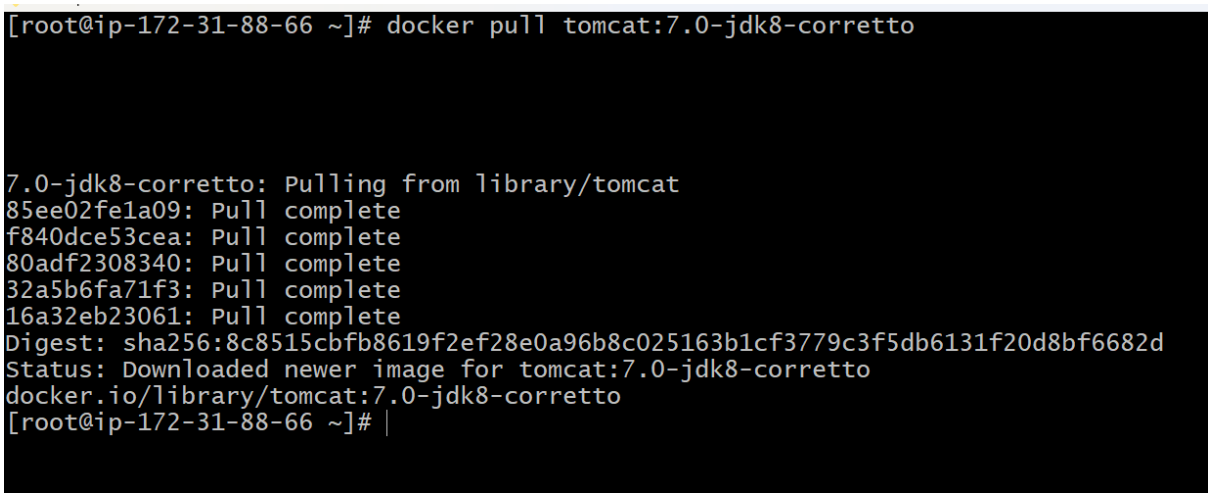
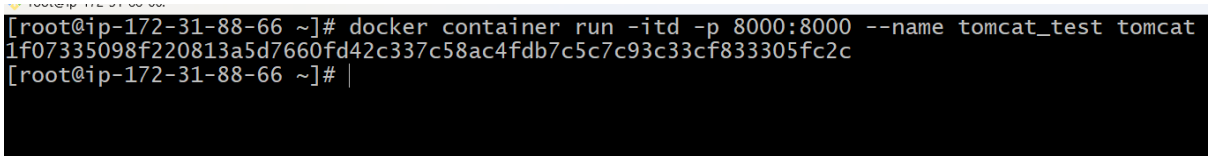
1. **Create a tomcat container on 8080 and deploy sample application in tomcat.**

- docker pull tomcat:7.0-jdk8-corretto

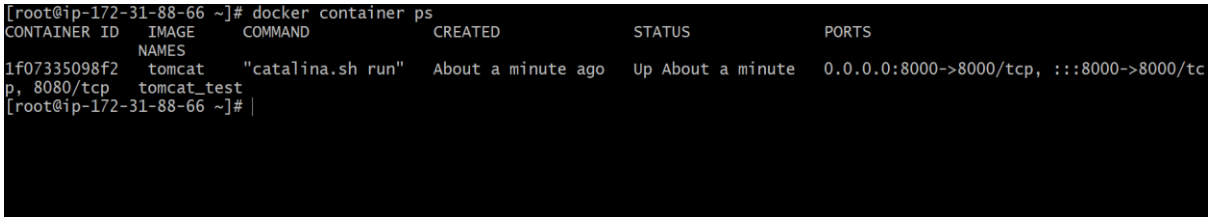


- docker container run -itd -p 8000:8000 --name tomcat_test tomcat

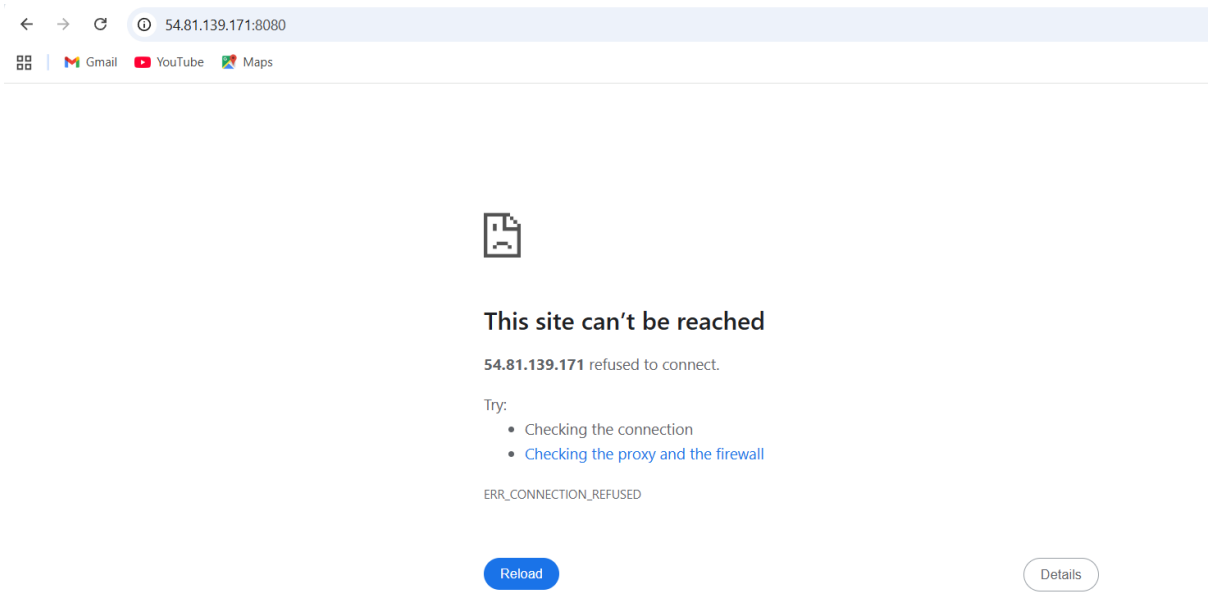
A container has been created for tomcat



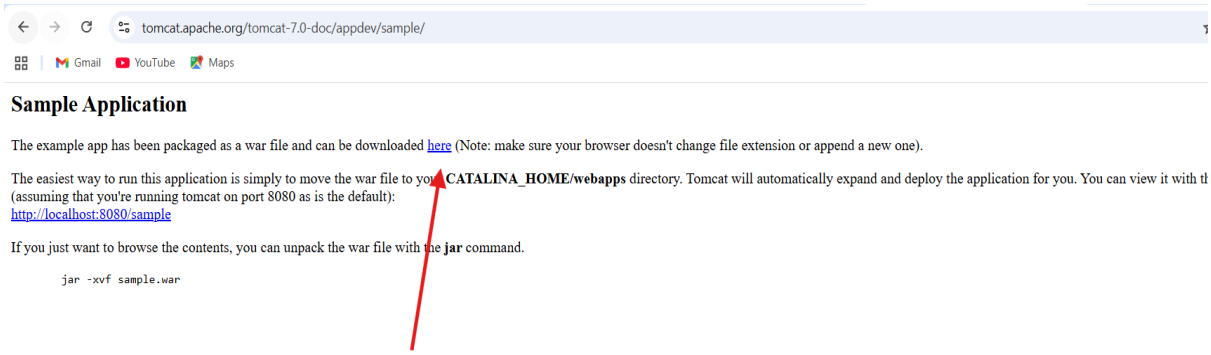
- docker container ps



Copy public ip and port number 8080 if we find error.



Login to your container and deploy a sample.war file



- download sample war in root location

```
[root@ip-172-31-75-180 ~]# wget https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/
--2025-11-24 14:18:41--  https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.
Resolving tomcat.apache.org (tomcat.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to tomcat.apache.org (tomcat.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4606 (4.5K)
Saving to: 'sample.war'

sample.war                          100%[=================================================

2025-11-24 14:18:41 (60.9 MB/s) - 'sample.war' saved [4606/4606]
```

Go into your container by using

- docker exec -it container_id /bin/bash

```
[root@ip-172-31-75-180 ~]# docker exec -it 89b0eff2eb90 /bin/bash
root@89b0eff2eb90:/usr/local/tomcat# ls
bin             CONTRIBUTING.md  LICENSE         NOTICE          RUNNING.txt     webapps
BUILDING.txt    filtered-KEYS    logs            README.md       temp            webapps.dist
conf            lib              native-jni-lib  RELEASE-NOTES   upstream-KEYS   work
root@89b0eff2eb90:/usr/local/tomcat# cd webapps
root@89b0eff2eb90:/usr/local/tomcat/webapps# ll
total 0
drwxr-xr-x. 2 root root  6 Nov 14 00:41 ./
drwxr-xr-x. 1 root root 30 Nov 14 00:41 ../
```

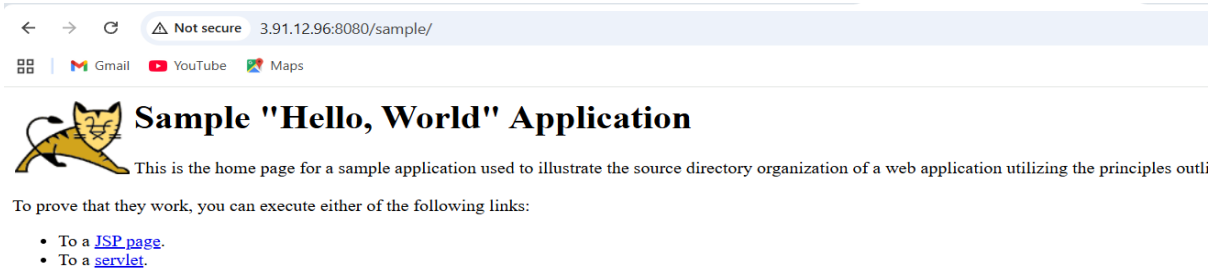
Copy the sample war and move to your container.

- docker cp sample.war container_name:/usr/local/tomcat/webapps/

```
[root@ip-172-31-75-180 ~]# docker cp sample.war tomcat_test:/usr/local/tomcat/webapps/
Successfully copied 6.14kB to tomcat_test:/usr/local/tomcat/webapps/
[root@ip-172-31-75-180 ~]# docker exec -it 89b0eff2eb90 /bin/bash
root@89b0eff2eb90:/usr/local/tomcat# cd /webapps
bash: cd: /webapps: No such file or directory
root@89b0eff2eb90:/usr/local/tomcat# ll
total 220
drwxr-xr-x. 1 root root    57 Nov 14 00:41 ./
drwxr-xr-x. 1 root root    20 Nov 14 00:40 ../
drwxr-xr-x. 2 root root 16384 Nov 14 00:41 bin/
-rw-r--r--. 1 root root 24262 Nov  5 19:03 BUILDING.txt
```

```
drwxr-xr-x. 7 root root    81 Nov  5 19:03 webapps.dist/
drwxrwxrwt. 1 root root    22 Nov 24 14:21 work/
root@89b0eff2eb90:/usr/local/tomcat# cd webapps
root@89b0eff2eb90:/usr/local/tomcat/webapps# ls
sample  sample.war
root@89b0eff2eb90:/usr/local/tomcat/webapps#
```

Go to browser search by port number 8080 and /sample



2. Create volume and deploy tomcat container on port 8081.

* docker volume create deployment

```
[root@ip-172-31-75-180 ~]# docker volume create deployment
deployment
[root@ip-172-31-75-180 ~]# docker volume ls
DRIVER    VOLUME NAME
local     deployment
[root@ip-172-31-75-180 ~]#
```

Go to volumes

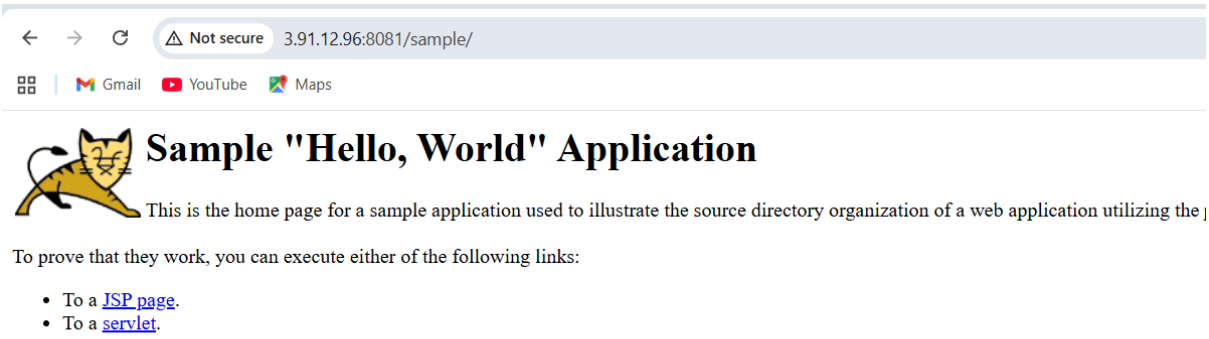
- cd /var/lib/docker/volumes/deployment/_data
- wget samplewar

```
[root@ip-172-31-75-180 ~]# cd /var/lib/docker/volumes
[root@ip-172-31-75-180 volumes]# ls
backingFsBlockDev  deployment  metadata.db
[root@ip-172-31-75-180 volumes]# cd deployment/
[root@ip-172-31-75-180 deployment]# ls
_data
[root@ip-172-31-75-180 deployment]# cd _data/
[root@ip-172-31-75-180 _data]# wget https://tomcat.apache.org/tomcat-7.0-doc
--2025-11-24 14:38:25--  https://tomcat.apache.org/tomcat-7.0-doc/appdev/sam
Resolving tomcat.apache.org (tomcat.apache.org)... 151.101.2.132, 2a04:4e42:
Connecting to tomcat.apache.org (tomcat.apache.org)|151.101.2.132|:443... co
HTTP request sent, awaiting response... 200 OK
Length: 4606 (4.5K)
Saving to: ‘sample.war’

sample.war                    100%[=====================================

2025-11-24 14:38:25 (62.4 MB/s) - ‘sample.war’ saved [4606/4606]

[root@ip-172-31-75-180 _data]#
```

To map your container to volume.

- docker run -itd -p 8081:8080 -v /var/lib/docker/volumes/deployment/_data:/usr/local/tomcat/webapps tomcat

```
[root@ip-172-31-75-180 ~]# docker run -itd -p 8081:8080 -v /var/lib/docker/volumes/deployment/_data:/usr/local/tomcat/webapps tomcat
66e01efd28a18c72dae1c24c6d307612e978cef552ea0fec822b81a365cbfa95
[root@ip-172-31-75-180 ~]#
```



3. Limit the nginx container to 500 MB.

* docker run -d --name nginx -p 80:80 --memory="500m" nginx:latest

```
[root@ip-172-31-75-180 ~]# docker run -d --name nginx -p 80:80 --memory="500m" nginx:latest
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
0e4bc2bd6656: Pull complete
b5feb73171bf: Pull complete
108ab8292820: Pull complete
53d743880af4: Pull complete
77fa2eb06317: Pull complete
192e2451f875: Pull complete
de57a609c9d5: Pull complete
Digest: sha256:553f64aecdc31b5bf944521731cd70e35da4faed96b2b7548a3d8e2598c52a42
Status: Downloaded newer image for nginx:latest
d2358797bd3c2d7c4e683310359094f86c8d844b5d41b4f5bc1e2d7d76cfb296
[root@ip-172-31-75-180 ~]#
```

- docker stats

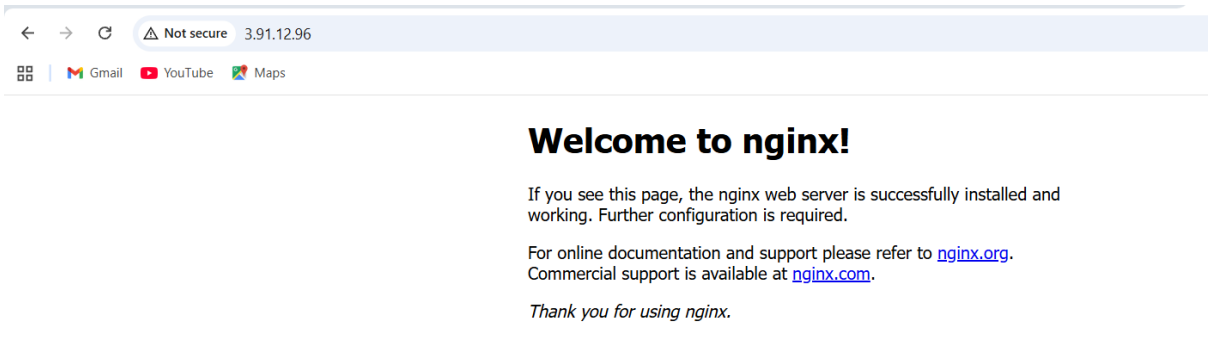
```
CONTAINER ID   NAME              CPU %     MEM USAGE / LIMIT     MEM %     NET I/O         BLOCK I/O         PIDS
d2358797bd3c   nginx             0.00%     13.32MiB / 500MiB     2.66%     726B / 0B       10.8MB / 12.3kB   3
66e01efd28a1   trusting_merkle   0.04%     74.7MiB / 904.8MiB    8.26%     21.2kB / 40.3kB 483kB / 815kB     29
89b0eff2eb90   tomcat_test       0.10%     80.18MiB / 904.8MiB   8.86%     28.5kB / 51.2kB 13.5MB / 1.07MB   29
```

- docker inspect nginx | grep -i memory

```
[root@ip-172-31-75-180 ~]# docker inspect nginx | grep -i memory
            "Memory": 524288000,
            "MemoryReservation": 0,
            "MemorySwap": 1048576000,
            "MemorySwappiness": null,
[root@ip-172-31-75-180 ~]#
```

```
            "MemorySwappiness": null,
[root@ip-172-31-75-180 ~]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS
     NAMES
d2358797bd3c   nginx:latest   "/docker-entrypoint.…"   3 minutes ago    Up 3 minutes    0.0.0.0:80->80/tcp, :::80->80/tcp
     nginx
66e01efd28a1   tomcat         "catalina.sh run"        55 minutes ago   Up 55 minutes   0.0.0.0:8081->8080/tcp, :::8081->808
tcp   trusting_merkle
89b0eff2eb90   tomcat         "catalina.sh run"        2 hours ago      Up 2 hours      0.0.0.0:8080->8080/tcp, :::8080->808
tcp   tomcat_test
[root@ip-172-31-75-180 ~]#
```

Search in browser with public_ip and portnumber 80



4. Create a sample docker file using below instructions.

a. Base module as amazonlinux:latest

b. Maintainer you name

c. Install nginx

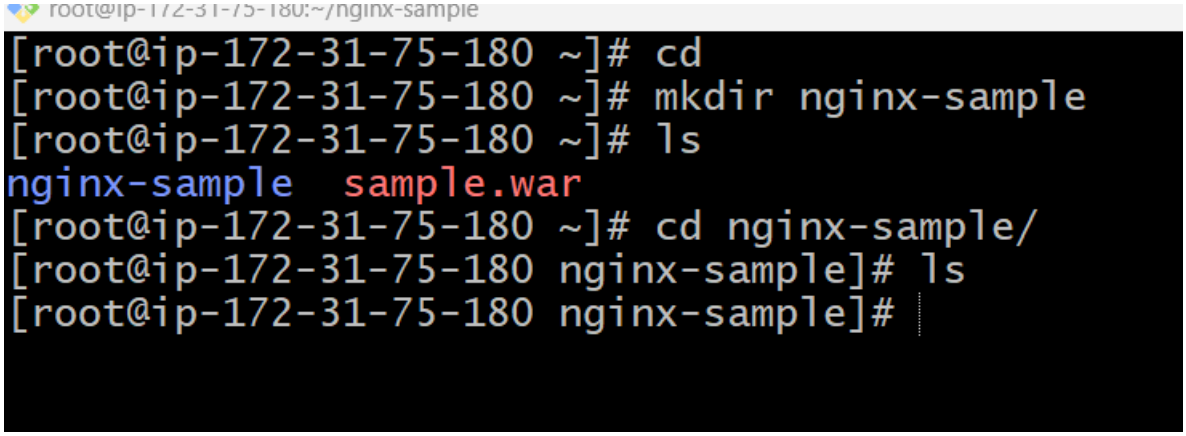
d. COPY one index.html file to image

e. Expose on port 80

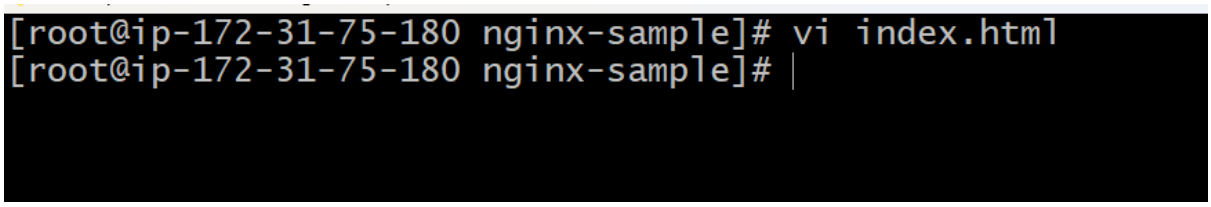
f. Command to start the nginx container

select root location

- cd
- mkdir nginx-sample
- cd nginx-sample



In the nginx-sample directory create a file vi index.html



```
<!doctype html>
<html>
<head>
<title>My Nginx Sample</title>
</head>
<body>
<h1>Hello from Nginx on Amazon Linux!</h1>
<p>This page is served inside a Docker container.</p>
</body>
</html>
~
~
~
~
```

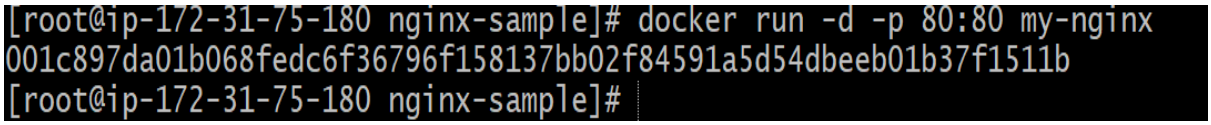
In the same location create another file named as vi dockerfile.

```
[root@ip-172-31-75-180 nginx-sample]# vi dockerfile
```

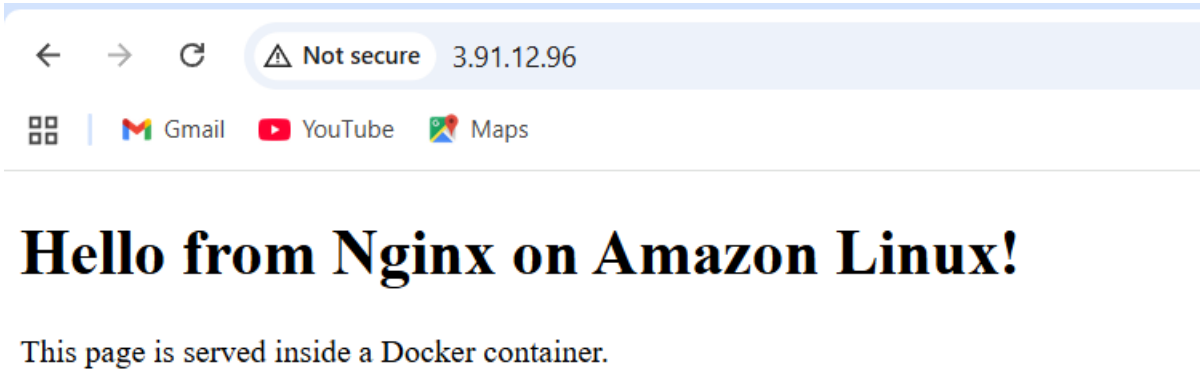
```
# a. Base image
FROM amazonlinux:latest

# b. Maintainer
LABEL maintainer="Shaik Mohammad Mujaheed"

# c. Install nginx
RUN yum update -y && \
    yum install -y nginx && \
    yum clean all

# d. Copy index.html into nginx default directory
COPY index.html /usr/share/nginx/html/index.html

# e. Expose port 80
EXPOSE 80

# f. Command to start nginx in the foreground
CMD ["nginx", "-g", "daemon off;"]
~
~
~
~
~
~
~
```

- docker build -t my-nginx .

- docker run -d -p 80:80 my-nginx

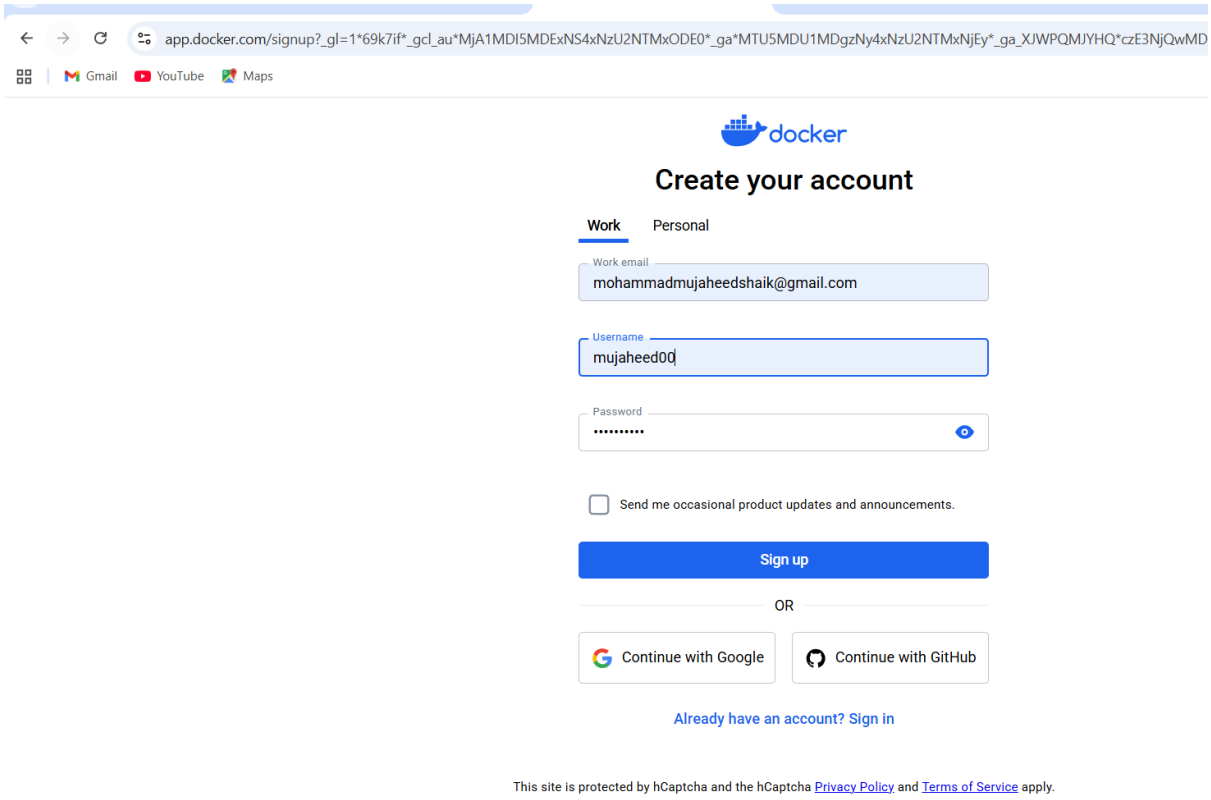


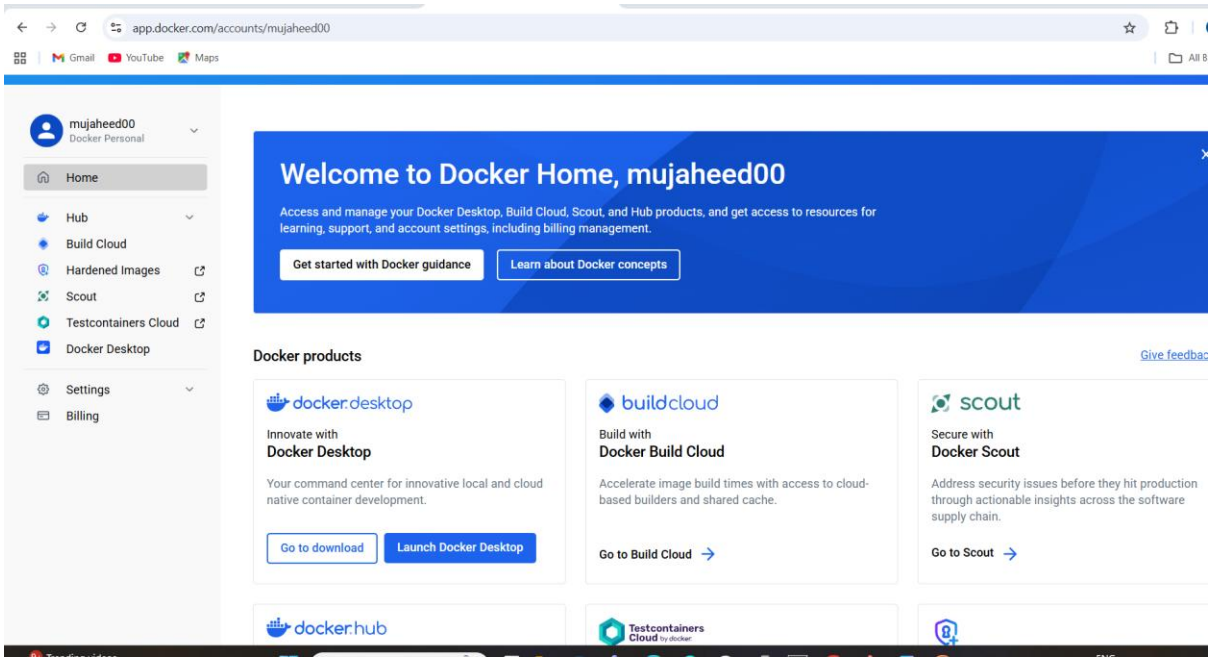
Search on browser with theport number 80.



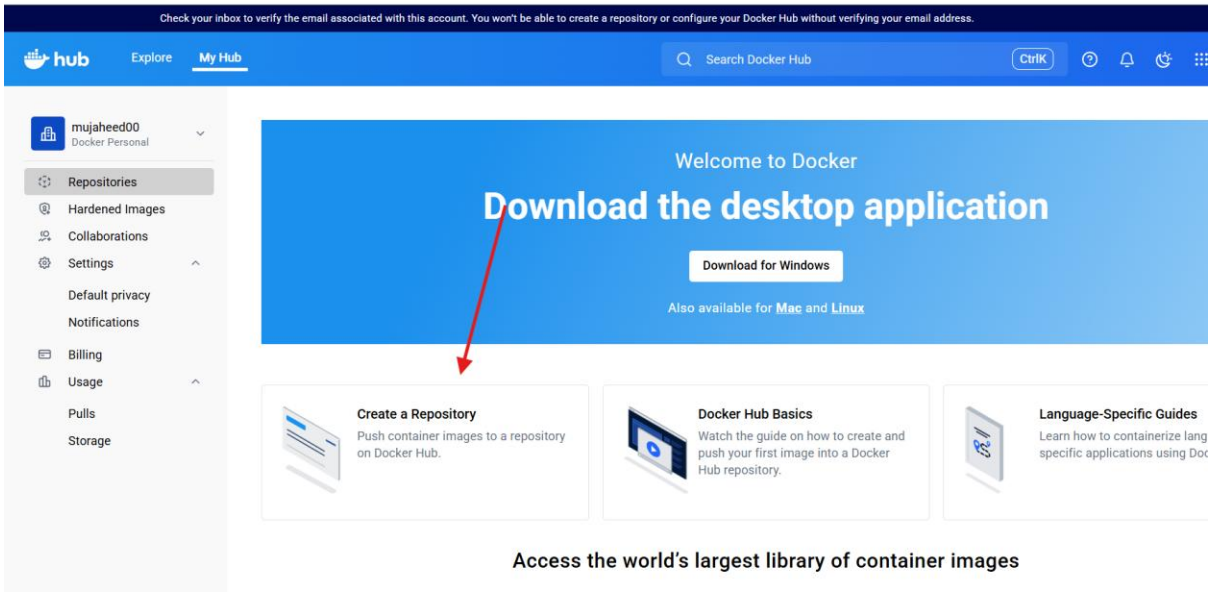
5 . Push image to Docker hub

Signup to dockerhub.

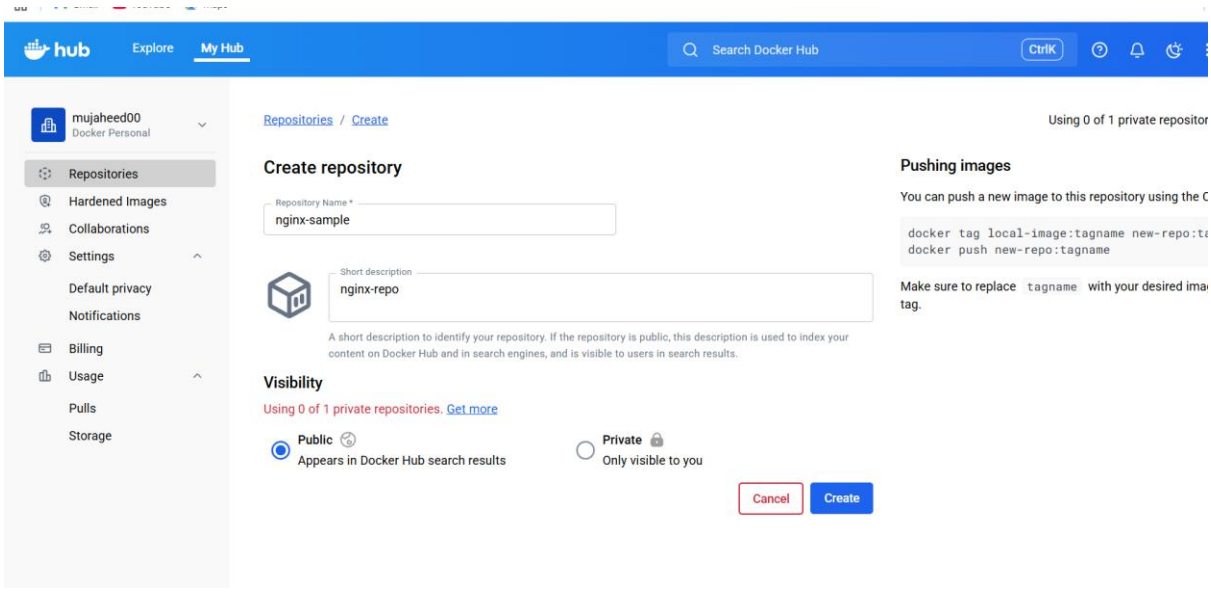




Go to hub click on repository,create repository.



Give name as nginx-sample click on create.



- docker login

give your username and password.

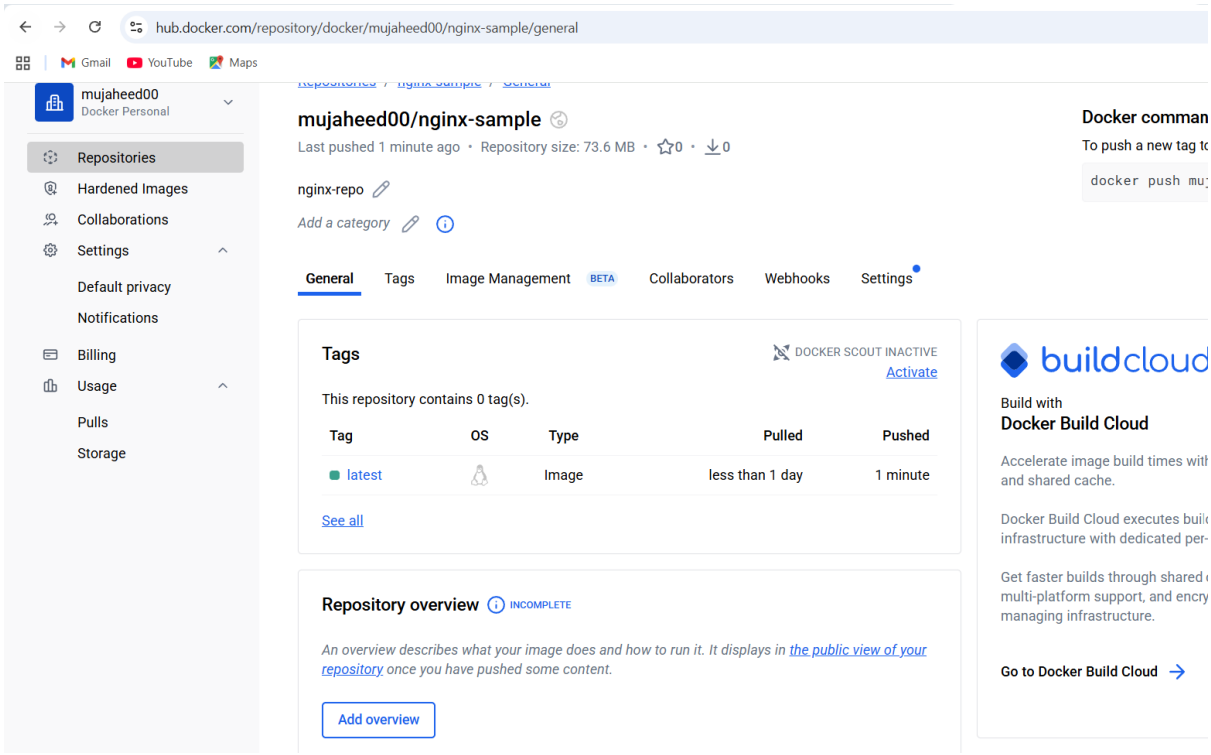
```
root@ip-172-31-75-180:~
[root@ip-172-31-75-180 ~]# docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have
to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants be
quired for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username: mujaheed00
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[root@ip-172-31-75-180 ~]#
```

- docker tag my-nginx mujaheed00/nginx-sample:latest

```
[root@ip-172-31-75-180 ~]# docker tag my-nginx mujaheed00/nginx-sample:latest
[root@ip-172-31-75-180 ~]#
```

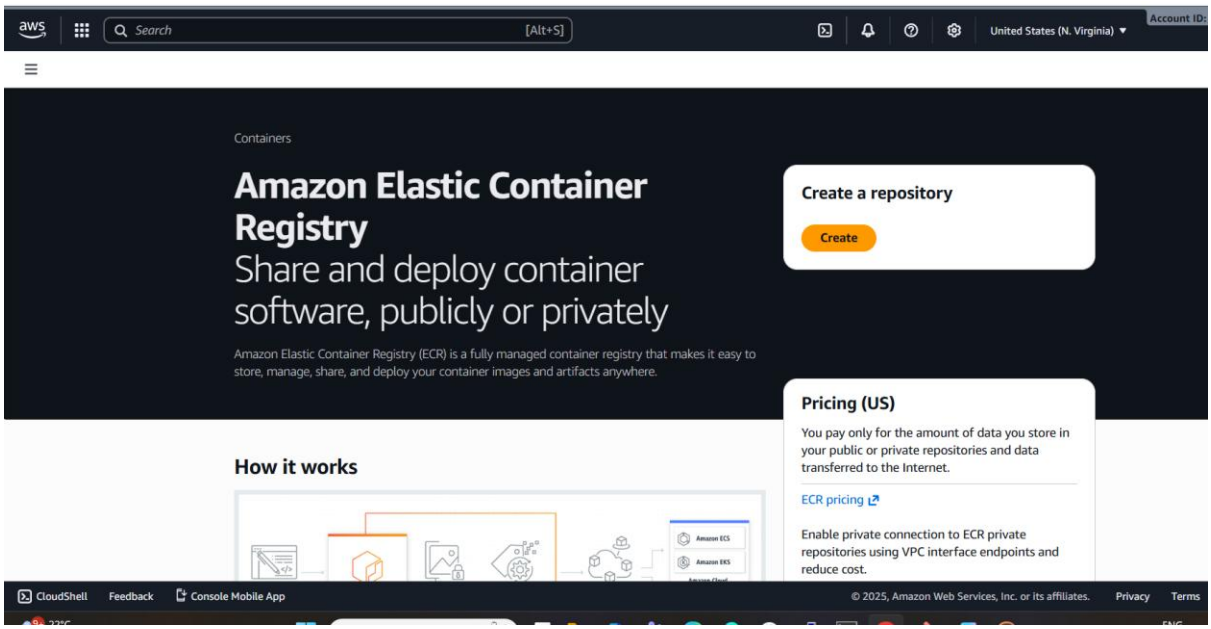
- docker push mujaheed00/nginx-sample:latest

```
[root@ip-172-31-75-180 ~]# docker tag my-nginx mujaheed00/nginx-sample:latest
[root@ip-172-31-75-180 ~]# docker push mujaheed00/nginx-sample:latest
The push refers to repository [docker.io/mujaheed00/nginx-sample]
e3dda3eead3d: Pushed
f0224e513680: Pushed
0e7c6314d998: Mounted from library/amazonlinux
latest: digest: sha256:f6377b28d28e64b83d0a5af4066dd829b085c6eecb60e7a2d53192ca3b7082d1 size: 948
[root@ip-172-31-75-180 ~]#
```

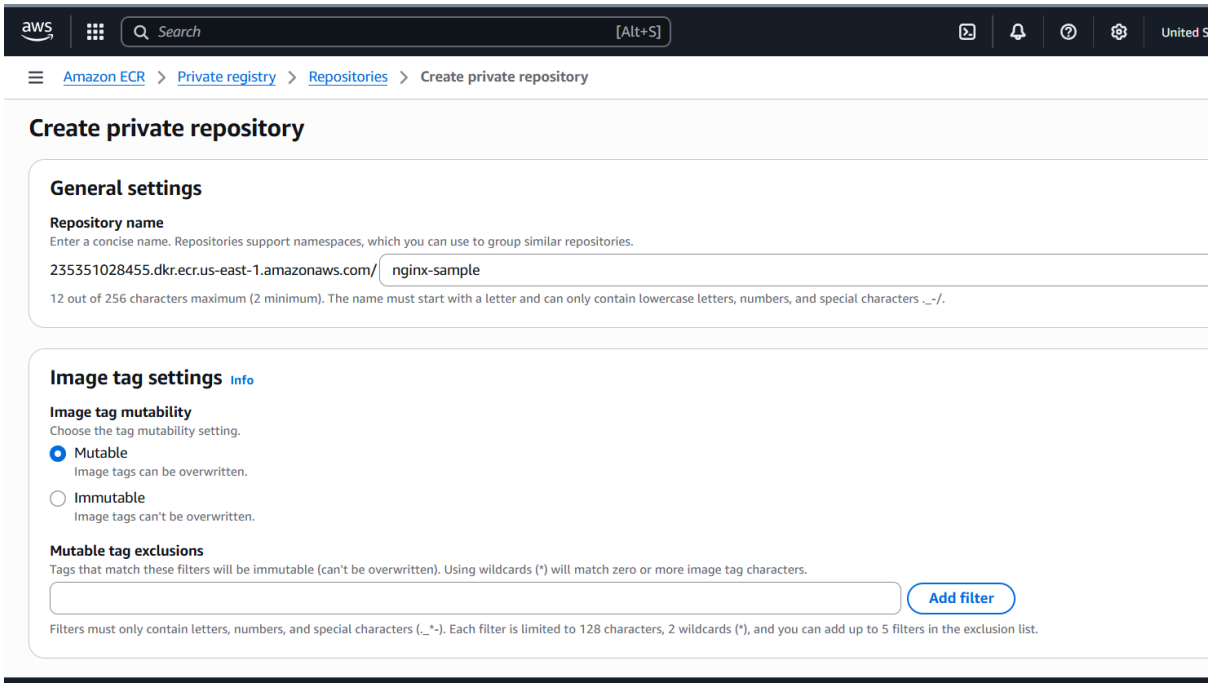


6. push image to AWS ECR

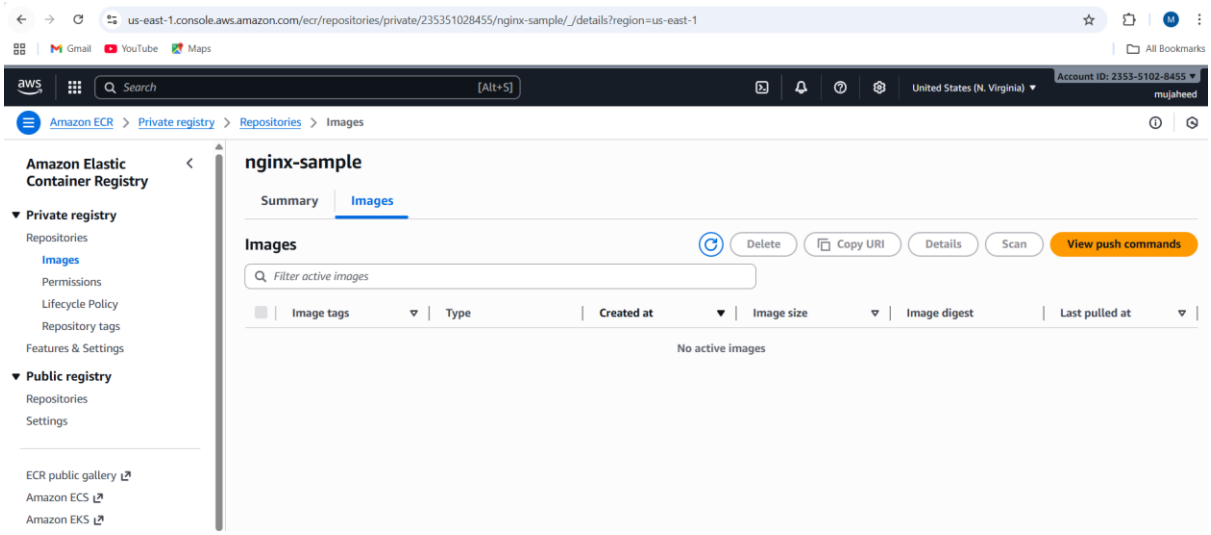
Go to aws ECR create a repository.



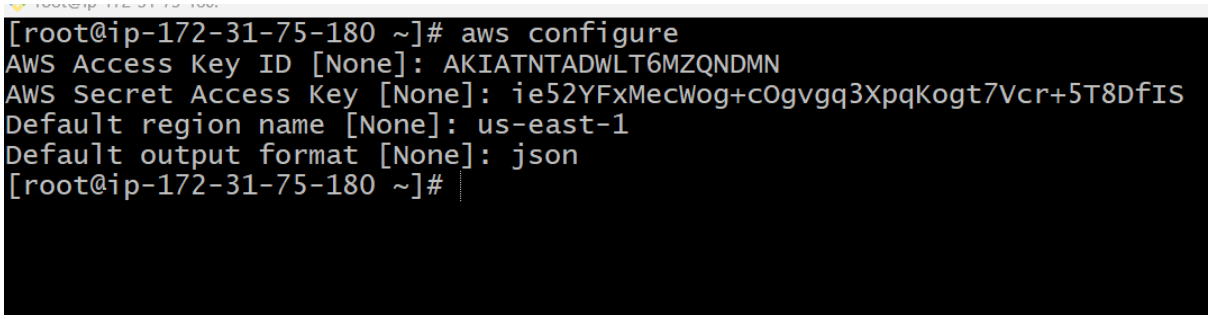
Give the repository name.



There are no images.



- aws configure



- aws ecr get-login-password --region us-east-1 \| docker login --username AWS --password-stdin 235351028455.dkr.ecr.us-east-1.amazonaws.com

```
[root@ip-172-31-75-180 ~]# aws ecr get-login-password --region us-east-1 \
| docker login --username AWS --password-stdin 235351028455.dkr.ecr.us-east-1.amazonaws.com
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[root@ip-172-31-75-180 ~]#
```

- docker tag my-nginx:latest 235351028455.dkr.ecr.us-east-1.amazonaws.com/nginx-sample:latest
- docker push 235351028455.dkr.ecr.us-east-1.amazonaws.com/nginx-sample:latest

```
[root@ip-172-31-75-180 ~]# docker tag my-nginx:latest 235351028455.dkr.ecr.us-east-1.amazonaws.com/nginx-sample:latest
[root@ip-172-31-75-180 ~]# docker push 235351028455.dkr.ecr.us-east-1.amazonaws.com/nginx-sample:latest
The push refers to repository [235351028455.dkr.ecr.us-east-1.amazonaws.com/nginx-sample]
e3dda3eead3d: Pushed
f0224e513680: Pushed
0e7c6314d998: Pushed
latest: digest: sha256:f6377b28d28e64b83d0a5af4066dd829b085c6eecb60e7a2d53192ca3b7082d1 size: 948
[root@ip-172-31-75-180 ~]#
```

The image was pushed to ECR