

DOCKER-03

1) Create a image from running container.

→ LETS FIRST START ONE CONTAINER

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

```
# docker ps
```

```
[root@ip-10-0-0-169 ~]# docker container run -itd -p 80:80 nginx:latest
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
bc0965b23a04: Pull complete
650ee30bbe5e: Pull complete
8cc1569e58f5: Pull complete
362f35df001b: Pull complete
13e320bf29cd: Pull complete
7b50399908e1: Pull complete
57b64962dd94: Pull complete
Digest: sha256:fb197595ebe76b9c0c14ab68159fd3c08bd067ec62300583543f0ebda353b5be
Status: Downloaded newer image for nginx:latest
32cad684aa9c903e14677f79c3ac99157ed14586309e8f0fc49a75ff41cb5266
[root@ip-10-0-0-169 ~]# docker ps
```


| CONTAINER ID | IMAGE | COMMAND | NAMES | CREATED | STATUS |
|--------------|--------------|-------------------------|------------|----------------|---------------|
| 32cad684aa9c | nginx:latest | "/docker-entrypoint..." | keen_wiles | 11 seconds ago | Up 10 seconds |

→ CREATING AND IMAGE OF THIS RUNNING CONTAINER

```
# docker commit 32ca nginx:dev
```

```
[root@ip-10-0-0-169 ~]# docker commit 32ca nginx:dev
sha256:97e281f351b23629e951bf48cf59def629a11633d433c1e9e99b25e2b6aca477
[root@ip-10-0-0-169 ~]# docker images
```

| REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
|------------|--------|--------------|---------------|-------|
| nginx | dev | 97e281f351b2 | 5 seconds ago | 192MB |
| nginx | latest | 66f8bdd3810c | 3 weeks ago | 192MB |



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

→ LETS FIRST DOWNLOAD DOCKER DESKTOP AND PULL ONE HTTPD IMAGE IN LOCAL MACHINE AND SEND IT TO OUR EC2 SERVER

→ IN LOCAL MACHINE:

```
# docker pull httpd
```

```
# docker images
```

```
syedf@Lenovo-i5 MINGW64 ~/Desktop
$ docker images
```

| REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
|------------|--------|--------------|--------------|-------|
| httpd | latest | f4c5139eda46 | 5 months ago | 220MB |

```
# docker save -o httpd.tar httpd:latest
# scp -i virginia.pem httpd.tar ec2-user@54.80.230.194:/tmp
```

```
syedf@Lenovo-i5 MINGW64 ~/Desktop
$ scp -i virginia.pem httpd.tar ec2-user@54.80.230.194:/tmp
httpd.tar 100% 56MB 62.6KB/s 15:09
```

→ ON EC2 SERVER

```
# cd /tmp
```

```
# ls
```

```
[root@ip-10-0-0-169 ~]# cd /tmp/
[root@ip-10-0-0-169 tmp]# ls
httpd.tar  systemd-private-fdd0b1888c424b27b41e08dae4ad3370-chronyd.service-I72qeM
```

```
# docker load -i httpd.tar
```

```
[root@ip-10-0-0-169 tmp]# docker load -i httpd.tar
ea0dc0ce5a73: Loading layer [=====>] 145B/145B
5f70bf18a086: Loading layer [=====>] 32B/32B
ee527e6a3aab: Loading layer [=====>] 4.008MB/4.008MB
dac6ca30c6d4: Loading layer [=====>] 26.04MB/26.04MB
d284c6380ce3: Loading layer [=====>] 292B/292B
Loaded image: httpd:latest
[root@ip-10-0-0-169 tmp]# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
alpine-tomcat        latest          6deee84d0bff   10 minutes ago 231MB
nginx                dev            97e281f351b2   33 minutes ago 192MB
nginx                latest         66f8bdd3810c   3 weeks ago    192MB
httpd                latest         494b2b45fd74   5 months ago   147MB
```

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

→ CREATE A **Dockerfile** AND ADD THE BELOW # vi Dockerfile

```
FROM alpine:latest
RUN apk add --no-cache openjdk11-jre
RUN wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.34/bin/apache-tomcat-10.1.34.tar.gz
RUN tar xvf ap*.tar.gz -C /usr/local/
RUN mv /usr/local/apach* /usr/local/tomcat

EXPOSE 8080
CMD ["/usr/local/tomcat/bin/catalina.sh", "run"]
```

```
# docker build -t alpine-tomcat .
```

```
# docker images
```

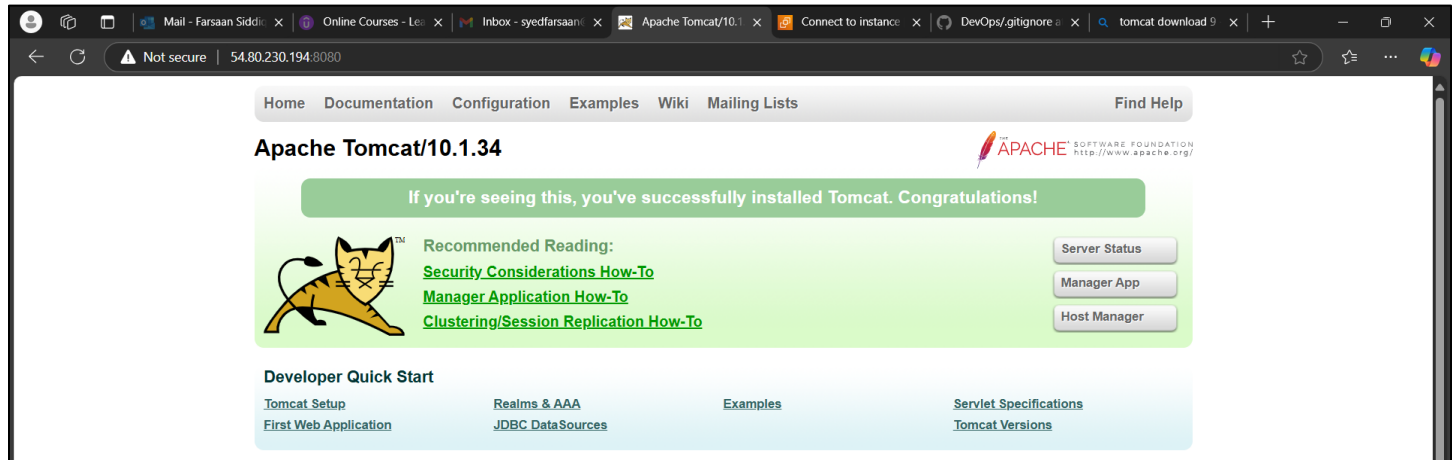
```
[root@ip-10-0-0-169 ~]# vi Dockerfile
[root@ip-10-0-0-169 ~]# docker build -t alpine-tomcat .
[+] Building 5.7s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 368B
=> [internal] load metadata for docker.io/library/alpine:latest
=> [auth] library/alpine:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/alpine:latest@sha256:21dc6063fd678b478f57c0e13f47560d0ea4eeba26dfc947b2a4f81f686b9f45
=> resolve docker.io/library/alpine:latest@sha256:21dc6063fd678b478f57c0e13f47560d0ea4eeba26dfc947b2a4f81f686b9f45
=> sha256:4048db5d36726a313ab8f7ffccf2362a34cba69e4cdd49119713483a68641fce 581B / 581B
=> sha256:38a8310d387e375e0ec6fabe047a9149e8eb214073db9f461fee6251fd936a75 3.64MB / 3.64MB
=> sha256:21dc6063fd678b478f57c0e13f47560d0ea4eeba26dfc947b2a4f81f686b9f45 9.22kB / 9.22kB
=> sha256:2c43f33bd1502ec7818bce9eea60a062d04eeadc4aa31cad9dabecb1e48b647b 1.02kB / 1.02kB
=> extracting sha256:38a8310d387e375e0ec6fabe047a9149e8eb214073db9f461fee6251fd936a75
=> [2/5] RUN apk add --no-cache openjdk11-jre
=> [3/5] RUN wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.34/bin/apache-tomcat-10.1.34.tar.gz
=> [4/5] RUN tar xvf apache-tomcat-10.1.34.tar.gz -C /usr/local/
=> [5/5] RUN mv /usr/local/apache-tomcat-10.1.34 /usr/local/tomcat
=> exporting to image
=> exporting layers
=> writing image sha256:6deee84d0bff057136bef2e902fa92ef671faf77eb99c54afa7416e2a1f6b97e
=> naming to docker.io/library/alpine-tomcat
[root@ip-10-0-0-169 ~]# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
alpine-tomcat        latest          6deee84d0bff   8 seconds ago   231MB
nginx                dev            97e281f351b2   23 minutes ago 192MB
nginx                latest         66f8bdd3810c   3 weeks ago    192MB
```

→ RUN THE CONTAINER AND CHECK ON BROWSER

```
# docker container run -itd -p 8080:8080 alpine-tomcat:latest
```

```
[root@ip-10-0-0-169 ~]# docker container run -itd -p 8080:8080 alpine-tomcat:latest
a7c9eb77e37d097a4d4b753c8180cdb2b809db63d16abc1392b4dd1b975e3894
```

BROWSER:



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

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

→single stage Dockerfile

➔ CLONE THE REPO IN DOCKER-SERVER

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

```
# cd multi-stage-example/
```

```
# vi Dockerfile
```

```
FROM openjdk:8-jdk-alpine
RUN mkdir -p /app/source
COPY . /app/source
WORKDIR /app/source
RUN ./mvnw clean package
RUN cp /app/source/target/*.jar /app/app.jar

EXPOSE 8080
ENTRYPOINT ["java", "-Djava.security.egd=file:/dev/./urandom", "-jar", "/app/app.jar"]
```

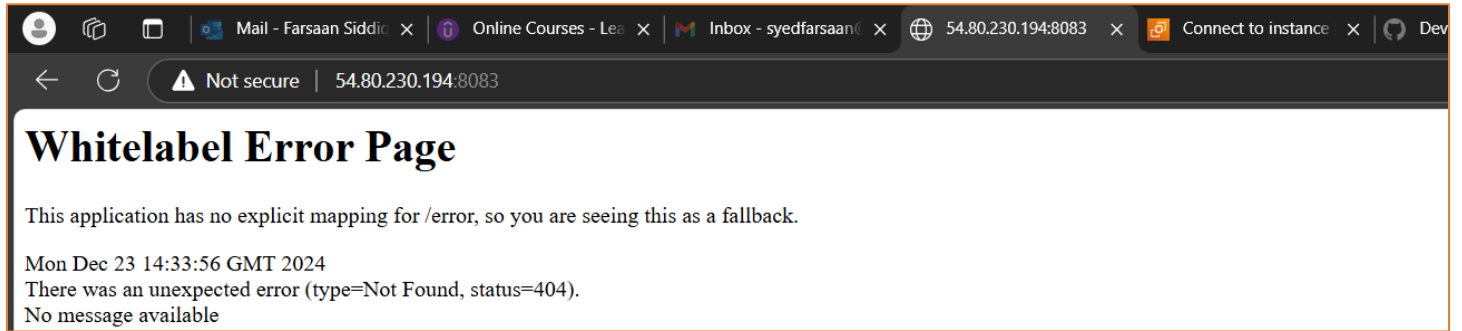
```
# docker build -t my-app:v2
```

```
[root@ip-10-0-0-169 multi-stage-example]# vi Dockerfile
[root@ip-10-0-0-169 multi-stage-example]# docker build -t my-app:v2 .
[+] Building 24.2s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 299B
=> [internal] load metadata for docker.io/library/openjdk:8-jdk-alpine
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/6] FROM docker.io/library/openjdk:8-jdk-alpine@sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f2283343547b3
=> [internal] load build context
=> => transferring context: 3.27kB
=> CACHED [2/6] RUN mkdir -p /app/source
=> [3/6] COPY . /app/source
=> [4/6] WORKDIR /app/source
=> [5/6] RUN ./mvnw clean package
=> [6/6] RUN cp /app/source/target/*.jar /app/app.jar
=> exporting to image
=> => exporting layers
=> => writing image sha256:6fe1c3d508813a6219bce5772a433e79c26a0e332bb473f2038e7eb1a485f94a
=> => naming to docker.io/library/my-app:v2
```

```
# docker container run -itd -p 8083:8080 my-app:v2
```

```
[root@ip-10-0-0-169 multi-stage-example]# docker container run -itd -p 8083:8080 my-app:v2
2f4303baccbc175fcf5e802286affe16b4c4d6cbfb7932ce84139b89c158e927
```

→check port 8083 followed by public Ip



→Multi-stage Dockerfile

→CLONE THE REPO IN DOCKER-SERVER

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

```
# cd multi-stage-example/
```

```
# vi Dockerfile
```

```
FROM openjdk:8-jdk-alpine as builder
```

```
RUN mkdir -p /app/source
```

```
COPY ./app/source
```

```
WORKDIR /app/source
```

```
RUN ./mvnw clean package
```

```
FROM builder
```

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


```
EXPOSE 8080
```

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

docker build -t my-app:v1

```
[root@ip-10-0-0-169 multi-stage-example]# docker build -t my-app:v1 .
[+] Building 27.3s (12/12) FINISHED
=> [internal] load build definition from Dockerfile
=> transferring dockerfile: 336B
=> [internal] load metadata for docker.io/library/openjdk:8-jdk-alpine
=> [auth] library/openjdk:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> transferring context: 2B
=> [builder 1/5] FROM docker.io/library/openjdk:8-jdk-alpine@sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f2283343547b3
=> resolve docker.io/library/openjdk:8-jdk-alpine@sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f2283343547b3
=> sha256:f910a506b6cb1dbec766725d70356f695ae2bf2bea6224dbe8c7c6ad4f3664a2 238B / 238B
=> sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 70.73MB / 70.73MB
=> sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f2283343547b3 1.64kB / 1.64kB
=> sha256:44b3cea369c947527e266275cee85c71a81f20fc5076f6ebb5a13f19015dce71 947B / 947B
=> sha256:a3562aa0b991a80cfe8172847c8be6dbf6e46340b759c2b782f8b8be45342717 3.40kB / 3.40kB
=> sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 2.76MB / 2.76MB
=> extracting sha256:e7c96db7181be991f19a9fb6975cddb73c65f4a2681348e63a141a2192a5f10 0.2s
=> extracting sha256:f910a506b6cb1dbec766725d70356f695ae2bf2bea6224dbe8c7c6ad4f3664a2 0.0s
=> extracting sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 1.2s
=> [internal] load build context
=> transferring context: 162.20kB
=> [builder 2/5] RUN mkdir -p /app/source
=> [builder 3/5] COPY ./app/source
=> [builder 4/5] WORKDIR /app/source
=> [builder 5/5] RUN ./mvnw clean package
=> [stage-1 1/1] COPY --from=builder /app/source/target/*.jar /app/app.jar
=> exporting to image
=> exporting layers
=> writing image sha256:9f3379643ccfd5d387d92d9a2e4e3862e9933028128192ede7fa7c71cc2759ac
=> naming to docker.io/library/my-app:v1
[root@ip-10-0-0-169 multi-stage-example]# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
my-app               v1             9f3379643ccf   2 minutes ago   218MB
```

```
[root@ip-10-0-0-169 multi-stage-example]# docker container run -itd -p 8085:8080 my-app:v1
1957fe1cd9f3769f4b4b12ad7fe55718f437c8267d8255b4b1eaa9e5f6821d97
```



The screenshot shows a web browser window with a dark theme. The address bar at the top displays a warning icon, the text 'Not secure', and the URL '54.80.230.194:8085'. Several tabs are open, including 'Mail - Farsaan Siddiq', 'Online Courses - Lea', 'Inbox - syedfarsaan', and 'Connect to instance'. The main content area has a white background and displays the following text:

Whitelabel Error Page

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

Mon Dec 23 14:38:08 GMT 2024

There was an unexpected error (type=Not Found, status=404).

No message available

```
# sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```

root@ip-10-0-0-169 ~]# sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/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 --:--:-- --:--:-- --:--:--    0
100 12.1M 100 12.1M    0     0  78.4M    0 --:--:-- --:--:-- --:--:-- 102M
root@ip-10-0-0-169 ~]# sudo chmod +x /usr/local/bin/docker-compose
root@ip-10-0-0-169 ~]# docker-compose --version
bash: docker-compose: command not found
root@ip-10-0-0-169 ~]# docker compose --version
Docker version 25.0.5, build 5dc9bcc

```

```
version: '3.3'
services:
  wordpress:
    image: wordpress:latest
    ports:
      - "8000:80"
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
  db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: somewordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress
```

docker-compose up

```
root@ip-10-0-0-169:~
root-db-1 2024-12-25T21:15:11.641125Z 0 [Note] Shutting down plugin 'INNODB_TRX'
root-db-1 2024-12-25T21:15:11.641128Z 0 [Note] Shutting down plugin 'InnoDB'
root-db-1 2024-12-25T21:15:11.641194Z 0 [Note] InnoDB: FTS optimize thread exiting.
root-db-1 2024-12-25T21:15:11.641742Z 0 [Note] InnoDB: Starting shutdown...
root-db-1 2024-12-25T21:15:11.742058Z 0 [Note] InnoDB: Dumping buffer pool(s) to /var/lib/mysql/ib_buffer_pool
root-db-1 2024-12-25T21:15:11.742522Z 0 [Note] InnoDB: Buffer pool(s) dump completed at 241225 21:15:11
root-db-1 2024-12-25T21:15:12.950731Z 0 [Note] InnoDB: Shutdown completed; log sequence number 12219419
root-db-1 2024-12-25T21:15:12.951532Z 0 [Note] InnoDB: Removed temporary tablespace data file: "ibtmp1"
root-db-1 2024-12-25T21:15:12.951582Z 0 [Note] Shutting down plugin 'MEMORY'
root-db-1 2024-12-25T21:15:12.951587Z 0 [Note] Shutting down plugin 'CSV'
root-db-1 2024-12-25T21:15:12.951590Z 0 [Note] Shutting down plugin 'sha256_password'
root-db-1 2024-12-25T21:15:12.951593Z 0 [Note] Shutting down plugin 'mysql_native_password'
root-db-1 2024-12-25T21:15:12.952732Z 0 [Note] Shutting down plugin 'binlog'
root-db-1 2024-12-25T21:15:12.953229Z 0 [Note] mysqld: Shutdown complete

root-db-1 2024-12-25 21:15:13+00:00 [Note] [Entrypoint]: Temporary server stopped


root-db-1 2024-12-25 21:15:13+00:00 [Note] [Entrypoint]: MySQL init process done. Ready for start up.

root-db-1 2024-12-25T21:15:13.834539Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
root-db-1 2024-12-25T21:15:13.836706Z 0 [Note] mysqld (mysqld 5.7.44) starting as process 1 ...
root-db-1 2024-12-25T21:15:13.839246Z 0 [Note] InnoDB: PUNCH HOLE support available
root-db-1 2024-12-25T21:15:13.839271Z 0 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
root-db-1 2024-12-25T21:15:13.839272Z 0 [Note] InnoDB: Uses event mutexes
root-db-1 2024-12-25T21:15:13.839278Z 0 [Note] InnoDB: GCC builtin __atomic_thread_fence() is used for memory barrier
root-db-1 2024-12-25T21:15:13.839282Z 0 [Note] InnoDB: Compressed tables use zlib 1.2.13
root-db-1 2024-12-25T21:15:13.839283Z 0 [Note] InnoDB: Using Linux native AIO
root-db-1 2024-12-25T21:15:13.839468Z 0 [Note] InnoDB: Number of pools: 1
root-db-1 2024-12-25T21:15:13.839732Z 0 [Note] InnoDB: Using CPU crc32 instructions
root-db-1 2024-12-25T21:15:13.841165Z 0 [Note] InnoDB: Initializing buffer pool, total size = 128M, instances = 1, chunk size = 128M
root-db-1 2024-12-25T21:15:13.848946Z 0 [Note] InnoDB: Completed initialization of buffer pool
root-db-1 2024-12-25T21:15:13.850853Z 0 [Note] InnoDB: If the mysqld execution user is authorized, page cleaner thread priority can be changed. See the man page of setpriority().
root-db-1 2024-12-25T21:15:13.860406Z 0 [Note] InnoDB: Highest supported file format is Barracuda.
root-db-1 2024-12-25T21:15:13.871436Z 0 [Note] InnoDB: Creating shared tablespace for temporary tables
root-db-1 2024-12-25T21:15:13.871687Z 0 [Note] InnoDB: Setting file './ibtmp1' size to 12 MB. Physically writing the file full; Please wait ...
root-db-1 2024-12-25T21:15:13.900739Z 0 [Note] InnoDB: File './ibtmp1' size is now 12 MB.
root-db-1 2024-12-25T21:15:13.901531Z 0 [Note] InnoDB: 96 redo rollback segment(s) found. 96 redo rollback segment(s) are active.
root-db-1 2024-12-25T21:15:13.901546Z 0 [Note] InnoDB: 32 non-redo rollback segment(s) are active.
root-db-1 2024-12-25T21:15:13.901881Z 0 [Note] InnoDB: Waiting for purge to start
root-db-1 2024-12-25T21:15:13.952070Z 0 [Note] InnoDB: 5.7.44 started; log sequence number 12219419
root-db-1 2024-12-25T21:15:13.952485Z 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
root-db-1 2024-12-25T21:15:13.952504Z 0 [Note] Plugin 'FEDERATED' is disabled.
root-db-1 2024-12-25T21:15:13.955081Z 0 [Note] InnoDB: Buffer pool(s) load completed at 241225 21:15:13
root-db-1 2024-12-25T21:15:13.958962Z 0 [Note] Found ca.pem, server-cert.pem and server-key.pem in data directory. Trying to enable SSL support using them.
root-db-1 2024-12-25T21:15:13.958914Z 0 [Note] Skipping generation of SSL certificates as certificate files are present in data directory.
root-db-1 2024-12-25T21:15:13.958918Z 0 [Warning] A deprecated TLS version TLSv1 is enabled. Please use TLSv1.2 or higher.
root-db-1 2024-12-25T21:15:13.958921Z 0 [Warning] A deprecated TLS version TLSv1.1 is enabled. Please use TLSv1.2 or higher.
root-db-1 2024-12-25T21:15:13.959487Z 0 [Warning] CA certificate ca.pem is self signed.
root-db-1 2024-12-25T21:15:13.959588Z 0 [Note] Skipping generation of RSA key pair as key files are present in data directory.
root-db-1 2024-12-25T21:15:13.959820Z 0 [Note] Server hostname (bind-address): '*'; port: 3306
root-db-1 2024-12-25T21:15:13.959932Z 0 [Note] IPv6 is available.
root-db-1 2024-12-25T21:15:13.960013Z 0 [Note] ' ':: resolves to '::'.
root-db-1 2024-12-25T21:15:13.960092Z 0 [Note] Server socket created on IP: '::'.
root-db-1 2024-12-25T21:15:13.961630Z 0 [Warning] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.
root-db-1 2024-12-25T21:15:13.969688Z 0 [Note] Event Scheduler: Loaded 0 events
root-db-1 2024-12-25T21:15:13.969923Z 0 [Note] mysqld: ready for connections.
Version: '5.7.44' socket: '/var/run/mysqld/mysqld.sock' port: 3306 MySQL Community Server (GPL)
```

Open your browser and navigate to <http://public-ip:8080/>

Mail - Farsaan Siddiqui - x Online Courses - Learn Ai x Inbox - syedfarsaan@gmail.com x Connect to instance | EC2 x Farsaan-tech (Farsaan Siddiqui) x WordPress - Installation x

Not secure | 3.208.26.87/8000/wp-admin/install.php?step=1



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

Username

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

Password

Strong

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

Your Email

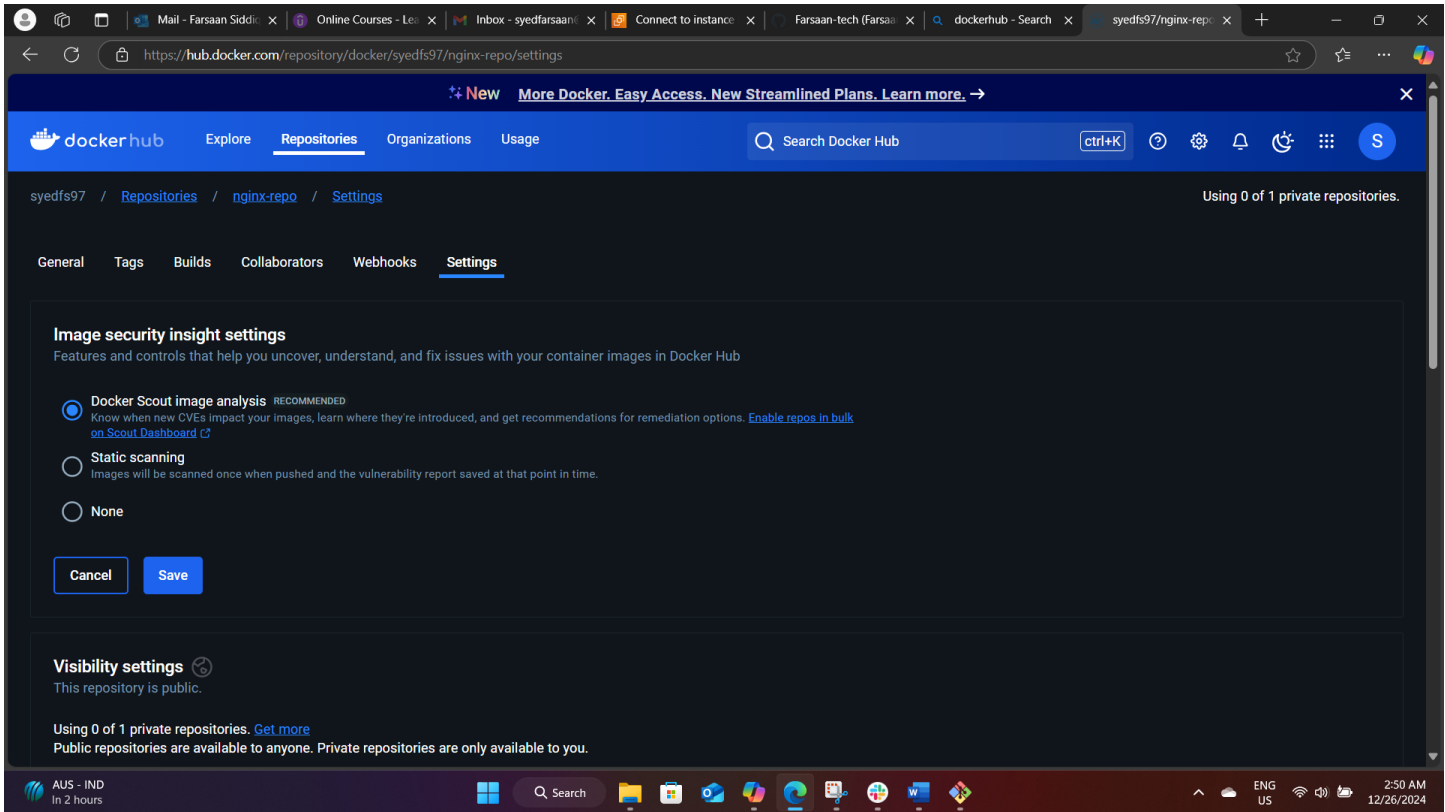
Double-check your email address before continuing.

Search engine ☐ Discourage search engines from indexing this site

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

→ OPEN DOCKERHUB AND NAVIGATE TO REPOSITORY

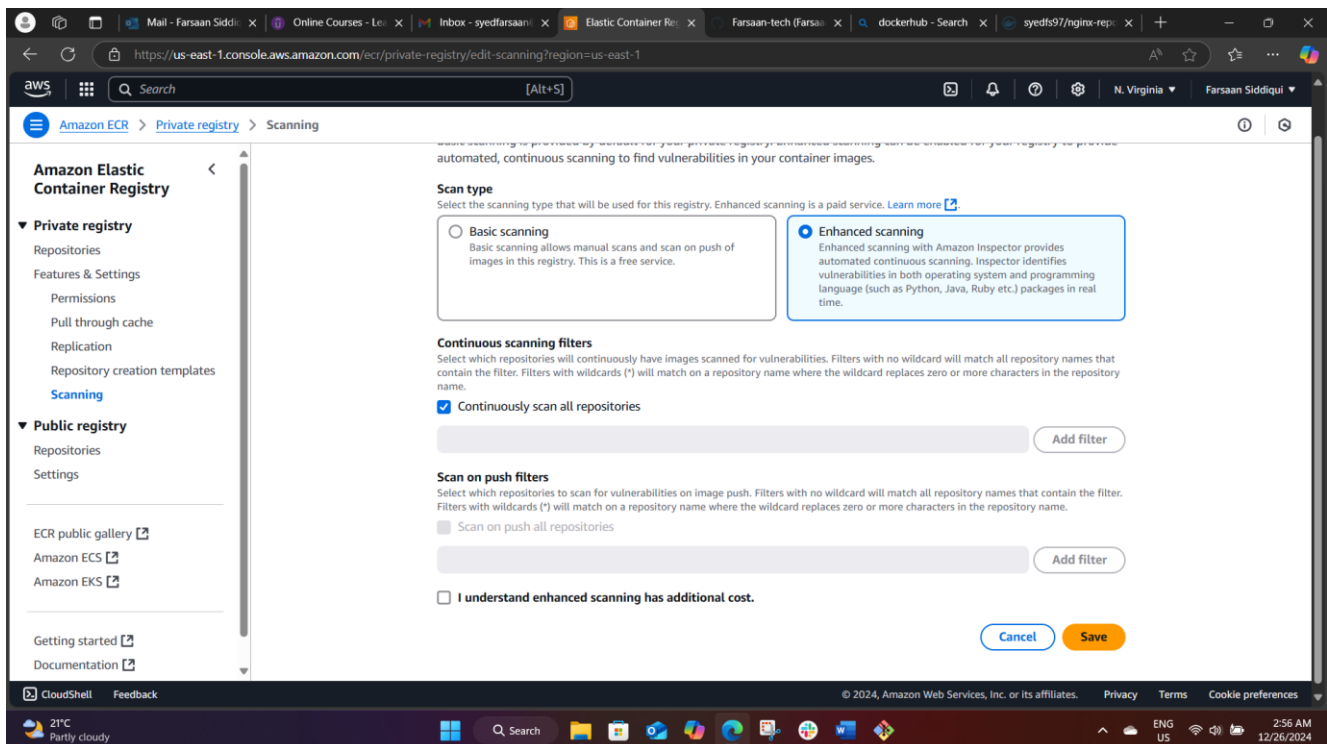
repositories > repo name > settings > Docker Scout image analysis > save



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

→ OPEN ECR IN AWS CONSOLE

amazon ecr > private registry > features and settings > scanning > enhanced scanning > save



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

→ ON JENKINS SERVER INSTALL DOCKER AND START THE DOCKER SERVICE

```
# mkdir my-app
```

```
# cd my-app
```

```
# vi Dockerfile
```

```
FROM nginx:latest
LABEL maintainer="syedfs97@outlook.com"
LABEL description="A Dockerfile for an Nginx web server serving a sample application."

# Update and install any required packages (if needed)
RUN apt-get update && apt-get install -y \
    nano \
    curl

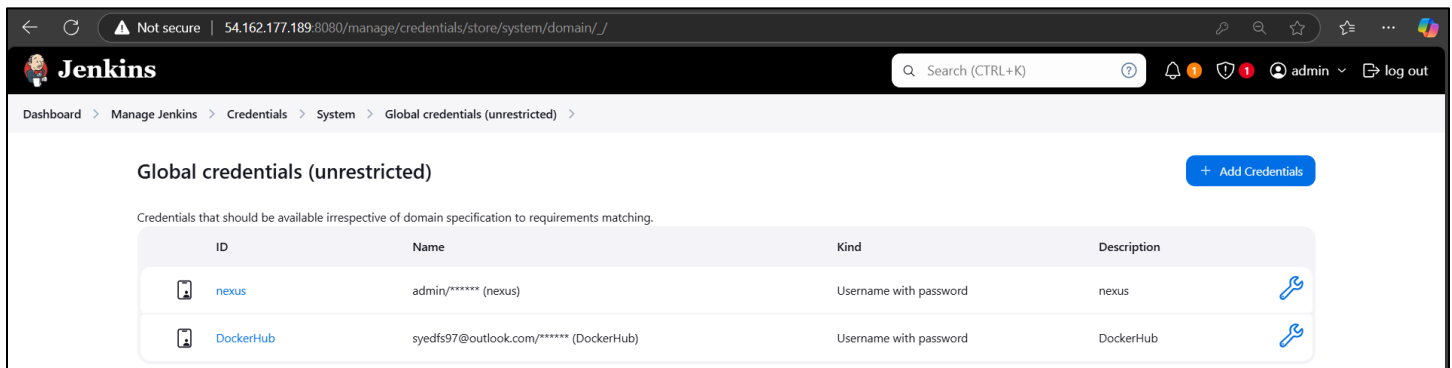
# Expose port 80 to the outside world
EXPOSE 80

# Start Nginx when the container launches
CMD ["nginx", "-g", "daemon off;"]
```

```
# sudo chown -R jenkins:jenkins /home/ec2-user/my-app
```

```
# sudo chmod -R 755 /home/ec2-user/my-app
```

→ ADD CREDENTIALS IN JENKINS GUI



Create a jenkins job

Dashboard > new item > item name > select pipeline > add below script > save

```
pipeline {
  agent any
  environment {
    DOCKERHUB_CREDENTIALS = credentials('DockerHub')
    UNIQUE_TAG = "${new Date().format('yyyyMMddHHmmss')}"
  }
  stages {
    stage('Verify Directory') {
      steps {
        dir('/home/ec2-user/my-app') {
          script {
            sh 'echo "Listing contents of /home/ec2-user/my-app"'
            sh 'ls -al'
          }
        }
      }
    }
  }
}
```



```

}
stage('Build Docker Image') {
  steps {
    dir('/home/ec2-user/my-app') {
      sh 'docker build -t syedfs97/nginx-repo:$UNIQUE_TAG .'
    }
  }
}
stage('Login to Docker Hub') {
  steps {
    sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
  }
}
stage('Push Docker Image') {
  steps {
    sh 'docker push syedfs97/nginx-repo:$UNIQUE_TAG'
  }
}
}
post {
  always {
    sh 'docker logout'
  }
}
}

```

→ BUILD

The screenshot shows the Jenkins Pipeline Console for Build #8. The build is successful, having completed in 1 minute and 42 seconds. The pipeline consists of several stages: 'Create Dockerfile', 'Build Docker Image', 'Login to Docker Hub', 'Push Docker Image', and 'Post Actions'. The 'Post Actions' stage is currently selected, showing a 'docker logout' step that took 0.28 seconds. The console output for this step shows the command being executed and the removal of login credentials for the Docker Hub repository.

Build #8 Success 1 min 42 sec ago in 10 sec

- ✓ Create Dockerfile
- ✓ Build Docker Image
- ✓ Login to Docker Hub
- ✓ Push Docker Image
- ✓ **Post Actions**

Stage 'Post Actions'

- Started 1 min 51 sec ago
- Queued 0 ms
- Took 0.3 sec
- Success
- [View as plain text](#)

docker logout Shell Script 0.28 sec

```

0 + docker logout
1 Removing login credentials for https://index.docker.io/v1/

```

→ CHECK THE DOCKER HUB

The screenshot shows the Docker Hub repository page for 'syedfs97/nginx-repo'. The repository is tagged with '20241225221620'. The page displays the repository's manifest digest, OS/architecture (linux/amd64), compressed size (68.74 MB), last pushed time (2 minutes ago), and type (Image). The 'Image Layers' section is expanded, showing a single layer named 'debian.sh' with a size of 26.92 MB. The command for this layer is '# debian.sh --arch 'amd64' out/ 'bookworm' '01734912000''.

syedfs97/nginx-repo:20241225221620

MANIFEST DIGEST sha256:d9e0c56bbf8f3f2f1b8eca683549b2df6a13bf13897358ed67f3482629eaf42e

OS/ARCH: linux/amd64 | COMPRESSED SIZE: 68.74 MB | LAST PUSHED: 2 minutes ago by syedfs97 | TYPE: Image | MANIFEST DIGEST: sha256:d9e0c56b...

Image Layers

Image Layers

1 # debian.sh --arch 'amd64' out/ 26.92 MB

Command: # debian.sh --arch 'amd64' out/ 'bookworm' '01734912000'