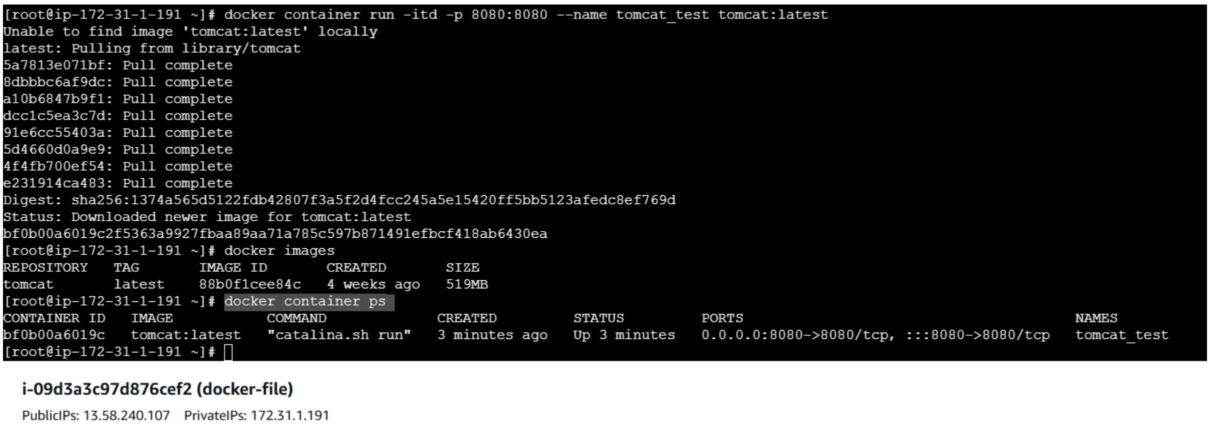
DOCKER 02

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

→Create docker container:

docker container run -itd -p 8080:8080 --name tomcat_test tomcat:latest



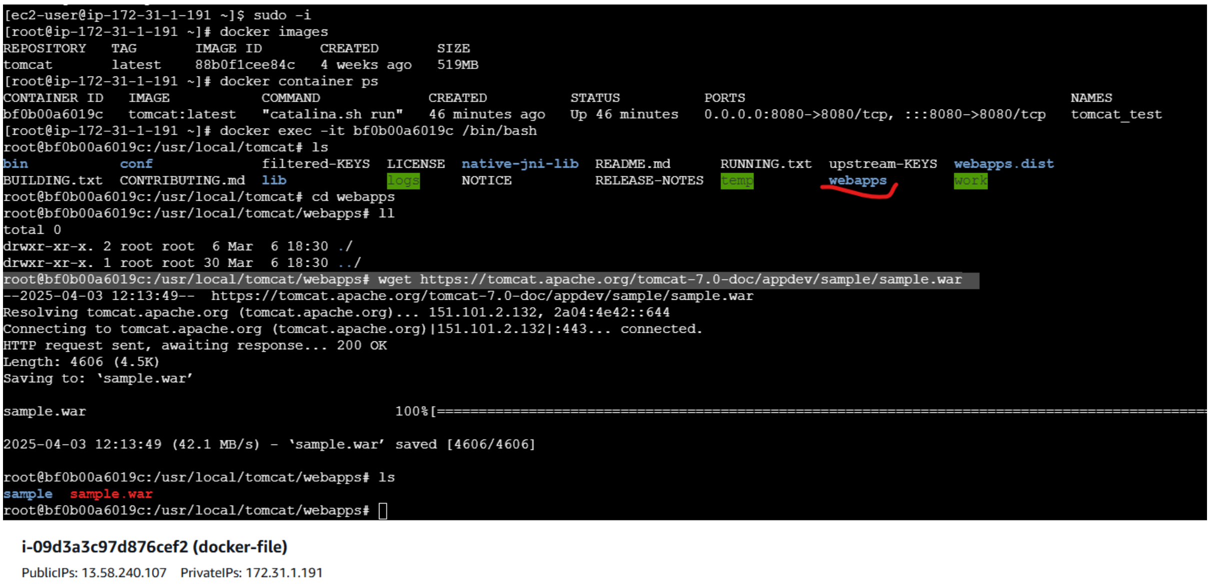
```
[root@ip-172-31-1-191 ~]# docker container run -itd -p 8080:8080 --name tomcat_test tomcat:latest
Unable to find image 'tomcat:latest' locally
latest: Pulling from library/tomcat
5a7813e071bf: Pull complete
8dbbbc6af9dc: Pull complete
a10b6847b9f1: Pull complete
dcc1c5ea3c7d: Pull complete
91e6cc55403a: Pull complete
5d4660d0a9e9: Pull complete
4f4fb700ef54: Pull complete
e231914ca483: Pull complete
Digest: sha256:1374a565d5122fdb42807f3a5f2d4fcc245a5e15420ff5bb5123afedc8ef769d
Status: Downloaded newer image for tomcat:latest
bf0b00a6019c2f5363a9927fbaa89aa71a785c597b871491efbcf418ab6430ea
[root@ip-172-31-1-191 ~]# docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
tomcat       latest    88b0f1cee84c   4 weeks ago   519MB
[root@ip-172-31-1-191 ~]# docker container ps
CONTAINER ID   IMAGE           COMMAND             CREATED         STATUS         PORTS                                       NAMES
bf0b00a6019c   tomcat:latest   "catalina.sh run"   3 minutes ago   Up 3 minutes   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   tomcat_test
[root@ip-172-31-1-191 ~]# 
```

i-09d3a3c97d876cef2 (docker-file)

PublicIPs: 13.58.240.107   PrivateIPs: 172.31.1.191

→now login the container:

docker exec -it container id /bin/bash

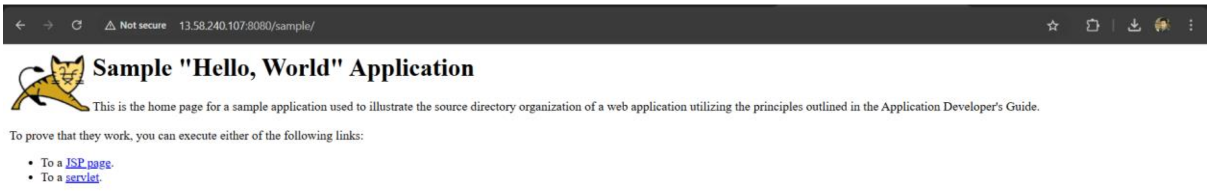


```
[ec2-user@ip-172-31-1-191 ~]$ sudo -i
[root@ip-172-31-1-191 ~]# docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
tomcat       latest    88b0f1cee84c   4 weeks ago   519MB
[root@ip-172-31-1-191 ~]# docker container ps
CONTAINER ID   IMAGE           COMMAND             CREATED          STATUS          PORTS                                       NAMES
bf0b00a6019c   tomcat:latest   "catalina.sh run"   46 minutes ago   Up 46 minutes   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   tomcat_test
[root@ip-172-31-1-191 ~]# docker exec -it bf0b00a6019c /bin/bash
root@bf0b00a6019c:/usr/local/tomcat# ls
bin           conf             filtered-KEYS  LICENSE  native-jni-lib  README.md      RUNNING.txt  upstream-KEYS  webapps.dist
BUILDING.txt  CONTRIBUTING.md  lib            logs     NOTICE          RELEASE-NOTES  temp         webapps        work
root@bf0b00a6019c:/usr/local/tomcat# cd webapps
root@bf0b00a6019c:/usr/local/tomcat/webapps# ll
total 0
drwxr-xr-x. 2 root root  6 Mar  6 18:30 ./
drwxr-xr-x. 1 root root 30 Mar  6 18:30 ../
root@bf0b00a6019c:/usr/local/tomcat/webapps# wget https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war
--2025-04-03 12:13:49--  https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war
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%[==========================================================================================

2025-04-03 12:13:49 (42.1 MB/s) - 'sample.war' saved [4606/4606]

root@bf0b00a6019c:/usr/local/tomcat/webapps# ls
sample  sample.war
root@bf0b00a6019c:/usr/local/tomcat/webapps# 
```

i-09d3a3c97d876cef2 (docker-file)

PublicIPs: 13.58.240.107   PrivateIPs: 172.31.1.191

→ run public ip:8080/sample on browser:



Sample "Hello, World" Application

This is the home page for a sample application used to illustrate the source directory organization of a web application utilizing the principles outlined in the Application Developer's Guide.

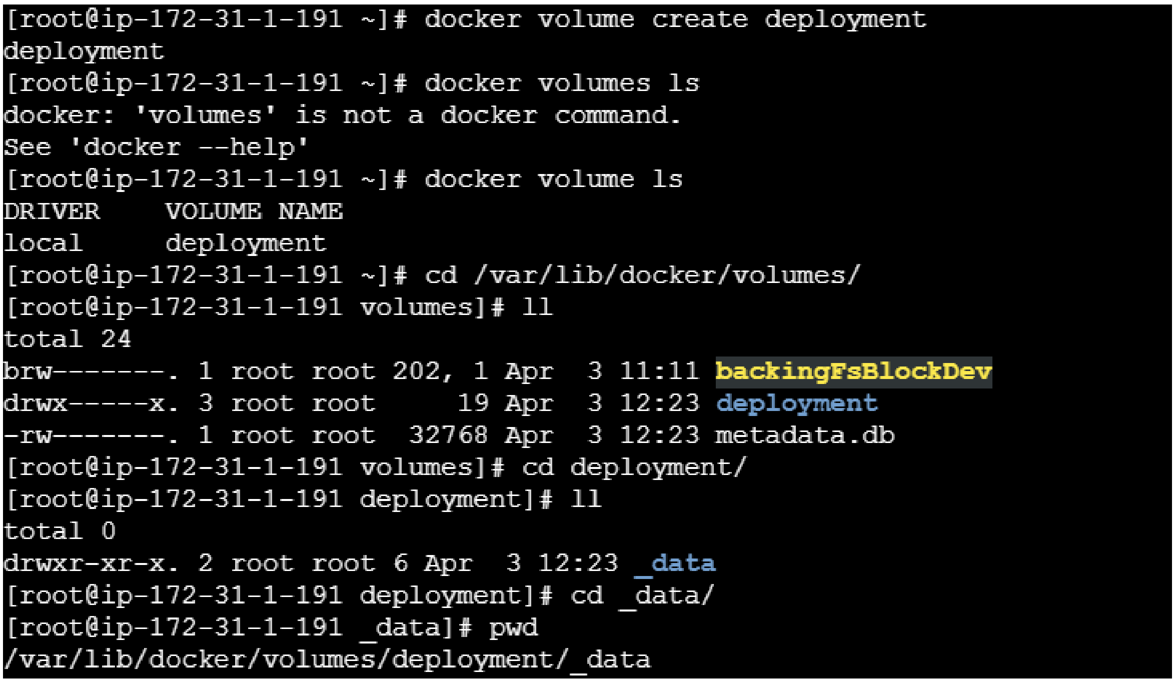
To prove that they work, you can execute either of the following links:

- To a JSP page.
- To a servlet.

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

→to create docker volume use below command:

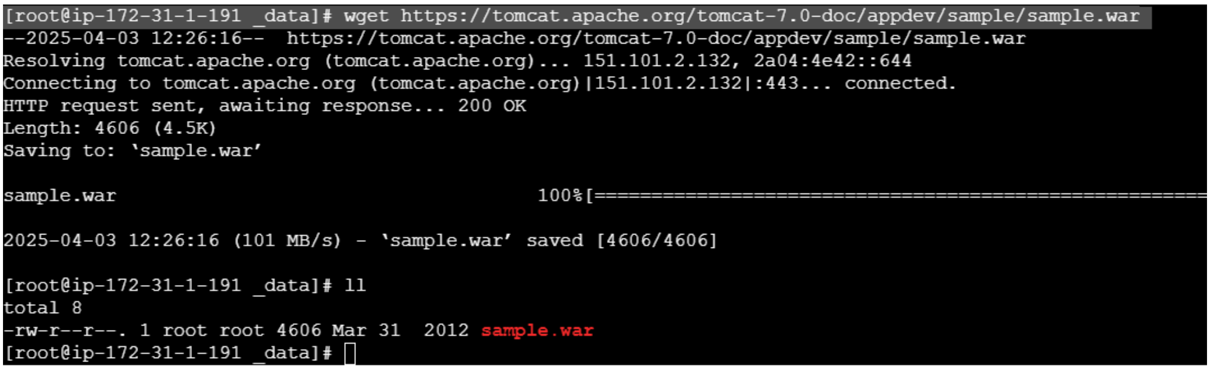
docker volume create deployment



```
[root@ip-172-31-1-191 ~]# docker volume create deployment
deployment
[root@ip-172-31-1-191 ~]# docker volumes ls
docker: 'volumes' is not a docker command.
See 'docker --help'
[root@ip-172-31-1-191 ~]# docker volume ls
DRIVER    VOLUME NAME
local     deployment
[root@ip-172-31-1-191 ~]# cd /var/lib/docker/volumes/
[root@ip-172-31-1-191 volumes]# ll
total 24
brw-------. 1 root root 202, 1 Apr  3 11:11 backingFsBlockDev
drwx-----x. 3 root root     19 Apr  3 12:23 deployment
-rw-------. 1 root root  32768 Apr  3 12:23 metadata.db
[root@ip-172-31-1-191 volumes]# cd deployment/
[root@ip-172-31-1-191 deployment]# ll
total 0
drwxr-xr-x. 2 root root 6 Apr  3 12:23 _data
[root@ip-172-31-1-191 deployment]# cd _data/
[root@ip-172-31-1-191 _data]# pwd
/var/lib/docker/volumes/deployment/_data
```

→deployed sample.war file in volume-deployment:

wget https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war



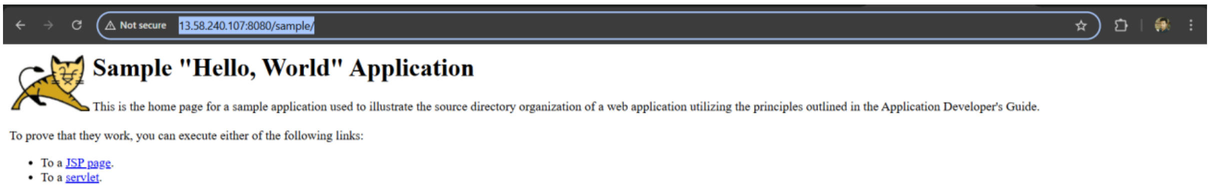
```
[root@ip-172-31-1-191 _data]# wget https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war
--2025-04-03 12:26:16--  https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war
Resolving tomcat.apache.org (tomcat.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to tomcat.apache.org (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%[=====================================================

2025-04-03 12:26:16 (101 MB/s) - ‘sample.war’ saved [4606/4606]

[root@ip-172-31-1-191 _data]# ll
total 8
-rw-r--r--. 1 root root 4606 Mar 31  2012 sample.war
[root@ip-172-31-1-191 _data]#
```

i-09d3a3c97d876cef2 (docker-file)

PublicIPs: 13.58.240.107   PrivateIPs: 172.31.1.191

→calling volume for deploying application in webapps while creating tomcat container:

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



Sample "Hello, World" Application

This is the home page for a sample application used to illustrate the source directory organization of a web application utilizing the principles outlined in the Application Developer's Guide.

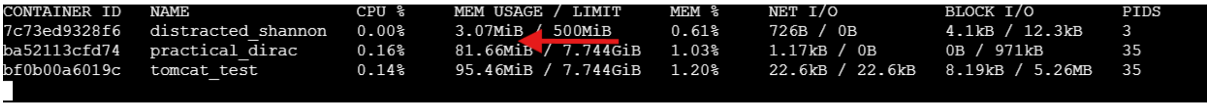
To prove that they work, you can execute either of the following links:

- To a JSP page.
- To a servlet.

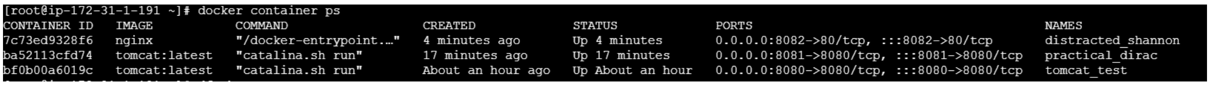
3) Limit the nginx container to 500 MB.

→to Limit the nginx container to 500 MB use below command so its create container with a limit of 500mb:

```
docker container run -itd -p 8082:80 –memory=500m nginx
```



```
CONTAINER ID   NAME                 CPU %     MEM USAGE / LIMIT     MEM %     NET I/O           BLOCK I/O         PIDS
7c73ed9328f6   distracted_shannon   0.00%     3.07MiB / 500MiB      0.61%     726B / 0B         4.1kB / 12.3kB    3
ba52113cfd74   practical_dirac      0.16%     81.66MiB / 7.744GiB   1.03%     1.17kB / 0B       0B / 971kB        35
bf0b00a6019c   tomcat_test          0.14%     95.46MiB / 7.744GiB   1.20%     22.6kB / 22.6kB   8.19kB / 5.26MB   35
```



```
[root@ip-172-31-1-191 ~]# docker container ps
CONTAINER ID   IMAGE           COMMAND                  CREATED             STATUS             PORTS                                       NAMES
7c73ed9328f6   nginx           "/docker-entrypoint.…"   4 minutes ago       Up 4 minutes       0.0.0.0:8082->80/tcp, :::8082->80/tcp       distracted_shannon
ba52113cfd74   tomcat:latest   "catalina.sh run"        17 minutes ago      Up 17 minutes      0.0.0.0:8081->8080/tcp, :::8081->8080/tcp   practical_dirac
bf0b00a6019c   tomcat:latest   "catalina.sh run"        About an hour ago   Up About an hour   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   tomcat_test
```

4) Create a sample docker file using below instructions.

1) Base module as amazonlinux:latest

2) Maintainer you name

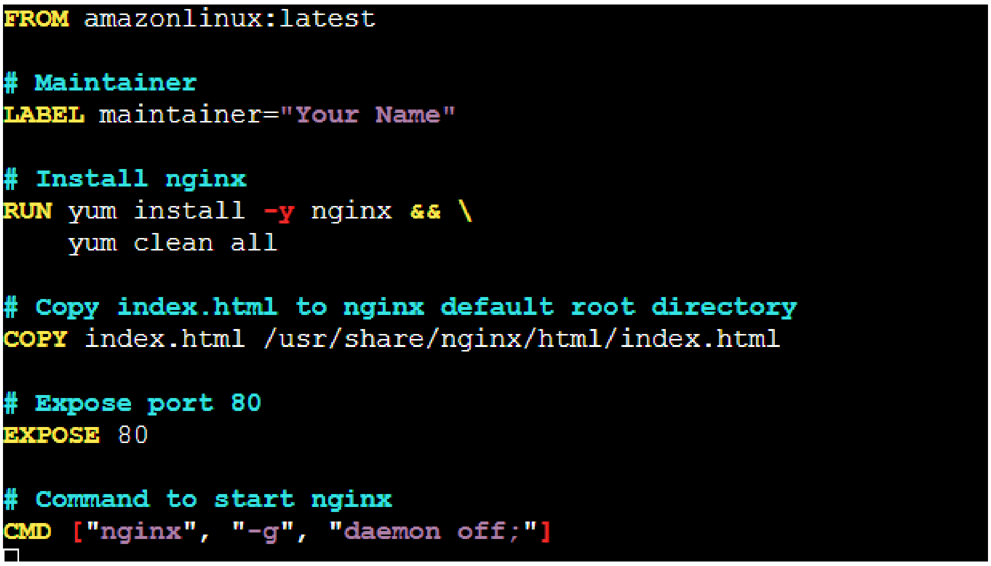
3) Install nginx

4) COPY one index.html file to image

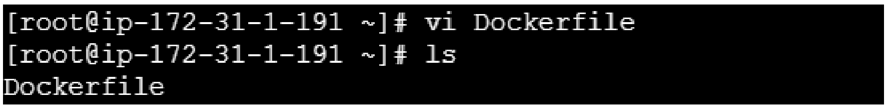
5) EXpose on port 80

6) Command to start the nginx container

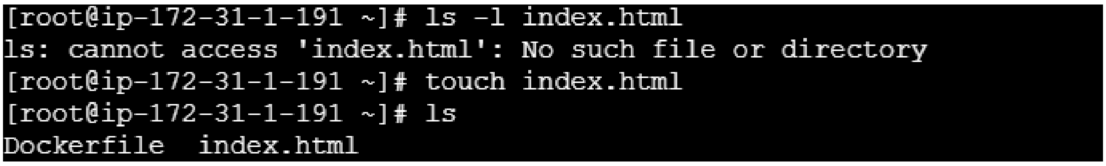
```
FROM amazonlinux:latest

# Maintainer

LABEL maintainer="Your Name"


# Install nginx

RUN yum install -y nginx && \

    yum clean all


# Copy index.html to nginx default root directory

COPY index.html /usr/share/nginx/html/index.html


# Expose port 80

EXPOSE 80


# Command to start nginx

CMD ["nginx", "-g", "daemon off;"]
```



```
FROM amazonlinux:latest

# Maintainer
LABEL maintainer="Your Name"

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

# Copy index.html to nginx default root directory
COPY index.html /usr/share/nginx/html/index.html

# Expose port 80
EXPOSE 80

# Command to start nginx
CMD ["nginx", "-g", "daemon off;"]
```

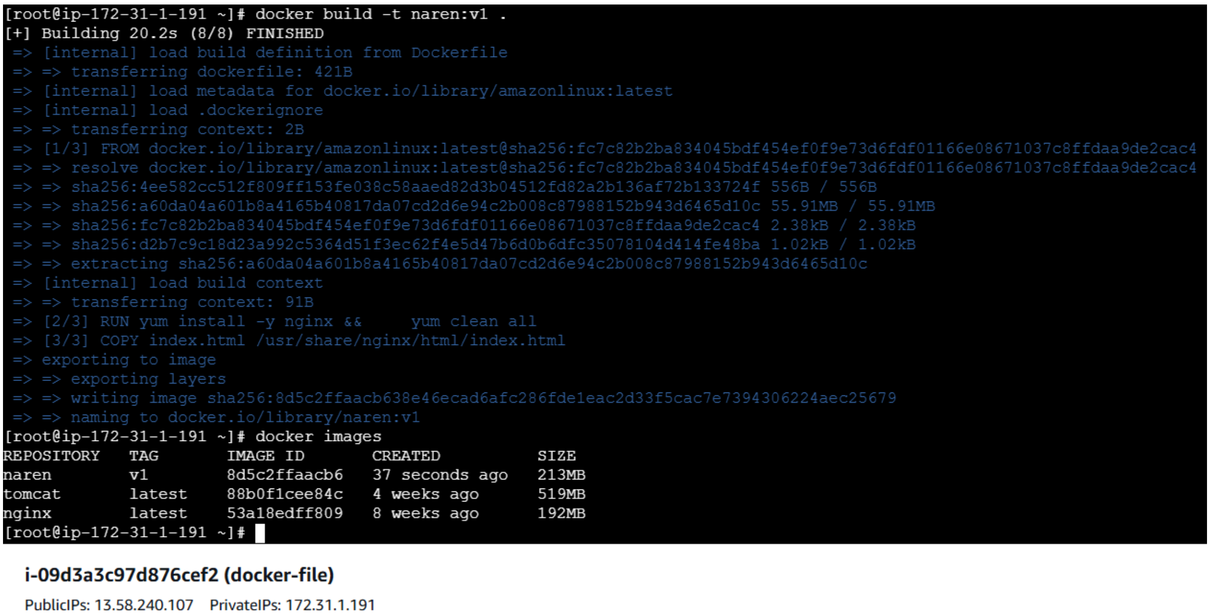


```
[root@ip-172-31-1-191 ~]# vi Dockerfile
[root@ip-172-31-1-191 ~]# ls
Dockerfile
```



```
[root@ip-172-31-1-191 ~]# ls -l index.html
ls: cannot access 'index.html': No such file or directory
[root@ip-172-31-1-191 ~]# touch index.html
[root@ip-172-31-1-191 ~]# ls
Dockerfile  index.html
```

→Created image naren with docker file:



```
[root@ip-172-31-1-191 ~]# docker build -t naren:v1 .
[+] Building 20.2s (8/8) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 421B
 => [internal] load metadata for docker.io/library/amazonlinux:latest
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/3] FROM docker.io/library/amazonlinux:latest@sha256:fc7c82b2ba834045bdf454ef0f9e73d6fdf01166e08671037c8ffdaa9de2cac4
 => => resolve docker.io/library/amazonlinux:latest@sha256:fc7c82b2ba834045bdf454ef0f9e73d6fdf01166e08671037c8ffdaa9de2cac4
 => => sha256:4ee582cc512f809ff153fe038c58aaed82d3b04512fd82a2b136af72b133724f 556B / 556B
 => => sha256:a60da04a601b8a4165b40817da07cd2d6e94c2b008c87988152b943d6465d10c 55.91MB / 55.91MB
 => => sha256:fc7c82b2ba834045bdf454ef0f9e73d6fdf01166e08671037c8ffdaa9de2cac4 2.38kB / 2.38kB
 => => sha256:d2b7c9c18d23a992c5364d51f3ec62f4e5d47b6d0b6dfc35078104d414fe48ba 1.02kB / 1.02kB
 => => extracting sha256:a60da04a601b8a4165b40817da07cd2d6e94c2b008c87988152b943d6465d10c
 => [internal] load build context
 => => transferring context: 91B
 => [2/3] RUN yum install -y nginx &&     yum clean all
 => [3/3] COPY index.html /usr/share/nginx/html/index.html
 => exporting to image
 => => exporting layers
 => => writing image sha256:8d5c2ffaacb638e46ecad6afc286fde1eac2d33f5cac7e7394306224aec25679
 => => naming to docker.io/library/naren:v1
[root@ip-172-31-1-191 ~]# docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
naren        v1        8d5c2ffaacb6   37 seconds ago   213MB
tomcat       latest    88b0f1cee84c   4 weeks ago      519MB
nginx        latest    53a18edff809   8 weeks ago      192MB
[root@ip-172-31-1-191 ~]# 
```

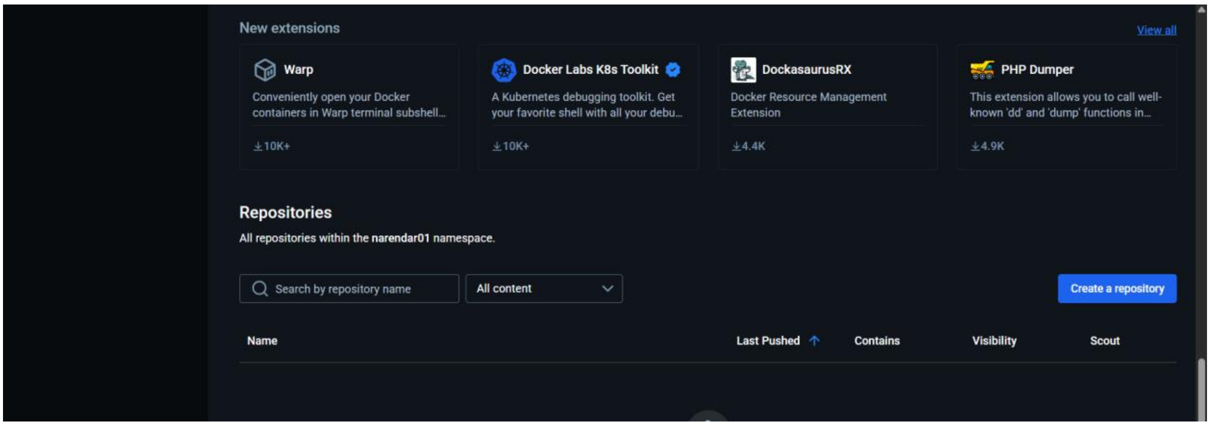
i-09d3a3c97d876cef2 (docker-file)

PublicIPs: 13.58.240.107   PrivateIPs: 172.31.1.191

5) Push image to dockerhub

→create docker hub account:

→goto my hub create repository:



New extensions

Warp
Conveniently open your Docker containers in Warp terminal subshell...
⤓10K+

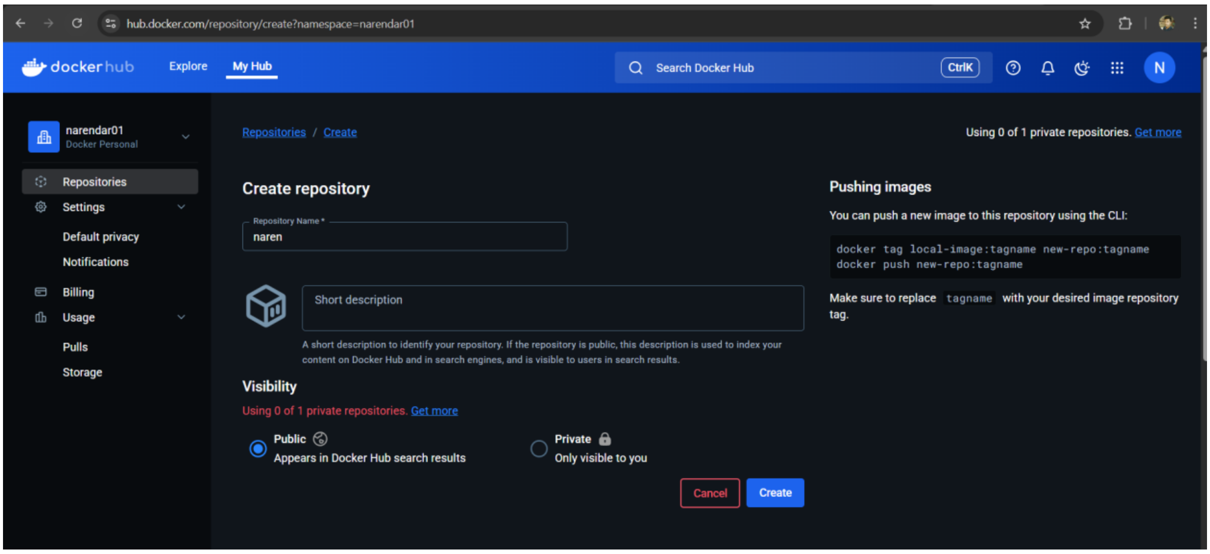
Docker Labs K8s Toolkit
A Kubernetes debugging toolkit. Get your favorite shell with all your debu...
⤓10K+

DockasaurusRX
Docker Resource Management Extension
⤓4.4K

PHP Dumper
This extension allows you to call well-known 'dd' and 'dump' functions in...
⤓4.9K

Repositories

All repositories within the narendar01 namespace.



Using 0 of 1 private repositories. Get more

Create repository

Repository Name *
naren

Short description

A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.

Visibility

Using 0 of 1 private repositories. Get more

● Public
Appears in Docker Hub search results

○ Private
Only visible to you

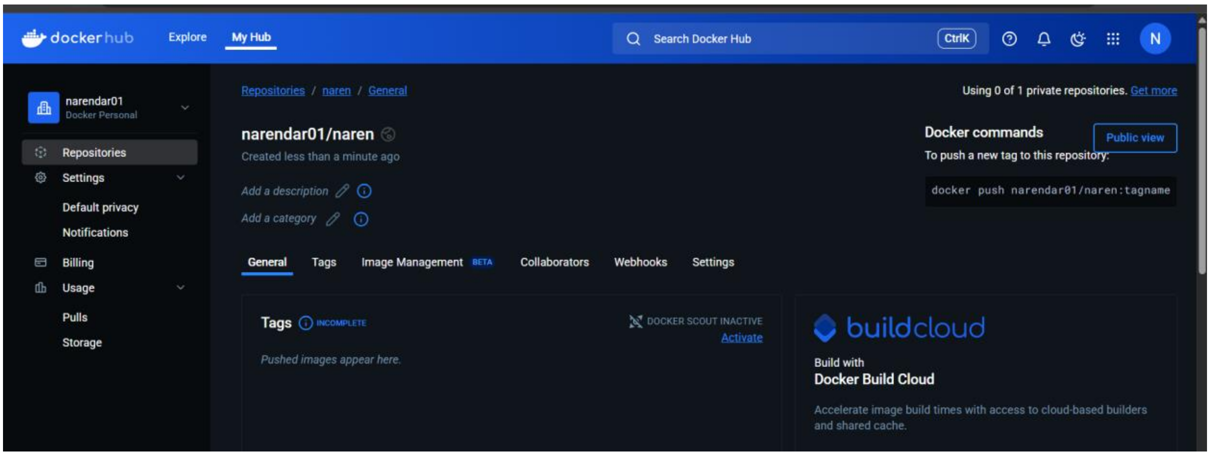
Pushing images

You can push a new image to this repository using the CLI:

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

Make sure to replace `tagname` with your desired image repository tag.

→created repository:



Using 0 of 1 private repositories. Get more

narendar01/naren

Created less than a minute ago

Add a description

Add a category

Tags INCOMPLETE

DOCKER SCOUT INACTIVE
Activate

Pushed images appear here.

Docker commands

To push a new tag to this repository:

```
docker push narendar01/naren:tagname
```

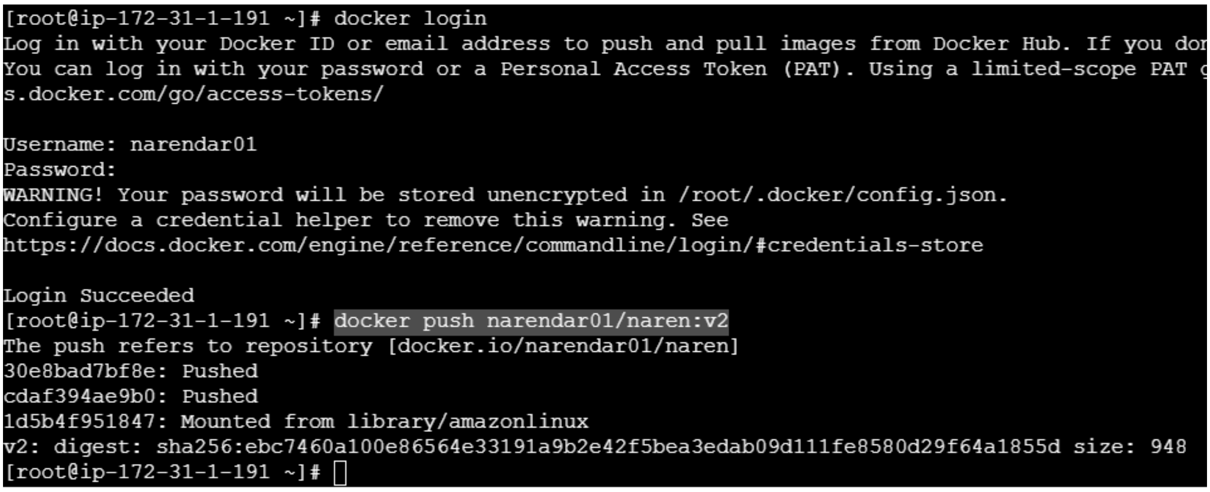
buildcloud

Build with
Docker Build Cloud

Accelerate image build times with access to cloud-based builders and shared cache.

-- docker login

-- docker push narendar01/naren:v2

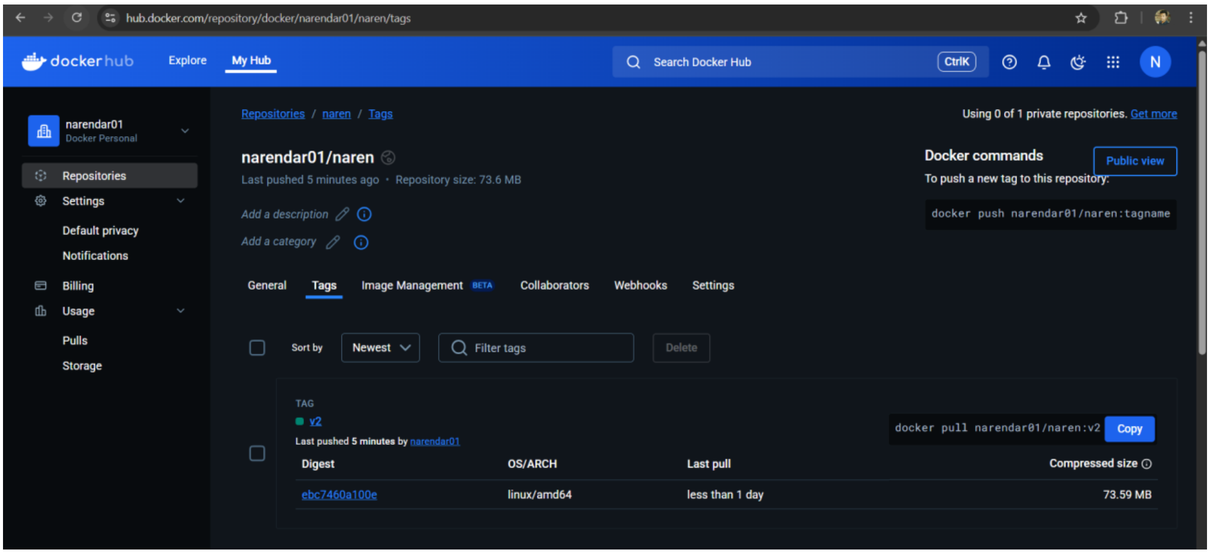


```
[root@ip-172-31-1-191 ~]# docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you do
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT g
s.docker.com/go/access-tokens/

Username: narendar01
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-172-31-1-191 ~]# docker push narendar01/naren:v2
The push refers to repository [docker.io/narendar01/naren]
30e8bad7bf8e: Pushed
cdaf394ae9b0: Pushed
1d5b4f951847: Mounted from library/amazonlinux
v2: digest: sha256:ebc7460a100e86564e33191a9b2e42f5bea3edab09d111fe8580d29f64a1855d size: 948
[root@ip-172-31-1-191 ~]#
```

i-09d3a3c97d876cef2 (docker-file)

PublicIPs: 13.58.240.107   PrivateIPs: 172.31.1.191

→ Pushed image to dockerhub:



Using 0 of 1 private repositories. Get more

narendar01/naren

Last pushed 5 minutes ago · Repository size: 73.6 MB

Docker commands

To push a new tag to this repository:

```
docker push narendar01/naren:tagname
```

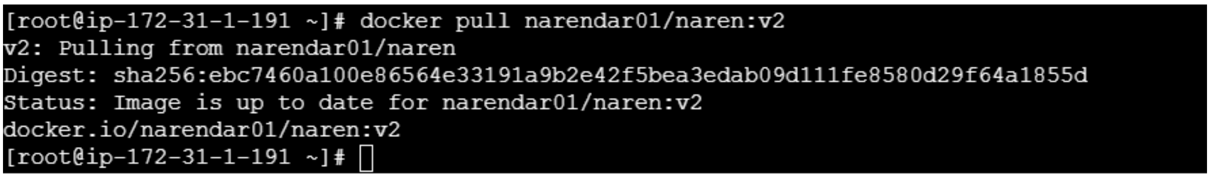
TAG

v2

Last pushed 5 minutes by narendar01

```
docker pull narendar01/naren:v2
```

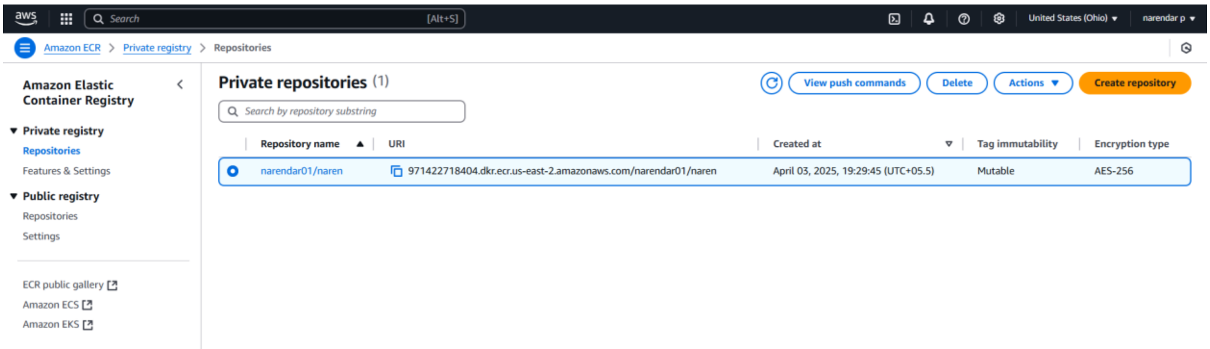
Digest	OS/ARCH	Last pull	Compressed size
ebc7460a100e	linux/amd64	less than 1 day	73.59 MB



```
[root@ip-172-31-1-191 ~]# docker pull narendar01/naren:v2
v2: Pulling from narendar01/naren
Digest: sha256:ebc7460a100e86564e33191a9b2e42f5bea3edab09d111fe8580d29f64a1855d
Status: Image is up to date for narendar01/naren:v2
docker.io/narendar01/naren:v2
[root@ip-172-31-1-191 ~]# 
```

6) push image to aws ecr

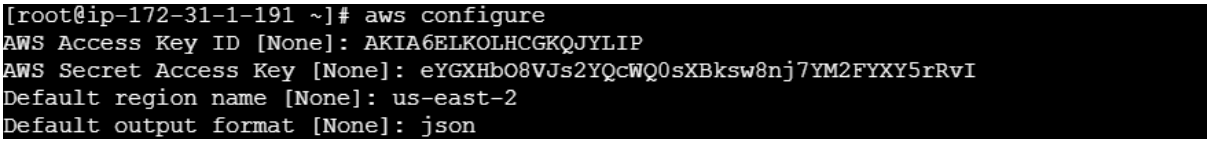
→create repository in AWS ECR:



Private repositories (1)

	Repository name	URI	Created at	Tag immutability	Encryption type
●	narendar01/naren	971422718404.dkr.ecr.us-east-2.amazonaws.com/narendar01/naren	April 03, 2025, 19:29:45 (UTC+05.5)	Mutable	AES-256

→do aws configure in ec2 docker server:



```
[root@ip-172-31-1-191 ~]# aws configure
AWS Access Key ID [None]: AKIA6ELKOLHCGKQJYLIP
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-2
Default output format [None]: json
```

→then goto repository in AWS ECR click on view push commands

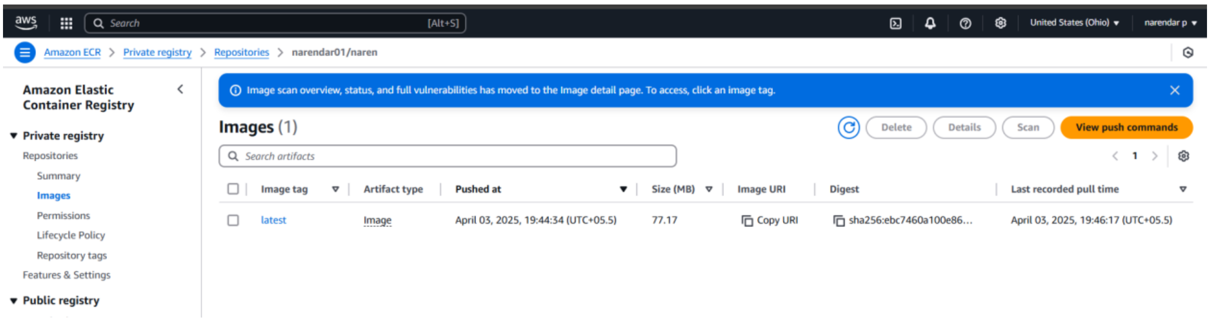
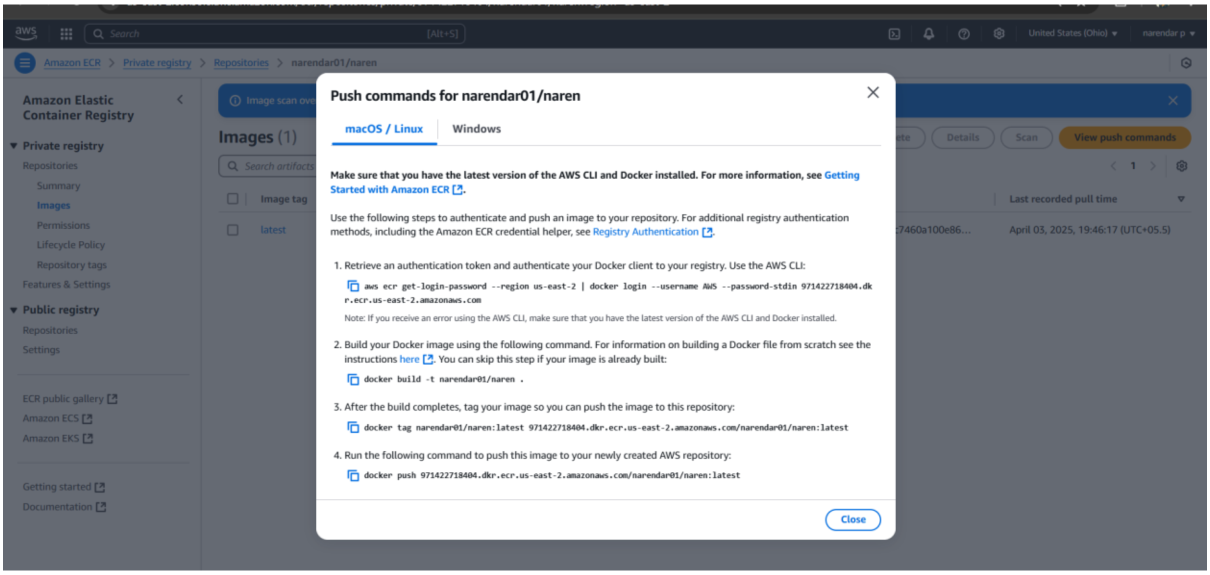


Image scan overview, status, and full vulnerabilities has moved to the Image detail page. To access, click an image tag.

Images (1)

□	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
□	latest	Image	April 03, 2025, 19:44:34 (UTC+05.5)	77.17	Copy URI	sha256:ebc7460a100e86...	April 03, 2025, 19:46:17 (UTC+05.5)

→run all commands in docker server:



Push commands for narendar01/naren

Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see Getting Started with Amazon ECR.

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see Registry Authentication.

1. Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS CLI:

```
aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 971422718404.dk
r.ecr.us-east-2.amazonaws.com
```

   Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.

2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions here. You can skip this step if your image is already built:

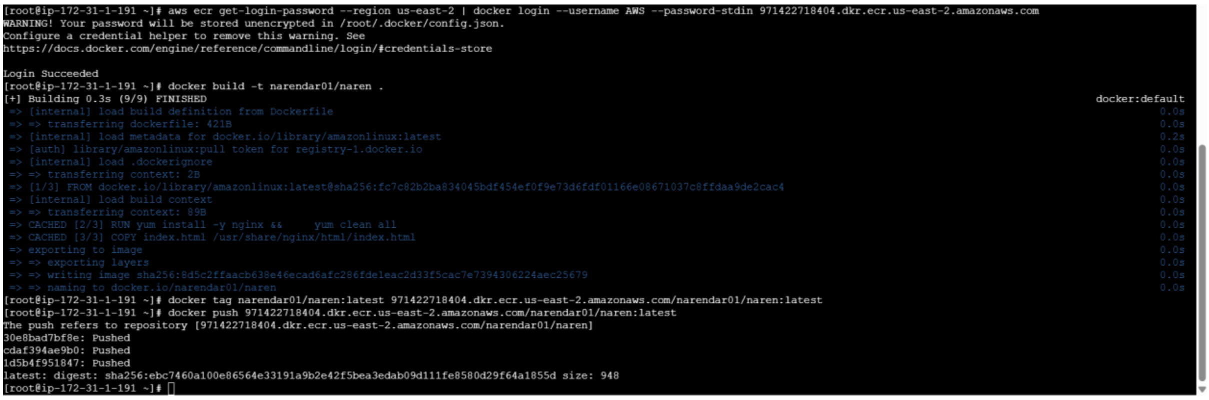
```
docker build -t narendar01/naren .
```

3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag narendar01/naren:latest 971422718404.dkr.ecr.us-east-2.amazonaws.com/narendar01/naren:latest
```

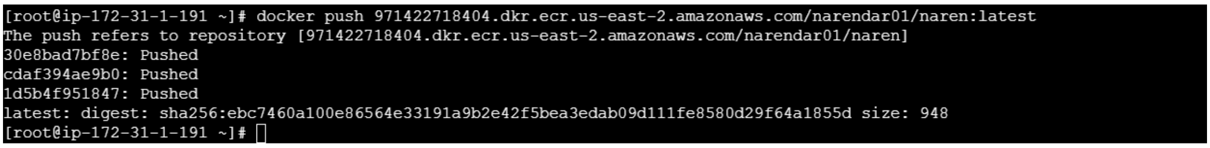
4. Run the following command to push this image to your newly created AWS repository:

```
docker push 971422718404.dkr.ecr.us-east-2.amazonaws.com/narendar01/naren:latest
```



```
[root@ip-172-31-1-191 ~]# aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 971422718404.dkr.ecr.us-east-2.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-172-31-1-191 ~]# docker build -t narendar01/naren .
[+] Building 0.3s (9/9) FINISHED                                                                          docker:default
 => [internal] load build definition from Dockerfile                                                                0.0s
 => => transferring dockerfile: 421B                                                                                0.0s
 => [internal] load metadata for docker.io/library/amazonlinux:latest                                               0.2s
 => [auth] library/amazonlinux:pull token for registry-1.docker.io                                                  0.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [1/3] FROM docker.io/library/amazonlinux:latest@sha256:fc7c82b2ba834045bdf454ef0f9e73d6fdf01166e08671037c8ffdaa9de2cac4  0.0s
 => [internal] load build context                                                                                   0.0s
 => => transferring context: 89B                                                                                    0.0s
 => CACHED [2/3] RUN yum install -y nginx &&     yum clean all                                                      0.0s
 => CACHED [3/3] COPY index.html /usr/share/nginx/html/index.html                                                   0.0s
 => exporting to image                                                                                              0.0s
 => => exporting layers                                                                                             0.0s
 => => writing image sha256:8d5c2ffaacb638e46ecad6afc286fde1eac2d33f5cac7e7394306224aec25679                        0.0s
 => => naming to docker.io/narendar01/naren                                                                         0.0s
[root@ip-172-31-1-191 ~]# docker tag narendar01/naren:latest 971422718404.dkr.ecr.us-east-2.amazonaws.com/narendar01/naren:latest
[root@ip-172-31-1-191 ~]# docker push 971422718404.dkr.ecr.us-east-2.amazonaws.com/narendar01/naren:latest
The push refers to repository [971422718404.dkr.ecr.us-east-2.amazonaws.com/narendar01/naren]
30e8bad7bf8e: Pushed
cdaf394ae9b0: Pushed
1d5b4f951847: Pushed
latest: digest: sha256:ebc7460a100e86564e33191a9b2e42f5bea3edab09d111fe8580d29f64a1855d size: 948
[root@ip-172-31-1-191 ~]# 
```

→pushed image to aws ecr:



```
[root@ip-172-31-1-191 ~]# docker push 971422718404.dkr.ecr.us-east-2.amazonaws.com/narendar01/naren:latest
The push refers to repository [971422718404.dkr.ecr.us-east-2.amazonaws.com/narendar01/naren]
30e8bad7bf8e: Pushed
cdaf394ae9b0: Pushed
1d5b4f951847: Pushed
latest: digest: sha256:ebc7460a100e86564e33191a9b2e42f5bea3edab09d111fe8580d29f64a1855d size: 948
[root@ip-172-31-1-191 ~]# 
```

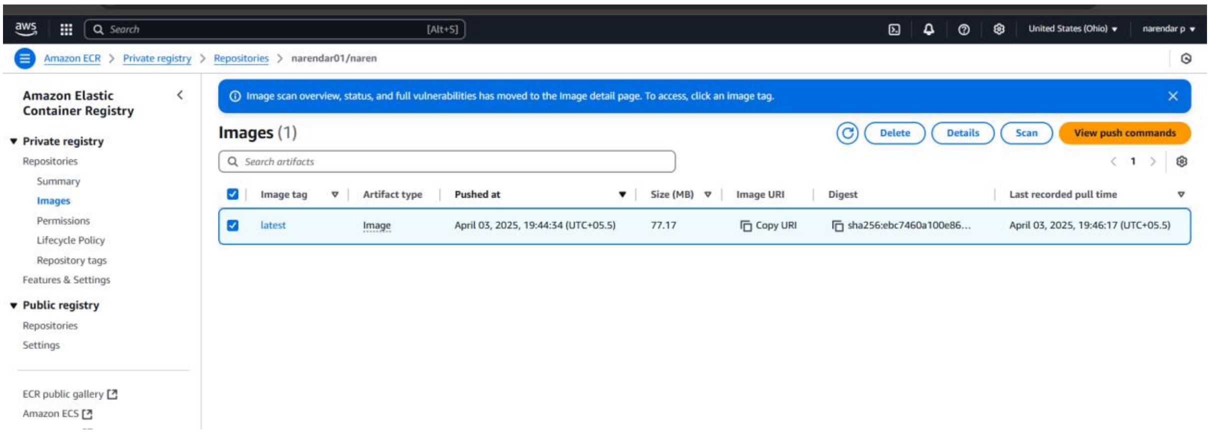
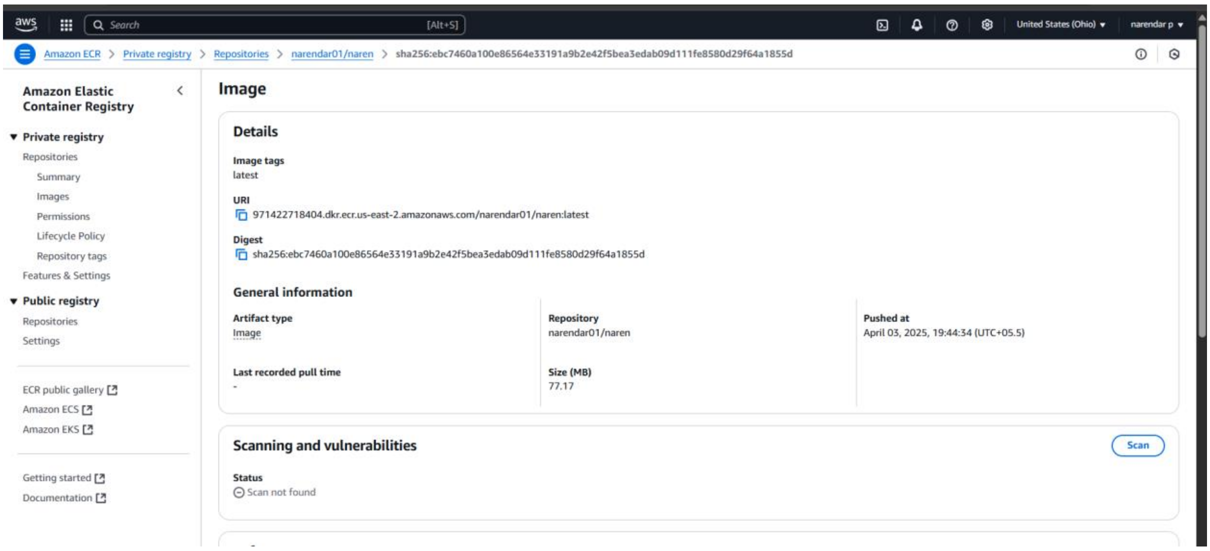


Image scan overview, status, and full vulnerabilities has moved to the Image detail page. To access, click an image tag.

Images (1)

☑	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
☑	latest	Image	April 03, 2025, 19:44:34 (UTC+05.5)	77.17	Copy URI	sha256:ebc7460a100e86...	April 03, 2025, 19:46:17 (UTC+05.5)



Image

Details

Image tags
latest

URI
971422718404.dkr.ecr.us-east-2.amazonaws.com/narendar01/naren:latest

Digest
sha256:ebc7460a100e86564e33191a9b2e42f5bea3edab09d111fe8580d29f64a1855d

General information

Artifact type
Image

Repository
narendar01/naren

Pushed at
April 03, 2025, 19:44:34 (UTC+05.5)

Last recorded pull time
-

Size (MB)
77.17

Scanning and vulnerabilities

Status
Scan not found