nubmer*, Common for exposing web apps.



```
root@ip-172-31-12-240:~# docker run -d -p 8000:80 nginx
f4830b3019ed0f5c332e28699ad25586db42867b3ee9567337a263d808688222
root@ip-172-31-12-240:~# docker ps
CONTAINER ID   IMAGE         COMMAND                  CREATED          STATUS          PORTS                                         NAMES
f4830b3019ed   nginx         "/docker-entrypoint.…"   4 seconds ago    Up 4 seconds    0.0.0.0:8000->80/tcp, [::]:8000->80/tcp       xenodochial_volhard
dcf596a26c3a   nginx         "/docker-entrypoint.…"   10 minutes ago   Up 10 minutes   80/tcp                                        mynginx
f45c36b7977c   node-app:V1   "docker-entrypoint.s…"   39 minutes ago   Up 39 minutes   0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp   jovial_darwin
root@ip-172-31-12-240:~# 
```

5. Running a Container with Volume Mount

CMD: docker run -d -v <Persistence volume>:<Container Path> <Image> -- Mounts host directory into container Path. Data persists even if container is removed. Useful for configuration files or logs**.**



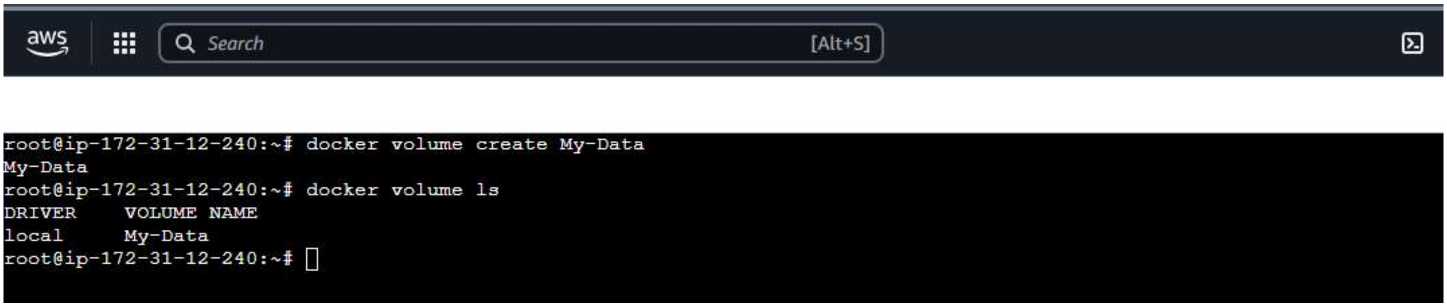
```
root@ip-172-31-12-240:~# docker run -d -p 8001:80 -v Data:/usr/share/nginx/html nginx
16de80d243a379ec0b640a0f363ce2d00de76aa102e70c70d9cba245d1347eaf
root@ip-172-31-12-240:~# docker ps
CONTAINER ID   IMAGE         COMMAND                  CREATED          STATUS          PORTS                                         NAMES
16de80d243a3   nginx         "/docker-entrypoint.…"   7 seconds ago    Up 7 seconds    0.0.0.0:8001->80/tcp, [::]:8001->80/tcp       happy_swanson
f4830b3019ed   nginx         "/docker-entrypoint.…"   7 minutes ago    Up 7 minutes    0.0.0.0:8000->80/tcp, [::]:8000->80/tcp       xenodochial_volhard
dcf596a26c3a   nginx         "/docker-entrypoint.…"   17 minutes ago   Up 17 minutes   80/tcp                                        mynginx
f45c36b7977c   node-app:V1   "docker-entrypoint.s…"   47 minutes ago   Up 47 minutes   0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp   jovial_darwin
root@ip-172-31-12-240:~# docker volume ls
DRIVER    VOLUME NAME
local     Data
root@ip-172-31-12-240:~# 
```

4. Docker volumes:

Scenario 1: Simple Docker Volume (Read-Write)

1. Create a Docker Volume.

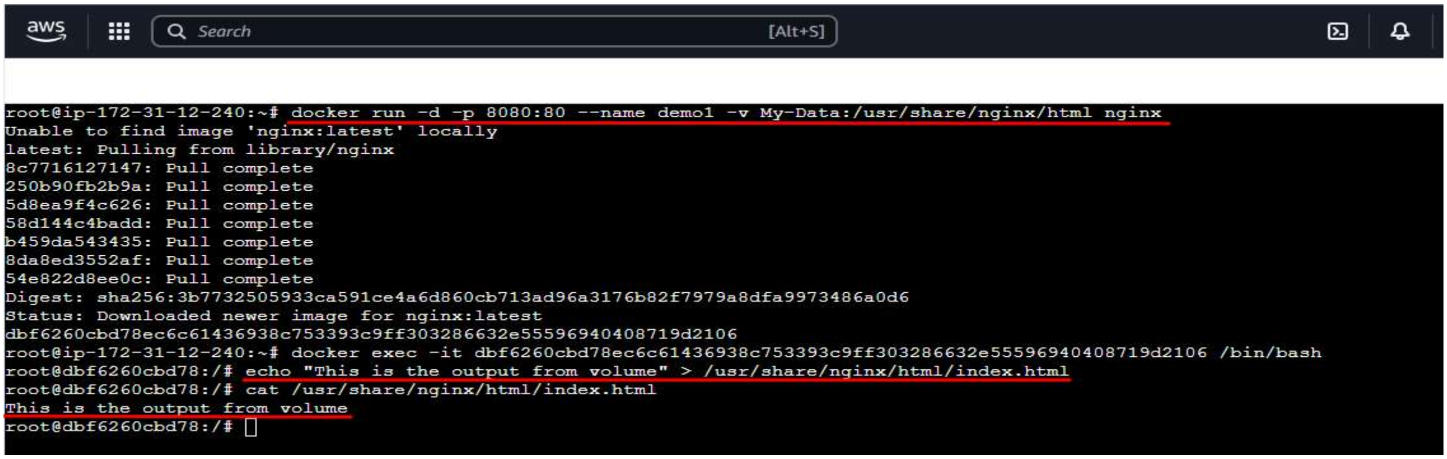
CMD: docker volume create <Volume-Name>



```
root@ip-172-31-12-240:~# docker volume create My-Data
My-Data
root@ip-172-31-12-240:~# docker volume ls
DRIVER    VOLUME NAME
local     My-Data
root@ip-172-31-12-240:~# 
```

2. Run a Container with Volume.

CMD: docker run -d -p <Port-Number> --name <Container-Name> -v <Volume>:<Container-Path> <Image>

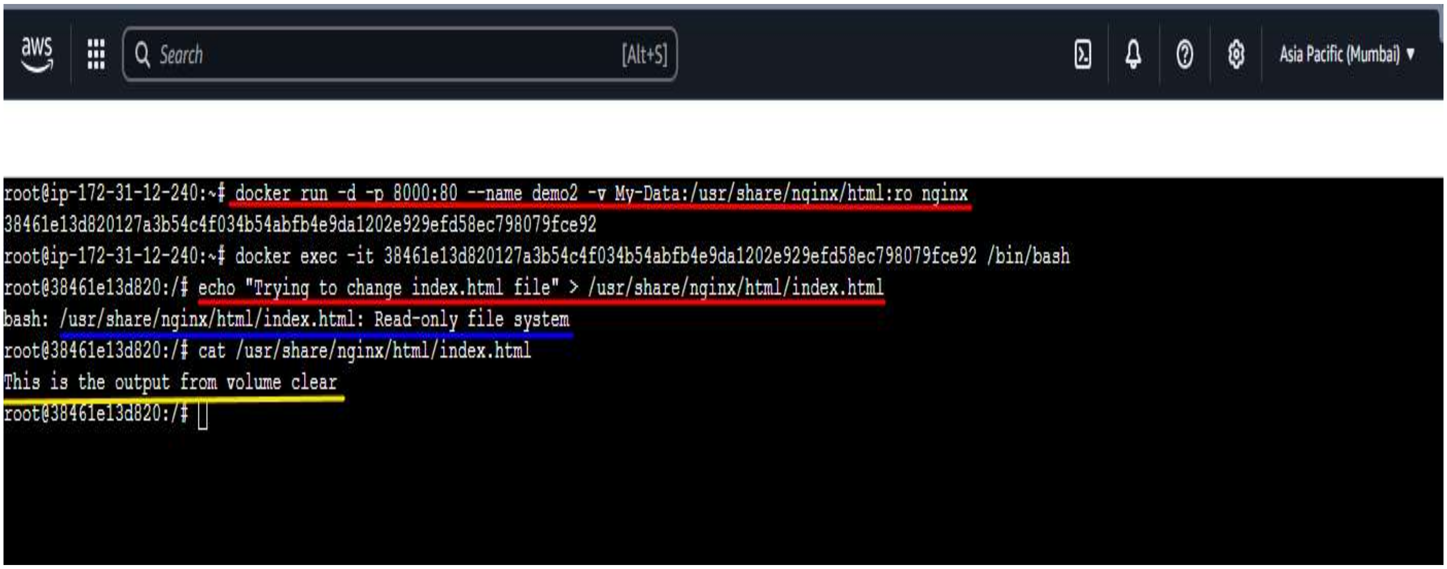


```
root@ip-172-31-12-240:~# docker run -d -p 8080:80 --name demo1 -v My-Data:/usr/share/nginx/html nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
8c7716127147: Pull complete
250b90fb2b9a: Pull complete
5d8ea9f4c626: Pull complete
58d144c4badd: Pull complete
b459da543435: Pull complete
8da8ed3552af: Pull complete
54e822d8ee0c: Pull complete
Digest: sha256:3b7732505933ca591ce4a6d860cb713ad96a3176b82f7979a8dfa9973486a0d6
Status: Downloaded newer image for nginx:latest
dbf6260cbd78ec6c61436938c753393c9ff303286632e55596940408719d2106
root@ip-172-31-12-240:~# docker exec -it dbf6260cbd78ec6c61436938c753393c9ff303286632e55596940408719d2106 /bin/bash
root@dbf6260cbd78:/# echo "This is the output from volume" > /usr/share/nginx/html/index.html
root@dbf6260cbd78:/# cat /usr/share/nginx/html/index.html
This is the output from volume
root@dbf6260cbd78:/# 
```

Scenario 2: Read-Only Volume.

1. Run a Container with Read-Only Volume and Try to Write to Volume.

CMD: docker run -d -p <Port-Number> --name <Container-Name> -v <Volume>:<Container-Path>:ro <Image>



```
root@ip-172-31-12-240:~# docker run -d -p 8000:80 --name demo2 -v My-Data:/usr/share/nginx/html:ro nginx
38461e13d820127a3b54c4f034b54abfb4e9da1202e929efd58ec798079fce92
root@ip-172-31-12-240:~# docker exec -it 38461e13d820127a3b54c4f034b54abfb4e9da1202e929efd58ec798079fce92 /bin/bash
root@38461e13d820:/# echo "Trying to change index.html file" > /usr/share/nginx/html/index.html
bash: /usr/share/nginx/html/index.html: Read-only file system
root@38461e13d820:/# cat /usr/share/nginx/html/index.html
This is the output from volume clear
root@38461e13d820:/# 
```

Conclusion

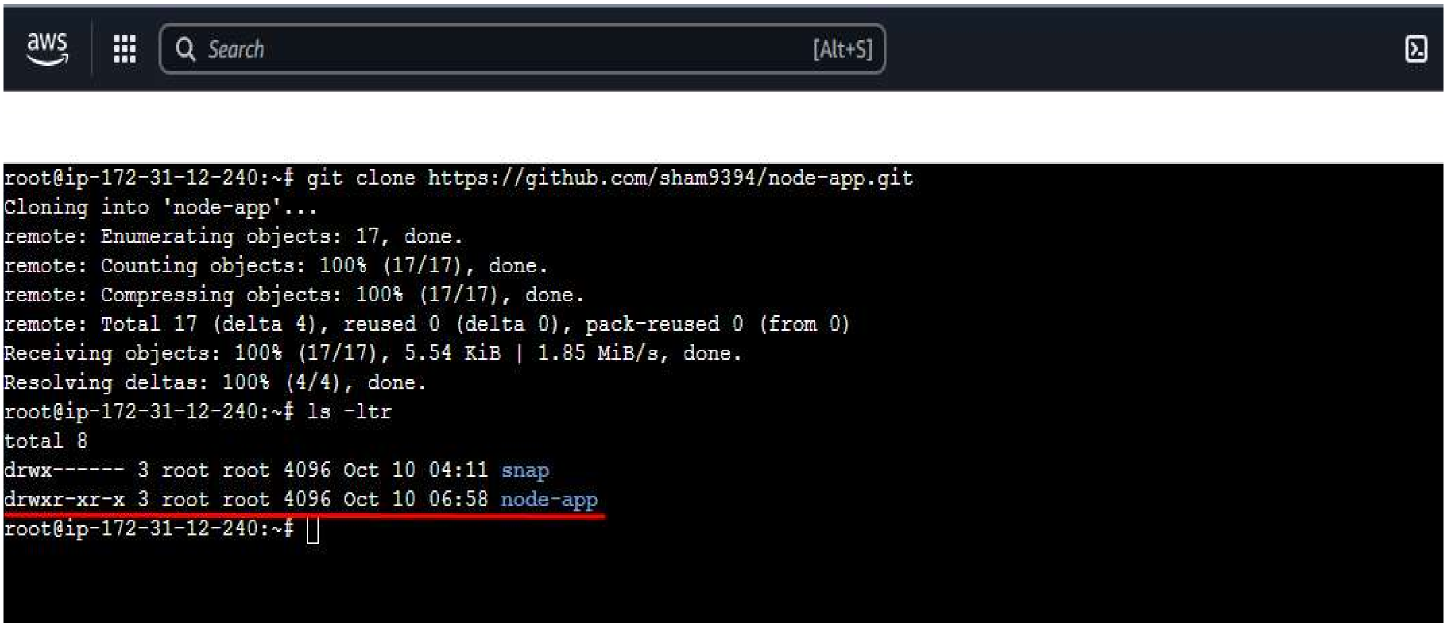
-V <Volume>:<Container-Path> is can provide read and write access.

-V <Volume>:<Container-Path>:ro > is Can only read, cannot modify.

5. Deploying a Node.js application.

1. Pull the code from GitHub.

CMD: Git clone <GitHub Repository link>



```
root@ip-172-31-12-240:~# git clone https://github.com/sham9394/node-app.git
Cloning into 'node-app'...
remote: Enumerating objects: 17, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 17 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (17/17), 5.54 KiB | 1.85 MiB/s, done.
Resolving deltas: 100% (4/4), done.
root@ip-172-31-12-240:~# ls -ltr
total 8
drwx------ 3 root root 4096 Oct 10 04:11 snap
drwxr-xr-x 3 root root 4096 Oct 10 06:58 node-app
root@ip-172-31-12-240:~# 
```

2. Create a Dockerfile.

CMD: vi Dockerfile / nano Dockerfile

FROM node:18-alpine

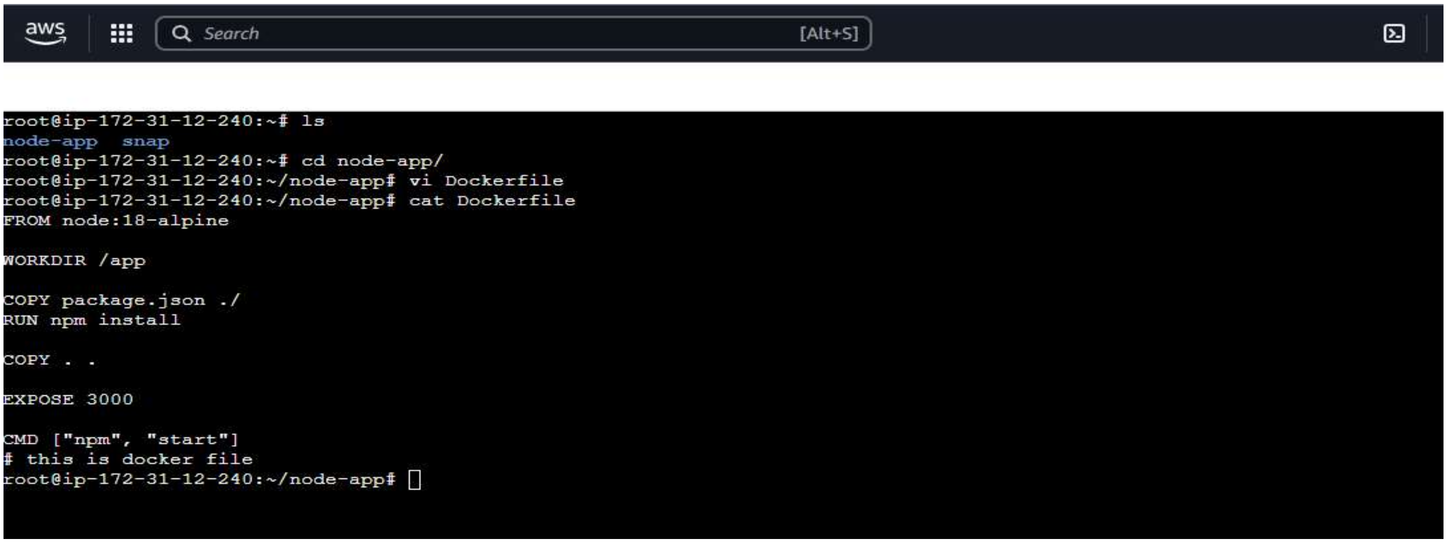
WORKDIR /app

COPY package.json ./
RUN npm install

COPY . .

EXPOSE 3000

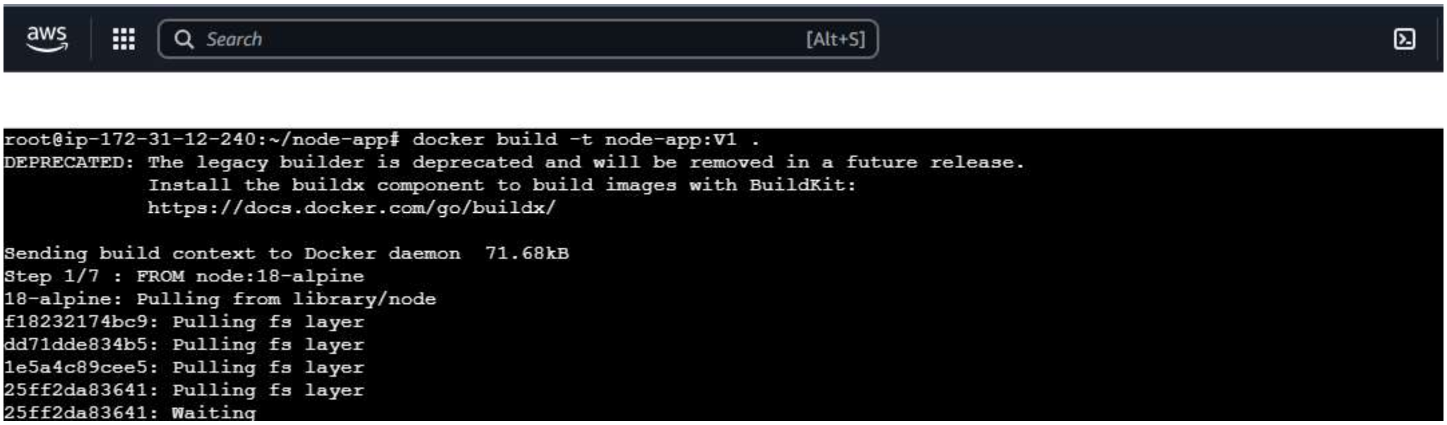
CMD ["npm", "start"]



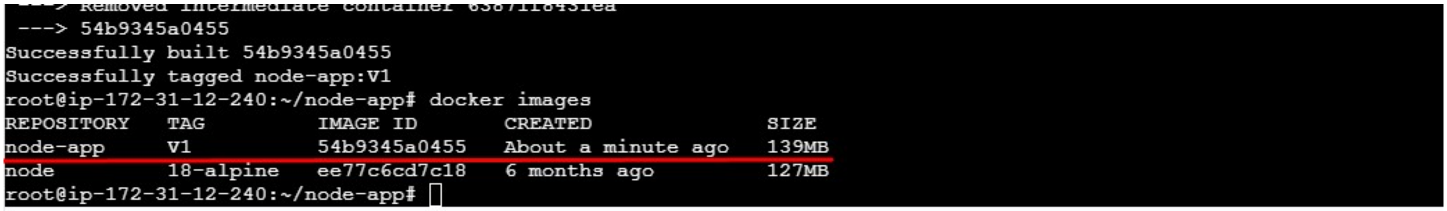
```
root@ip-172-31-12-240:~# ls
node-app  snap
root@ip-172-31-12-240:~# cd node-app/
root@ip-172-31-12-240:~/node-app# vi Dockerfile
root@ip-172-31-12-240:~/node-app# cat Dockerfile
FROM node:18-alpine

WORKDIR /app

COPY package.json ./
RUN npm install

COPY . .

EXPOSE 3000

CMD ["npm", "start"]
# this is docker file
root@ip-172-31-12-240:~/node-app# 
```

3. Build the Docker Image.

CMD: docker build -t <Image-Name:Tags> .

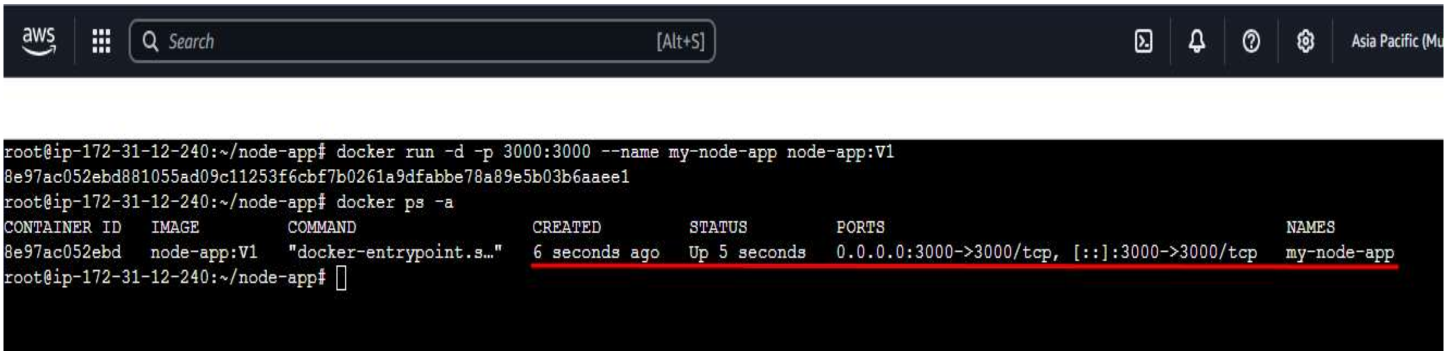


```
root@ip-172-31-12-240:~/node-app# docker build -t node-app:V1 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  71.68kB
Step 1/7 : FROM node:18-alpine
18-alpine: Pulling from library/node
f18232174bc9: Pulling fs layer
dd71dde834b5: Pulling fs layer
1e5a4c89cee5: Pulling fs layer
25ff2da83641: Pulling fs layer
25ff2da83641: Waiting
```



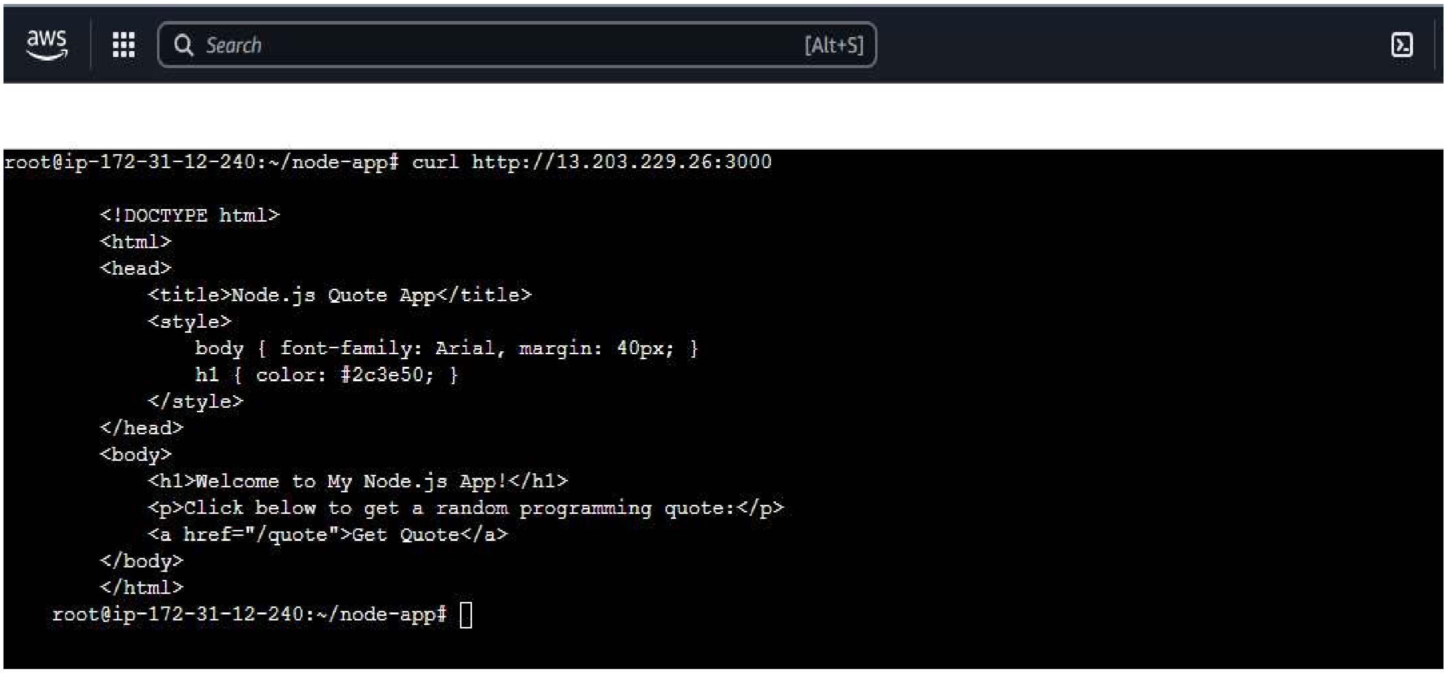
```
 ---> Removed intermediate container 6387118431ea
 ---> 54b9345a0455
Successfully built 54b9345a0455
Successfully tagged node-app:V1
root@ip-172-31-12-240:~/node-app# docker images
REPOSITORY   TAG         IMAGE ID       CREATED              SIZE
node-app     V1          54b9345a0455   About a minute ago   139MB
node         18-alpine   ee77c6cd7c18   6 months ago         127MB
root@ip-172-31-12-240:~/node-app# 
```

4. **Run the Container using Build Image.**
   CMD: docker run -d -p <Port-Number> --name <Container-Name> <Image-name:Tags>



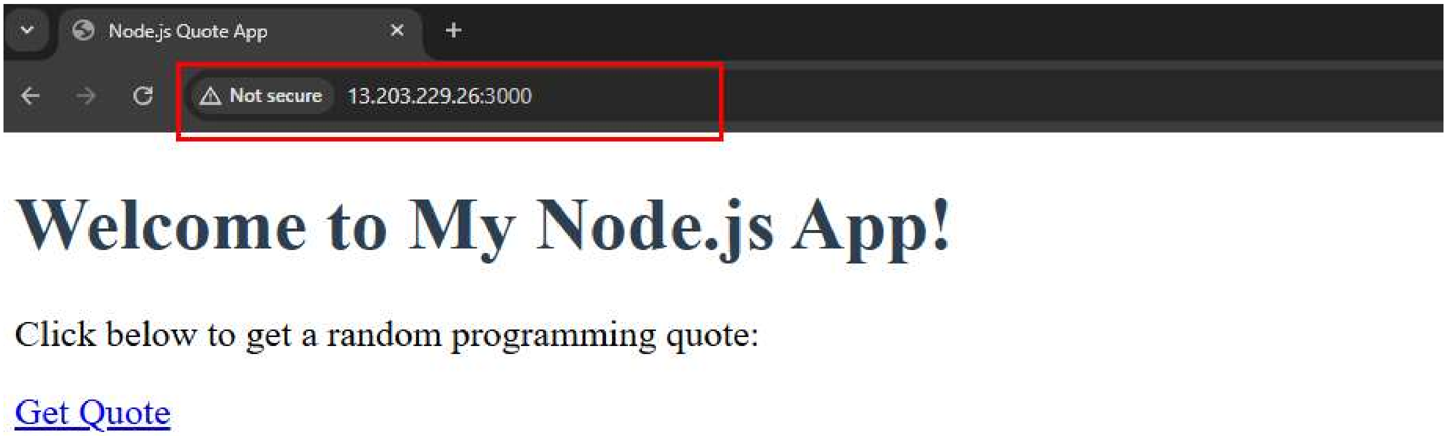
```
root@ip-172-31-12-240:~/node-app# docker run -d -p 3000:3000 --name my-node-app node-app:V1
8e97ac052ebd881055ad09c11253f6cbf7b0261a9dfabbe78a89e5b03b6aaee1
root@ip-172-31-12-240:~/node-app# docker ps -a
CONTAINER ID   IMAGE         COMMAND                  CREATED         STATUS         PORTS                                         NAMES
8e97ac052ebd   node-app:V1   "docker-entrypoint.s…"   6 seconds ago   Up 5 seconds   0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp   my-node-app
root@ip-172-31-12-240:~/node-app# 
```

5. **Test the Application.**
   CMD: curl http://<Instance-IP>:<Port-Number>
   OR use Brower: http://<Instance-IP>:<Port-Number>



```
root@ip-172-31-12-240:~/node-app# curl http://13.203.229.26:3000

        <!DOCTYPE html>
        <html>
        <head>
            <title>Node.js Quote App</title>
            <style>
                body { font-family: Arial, margin: 40px; }
                h1 { color: #2c3e50; }
            </style>
        </head>
        <body>
            <h1>Welcome to My Node.js App!</h1>
            <p>Click below to get a random programming quote:</p>
            <a href="/quote">Get Quote</a>
        </body>
        </html>
    root@ip-172-31-12-240:~/node-app# 
```

OR



Welcome to My Node.js App!

Click below to get a random programming quote:

Get Quote