DOCKER 3

1) Create a image from running container.

--create container:

docker container run -itd -p 80:80 nginx:latest

```
[root@ip-172-31-3-201 ~]# docker container ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[root@ip-172-31-3-201 ~]# docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
[root@ip-172-31-3-201 ~]# docker container run -itd -p 80:80 nginx:latest
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
6e909acdb790: Pull complete
5eaa34f5b9c2: Pull complete
417c4bccf534: Pull complete
e7e0ca015e55: Pull complete
```

```
[root@ip-172-31-3-201 ~]# docker container ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS                                   NAMES
6b764470af6f   nginx:latest   "/docker-entrypoint.…"   13 seconds ago   Up 12 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp   elegant_brahmagupta
```

--to Create a image from running container use below command:

docker commit 6b764470af6f

```
[root@ip-172-31-3-201 ~]# docker commit 6b764470af6f
sha256:a262667520186246a7e8e90909870dd72bfa626dc09d44810e284798740d894c
```

--created image from running container:

```
[root@ip-172-31-3-201 ~]# docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
<none>       <none>    a26266752018   27 seconds ago   192MB
nginx        latest    53a18edff809   8 weeks ago      192MB
[root@ip-172-31-3-201 ~]#
```

2) Copy image from local machine to docker server and load the image.

--download docker desktop

--connect to lacal machine from pem key is their

```
MINGW64:/c/Users/Public/Downloads

naren@narendar MINGW64 /c/Users/Public/Downloads
$ docker --version
Docker version 28.0.1, build 068a01e
```

--pull nginx image

```
naren@narendar MINGW64 /c/Users/Public/Downloads
$ docker pull nginx:latest

latest: Pulling from library/nginx
c22eb46e871a: Pulling fs layer
5eaa34f5b9c2: Pulling fs layer
373fe654e984: Pulling fs layer
417c4bccf534: Pulling fs layer
e7e0ca015e55: Pulling fs layer
97f5c0f51d43: Pulling fs layer
6e909acdb790: Pulling fs layer
373fe654e984: Download complete
c22eb46e871a: Download complete
417c4bccf534: Download complete
e7e0ca015e55: Download complete
97f5c0f51d43: Download complete
5eaa34f5b9c2: Download complete
6e909acdb790: Download complete
6e909acdb790: Pull complete
417c4bccf534: Pull complete
5eaa34f5b9c2: Pull complete
373fe654e984: Pull complete
c22eb46e871a: Pull complete
e7e0ca015e55: Pull complete
97f5c0f51d43: Pull complete
Digest: sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056c52d3692baebc19
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest

naren@narendar MINGW64 /c/Users/Public/Downloads
$

naren@narendar MINGW64 /c/Users/Public/Downloads
$ docker images
REPOSITORY                TAG       IMAGE ID       CREATED          SIZE
nginx                     latest    124b44bfc9cc   8 weeks ago      279MB
docker/welcome-to-docker  latest    eedaff45e3c7   17 months ago    29.5MB
```

--create tar file

docker save -o nginx.tar nginx:latest

```
naren@narendar MINGW64 /c/Users/Public/Downloads
$ docker save -o nginx.tar nginx:latest

naren@narendar MINGW64 /c/Users/Public/Downloads
$ ls
 971422718404_CloudTrail_ap-northeast-2_20250304T0935Z_hegLZVW1RcPRrCbW.json
'ANSIBLE 01.pdf'
'ANSIBLE 02.pdf'
'ANSIBLE 03.pdf'
'ANSIBLE 04 (1).pdf'
'ANSIBLE 04.pdf'
'AUTOSCALING GROUPS TASK.pdf'
```

--created nginx.tar file

```
 docker.pem
 hyd.pem
 jenkins.pem
'linux 01.pdf'
 my-key.pem
 new-vpc.pem
 nginx.tar
'sample (1).war'
 sample.war
```

-- Copy image from local machine to docker server use below command

$ scp -i /c/Users/Public/Downloads/docker.pem nginx.tar ec2-user@18.226.17.190:/tmp

```
naren@narendar MINGW64 /c/Users/Public/Downloads
$ pwd
/c/Users/Public/Downloads

naren@narendar MINGW64 /c/Users/Public/Downloads
$ scp -i /c/Users/Public/Downloads/docker.pem nginx.tar ec2-user@18.226.17.190:/tmp
The authenticity of host '18.226.17.190 (18.226.17.190)' can't be established.
ED25519 key fingerprint is SHA256:73e9LV6NY5thM8/uCD0DtncoOQoQg81lB8z2XKkAK5A.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '18.226.17.190' (ED25519) to the list of known hosts.
nginx.tar

naren@narendar MINGW64 /c/Users/Public/Downloads
```

--check tarfile in ec2 docker server

```
[root@ip-172-31-3-201 ~]# cd /tmp
[root@ip-172-31-3-201 tmp]# ls
nginx.tar                                                                    sys
systemd-private-1e186bb4a12e4f859b6198b9f0458150-chronyd.service-bTAgXP      sys
systemd-private-1e186bb4a12e4f859b6198b9f0458150-dbus-broker.service-VpbRfe  sys
[root@ip-172-31-3-201 tmp]# docker images
```

--load the image

docker load -i nginx.tar

```
[root@ip-172-31-3-201 tmp]# docker load -i nginx.tar
Loaded image: nginx:latest
```

--image created

```
[root@ip-172-31-3-201 tmp]# docker images
REPOSITORY          TAG       IMAGE ID       CREATED        SIZE
<none>              <none>    a26266752018   3 hours ago    192MB
nginx               latest    53a18edff809   8 weeks ago    192MB
```

3) Create Docker image using alpine and customize with tomcat.

--Created Docker file

```
# Use Alpine as base image
FROM alpine:latest

# Set environment variables
ENV TOMCAT_VERSION=9.0.85 \
    CATALINA_HOME=/opt/tomcat \
    PATH="$PATH:/opt/tomcat/bin"

# Install dependencies: OpenJDK, curl, and tar
RUN apk update && apk add --no-cache openjdk17 curl tar

# Create Tomcat directory
RUN mkdir -p /opt/tomcat

# Download and extract Tomcat
RUN curl -fsSL https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.102/bin/apache-tomcat-9.0.102.tar.gz \
    | tar -xz -C /opt/tomcat --strip-components=1

# Expose Tomcat port
EXPOSE 8080

# Start Tomcat
CMD ["catalina.sh", "run"]
```

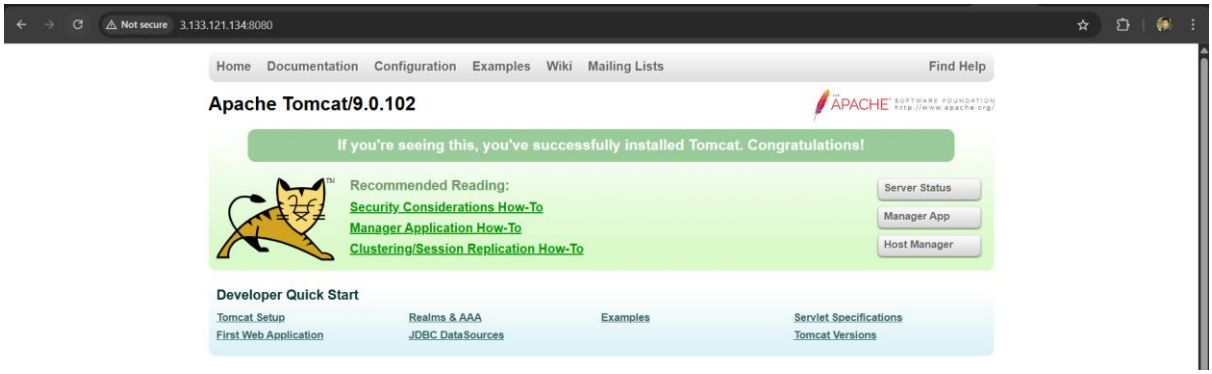
-- Docker image created using alpine

```
[root@ip-172-31-4-129 tomcat-alpine]# vi Dockerfile
[root@ip-172-31-4-129 tomcat-alpine]# docker build -t tomcat-alpine .
[+] Building 2.6s (9/9) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 682B
 => [internal] load metadata for docker.io/library/alpine:latest
 => [auth] library/alpine:pull token for registry-1.docker.io
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/4] FROM docker.io/library/alpine:latest@sha256:a8560b36e8b8210634f77d9f7f9efd7ffa463e380b75e2e74aff4511df3ef88c
 => CACHED [2/4] RUN apk update && apk add --no-cache openjdk17 curl tar
 => CACHED [3/4] RUN mkdir -p /opt/tomcat
 => [4/4] RUN curl -fsSL https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.102/bin/apache-tomcat-9.0.102.tar.gz     | tar -xz -C /opt/tomcat --strip-c
 => exporting to image
 => => exporting layers
 => => writing image sha256:26dfb72aebefb2f647a3772570d6ac503eb904a9d04e9663f0cb70cf153f796b
 => => naming to docker.io/library/tomcat-alpine
[root@ip-172-31-4-129 tomcat-alpine]# docker images
REPOSITORY                                                  TAG       IMAGE ID       CREATED              SIZE
tomcat-alpine                                               latest    26dfb72aebef   15 seconds ago       320MB
```

--Run the image

```
[root@ip-172-31-4-129 tomcat-alpine]# docker run -d -p 8080:8080 --name mytomcat tomcat-alpine
1bcd241de6d77742c5ddb9cfbe7fa4045cf45cc3f406bbaf231ffa05751d8275
```

--Run the Tomcat



4) Create single stage and multi stage docker file using the below source code.

https://github.com/betawins/multi-stage-example.git

--clone the url

git clone https://github.com/betawins/multi-stage-example.git

```
[root@ip-172-31-10-249 ~]# git --version
git version 2.47.1
[root@ip-172-31-10-249 ~]# git clone https://github.com/betawins/multi-stage-example.git
Cloning into 'multi-stage-example'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 31 (delta 2), reused 1 (delta 1), pack-reused 24 (from 1)
Receiving objects: 100% (31/31), 53.25 KiB | 2.42 MiB/s, done.
Resolving deltas: 100% (3/3), done.
[root@ip-172-31-10-249 ~]# ls
multi-stage-example
[root@ip-172-31-10-249 ~]# cd multi-stage-example/
[root@ip-172-31-10-249 multi-stage-example]# ls
Dockerfile  README.md  mvnw  mvnw.cmd  pom.xml  src
[root@ip-172-31-10-249 multi-stage-example]# vi Dockerfile
```

--Single stage Docker file:

FROM openjdk:8-jdk

RUN mkdir -p /app/source

COPY . /app/source

WORKDIR /app/source

RUN ./mvnw clean package

EXPOSE 8080

ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom", "-jar", "/app/source/target/multi-stage-example-0.0.1-SNAPSHOT.jar"]

```
FROM openjdk:8-jdk
RUN mkdir -p /app/source
COPY . /app/source
WORKDIR /app/source
RUN ./mvnw clean package
EXPOSE 8080
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom", "-jar", "/app/source/target/multi-stage-example-0.0.1-SNAPSHOT.jar"]
```

--now build dockerimage using docker file

```
[root@ip-172-31-10-249 multi-stage-example]# docker build -t singlestage:v1 .
[+] Building 0.2s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 352B
 => [internal] load metadata for docker.io/library/openjdk:8-jdk
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/5] FROM docker.io/library/openjdk:8-jdk@sha256:86e863cc57215cfb181bd319736d0baf625fe8f150577f9eb58bd937f5452cb8
 => [internal] load build context
 => => transferring context: 7.41kB
 => CACHED [2/5] RUN mkdir -p /app/source
 => CACHED [3/5] COPY . /app/source
 => CACHED [4/5] WORKDIR /app/source
 => CACHED [5/5] RUN ./mvnw clean package
 => exporting to image
 => => exporting layers
 => => writing image sha256:c8ef393064cdc5aad792212d79827a91367f0d5c51985144ff2d3cf2131b2024
 => => naming to docker.io/library/singlestage:v1
[root@ip-172-31-10-249 multi-stage-example]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED             SIZE
singlestage   v1        c8ef393064cd   About a minute ago  619MB
multistage    v1        c8ef393064cd   About a minute ago  619MB
multistage    v2        f34461e481d8   13 minutes ago      545MB
singlestage   v2        f34461e481d8   13 minutes ago      545MB
```

--create container from image

```
[root@ip-172-31-10-249 multi-stage-example]# docker container run -itd -p 8081:8080 singlestage:v1
daea9a38286e0e89c2d2461fb60093384791cab0b9f218189fe280b9ee57ef29
[root@ip-172-31-10-249 multi-stage-example]# docker ps
CONTAINER ID   IMAGE            COMMAND                  CREATED          STATUS         PORTS                                          NAMES
daea9a38286e   singlestage:v1   "java -Djava.securit…"   10 seconds ago   Up 9 seconds   0.0.0.0:8081->8080/tcp, :::8081->8080/tcp      heuristic_bohr
b1228b56714b   multistage:v2    "java -Djava.securit…"   9 minutes ago    Up 9 minutes   0.0.0.0:8082->8080/tcp, :::8082->8080/tcp      flamboyant_leavitt
```

--check it in browser

--Multistage Dockerfile:

FROM openjdk:8-jdk as builder

RUN mkdir -p /app/source

COPY . /app/source

WORKDIR /app/source

RUN ./mvnw clean package

#Run image

FROM openjdk:8-jdk

WORKDIR /app

COPY --from=builder /app/source/target/*.jar /app/app.jar

EXPOSE 8080

ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom", "-jar", "/app/app.jar"]

```
FROM openjdk:8-jdk as builder
RUN mkdir -p /app/source
COPY . /app/source
WORKDIR /app/source
RUN ./mvnw clean package

#Run image
FROM openjdk:8-jdk
WORKDIR /app
COPY --from=builder /app/source/target/*.jar /app/app.jar
EXPOSE 8080
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom", "-jar", "/app/app.jar"]

~
~
```

--build image from docker file

```
[root@ip-172-31-10-249 multi-stage-example]# docker build -t multistage:v2 .
[+] Building 0.2s (12/12) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 419B
 => [internal] load metadata for docker.io/library/openjdk:8-jdk
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load build context
 => => transferring context: 7.41kB
 => [builder 1/5] FROM docker.io/library/openjdk:8-jdk@sha256:86e863cc57215cfb181bd319736d0baf625fe8f150577f9eb58bd937f5452
 => CACHED [stage-1 2/3] WORKDIR /app
 => CACHED [builder 2/5] RUN mkdir -p /app/source
 => CACHED [builder 3/5] COPY . /app/source
 => CACHED [builder 4/5] WORKDIR /app/source
 => CACHED [builder 5/5] RUN ./mvnw clean package
 => CACHED [stage-1 3/3] COPY --from=builder /app/source/target/*.jar /app/app.jar
 => exporting to image
 => => exporting layers
 => => writing image sha256:f34461e481d8177db314b968890585a547e2cc180fa79c0a3c8afd09dd1ecda8
 => => naming to docker.io/library/multistage:v2
[root@ip-172-31-10-249 multi-stage-example]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
multistage    v2        f34461e481d8   2 minutes ago    545MB
singlestage   v2        f34461e481d8   2 minutes ago    545MB
```

--create container by running above image

```
[root@ip-172-31-10-249 multi-stage-example]# docker container run -itd -p 8082:8080 multistage:v2
b1228b56714babb407f92985f0e1369a56cceb6a365ca4a78d61d13ea6bb17f8
[root@ip-172-31-10-249 multi-stage-example]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS                                       NAMES
b1228b56714b   multistage:v2  "java -Djava.securit…"   8 seconds ago    Up 6 seconds    0.0.0.0:8082->8080/tcp, :::8082->8080/tcp   flamboyant_leavitt
[root@ip-172-31-10-249 multi-stage-example]# vi Dockerfile
```

--check it in browser

Not secure 3.145.83.65:8082

Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Tue Apr 08 07:27:09 UTC 2025
There was an unexpected error (type=Not Found, status=404).
No message available

5) Install docker compose and execute sample applciation.

--docker compose installed

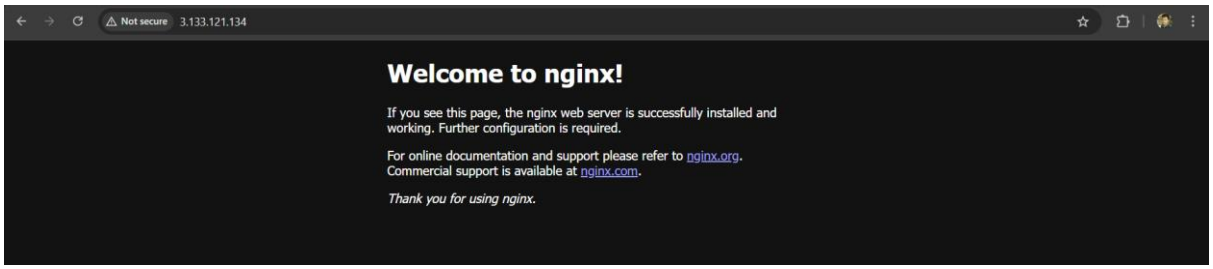
```
[root@ip-172-31-4-129 tomcat-alpine]# docker run -d -p 8080:8080 --name mytomcat tomcat-alpine
1bcd241de6d77742c5ddb9cfbe7fa4045cf45cc3f406bbaf231ffa05751d8275
[root@ip-172-31-4-129 tomcat-alpine]# vi Dockerfile
[root@ip-172-31-4-129 tomcat-alpine]# cd
[root@ip-172-31-4-129 ~]# DOCKER_CONFIG=${DOCKER_CONFIG:-$HOME/.docker}
mkdir -p $DOCKER_CONFIG/cli-plugins

curl -SL https://github.com/docker/compose/releases/latest/download/docker-compose-linux-x86_64 \
  -o $DOCKER_CONFIG/cli-plugins/docker-compose

chmod +x $DOCKER_CONFIG/cli-plugins/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 71.4M  100 71.4M    0     0  57.9M      0  0:00:01  0:00:01 --:--:-- 69.0M
[root@ip-172-31-4-129 ~]# docker compose version
Docker Compose version v2.34.0
```

--Executed Sample Application

```
services:
  web:
    image: nginx:alpine
    ports:
      - "80:80"
```

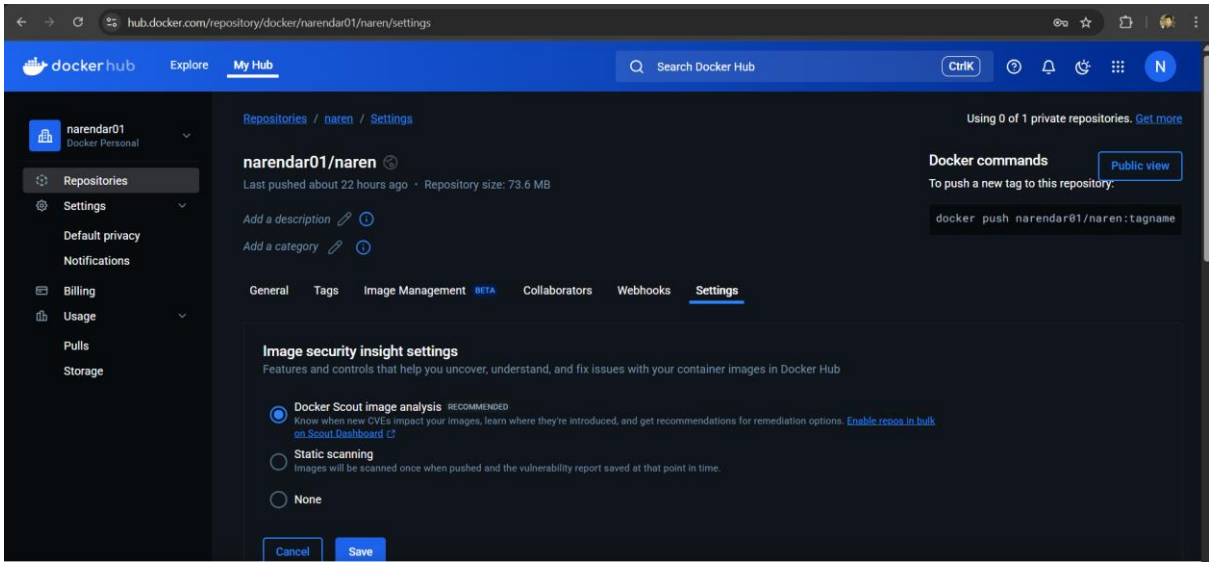
```
[root@ip-172-31-4-129 sample-app]# vi docker-compose.yml
[root@ip-172-31-4-129 sample-app]# docker compose up -d
[+] Running 9/9
 ✓ web Pulled
   ✓ f18232174bc9 Already exists
   ✓ ccc35e35d420 Pull complete
   ✓ 43f2ec460bdf Pull complete
   ✓ 984583bcf083 Pull complete
   ✓ 8d27c072a58f Pull complete
   ✓ ab3286a73463 Pull complete
   ✓ 6d79cc6084d4 Pull complete
   ✓ 0c7e4c092ab7 Pull complete
[+] Running 2/2
 ✓ Network sample-app_default  Created
 ✓ Container sample-app-web-1  Started
[root@ip-172-31-4-129 sample-app]#
```

--Access It on browser



6) Implement solution to scan images when pushed to docker registry.

--to do this first goto dockerhub-repositories-settings-enable Dcocker scout image analysis



--create tag:

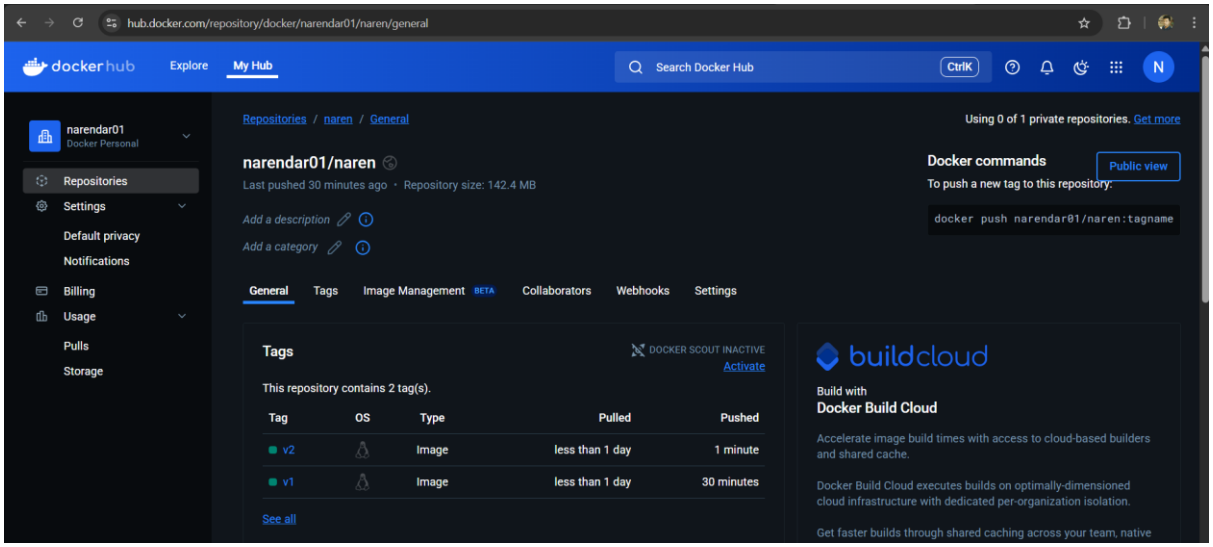
docker tag nginx:latest narendar01/naren:v2

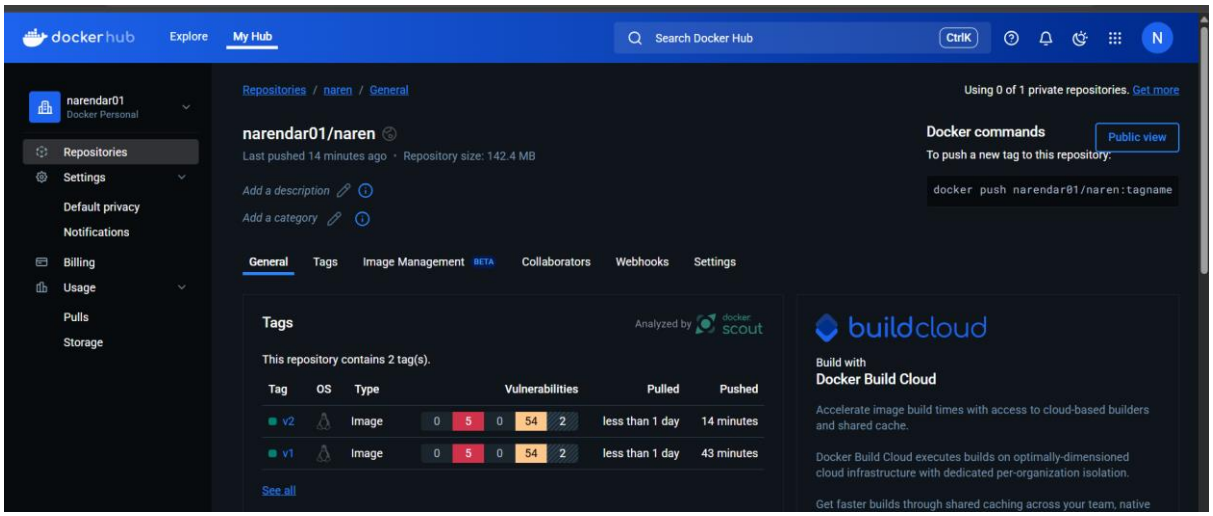
```
[root@ip-172-31-3-201 ~]# docker tag nginx:latest narendar01/naren:v2
[root@ip-172-31-3-201 ~]# docker images
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
<none>              <none>    a26266752018   30 minutes ago   192MB
nginx               latest    53a18edff809   8 weeks ago      192MB
narendar01/naren    v2        53a18edff809   8 weeks ago      192MB
[root@ip-172-31-3-201 ~]#
```

```
denied: requested access to the resource is denied
[root@ip-172-31-3-201 ~]# docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't h
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grant
 more at https://docs.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-172-31-3-201 ~]#
```

--push to docker registry:

-- docker push narendar01/naren:v2



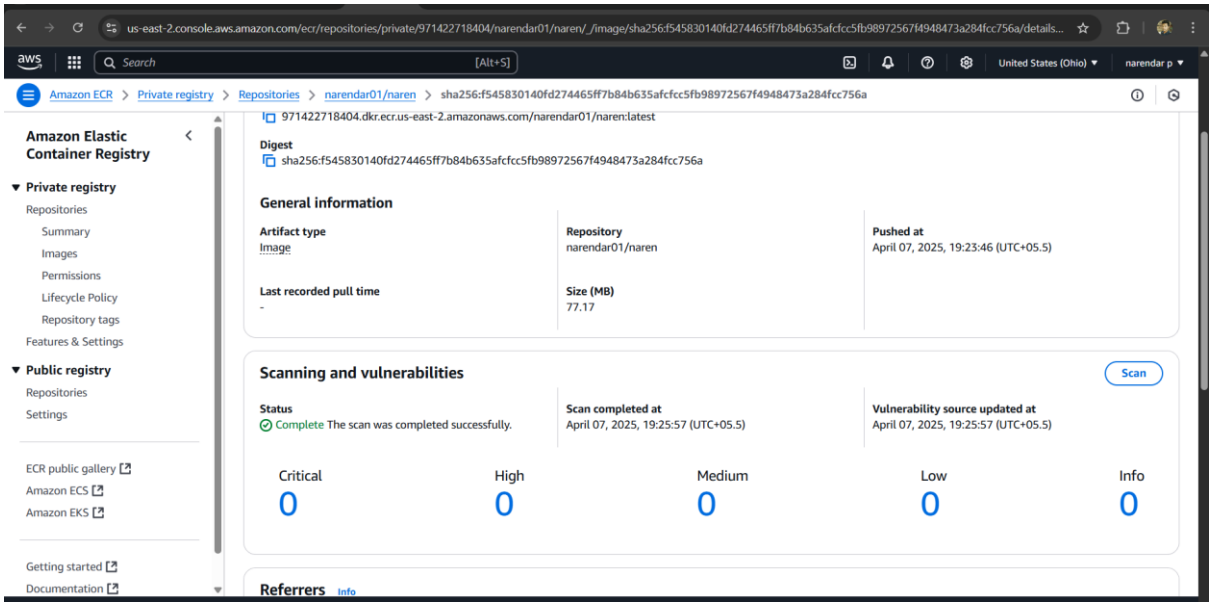


7) Implement solution to scan images when pushed to aws ecr.

--pushed to AWS ECR

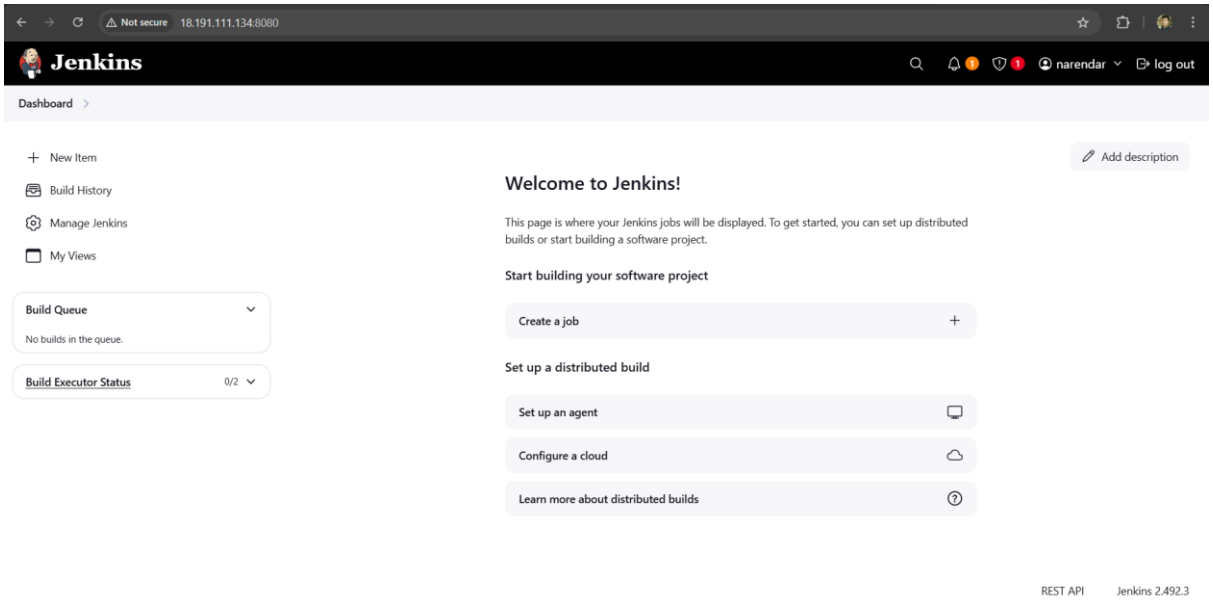
```
[root@ip-172-31-4-129 ~]# docker build -t narendar01/naren .
[+] Building 0.3s (9/9) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 306B
 => [internal] load metadata for docker.io/library/amazonlinux:latest
 => [auth] library/amazonlinux:pull token for registry-1.docker.io
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/3] FROM docker.io/library/amazonlinux:latest@sha256:fc7c82b2ba834045bdf454ef0f9e73d6fdf01166e08671037c8ffdaa9de2cac4
 => [internal] load build context
 => => transferring context: 89B
 => CACHED [2/3] RUN yum update -y &&    yum install -y nginx &&    yum clean all
 => CACHED [3/3] COPY index.html /usr/share/nginx/html/index.html
 => exporting to image
 => => exporting layers
 => => writing image sha256:1bd79aebeb92dc3b713068cbb570b635e76c65ac305aedae34e155b4da03a189
 => => naming to docker.io/narendar01/naren
[root@ip-172-31-4-129 ~]# docker tag narendar01/naren:latest 971422718404.dkr.ecr.us-east-2.amazonaws.com/narendar01/naren:latest
[root@ip-172-31-4-129 ~]# 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]
ae0060acedeb: Pushed
7cc3bf79ad1e: Pushed
1d5b4f951847: Pushed
latest: digest: sha256:f545830140fd274465ff7b84b635afcfcc5fb98972567f4948473a284fcc756a size: 948
[root@ip-172-31-4-129 ~]#
```

-- solution to scan images when pushed to aws ecr

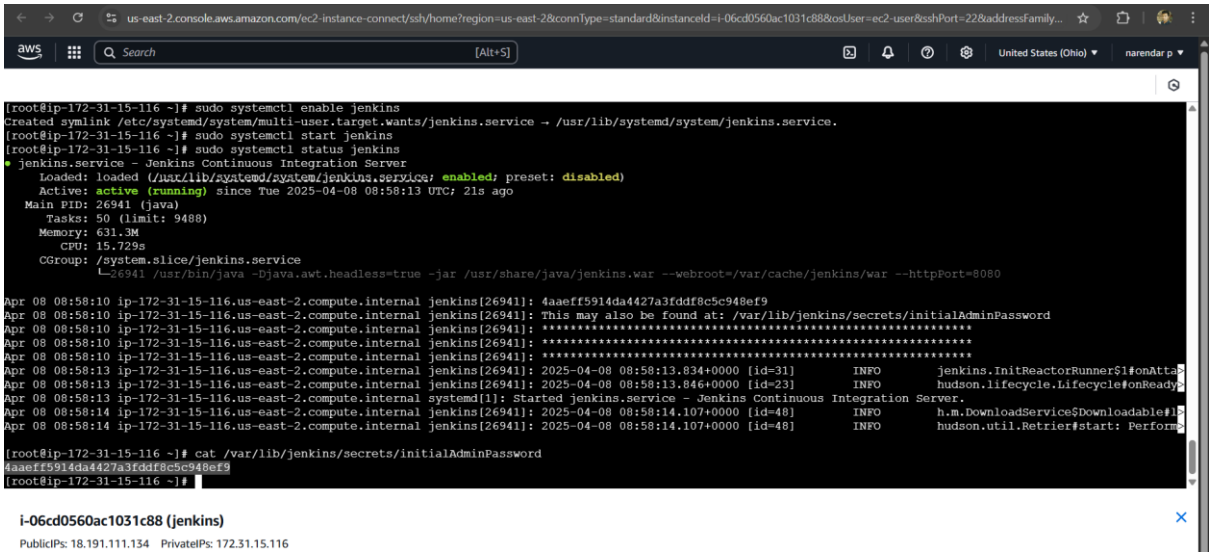


8) Create a jenkins pipeline to create a docker image and push the image to dockerhub.

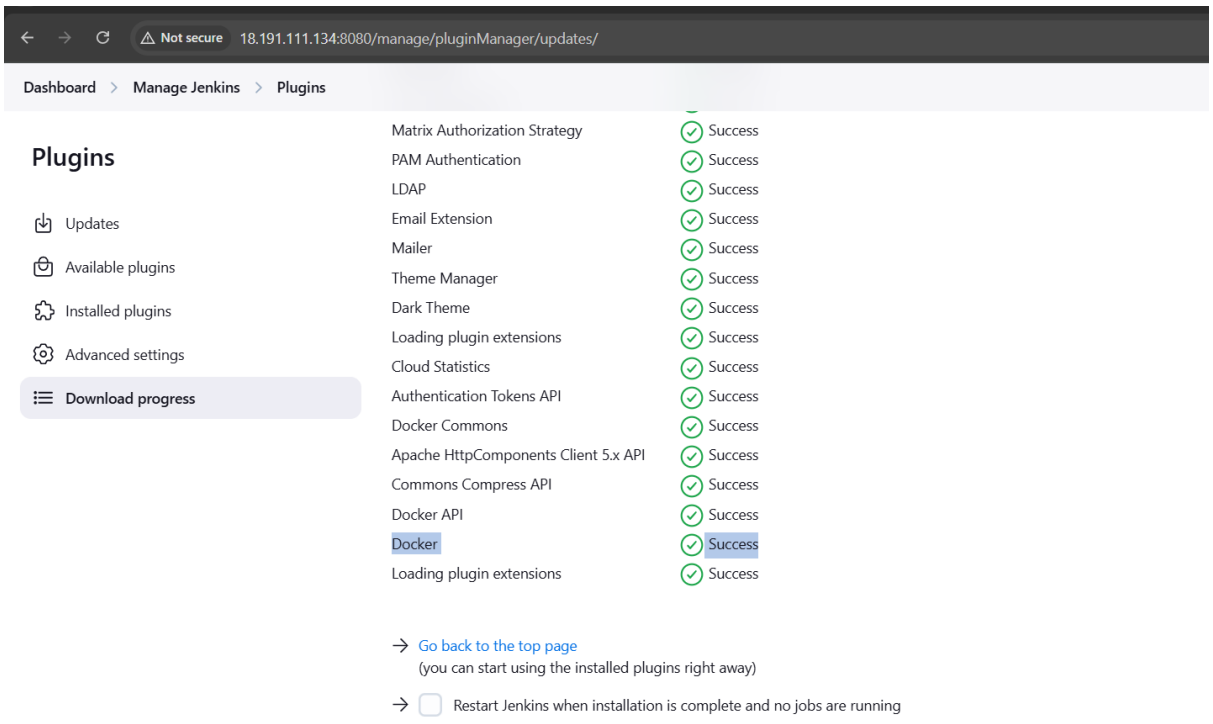
--create Jenkins:



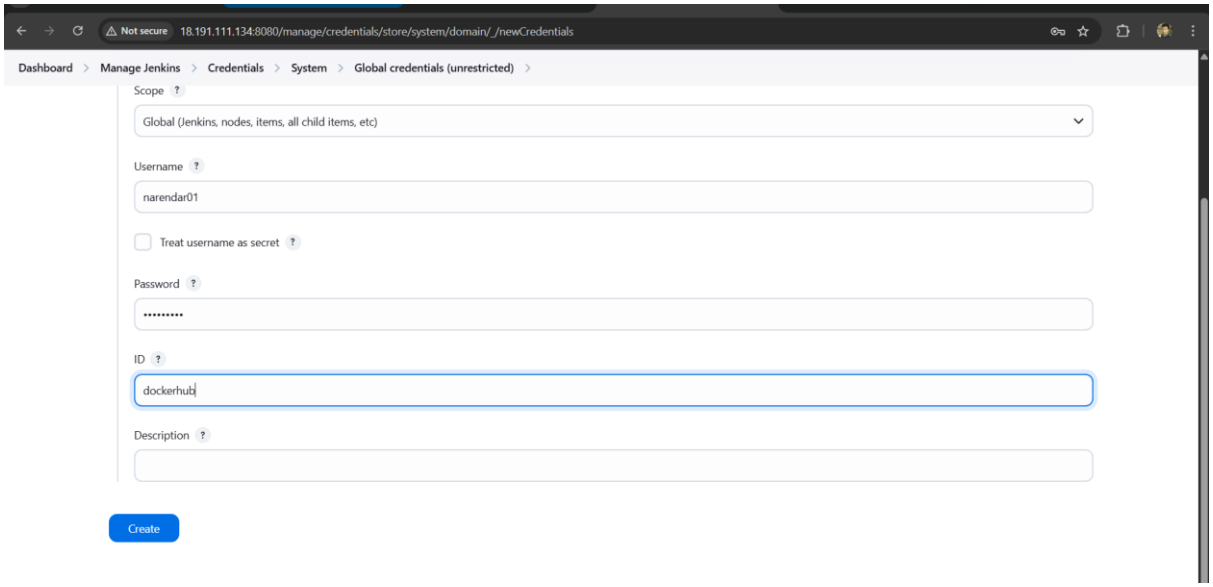
--jenkins running on server:



--install docker hub, dockerpipeline plugins in Jenkins:



--give credentials:

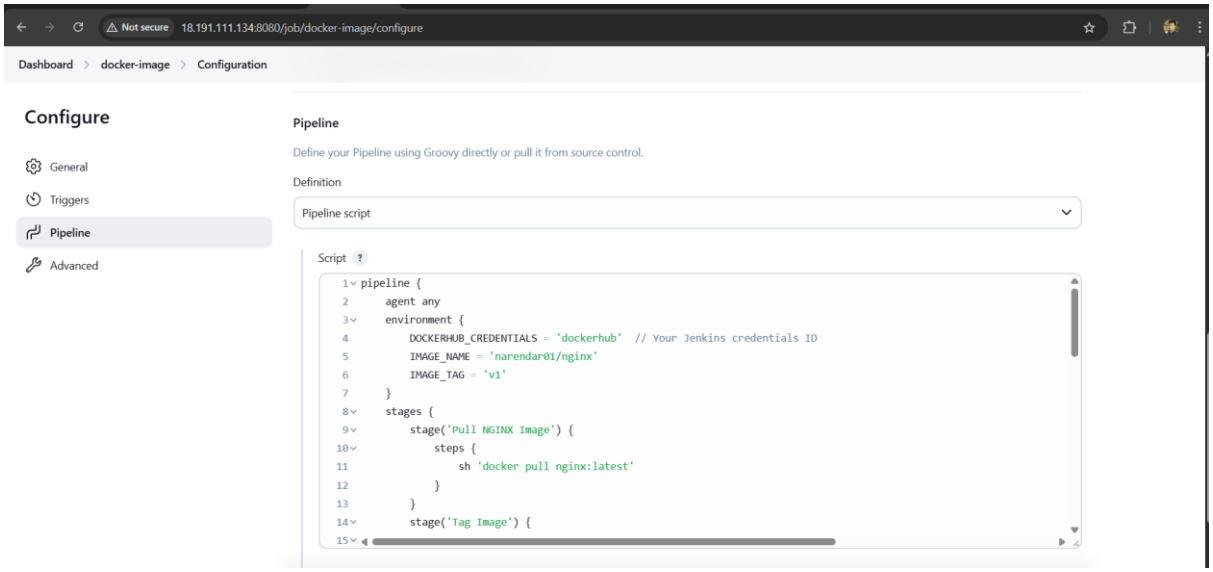


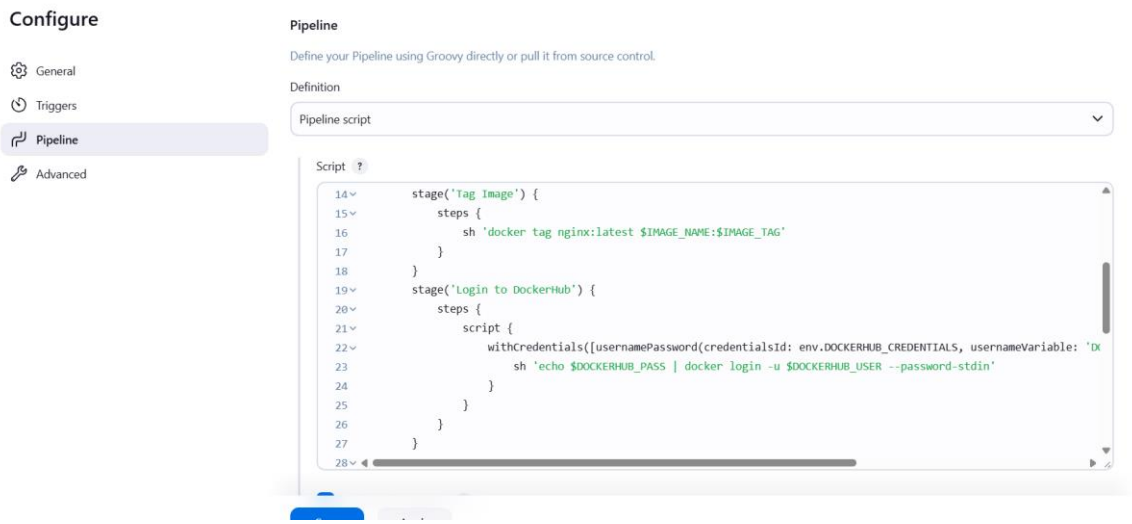
-- jenkins Declarative pipeline to create a docker image and push the image to dockerhub:

```
pipeline {
    agent any
    environment {
        DOCKERHUB_CREDENTIALS = 'dockerhub'  // Your Jenkins credentials ID
        IMAGE_NAME = 'narendar01/nginx'
        IMAGE_TAG = 'v1'
    }
    stages {
        stage('Pull NGINX Image') {
```

```
        steps {
            sh 'docker pull nginx:latest'
        }
    }
    stage('Tag Image') {
        steps {
            sh 'docker tag nginx:latest $IMAGE_NAME:$IMAGE_TAG'
        }
    }
    stage('Login to DockerHub') {
        steps {
            script {
                withCredentials([usernamePassword(credentialsId:
env.DOCKERHUB_CREDENTIALS, usernameVariable: 'DOCKERHUB_USER',
passwordVariable: 'DOCKERHUB_PASS')]) {
                    sh 'echo $DOCKERHUB_PASS | docker login -u
$DOCKERHUB_USER --password-stdin'
                }
            }
        }
    }
    stage('Push Image to DockerHub') {
        steps {
            sh 'docker push $IMAGE_NAME:$IMAGE_TAG'
        }
    }
```

```
        stage('Logout') {
            steps {
                sh 'docker logout'
            }
        }
    }
}
```





Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

Script ?

```
        }
        stage('Push Image to DockerHub') {
            steps {
                sh 'docker push $IMAGE_NAME:$IMAGE_TAG'
            }
        }
        stage('Logout') {
            steps {
                sh 'docker logout'
            }
        }
    }
}
```

--Pipeline Excuted:

Not secure 18.191.111.134:8080/job/docker-image/19/console

Jenkins narendar log out

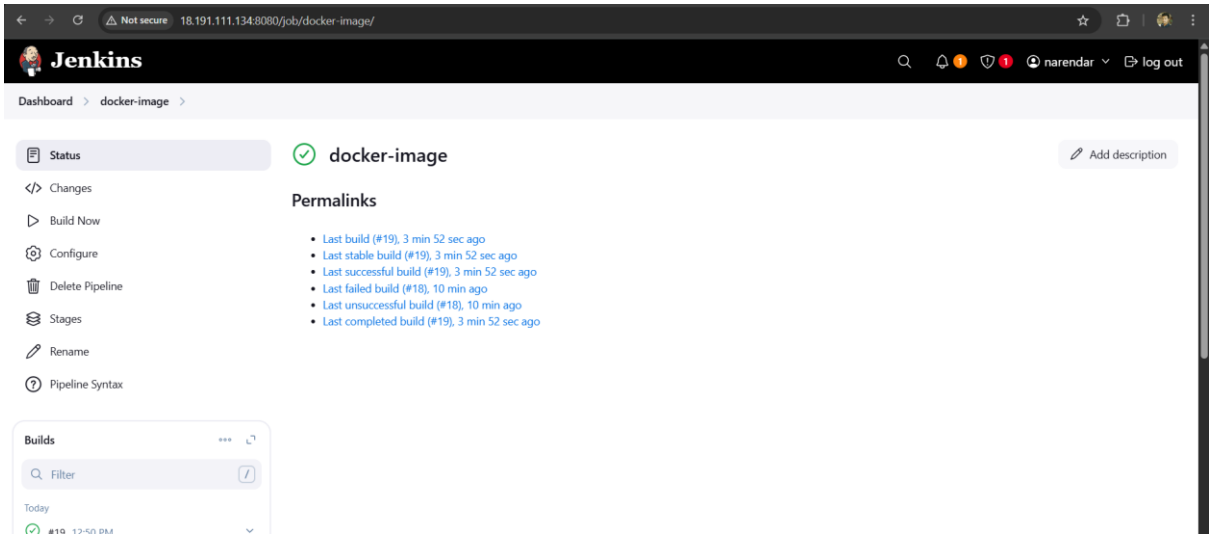
Dashboard > docker-image > #19

- Status
- Changes
- Console Output
- Edit Build Information
- Timings
- Pipeline Overview
- Pipeline Console
- Thread Dump
- Pause/resume
- Replay
- Pipeline Steps
- Workspaces
- Previous Build

Console Output Download Copy View as plain text

```
Started by user narendar
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/docker-image
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Pull NGINX Image)
[Pipeline] sh
+ docker pull nginx:latest
latest: Pulling from library/nginx
Digest: sha256:09369da6b10306312cd908661320086bf87fbae1b6b0c49a1f50ba531fef2eab
Status: Image is up to date for nginx:latest
docker.io/library/nginx:latest
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Tag Image)
[Pipeline] sh
+ docker tag nginx:latest narendar01/nginx:v1
[Pipeline] }
```

--Docker image created:



--pushed docker image to docker hub: