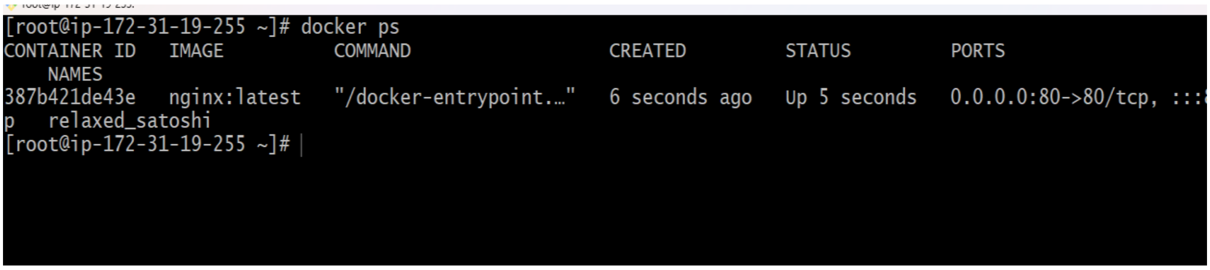
1. Create a image from running container.

- docker ps

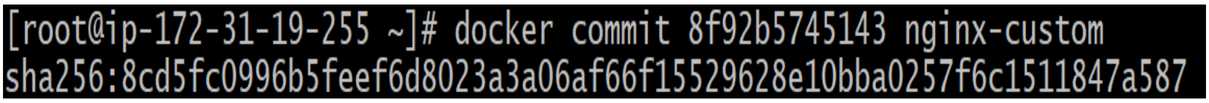


```
[root@ip-172-31-19-255 ~]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORTS
    NAMES
387b421de43e   nginx:latest   "/docker-entrypoint.…"   6 seconds ago   Up 5 seconds   0.0.0.0:80->80/tcp, :::
p   relaxed_satoshi
[root@ip-172-31-19-255 ~]#
```

A docker container is running

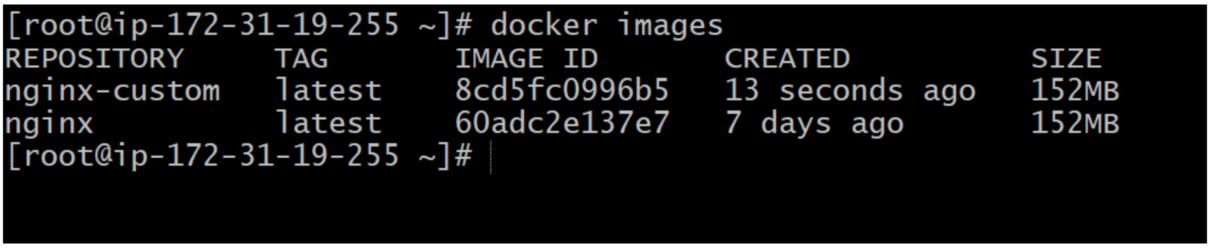
- docker commit commit_id name

it will create an image from running container



```
[root@ip-172-31-19-255 ~]# docker commit 8f92b5745143 nginx-custom
sha256:8cd5fc0996b5feef6d8023a3a06af66f15529628e10bba0257f6c1511847a587
```

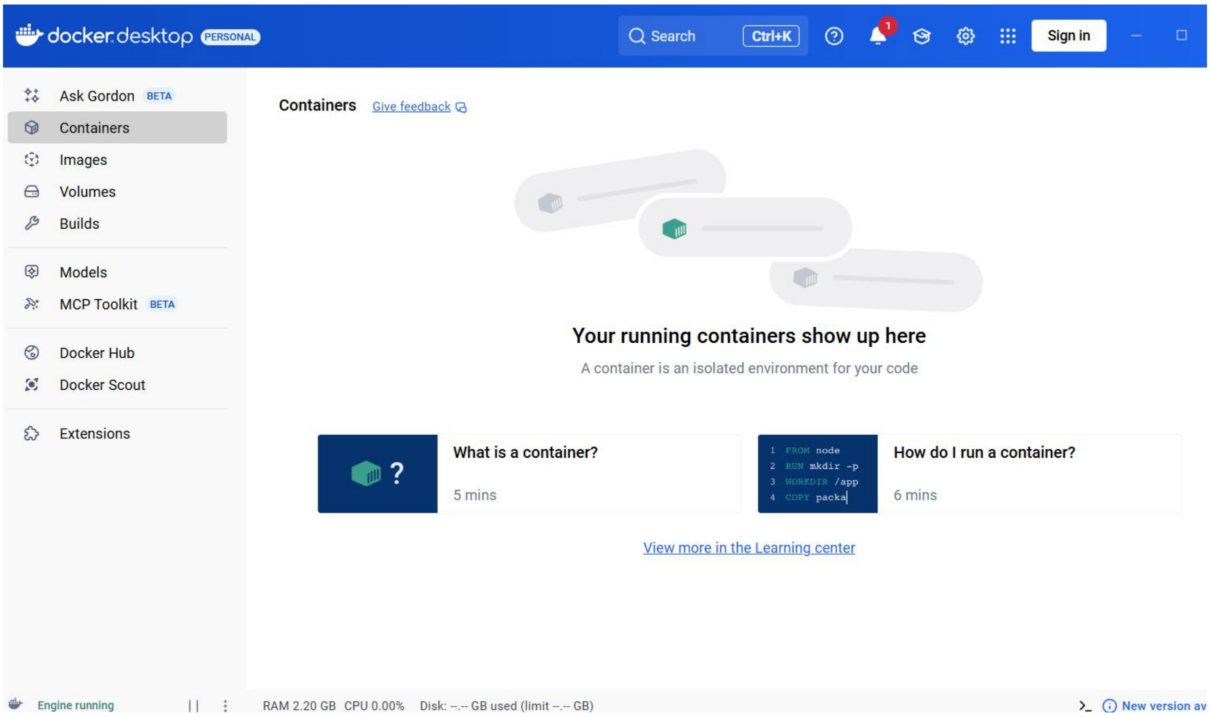
- docker images



```
[root@ip-172-31-19-255 ~]# docker images
REPOSITORY     TAG       IMAGE ID       CREATED          SIZE
nginx-custom   latest    8cd5fc0996b5   13 seconds ago   152MB
nginx          latest    60adc2e137e7   7 days ago       152MB
[root@ip-172-31-19-255 ~]#
```

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

Start docker desktop



Containers Give feedback

Your running containers show up here

A container is an isolated environment for your code

What is a container?

5 mins

```
1  FROM node
2  RUN mkdir -p
3  WORKDIR /app
4  COPY packa
```

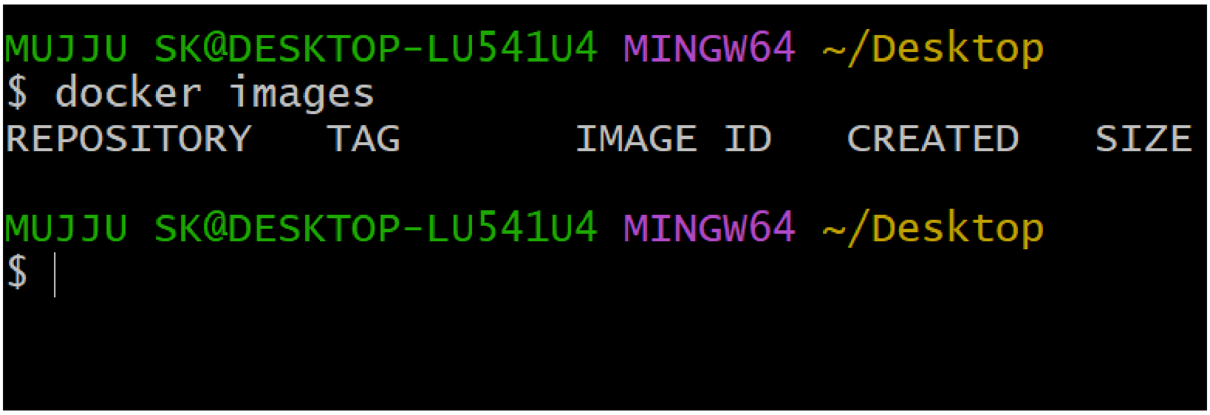
How do I run a container?

6 mins

View more in the Learning center

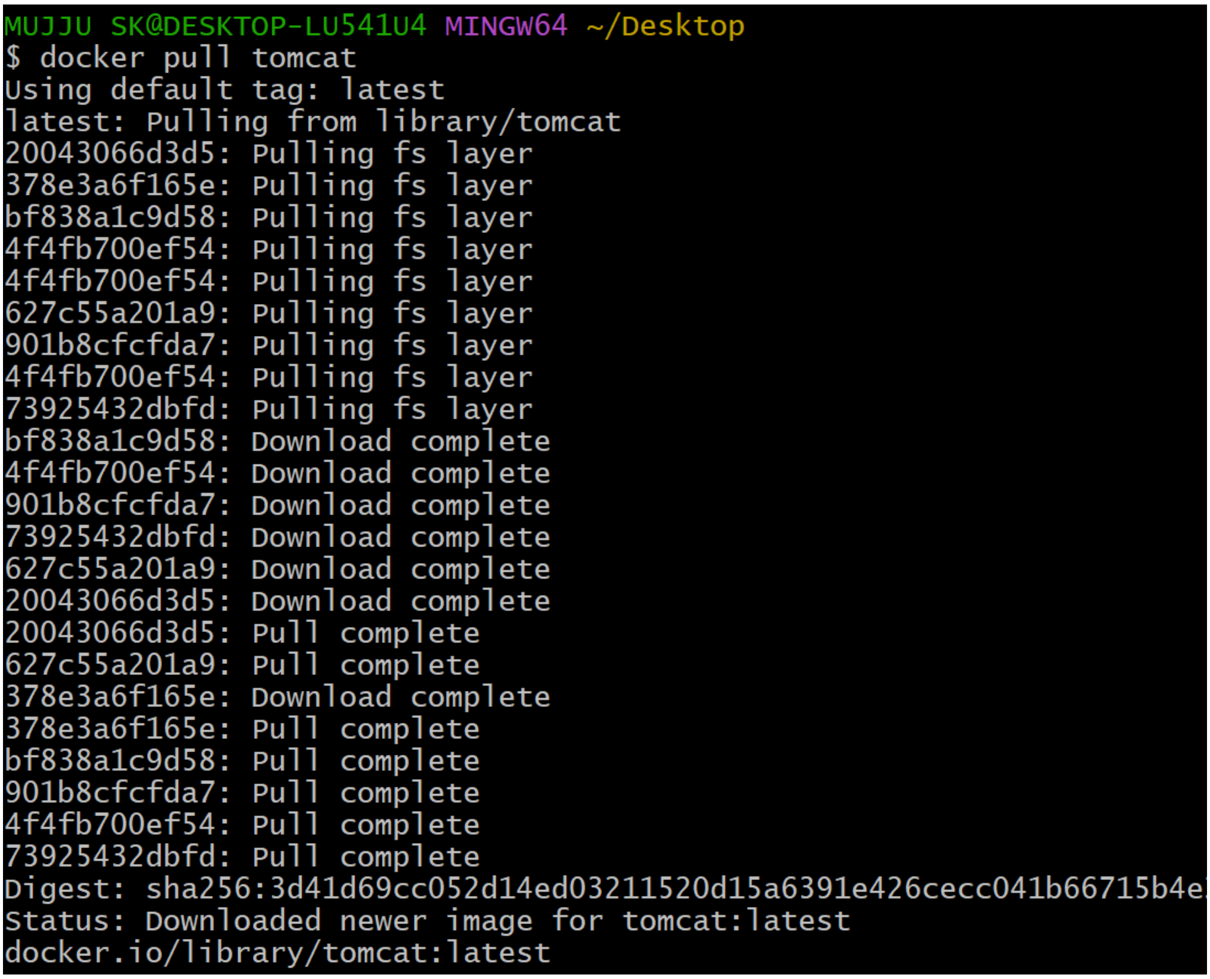
RAM 2.20 GB CPU 0.00% Disk: --.-- GB used (limit --.-- GB)

Open a git bash new terminal and pull the image.



```
MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Desktop
$ docker images
REPOSITORY    TAG         IMAGE ID    CREATED    SIZE

MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Desktop
$ 
```

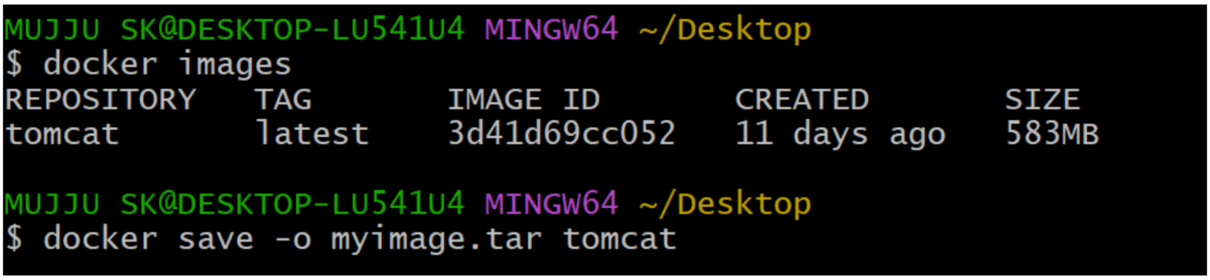
- docker pull tomcat



```
MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Desktop
$ docker pull tomcat
Using default tag: latest
latest: Pulling from library/tomcat
20043066d3d5: Pulling fs layer
378e3a6f165e: Pulling fs layer
bf838a1c9d58: Pulling fs layer
4f4fb700ef54: Pulling fs layer
4f4fb700ef54: Pulling fs layer
627c55a201a9: Pulling fs layer
901b8cfcfda7: Pulling fs layer
4f4fb700ef54: Pulling fs layer
73925432dbfd: Pulling fs layer
bf838a1c9d58: Download complete
4f4fb700ef54: Download complete
901b8cfcfda7: Download complete
73925432dbfd: Download complete
627c55a201a9: Download complete
20043066d3d5: Download complete
20043066d3d5: Pull complete
627c55a201a9: Pull complete
378e3a6f165e: Download complete
378e3a6f165e: Pull complete
bf838a1c9d58: Pull complete
901b8cfcfda7: Pull complete
4f4fb700ef54: Pull complete
73925432dbfd: Pull complete
Digest: sha256:3d41d69cc052d14ed03211520d15a6391e426cecc041b66715b4e
Status: Downloaded newer image for tomcat:latest
docker.io/library/tomcat:latest
```

Save the image by using

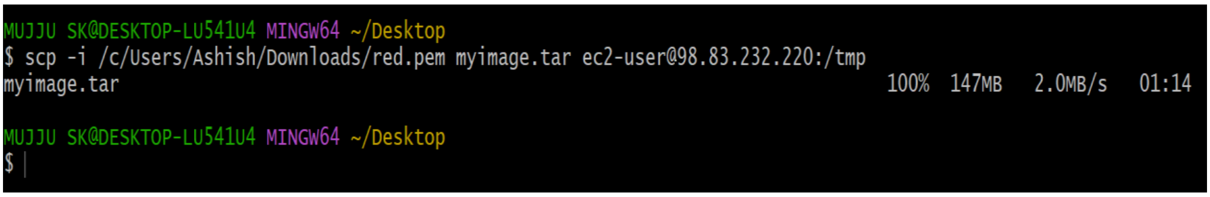
- docker save -o any_name.tar image_name



```
MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Desktop
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED       SIZE
tomcat        latest    3d41d69cc052   11 days ago   583MB

MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Desktop
$ docker save -o myimage.tar tomcat
```

Copy the image from local to ec2 instance

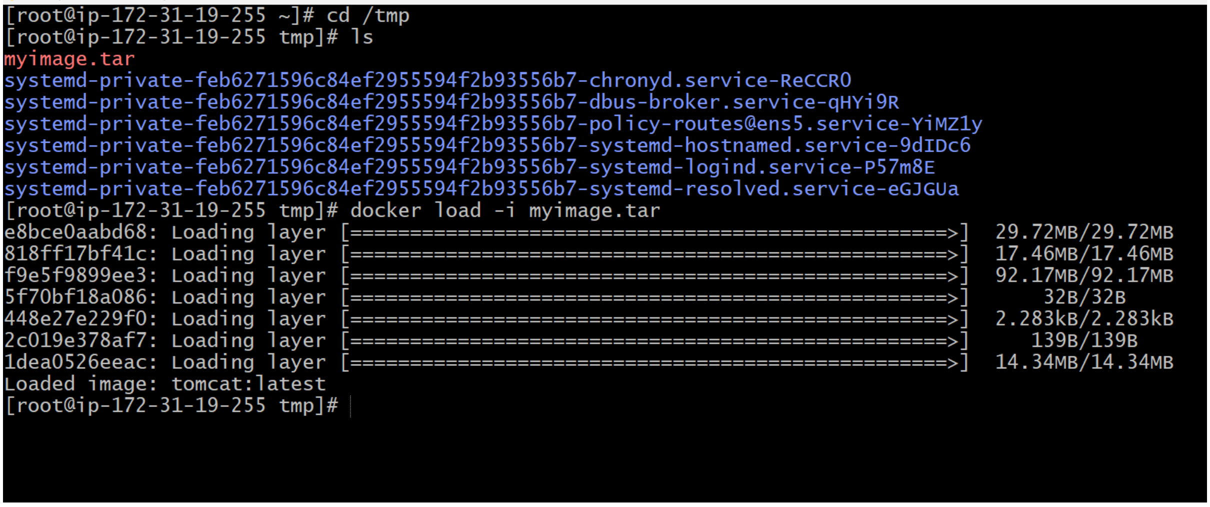
- scp -i /c/Users/Ashish/Downloads/red.pem myimage.tar ec2-user@98.83.232.220:/tmp



```
MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Desktop
$ scp -i /c/Users/Ashish/Downloads/red.pem myimage.tar ec2-user@98.83.232.220:/tmp
myimage.tar                                                                         100%  147MB   2.0MB/s   01:14

MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Desktop
$ 
```

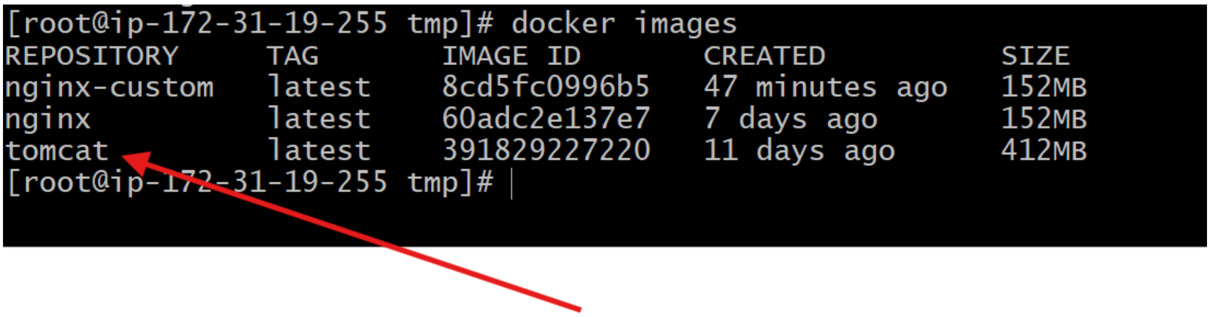
Go to your ec2 instance and change to tmp directory

- docker load -i myimage.tar



```
[root@ip-172-31-19-255 ~]# cd /tmp
[root@ip-172-31-19-255 tmp]# ls
myimage.tar
systemd-private-feb6271596c84ef2955594f2b93556b7-chronyd.service-ReCCR0
systemd-private-feb6271596c84ef2955594f2b93556b7-dbus-broker.service-qHYi9R
systemd-private-feb6271596c84ef2955594f2b93556b7-policy-routes@ens5.service-YiMZ1y
systemd-private-feb6271596c84ef2955594f2b93556b7-systemd-hostnamed.service-9dIDc6
systemd-private-feb6271596c84ef2955594f2b93556b7-systemd-logind.service-P57m8E
systemd-private-feb6271596c84ef2955594f2b93556b7-systemd-resolved.service-eGJGUa
[root@ip-172-31-19-255 tmp]# docker load -i myimage.tar
e8bce0aabd68: Loading layer [==================================================>]  29.72MB/29.72MB
818ff17bf41c: Loading layer [==================================================>]  17.46MB/17.46MB
f9e5f9899ee3: Loading layer [==================================================>]  92.17MB/92.17MB
5f70bf18a086: Loading layer [==================================================>]      32B/32B
448e27e229f0: Loading layer [==================================================>]  2.283kB/2.283kB
2c019e378af7: Loading layer [==================================================>]     139B/139B
1dea0526eeac: Loading layer [==================================================>]  14.34MB/14.34MB
Loaded image: tomcat:latest
[root@ip-172-31-19-255 tmp]# 
```

Tomcat has been copied

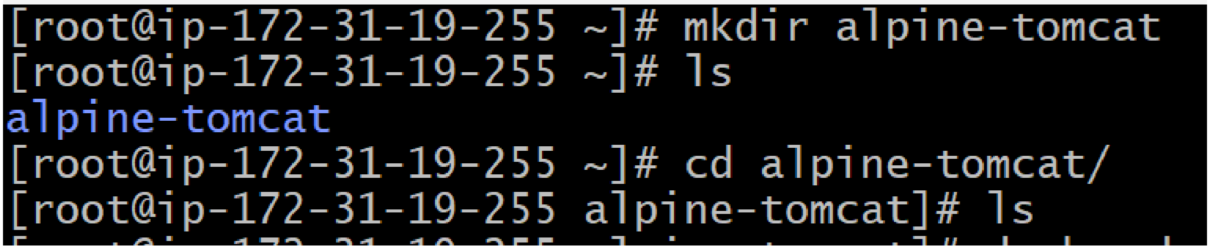


```
[root@ip-172-31-19-255 tmp]# docker images
REPOSITORY     TAG       IMAGE ID       CREATED          SIZE
nginx-custom   latest    8cd5fc0996b5   47 minutes ago   152MB
nginx          latest    60adc2e137e7   7 days ago       152MB
tomcat         latest    391829227220   11 days ago      412MB
[root@ip-172-31-19-255 tmp]# 
```

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

* mkdir alpine-tomcat

* cd alpine-tomcat



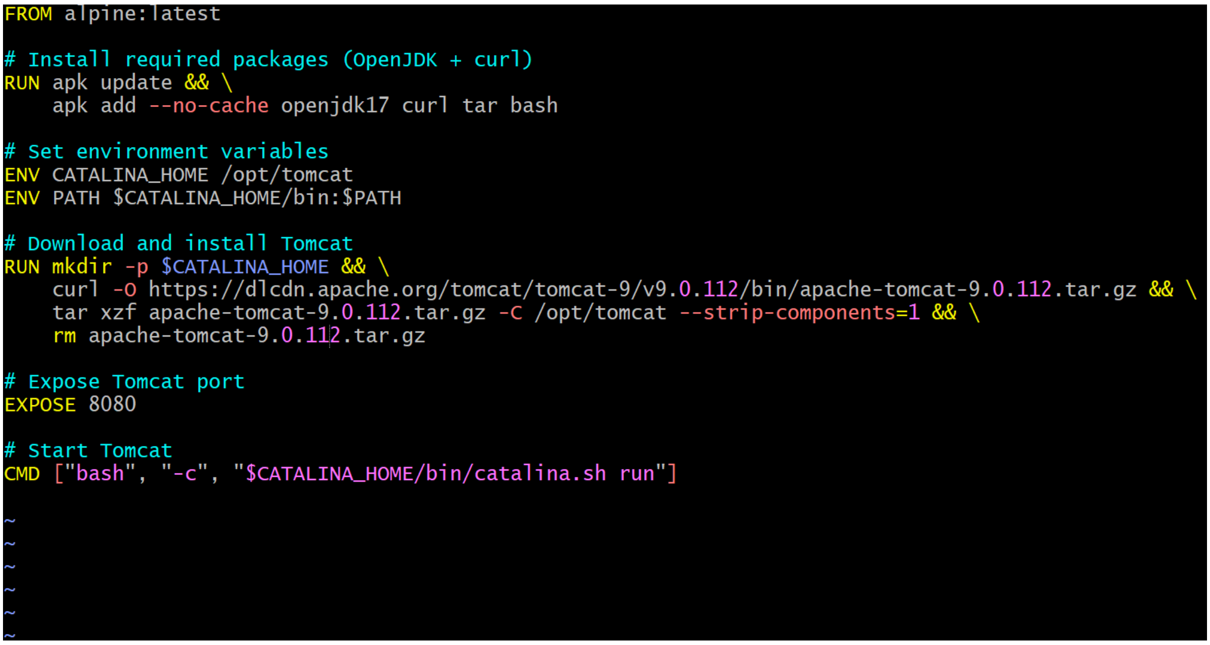
```
[root@ip-172-31-19-255 ~]# mkdir alpine-tomcat
[root@ip-172-31-19-255 ~]# ls
alpine-tomcat
[root@ip-172-31-19-255 ~]# cd alpine-tomcat/
[root@ip-172-31-19-255 alpine-tomcat]# ls
```

In that created directory create a file named as docker file and give this script.

- vi dockerfile

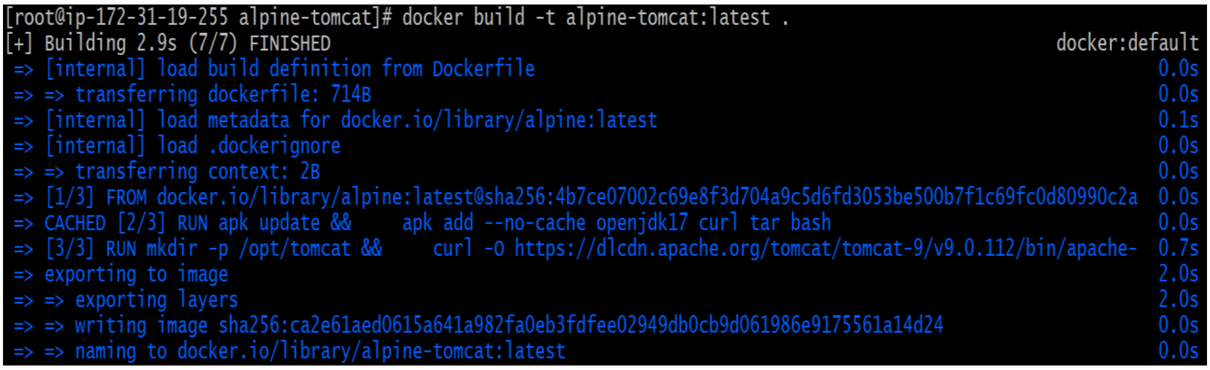
```
FROM alpine:latest

# Install required packages (OpenJDK + curl)
RUN apk update && \
    apk add --no-cache openjdk17 curl tar bash

# Set environment variables
ENV CATALINA_HOME /opt/tomcat
ENV PATH $CATALINA_HOME/bin:$PATH

# Download and install Tomcat
RUN mkdir -p $CATALINA_HOME && \
    curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.112/bin/apache-tomcat-9.0.112.tar.gz && \
    tar xzf apache-tomcat-9.0.112.tar.gz -C /opt/tomcat --strip-components=1 && \
    rm apache-tomcat-9.0.112.tar.gz

# Expose Tomcat port
EXPOSE 8080

# Start Tomcat
```

CMD ["bash", "-c", "$CATALINA_HOME/bin/catalina.sh run"]



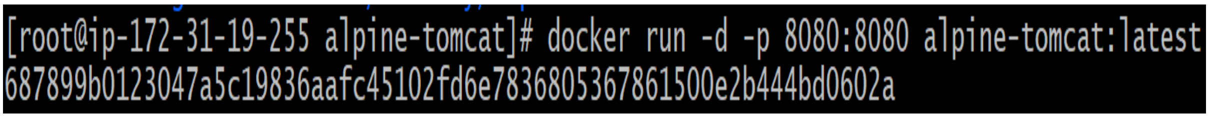
```
FROM alpine:latest

# Install required packages (OpenJDK + curl)
RUN apk update && \
    apk add --no-cache openjdk17 curl tar bash

# Set environment variables
ENV CATALINA_HOME /opt/tomcat
ENV PATH $CATALINA_HOME/bin:$PATH

# Download and install Tomcat
RUN mkdir -p $CATALINA_HOME && \
    curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.112/bin/apache-tomcat-9.0.112.tar.gz && \
    tar xzf apache-tomcat-9.0.112.tar.gz -C /opt/tomcat --strip-components=1 && \
    rm apache-tomcat-9.0.112.tar.gz

# Expose Tomcat port
EXPOSE 8080

# Start Tomcat
CMD ["bash", "-c", "$CATALINA_HOME/bin/catalina.sh run"]

~
~
~
~
~
~
```

- docker build -t alpine-tomcat:latest .



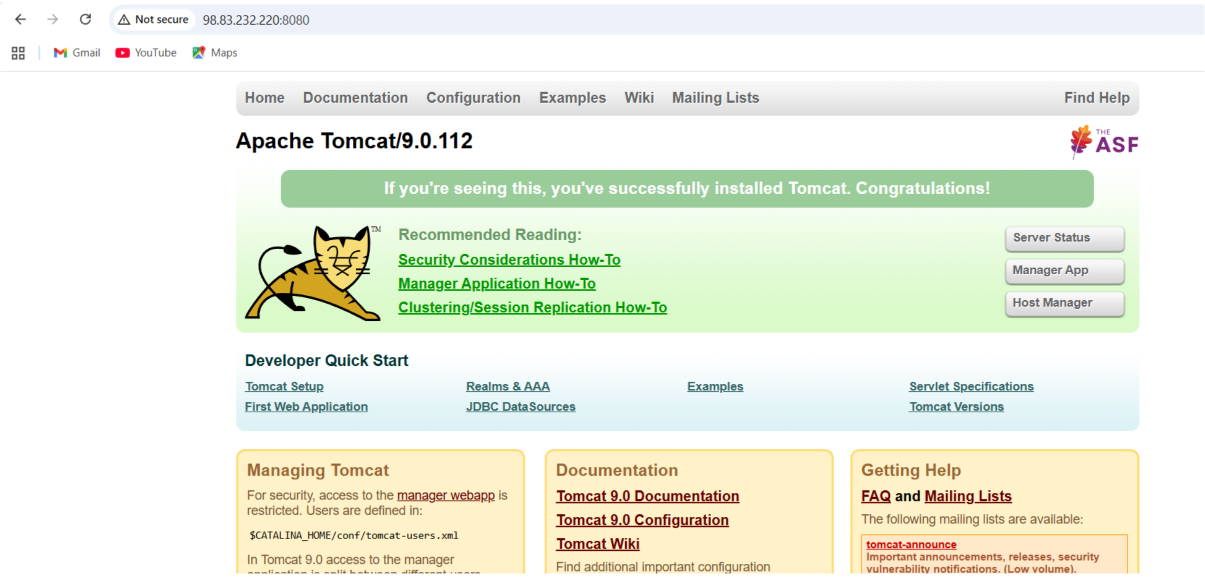
```
[root@ip-172-31-19-255 alpine-tomcat]# docker build -t alpine-tomcat:latest .
[+] Building 2.9s (7/7) FINISHED                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                          0.0s
 => => transferring dockerfile: 714B                                                                          0.0s
 => [internal] load metadata for docker.io/library/alpine:latest                                              0.1s
 => [internal] load .dockerignore                                                                             0.0s
 => => transferring context: 2B                                                                               0.0s
 => [1/3] FROM docker.io/library/alpine:latest@sha256:4b7ce07002c69e8f3d704a9c5d6fd3053be500b7f1c69fc0d80990c2a  0.0s
 => CACHED [2/3] RUN apk update &&     apk add --no-cache openjdk17 curl tar bash                             0.0s
 => [3/3] RUN mkdir -p /opt/tomcat &&     curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.112/bin/apache-  0.7s
 => exporting to image                                                                                        2.0s
 => => exporting layers                                                                                       2.0s
 => => writing image sha256:ca2e61aed0615a641a982fa0eb3fdfee02949db0cb9d061986e9175561a14d24                  0.0s
 => => naming to docker.io/library/alpine-tomcat:latest                                                       0.0s
```

- docker run -d -p 8080:8080 alpine-tomcat:latest



```
[root@ip-172-31-19-255 alpine-tomcat]# docker run -d -p 8080:8080 alpine-tomcat:latest
687899b0123047a5c19836aafc45102fd6e7836805367861500e2b444bd0602a
```

Browser your public_ip with port number 8080.



Not secure 98.83.232.220:8080

Gmail YouTube Maps

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/9.0.112

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To

Server Status
Manager App
Host Manager

Developer Quick Start

Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

$CATALINA_HOME/conf/tomcat-users.xml

In Tomcat 9.0 access to the manager

Documentation

Tomcat 9.0 Documentation
Tomcat 9.0 Configuration
Tomcat Wiki

Find additional important configuration

Getting Help

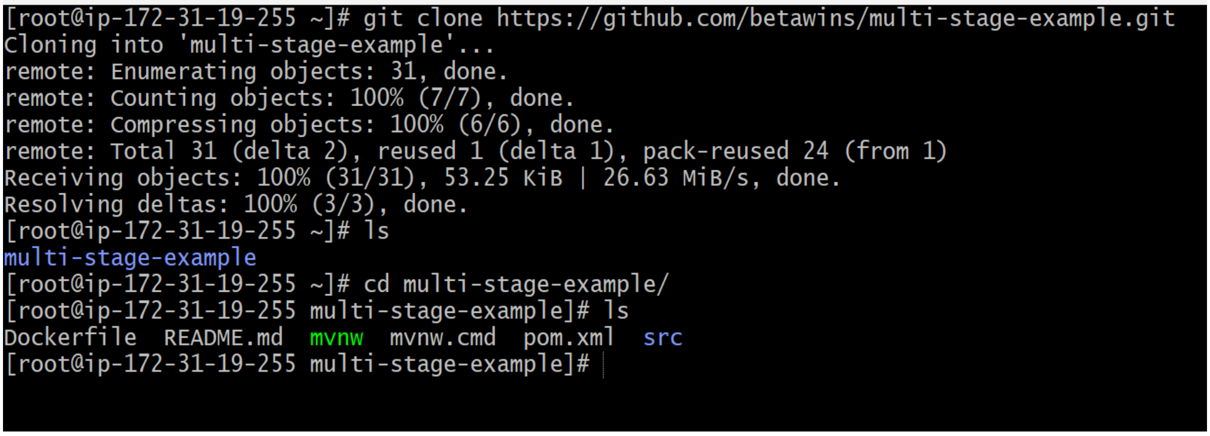
FAQ and Mailing Lists

The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

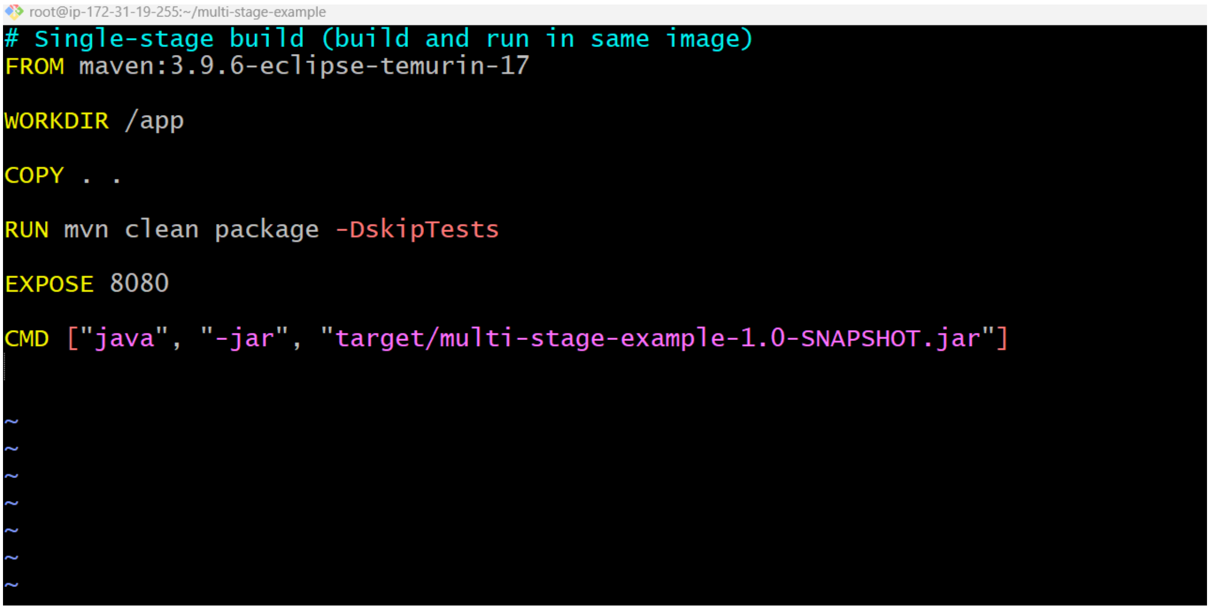
4. Create single stage and multi stage docker file using this source code: https://github.com/betawins/multi-stage-example.git

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



```
[root@ip-172-31-19-255 ~]# 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 | 26.63 MiB/s, done.
Resolving deltas: 100% (3/3), done.
[root@ip-172-31-19-255 ~]# ls
multi-stage-example
[root@ip-172-31-19-255 ~]# cd multi-stage-example/
[root@ip-172-31-19-255 multi-stage-example]# ls
Dockerfile  README.md  mvnw  mvnw.cmd  pom.xml  src
[root@ip-172-31-19-255 multi-stage-example]#
```

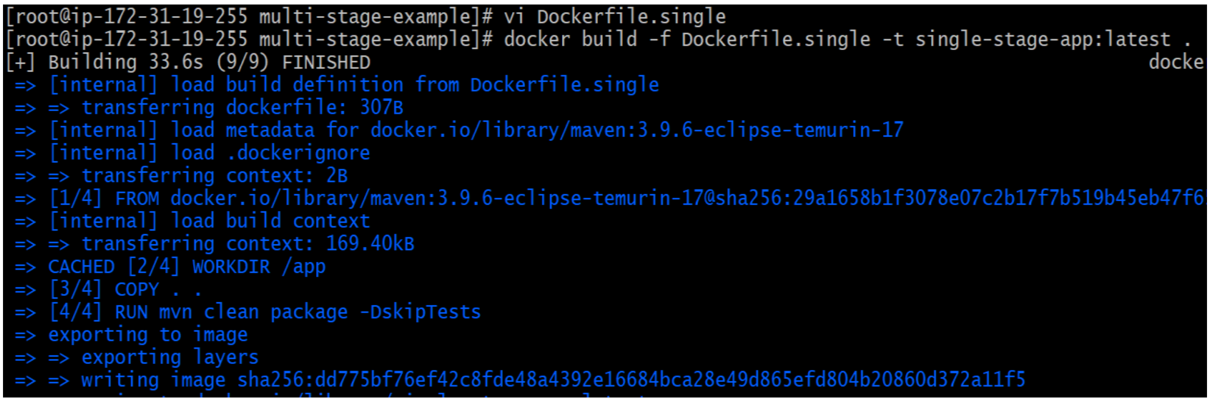
- cd multi-stage-example
- vi Dockerfile.single



root@ip-172-31-19-255:~/multi-stage-example

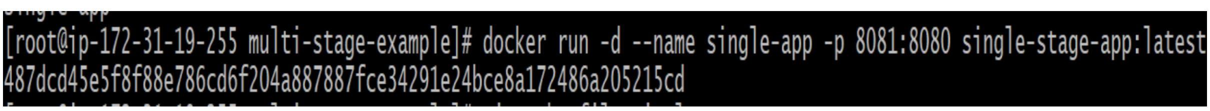
```
# Single-stage build (build and run in same image)
FROM maven:3.9.6-eclipse-temurin-17

WORKDIR /app

COPY . .

RUN mvn clean package -DskipTests

EXPOSE 8080

CMD ["java", "-jar", "target/multi-stage-example-1.0-SNAPSHOT.jar"]


~
~
~
~
~
~
~
```

- docker build -f Dockerfile.single -t single-stage-app:latest



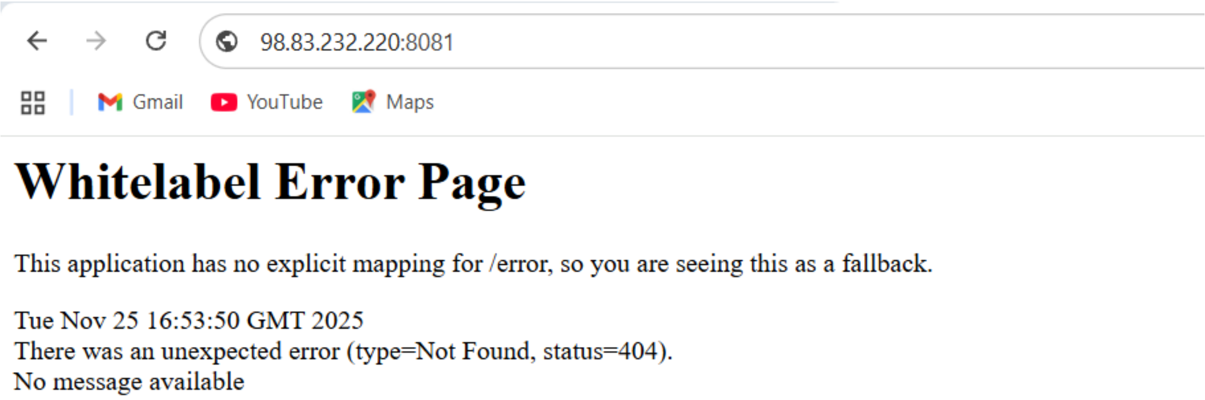
```
[root@ip-172-31-19-255 multi-stage-example]# vi Dockerfile.single
[root@ip-172-31-19-255 multi-stage-example]# docker build -f Dockerfile.single -t single-stage-app:latest .
[+] Building 33.6s (9/9) FINISHED                                                                     docke
 => [internal] load build definition from Dockerfile.single
 => => transferring dockerfile: 307B
 => [internal] load metadata for docker.io/library/maven:3.9.6-eclipse-temurin-17
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/4] FROM docker.io/library/maven:3.9.6-eclipse-temurin-17@sha256:29a1658b1f3078e07c2b17f7b519b45eb47f6
 => [internal] load build context
 => => transferring context: 169.40kB
 => CACHED [2/4] WORKDIR /app
 => [3/4] COPY . .
 => [4/4] RUN mvn clean package -DskipTests
 => exporting to image
 => => exporting layers
 => => writing image sha256:dd775bf76ef42c8fde48a4392e16684bca28e49d865efd804b20860d372a11f5
```

- docker run -d --name single-app -p 8081:8080 single-stage-app:latest



```
[root@ip-172-31-19-255 multi-stage-example]# docker run -d --name single-app -p 8081:8080 single-stage-app:latest
487dcd45e5f8f88e786cd6f204a887887fce34291e24bce8a172486a205215cd
```

Search in browser with the public_ip and port number 8081.



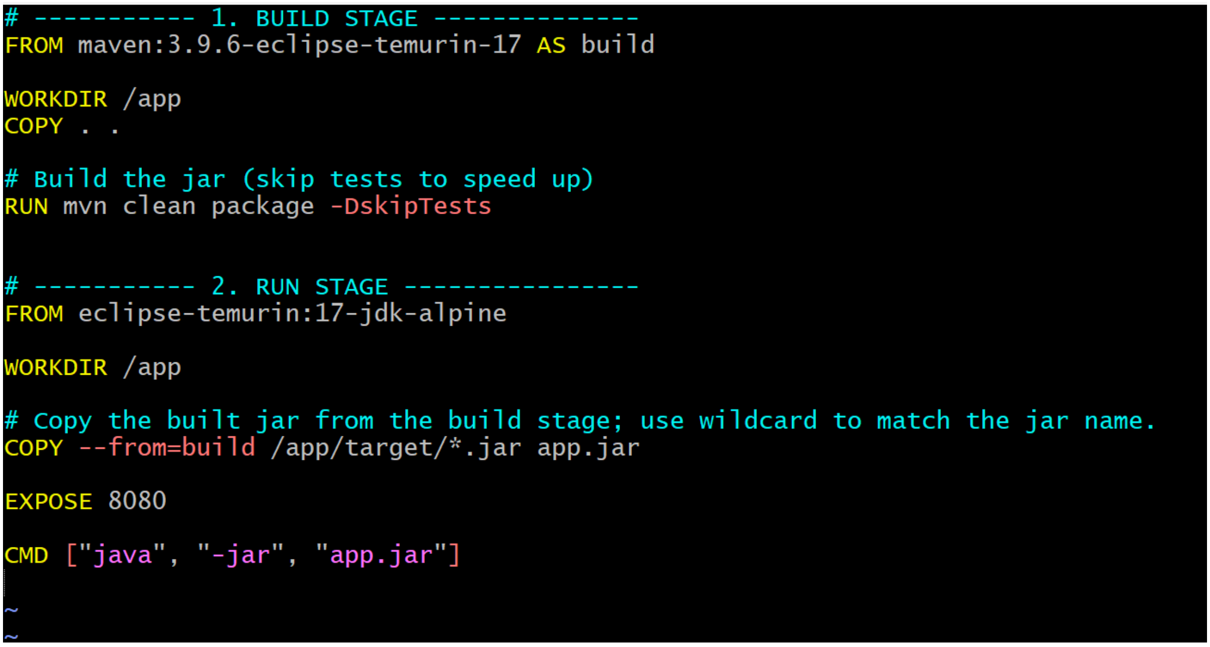
Whitelabel Error Page

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

Tue Nov 25 16:53:50 GMT 2025
There was an unexpected error (type=Not Found, status=404).
No message available

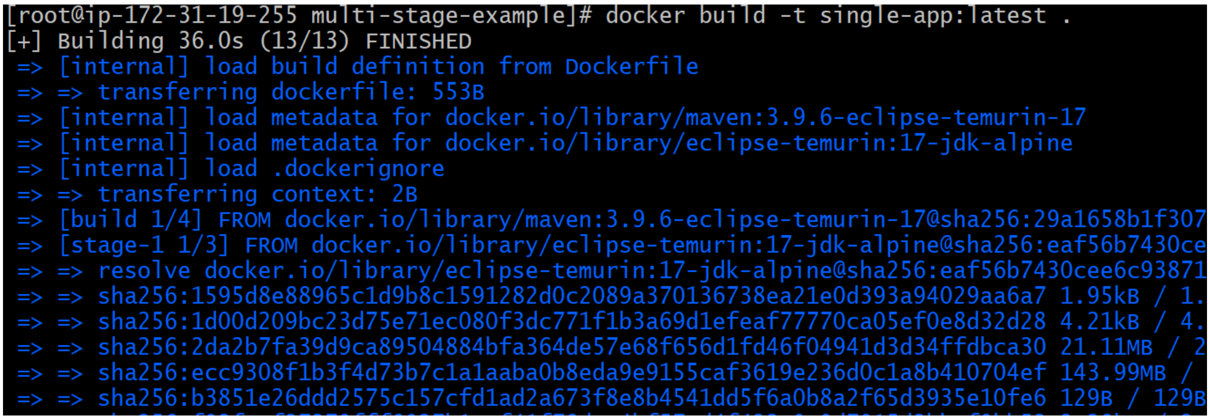
Multi-stage:

- vi Dockerfile add this script



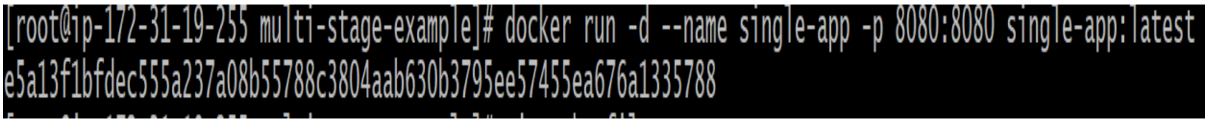
```
# ----------- 1. BUILD STAGE --------------
FROM maven:3.9.6-eclipse-temurin-17 AS build

WORKDIR /app
COPY . .

# Build the jar (skip tests to speed up)
RUN mvn clean package -DskipTests


# ----------- 2. RUN STAGE ----------------
FROM eclipse-temurin:17-jdk-alpine

WORKDIR /app

# Copy the built jar from the build stage; use wildcard to match the jar name.
COPY --from=build /app/target/*.jar app.jar

EXPOSE 8080

CMD ["java", "-jar", "app.jar"]

~
~
```

- docker build -t single-app:latest .



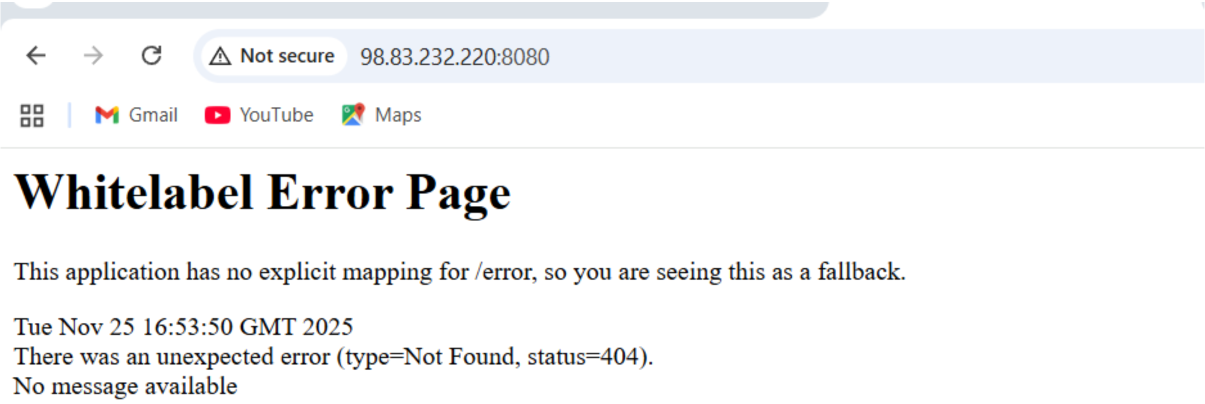
```
[root@ip-172-31-19-255 multi-stage-example]# docker build -t single-app:latest .
[+] Building 36.0s (13/13) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 553B
 => [internal] load metadata for docker.io/library/maven:3.9.6-eclipse-temurin-17
 => [internal] load metadata for docker.io/library/eclipse-temurin:17-jdk-alpine
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [build 1/4] FROM docker.io/library/maven:3.9.6-eclipse-temurin-17@sha256:29a1658b1f307
 => [stage-1 1/3] FROM docker.io/library/eclipse-temurin:17-jdk-alpine@sha256:eaf56b7430ce
 => => resolve docker.io/library/eclipse-temurin:17-jdk-alpine@sha256:eaf56b7430cee6c93871
 => => sha256:1595d8e88965c1d9b8c1591282d0c2089a370136738ea21e0d393a94029aa6a7 1.95kB / 1.
 => => sha256:1d00d209bc23d75e71ec080f3dc771f1b3a69d1efeaf77770ca05ef0e8d32d28 4.21kB / 4.
 => => sha256:2da2b7fa39d9ca89504884bfa364de57e68f656d1fd46f04941d3d34ffdbca30 21.11MB / 2
 => => sha256:ecc9308f1b3f4d73b7c1a1aaba0b8eda9e9155caf3619e236d0c1a8b410704ef 143.99MB /
 => => sha256:b3851e26ddd2575c157cfd1ad2a673f8e8b4541dd5f6a0b8a2f65d3935e10fe6 129B / 129B
```

- docker run -d --name single-app -p 8080:8080 single-app:latest



```
[root@ip-172-31-19-255 multi-stage-example]# docker run -d --name single-app -p 8080:8080 single-app:latest
e5a13f1bfdec555a237a08b55788c3804aab630b3795ee57455ea676a1335788
```

Search in browser with the public_ip and port number 8080.



Whitelabel Error Page

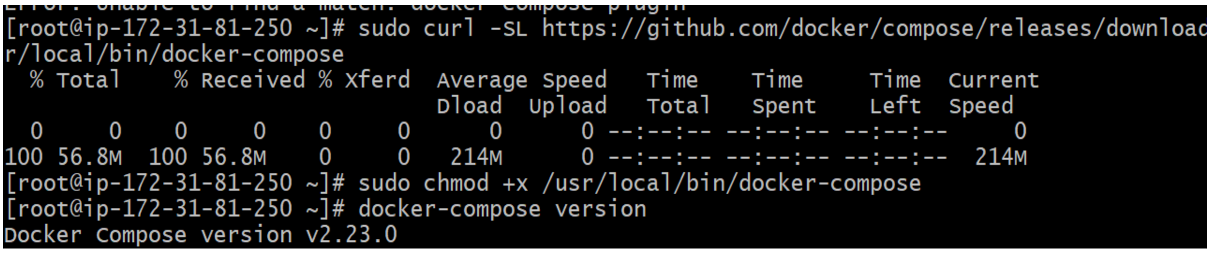
This application has no explicit mapping for /error, so you are seeing this as a fallback.

Tue Nov 25 16:53:50 GMT 2025
There was an unexpected error (type=Not Found, status=404).
No message available

5. Install docker compose and execute sample application.

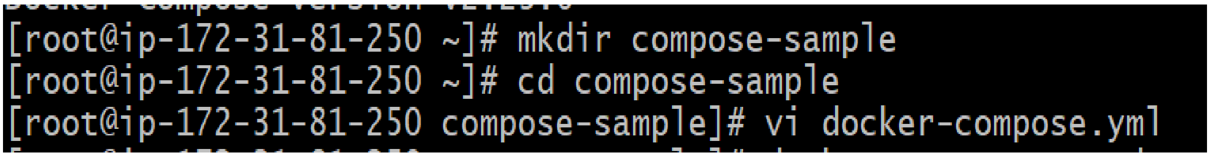
- curl -SL https://github.com/docker/compose/releases/download/v2.23.0/docker-compose-linux-x86_64 -o /usr/local/bin/docker-compose

- chmod +x /usr/local/bin/docker-compose



```
Error: Unable to find a match: docker-compose-plugin
[root@ip-172-31-81-250 ~]# sudo curl -SL https://github.com/docker/compose/releases/download
r/local/bin/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
100 56.8M  100 56.8M    0     0   214M      0 --:--:-- --:--:-- --:--:--  214M
[root@ip-172-31-81-250 ~]# sudo chmod +x /usr/local/bin/docker-compose
[root@ip-172-31-81-250 ~]# docker-compose version
Docker Compose version v2.23.0
```

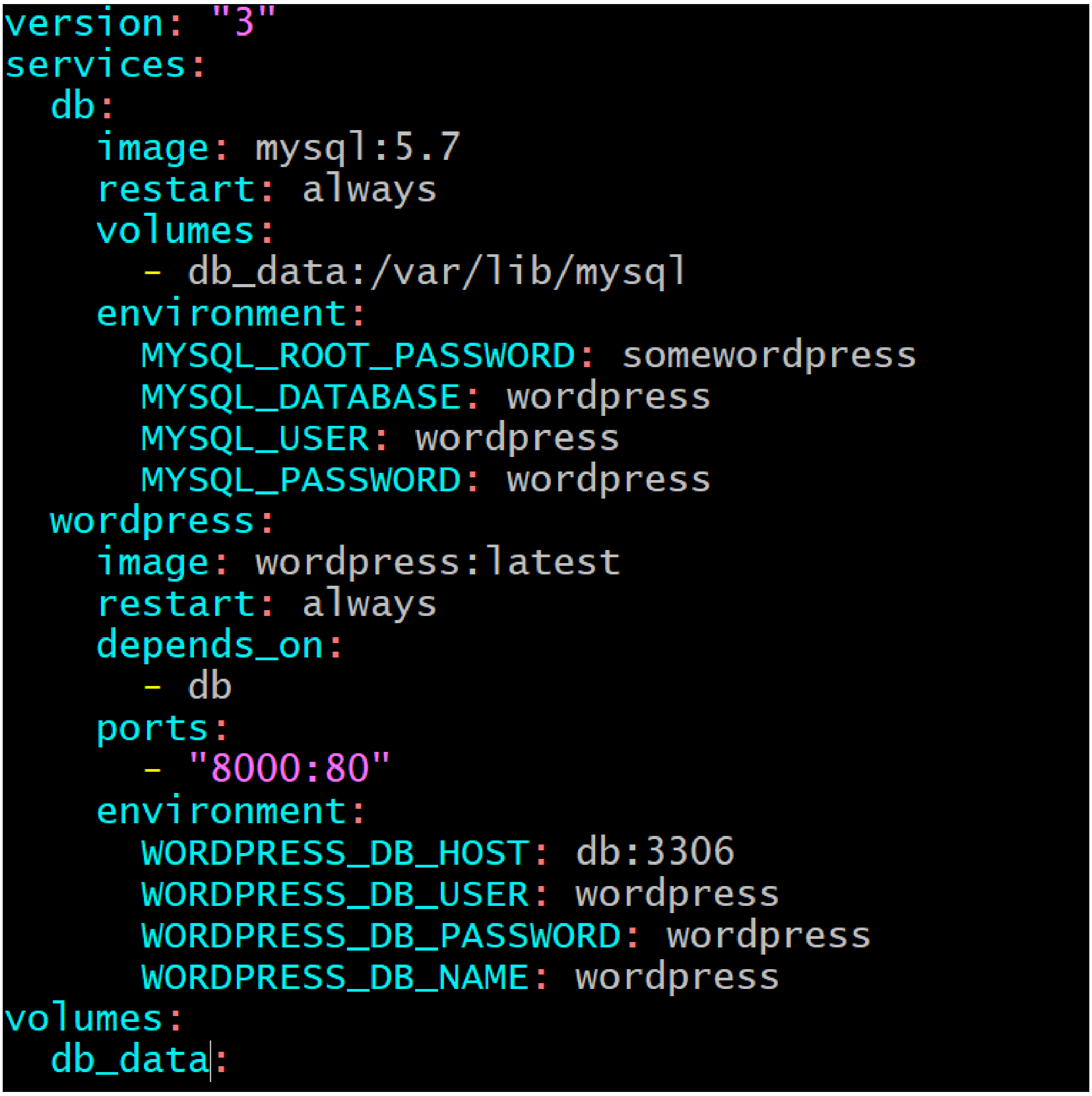
- mkdir compose-sample
- cd compose-sample



```
Docker Compose version v2.23.0
[root@ip-172-31-81-250 ~]# mkdir compose-sample
[root@ip-172-31-81-250 ~]# cd compose-sample
[root@ip-172-31-81-250 compose-sample]# vi docker-compose.yml
```

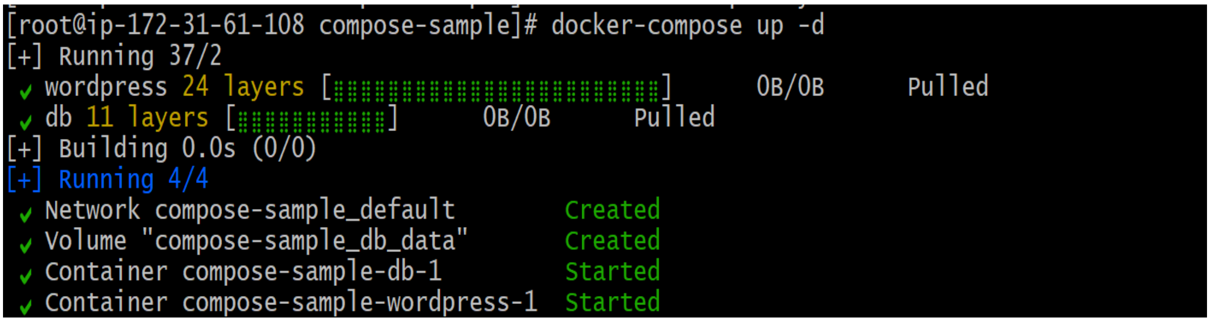
Create a file docker-compose.yml

```
version: "3"services:
  db:
    image: mysql:5.7
    restart: always
    volumes:
      - db_data:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: somewordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress  wordpress:
    image: wordpress:latest
    restart: always
    depends_on:
      - db
    ports:
      - "8000:80"
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpressvolumes:
  db_data:
```



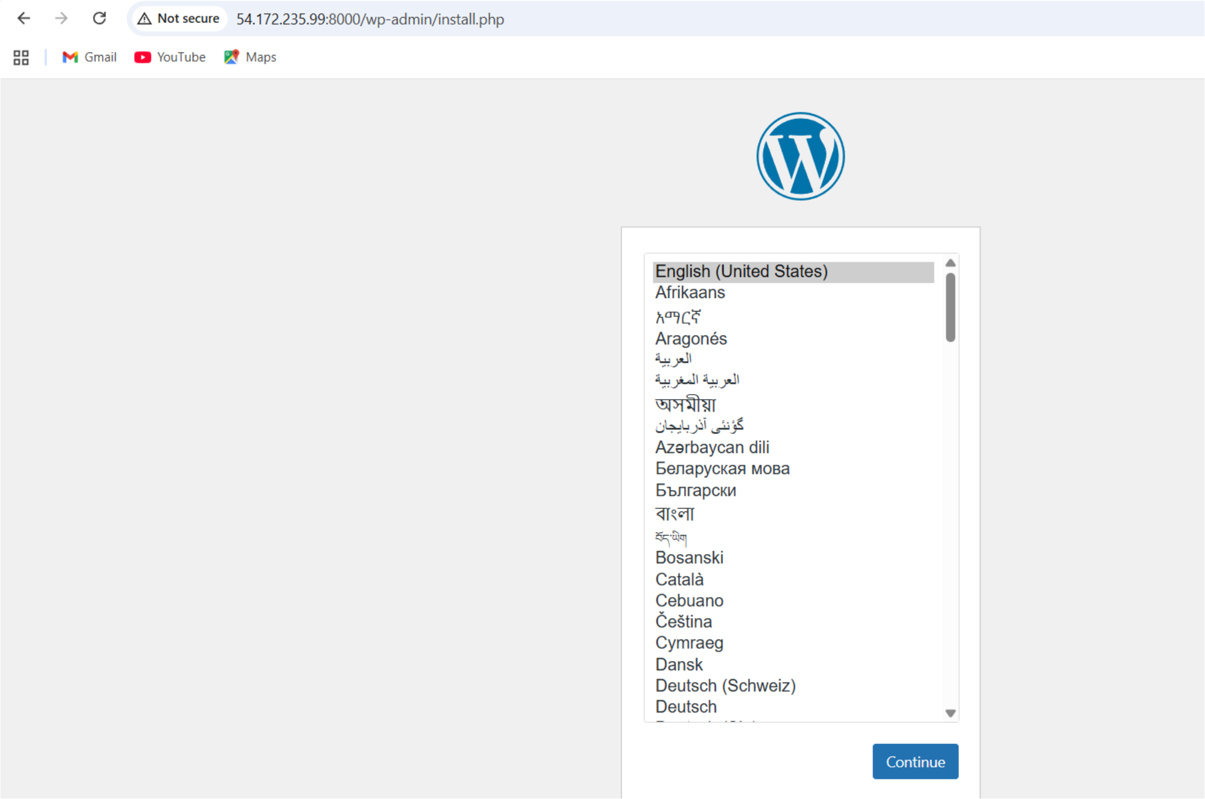
```
version: "3"
services:
  db:
    image: mysql:5.7
    restart: always
    volumes:
      - db_data:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: somewordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress
  wordpress:
    image: wordpress:latest
    restart: always
    depends_on:
      - db
    ports:
      - "8000:80"
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
volumes:
  db_data:
```

- docker-compose up -d



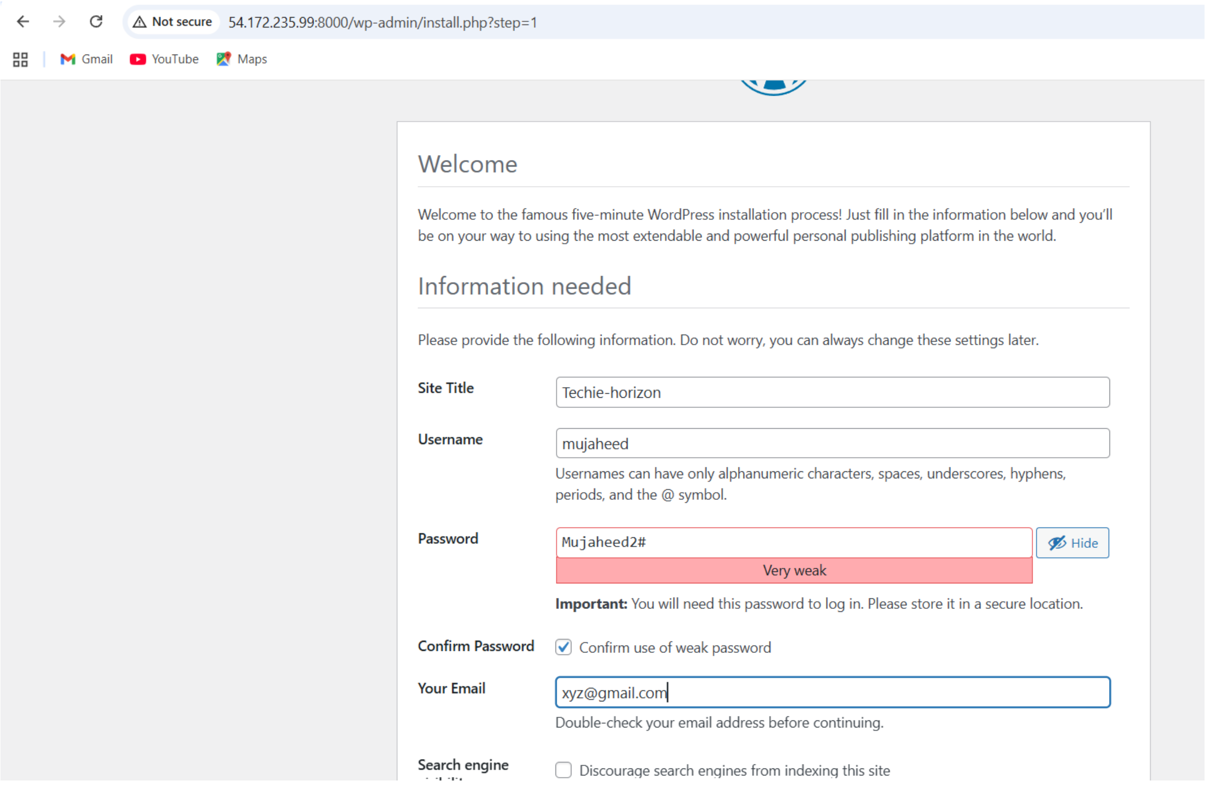
```
[root@ip-172-31-61-108 compose-sample]# docker-compose up -d
[+] Running 37/2
 ✓ wordpress 24 layers [⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿]      0B/0B      Pulled
 ✓ db 11 layers [⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿]      0B/0B      Pulled
[+] Building 0.0s (0/0)
[+] Running 4/4
 ✓ Network compose-sample_default         Created
 ✓ Volume "compose-sample_db_data"        Created
 ✓ Container compose-sample-db-1          Started
 ✓ Container compose-sample-wordpress-1  Started
```

Go to browser and search with public_ip and port number 8000.



⚠ Not secure 54.172.235.99:8000/wp-admin/install.php

English (United States)
Afrikaans
አማርኛ
Aragonés
العربية
العربية المغربية
অসমীয়া
گؤنئی آذربایجان
Azərbaycan dili
Беларуская мова
Български
বাংলা
བོད་ཡིག
Bosanski
Català
Cebuano
Čeština
Cymraeg
Dansk
Deutsch (Schweiz)
Deutsch



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title: Techie-horizon

Username: mujaheed

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password: Mujaheed2# Hide

Very weak

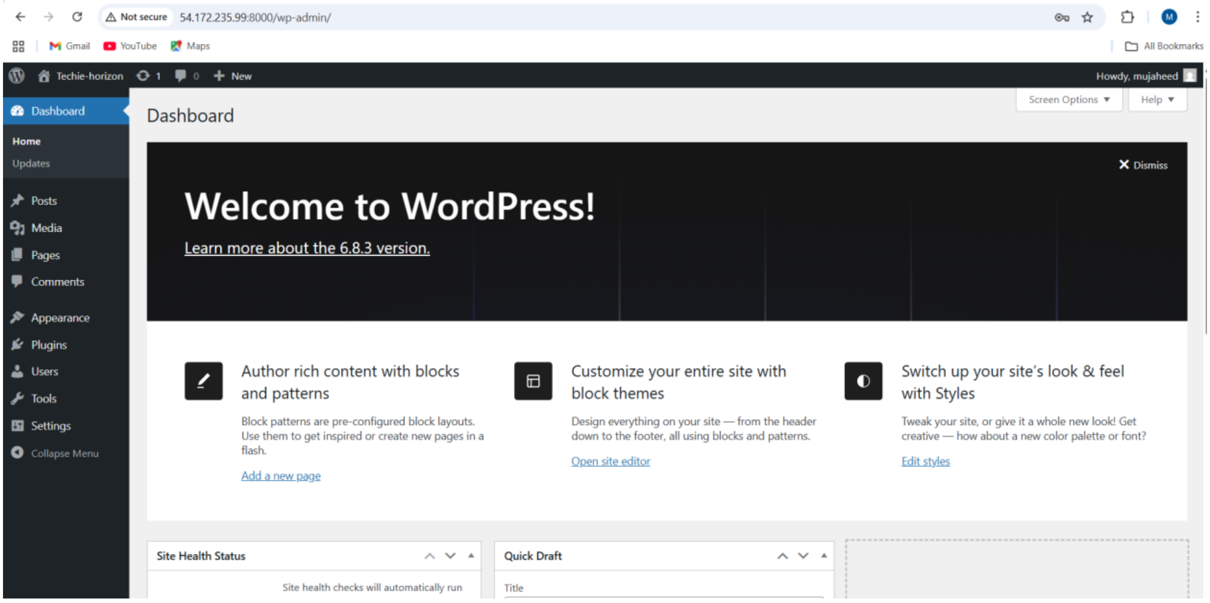
Important: You will need this password to log in. Please store it in a secure location.

Confirm Password: ☑ Confirm use of weak password

Your Email: xyz@gmail.com

Double-check your email address before continuing.

Search engine: □ Discourage search engines from indexing this site



Dashboard

Welcome to WordPress!

Learn more about the 6.8.3 version.

Author rich content with blocks and patterns

Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.

Add a new page

Customize your entire site with block themes

Design everything on your site — from the header down to the footer, all using blocks and patterns.

Open site editor

Switch up your site's look & feel with Styles

Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?

Edit styles

Site Health Status

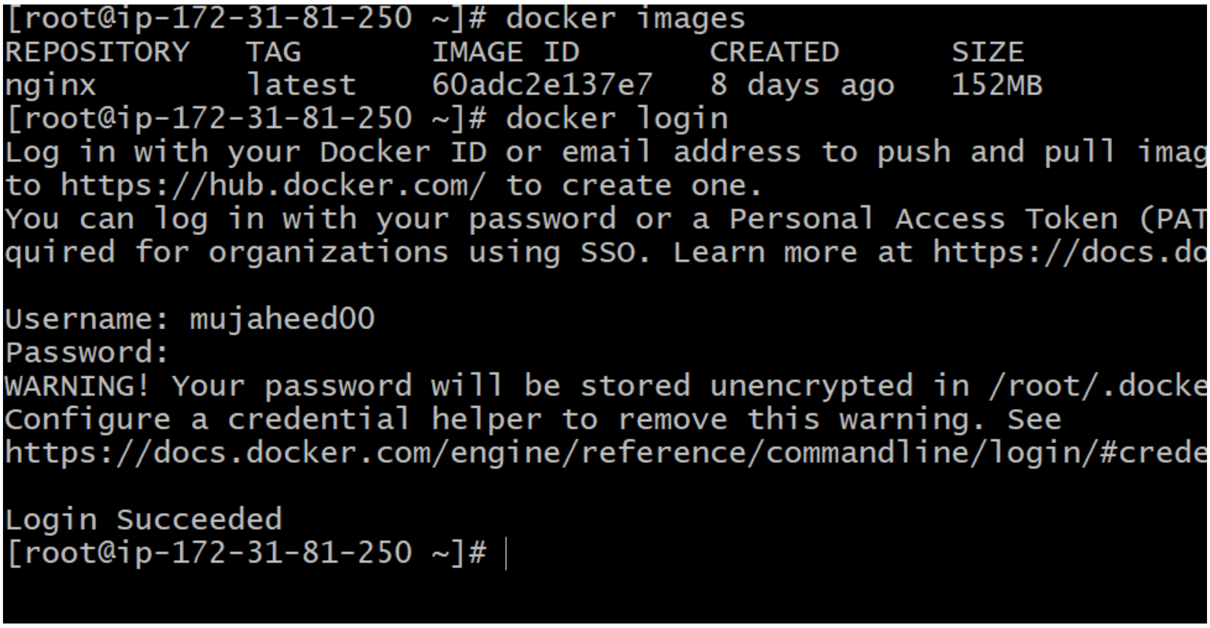
Site health checks will automatically run

Quick Draft

Title

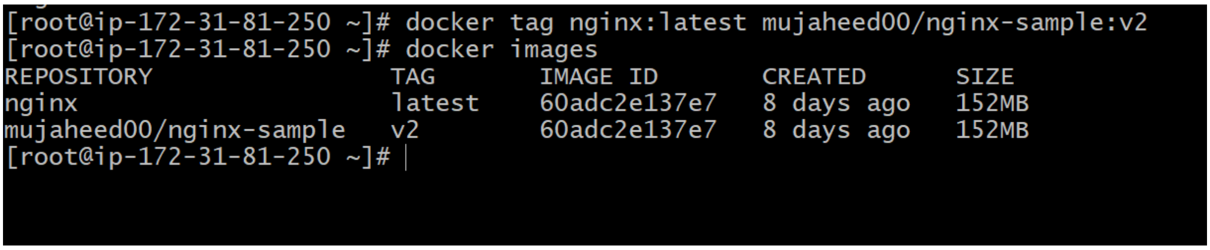
6. Implement solution to scan images when pushed to docker registry.

* docker login and provide username and password.



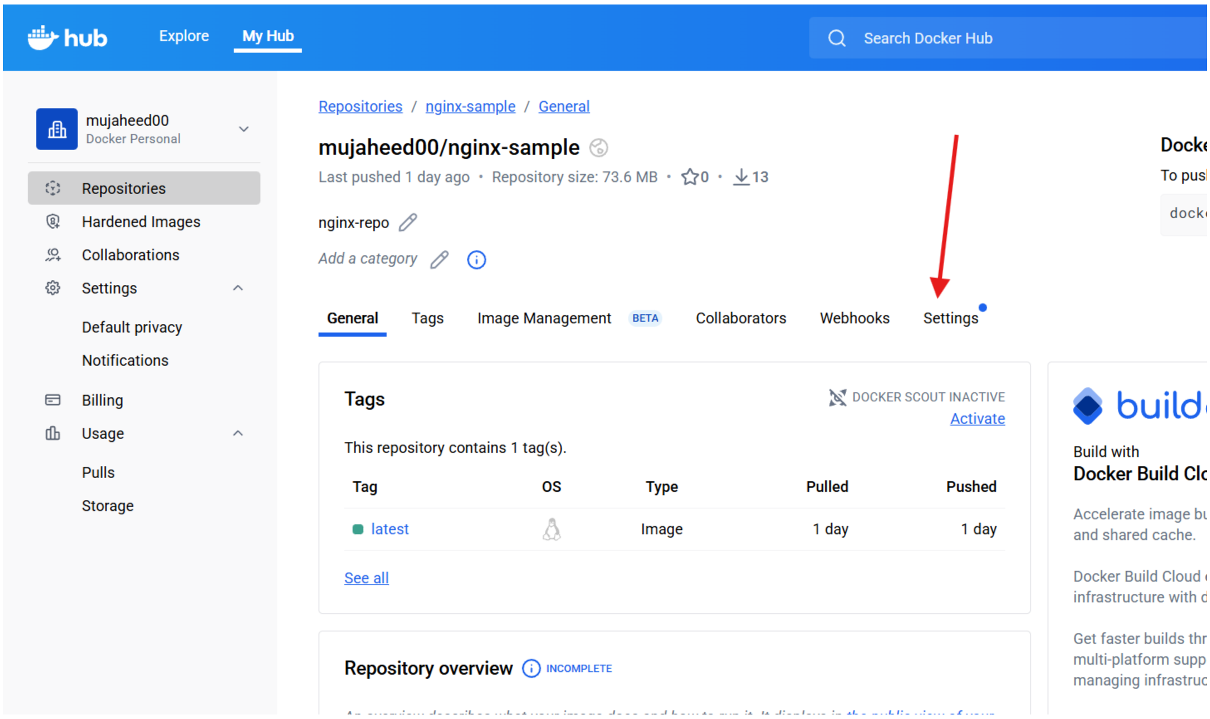
```
[root@ip-172-31-81-250 ~]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED       SIZE
nginx         latest    60adc2e137e7   8 days ago    152MB
[root@ip-172-31-81-250 ~]# docker login
Log in with your Docker ID or email address to push and pull imag
to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT
quired for organizations using SSO. Learn more at https://docs.do

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

Login Succeeded
[root@ip-172-31-81-250 ~]#
```

- docker tag image account_name/repo_name:tag
- docker tag nginx:latest mujaheed00/nginx-sample:v2



```
[root@ip-172-31-81-250 ~]# docker tag nginx:latest mujaheed00/nginx-sample:v2
[root@ip-172-31-81-250 ~]# docker images
REPOSITORY                TAG       IMAGE ID       CREATED       SIZE
nginx                     latest    60adc2e137e7   8 days ago    152MB
mujaheed00/nginx-sample   v2        60adc2e137e7   8 days ago    152MB
[root@ip-172-31-81-250 ~]#
```

Before pushing image to docker hub it need to be scan for that go to your repository and select settings.



Tag	OS	Type	Pulled	Pushed
latest		Image	1 day	1 day

Select docker scout image analysis and click on save.

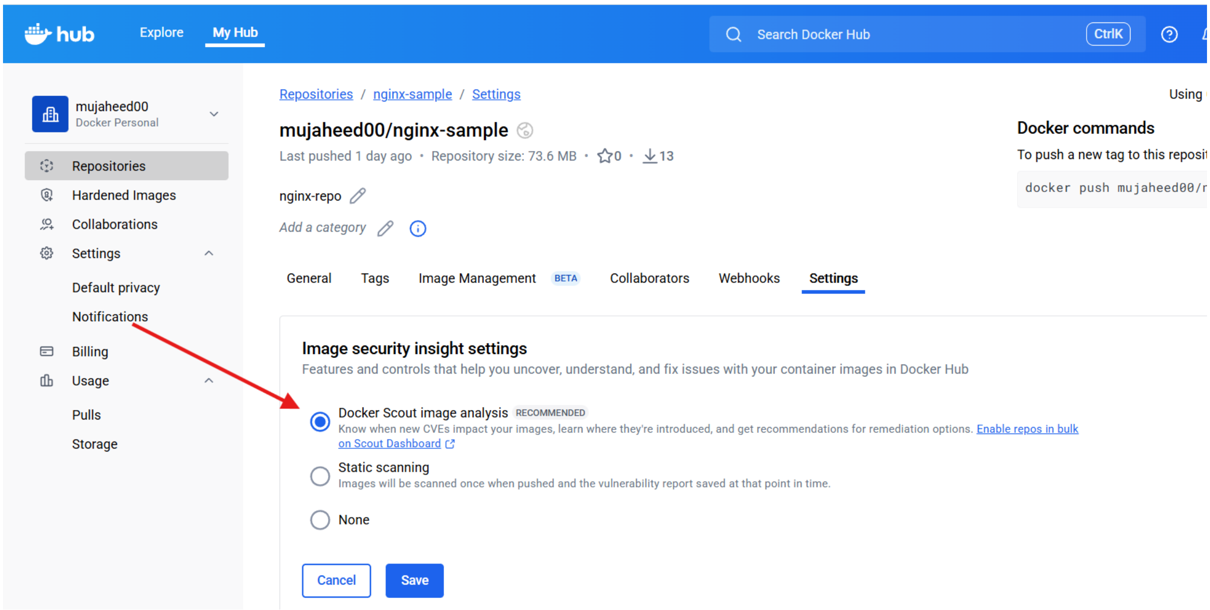


Image security insight settings

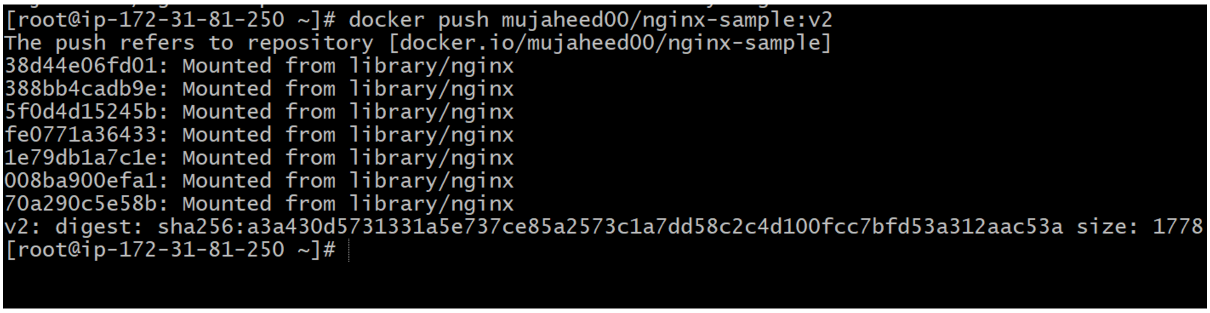
Features and controls that help you uncover, understand, and fix issues with your container images in Docker Hub

● Docker Scout image analysis RECOMMENDED
Know when new CVEs impact your images, learn where they're introduced, and get recommendations for remediation options. Enable repos in bulk on Scout Dashboard

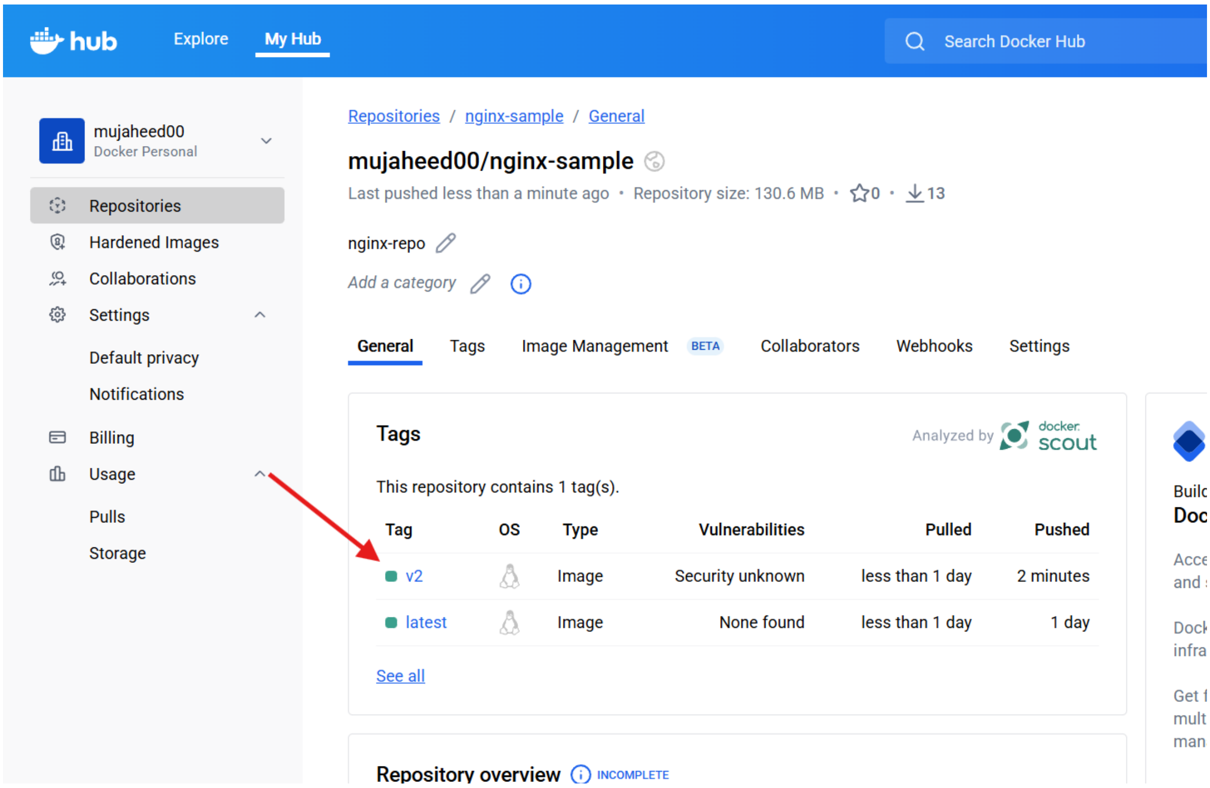
○ Static scanning
Images will be scanned once when pushed and the vulnerability report saved at that point in time.

○ None

- docker push mujaheed00/nginx-sample:v2



```
[root@ip-172-31-81-250 ~]# docker push mujaheed00/nginx-sample:v2
The push refers to repository [docker.io/mujaheed00/nginx-sample]
38d44e06fd01: Mounted from library/nginx
388bb4cadb9e: Mounted from library/nginx
5f0d4d15245b: Mounted from library/nginx
fe0771a36433: Mounted from library/nginx
1e79db1a7c1e: Mounted from library/nginx
008ba900efa1: Mounted from library/nginx
70a290c5e58b: Mounted from library/nginx
v2: digest: sha256:a3a430d5731331a5e737ce85a2573c1a7dd58c2c4d100fcc7bfd53a312aac53a size: 1778
[root@ip-172-31-81-250 ~]#
```



Repositories / nginx-sample / General

mujaheed00/nginx-sample

Last pushed less than a minute ago • Repository size: 130.6 MB • ☆0 • ⤓13

nginx-repo

Add a category

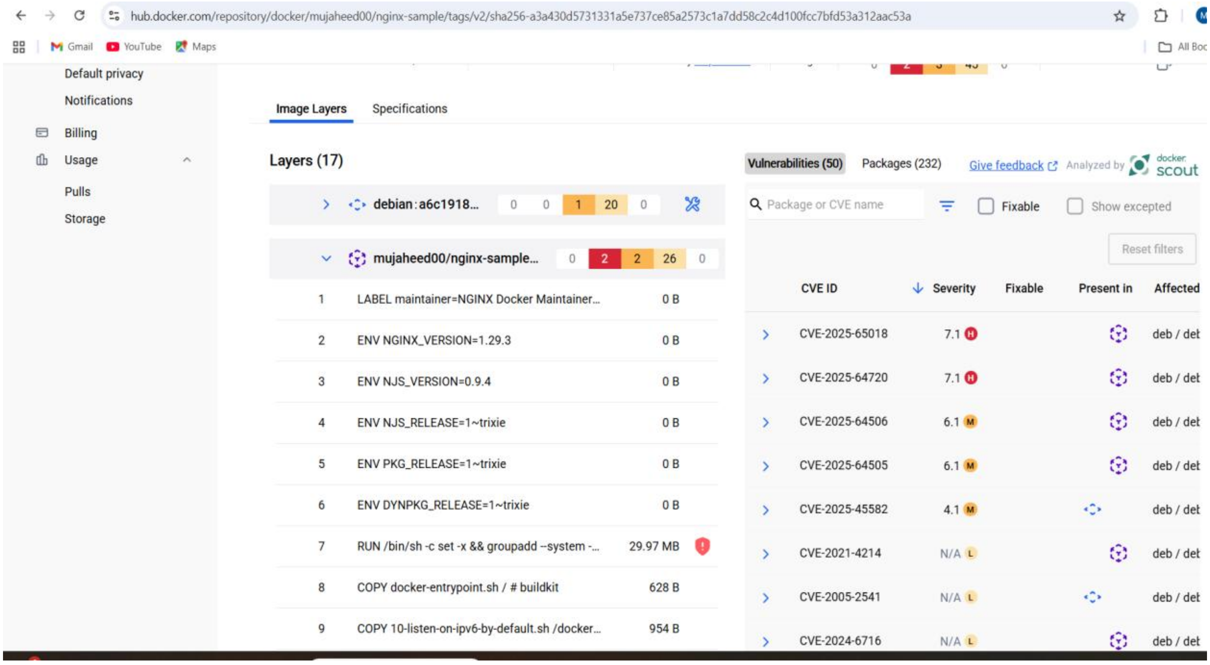
Tags

This repository contains 1 tag(s).

Tag	OS	Type	Vulnerabilities	Pulled	Pushed
v2		Image	Security unknown	less than 1 day	2 minutes
latest		Image	None found	less than 1 day	1 day

Repository overview INCOMPLETE

If you select your image you will find



Layers (17)

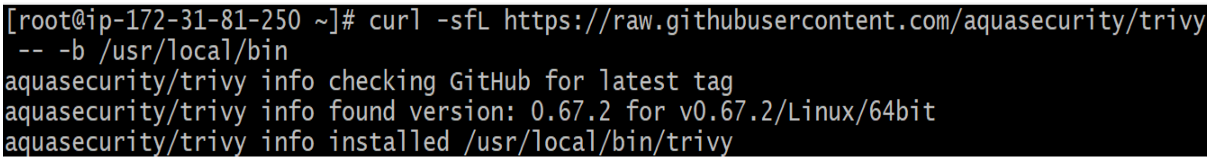
debian:a6c1918...	0	0	1	20	0
mujaheed00/nginx-sample...	0	2	2	26	0

1	LABEL maintainer=NGINX Docker Maintainer...	0 B
2	ENV NGINX_VERSION=1.29.3	0 B
3	ENV NJS_VERSION=0.9.4	0 B
4	ENV NJS_RELEASE=1~trixie	0 B
5	ENV PKG_RELEASE=1~trixie	0 B
6	ENV DYNPKG_RELEASE=1~trixie	0 B
7	RUN /bin/sh -c set -x && groupadd --system -...	29.97 MB
8	COPY docker-entrypoint.sh / # buildkit	628 B
9	COPY 10-listen-on-ipv6-by-default.sh /docker...	954 B

CVE ID	Severity	Fixable	Present in	Affected
CVE-2025-65018	7.1 H			deb / det
CVE-2025-64720	7.1 H			deb / det
CVE-2025-64506	6.1 M			deb / det
CVE-2025-64505	6.1 M			deb / det
CVE-2025-45582	4.1 M			deb / det
CVE-2021-4214	N/A L			deb / det
CVE-2005-2541	N/A L			deb / det
CVE-2024-6716	N/A L			deb / det

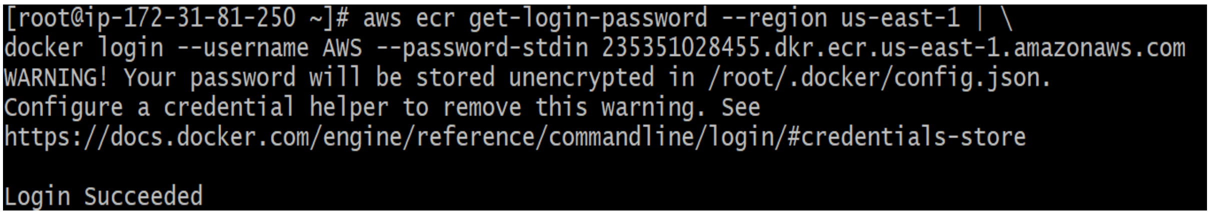
7. Implement solution to scan images when pushed to AWS ECR.

- curl -sfL https://raw.githubusercontent.com/aquasecurity/trivy/main/contrib/install.sh | sudo sh -s -- -b /usr/local/bin



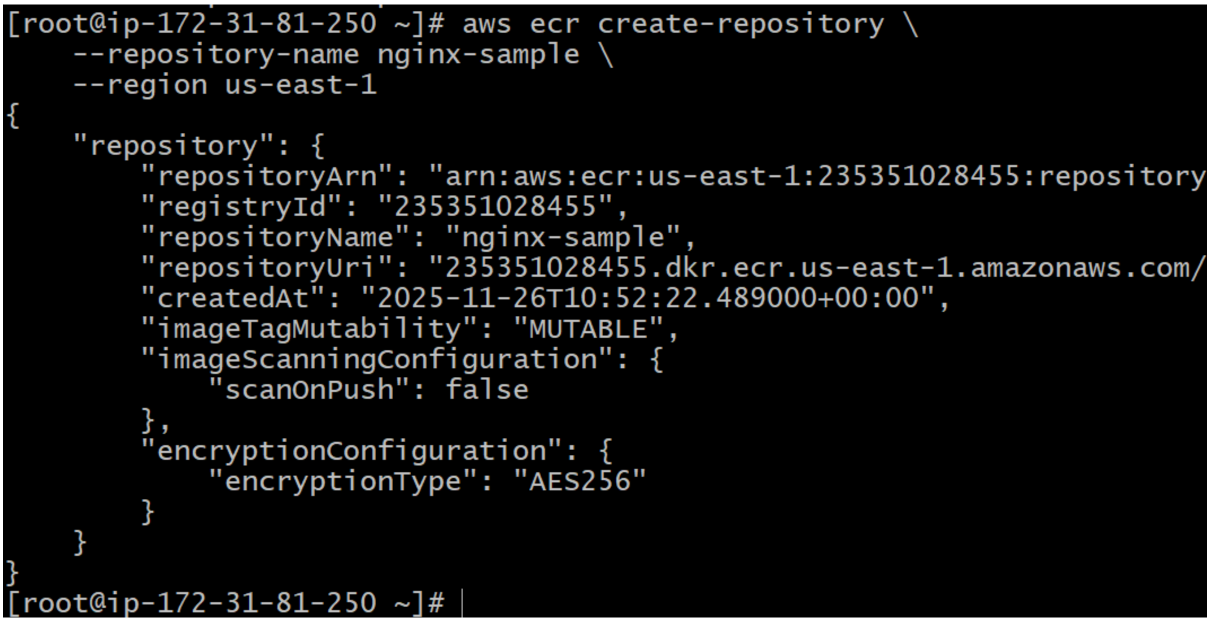
```
[root@ip-172-31-81-250 ~]# curl -sfL https://raw.githubusercontent.com/aquasecurity/trivy
 -- -b /usr/local/bin
aquasecurity/trivy info checking GitHub for latest tag
aquasecurity/trivy info found version: 0.67.2 for v0.67.2/Linux/64bit
aquasecurity/trivy info installed /usr/local/bin/trivy
```

- 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-172-31-81-250 ~]# 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
```

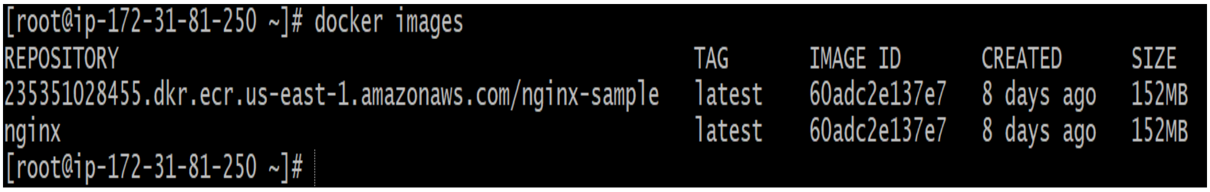
- aws ecr create-repository \ --repository-name nginx-sample \ --region us-east-1



```
[root@ip-172-31-81-250 ~]# aws ecr create-repository \
    --repository-name nginx-sample \
    --region us-east-1
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:us-east-1:235351028455:repository
        "registryId": "235351028455",
        "repositoryName": "nginx-sample",
        "repositoryUri": "235351028455.dkr.ecr.us-east-1.amazonaws.com/
        "createdAt": "2025-11-26T10:52:22.489000+00:00",
        "imageTagMutability": "MUTABLE",
        "imageScanningConfiguration": {
            "scanOnPush": false
        },
        "encryptionConfiguration": {
            "encryptionType": "AES256"
        }
    }
}
[root@ip-172-31-81-250 ~]#
```

Tag the image

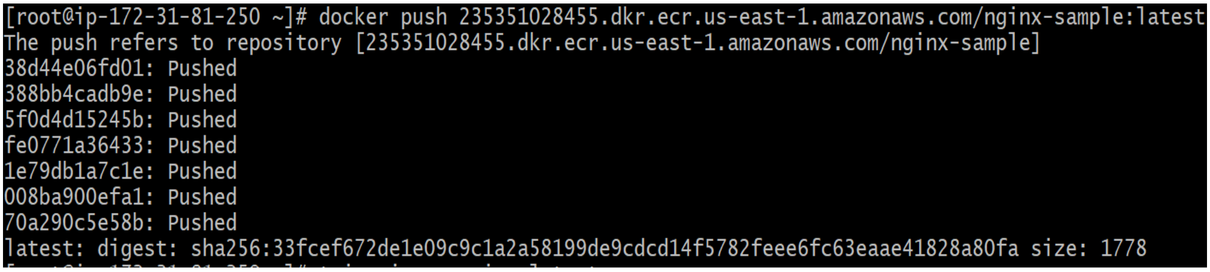
- docker tag nginx:latest 235351028455.dkr.ecr.us-east-1.amazonaws.com/nginx-sample:latest



```
[root@ip-172-31-81-250 ~]# docker images
REPOSITORY                                                TAG       IMAGE ID       CREATED       SIZE
235351028455.dkr.ecr.us-east-1.amazonaws.com/nginx-sample latest    60adc2e137e7   8 days ago    152MB
nginx                                                     latest    60adc2e137e7   8 days ago    152MB
[root@ip-172-31-81-250 ~]#
```

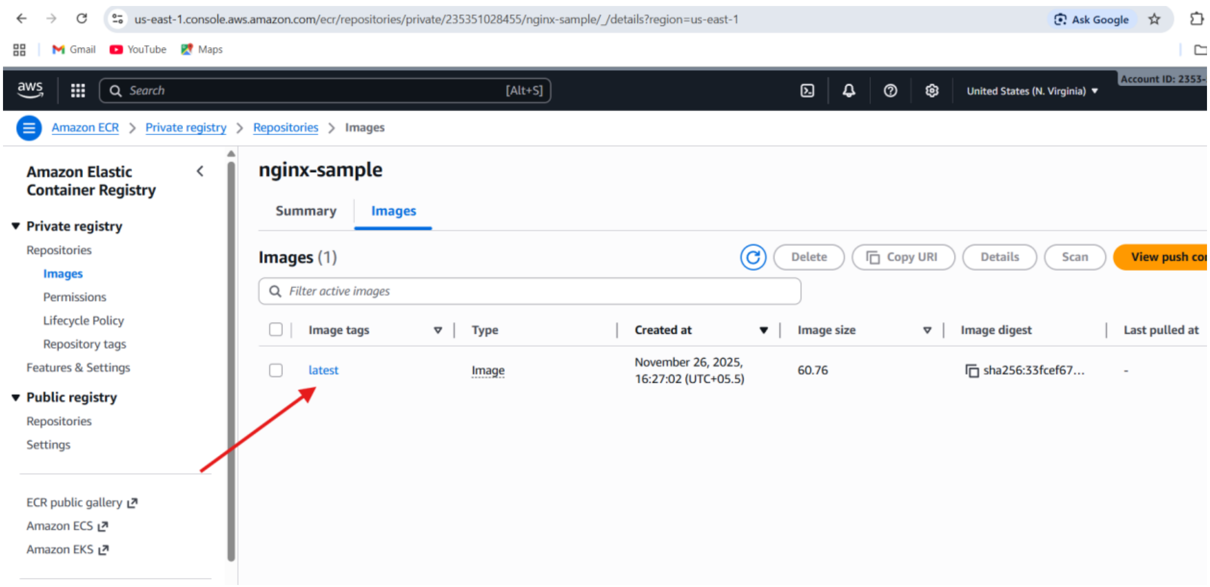
Docker push

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



```
[root@ip-172-31-81-250 ~]# 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]
38d44e06fd01: Pushed
388bb4cadb9e: Pushed
5f0d4d15245b: Pushed
fe0771a36433: Pushed
1e79db1a7c1e: Pushed
008ba900efa1: Pushed
70a290c5e58b: Pushed
latest: digest: sha256:33fcef672de1e09c9c1a2a58199de9cdcd14f5782feee6fc63eaae41828a80fa size: 1778
```

Click on image that you are pushed.

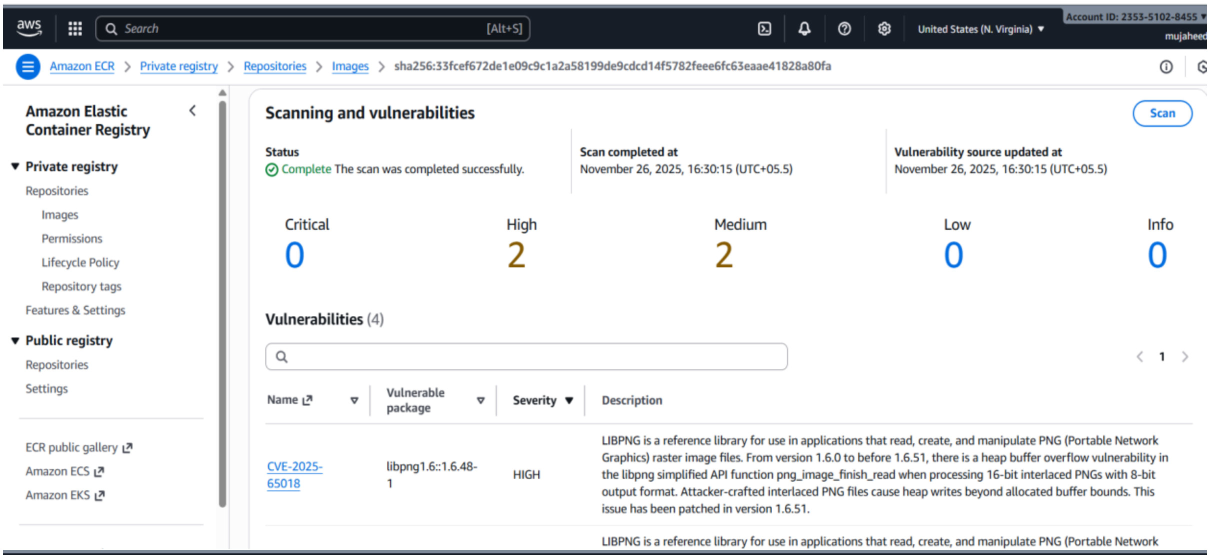


nginx-sample

Images (1)

□	Image tags	Type	Created at	Image size	Image digest	Last pulled at
□	latest	Image	November 26, 2025, 16:27:02 (UTC+05.5)	60.76	sha256:33fcef67...	-

If you click on scan



Scanning and vulnerabilities

Status
Complete The scan was completed successfully.

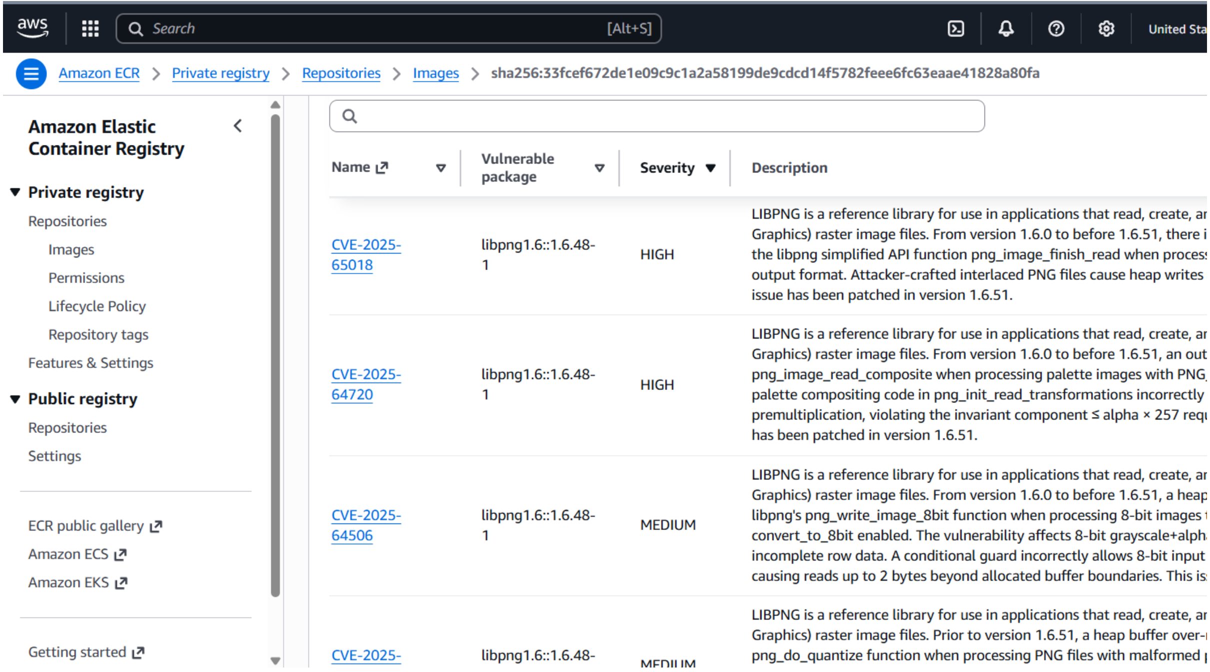
Scan completed at
November 26, 2025, 16:30:15 (UTC+05.5)

Vulnerability source updated at
November 26, 2025, 16:30:15 (UTC+05.5)

Critical	High	Medium	Low	Info
0	2	2	0	0

Vulnerabilities (4)

Name	Vulnerable package	Severity	Description
CVE-2025-65018	libpng1.6::1.6.48-1	HIGH	LIBPNG is a reference library for use in applications that read, create, and manipulate PNG (Portable Network Graphics) raster image files. From version 1.6.0 to before 1.6.51, there is a heap buffer overflow vulnerability in the libpng simplified API function png_image_finish_read when processing 16-bit interlaced PNGs with 8-bit output format. Attacker-crafted interlaced PNG files cause heap writes beyond allocated buffer bounds. This issue has been patched in version 1.6.51.
			LIBPNG is a reference library for use in applications that read, create, and manipulate PNG (Portable Network

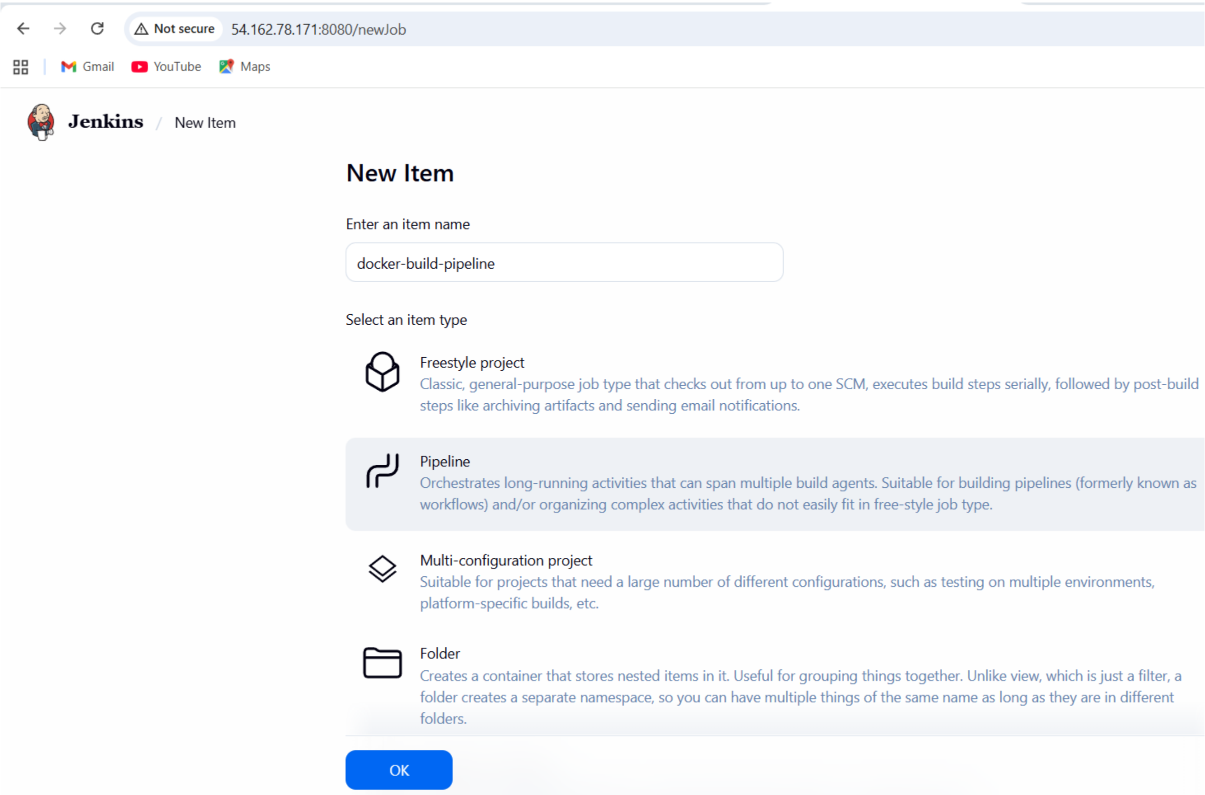


Name	Vulnerable package	Severity	Description
CVE-2025-65018	libpng1.6::1.6.48-1	HIGH	LIBPNG is a reference library for use in applications that read, create, ar Graphics) raster image files. From version 1.6.0 to before 1.6.51, there i the libpng simplified API function png_image_finish_read when proces output format. Attacker-crafted interlaced PNG files cause heap writes issue has been patched in version 1.6.51.
CVE-2025-64720	libpng1.6::1.6.48-1	HIGH	LIBPNG is a reference library for use in applications that read, create, ar Graphics) raster image files. From version 1.6.0 to before 1.6.51, an out png_image_read_composite when processing palette images with PNG_ palette compositing code in png_init_read_transformations incorrectly premultiplication, violating the invariant component ≤ alpha × 257 req has been patched in version 1.6.51.
CVE-2025-64506	libpng1.6::1.6.48-1	MEDIUM	LIBPNG is a reference library for use in applications that read, create, ar Graphics) raster image files. From version 1.6.0 to before 1.6.51, a heap libpng's png_write_image_8bit function when processing 8-bit images convert_to_8bit enabled. The vulnerability affects 8-bit grayscale+alph incomplete row data. A conditional guard incorrectly allows 8-bit input causing reads up to 2 bytes beyond allocated buffer boundaries. This is
CVE-2025-	libpng1.6::1.6.48-	MEDIUM	LIBPNG is a reference library for use in applications that read, create, ar Graphics) raster image files. Prior to version 1.6.51, a heap buffer over- png_do_quantize function when processing PNG files with malformed

8. Create a Jenkins pipeline to create a docker image and push the image to Docker hub.

Install Jenkins in the same docker server.

Go to Jenkins server click on new job give name as docker-build-pipeline and select type as pipeline.



New Item

Enter an item name

docker-build-pipeline

Select an item type

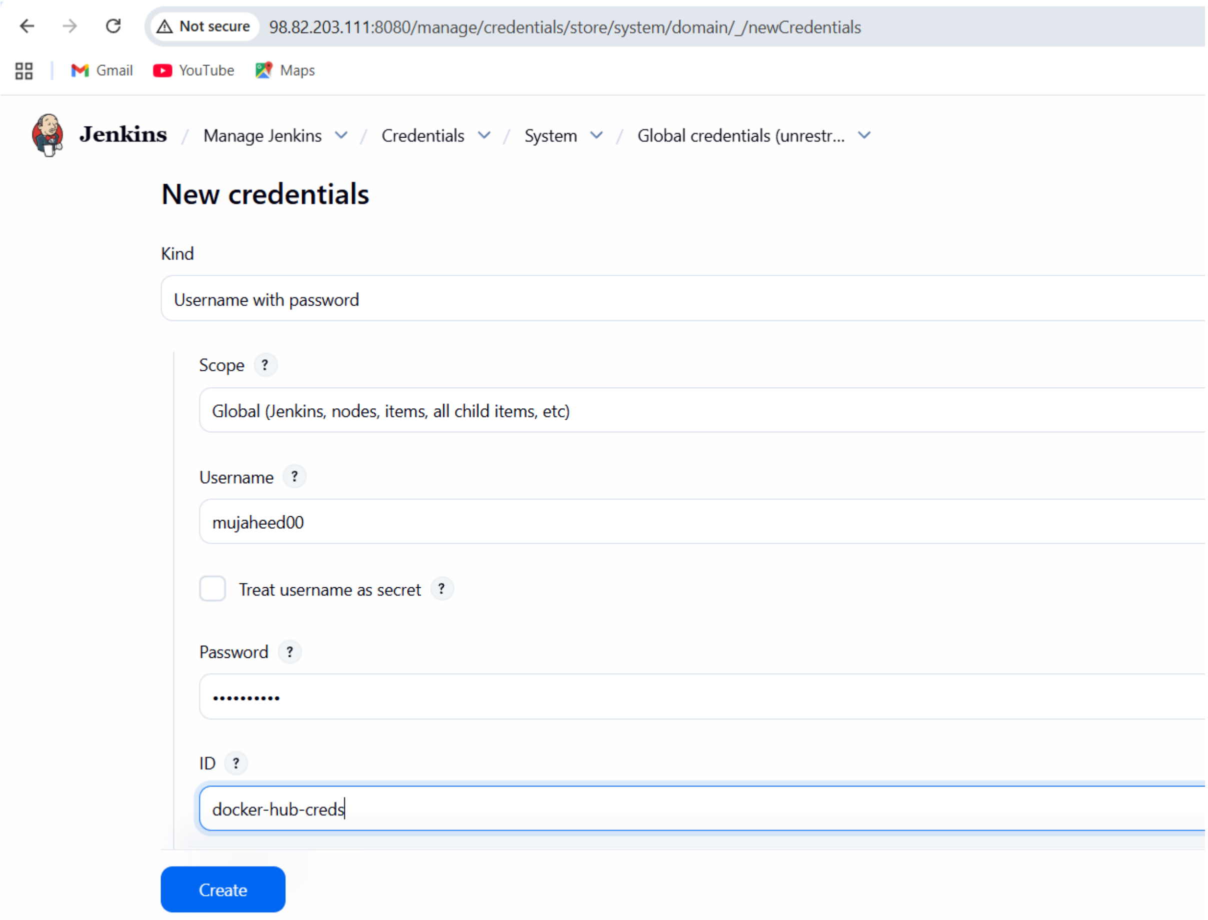
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Go to manage Jenkins,credentials,give docker hub username and password and id as docker-hub-creds.



Not secure
98.82.203.111:8080/manage/credentials/store/system/domain/_/newCredentials
Gmail
YouTube
Maps
Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...

New credentials

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

mujaheed00

□ Treat username as secret ?

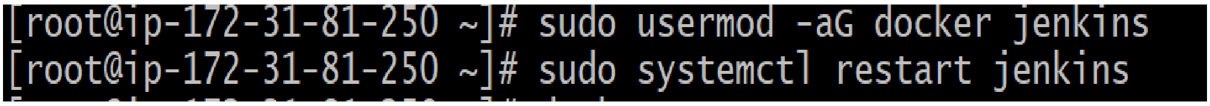
Password ?

••••••••••

ID ?

docker-hub-creds

- usermod -aG docker Jenkins
- systemctl restart Jenkins

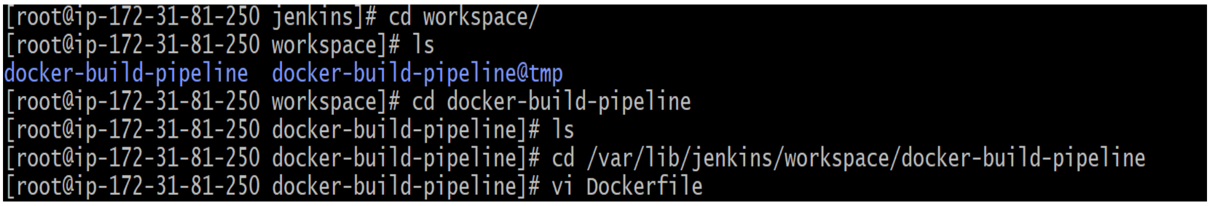


```
[root@ip-172-31-81-250 ~]# sudo usermod -aG docker jenkins
[root@ip-172-31-81-250 ~]# sudo systemctl restart jenkins
```

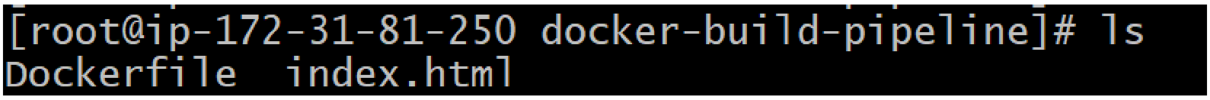
change to this directory

- cd /var/lib/jenkins/workspace/docker-build-pipeline

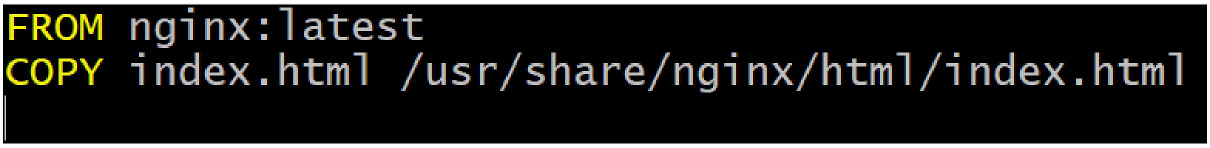
and create a file named as vi Dockerfile.



```
[root@ip-172-31-81-250 jenkins]# cd workspace/
[root@ip-172-31-81-250 workspace]# ls
docker-build-pipeline  docker-build-pipeline@tmp
[root@ip-172-31-81-250 workspace]# cd docker-build-pipeline
[root@ip-172-31-81-250 docker-build-pipeline]# ls
[root@ip-172-31-81-250 docker-build-pipeline]# cd /var/lib/jenkins/workspace/docker-build-pipeline
[root@ip-172-31-81-250 docker-build-pipeline]# vi Dockerfile
```

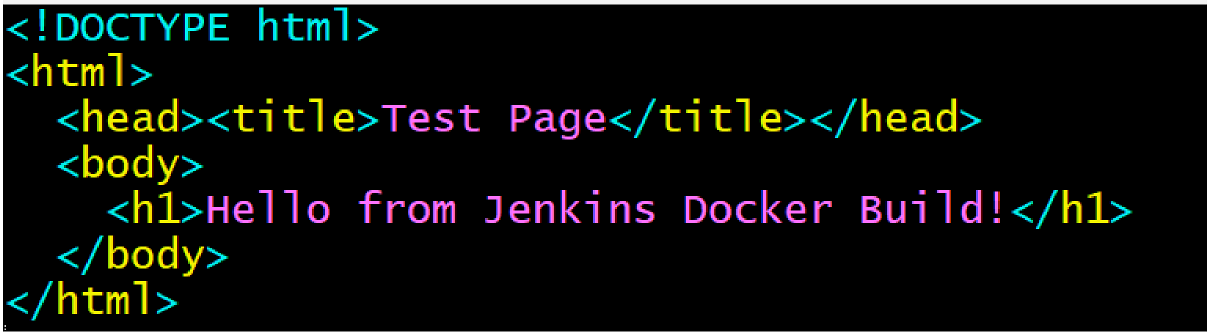


```
[root@ip-172-31-81-250 docker-build-pipeline]# ls
Dockerfile  index.html
```



```
FROM nginx:latest
COPY index.html /usr/share/nginx/html/index.html
```

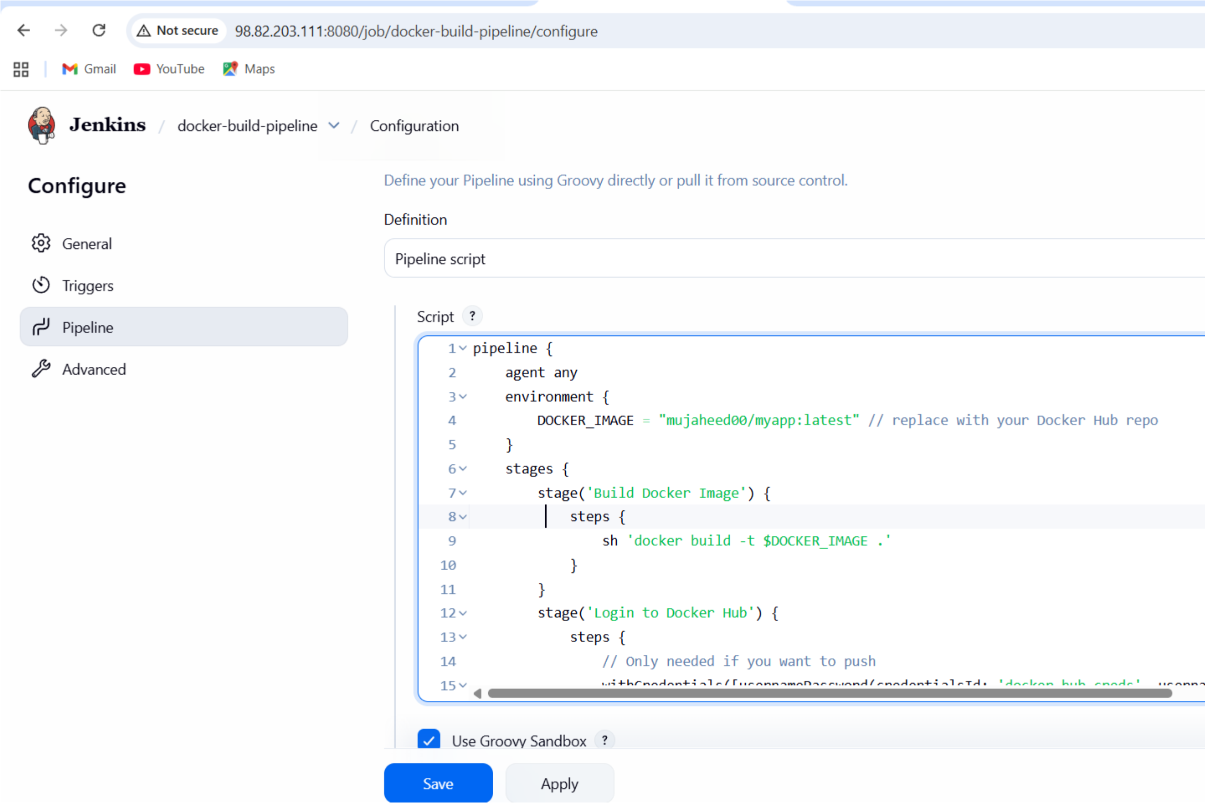
And create a file vi index.html



```
<!DOCTYPE html>
<html>
  <head><title>Test Page</title></head>
  <body>
    <h1>Hello from Jenkins Docker Build!</h1>
  </body>
</html>
```

Go to Jenkins job in configure give this script.

```
pipeline {
    agent any
    environment {
        DOCKER_IMAGE = "mujaheed00/myapp:latest" // replace with your Docker Hub repo
    }
    stages {
        stage('Build Docker Image') {
            steps {
                sh 'docker build -t $DOCKER_IMAGE .'
            }
        }
        stage('Login to Docker Hub') {
            steps {
                // Only needed if you want to push
                withCredentials([usernamePassword(credentialsId: 'docker-hub-creds', usernameVariable: 'DOCKER_USER', passwordVariable: 'DOCKER_PASS')]) {
                    sh 'echo $DOCKER_PASS | docker login -u $DOCKER_USER --password-stdin'
                }
            }
        }
        stage('Push Docker Image') {
            steps {
                sh 'docker push $DOCKER_IMAGE'
            }
        }
    }
}
```



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

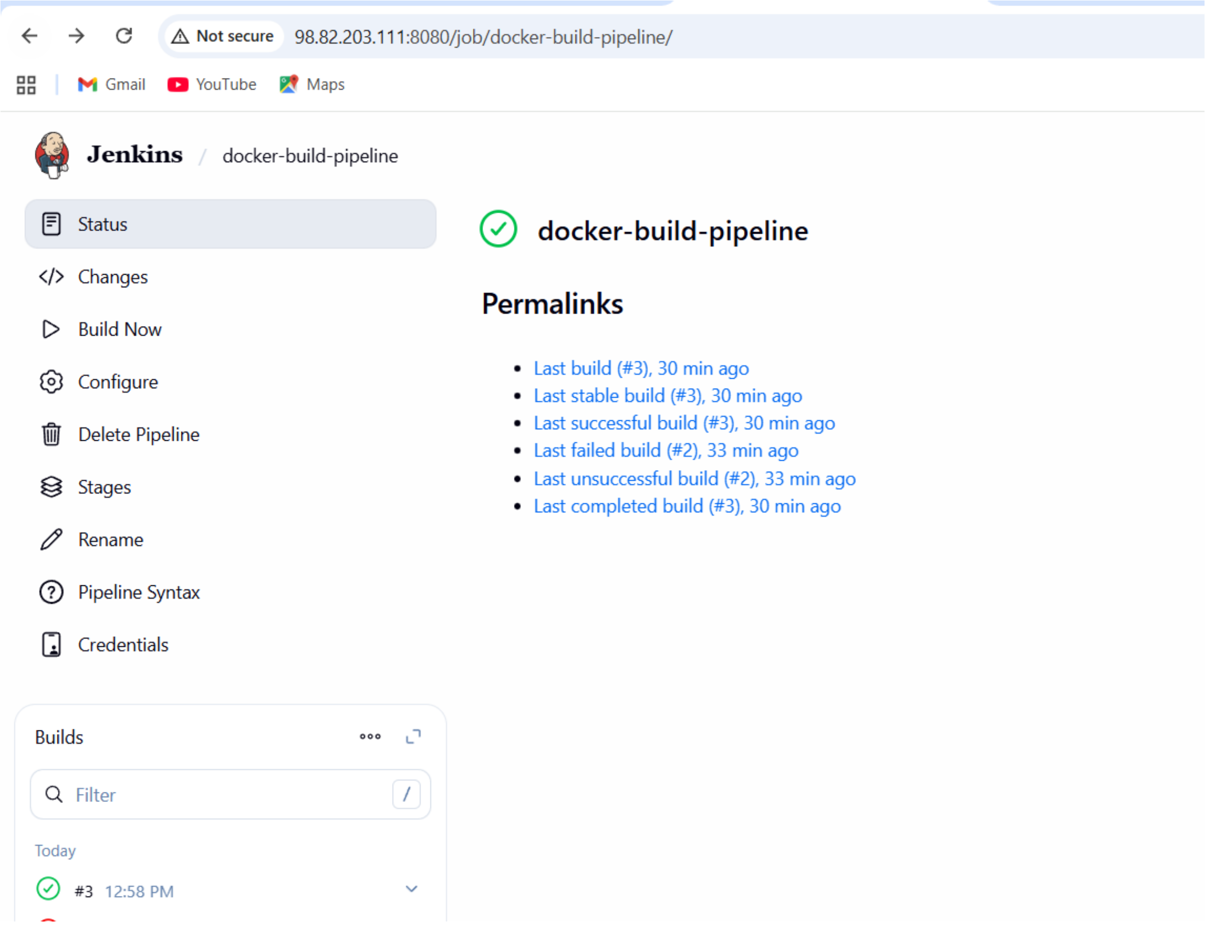
Definition: Pipeline script

Script

```
pipeline {
    agent any
    environment {
        DOCKER_IMAGE = "mujaheed00/myapp:latest" // replace with your Docker Hub repo
    }
    stages {
        stage('Build Docker Image') {
            steps {
                sh 'docker build -t $DOCKER_IMAGE .'
            }
        }
        stage('Login to Docker Hub') {
            steps {
                // Only needed if you want to push
                withCredentials([usernamePassword(credentialsId: 'docker-hub-creds', userna
```

☑ Use Groovy Sandbox

Click on save and build now.

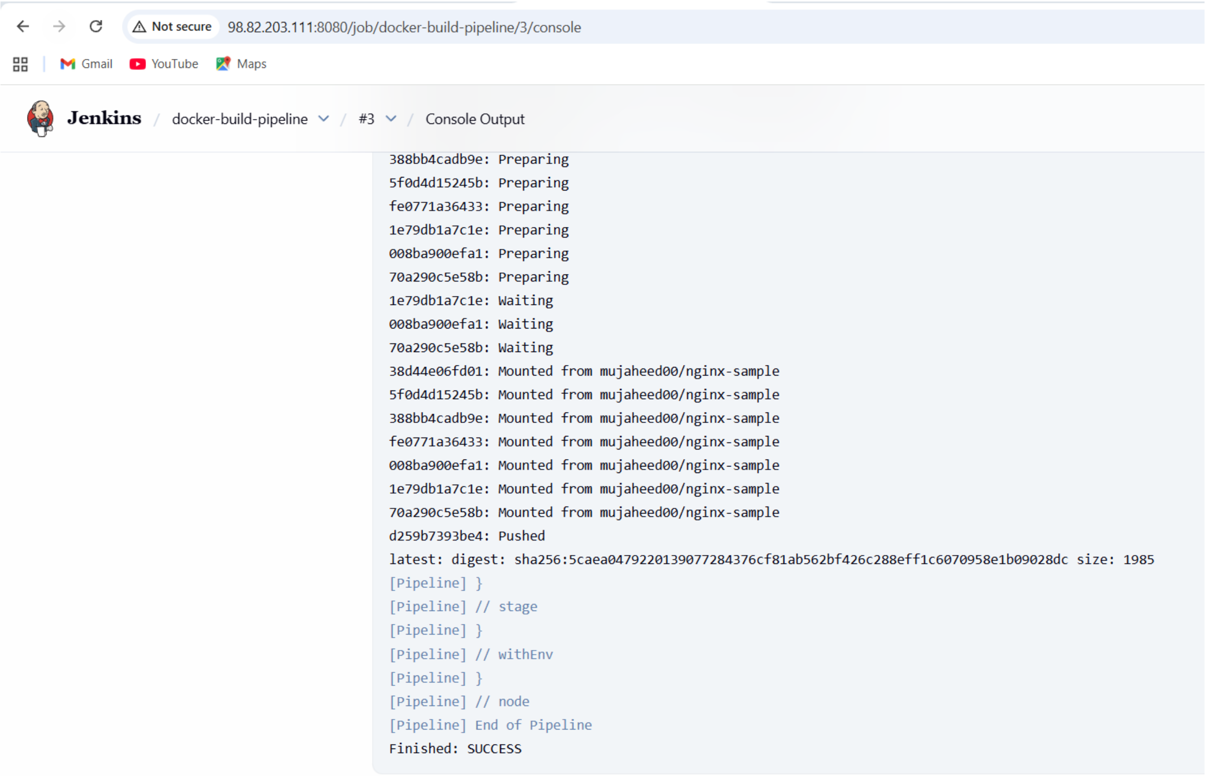


docker-build-pipeline

Permalinks

- Last build (#3), 30 min ago
- Last stable build (#3), 30 min ago
- Last successful build (#3), 30 min ago
- Last failed build (#2), 33 min ago
- Last unsuccessful build (#2), 33 min ago
- Last completed build (#3), 30 min ago

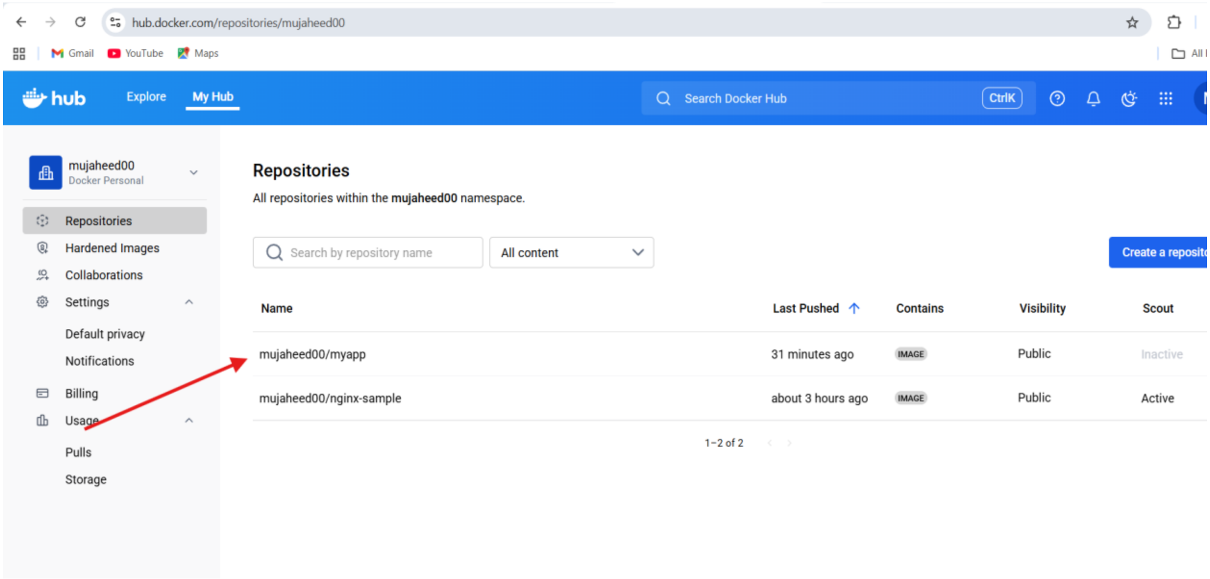
#3 12:58 PM



Not secure 98.82.203.111:8080/job/docker-build-pipeline/3/console
Gmail
YouTube
Maps
Jenkins / docker-build-pipeline / #3 / Console Output

```
388bb4cadb9e: Preparing
5f0d4d15245b: Preparing
fe0771a36433: Preparing
1e79db1a7c1e: Preparing
008ba900efa1: Preparing
70a290c5e58b: Preparing
1e79db1a7c1e: Waiting
008ba900efa1: Waiting
70a290c5e58b: Waiting
38d44e06fd01: Mounted from mujaheed00/nginx-sample
5f0d4d15245b: Mounted from mujaheed00/nginx-sample
388bb4cadb9e: Mounted from mujaheed00/nginx-sample
fe0771a36433: Mounted from mujaheed00/nginx-sample
008ba900efa1: Mounted from mujaheed00/nginx-sample
1e79db1a7c1e: Mounted from mujaheed00/nginx-sample
70a290c5e58b: Mounted from mujaheed00/nginx-sample
d259b7393be4: Pushed
latest: digest: sha256:5caea04792201390772843 76cf81ab562bf426c288eff1c6070958e1b09028dc size: 1985
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Go to docker hub a repository has been created with the image.



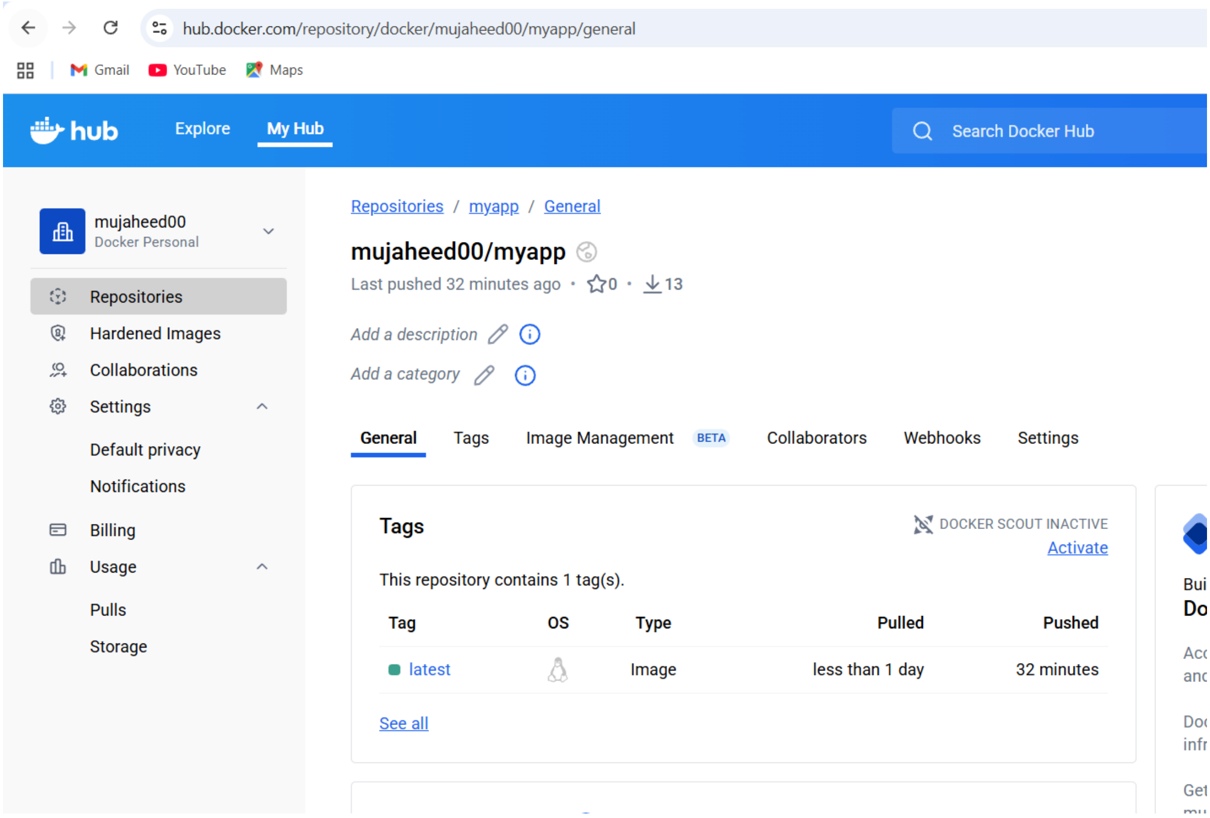
Repositories

All repositories within the **mujaheed00** namespace.

Search by repository name | All content

Name	Last Pushed	Contains	Visibility	Scout
mujaheed00/myapp	31 minutes ago	IMAGE	Public	Inactive
mujaheed00/nginx-sample	about 3 hours ago	IMAGE	Public	Active

1–2 of 2



mujaheed00/myapp

Last pushed 32 minutes ago • ☆0 • ⤓13

Add a description

Add a category

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	less than 1 day	32 minutes