Deploying a Java Web Application Using Docker on EC2

Overview

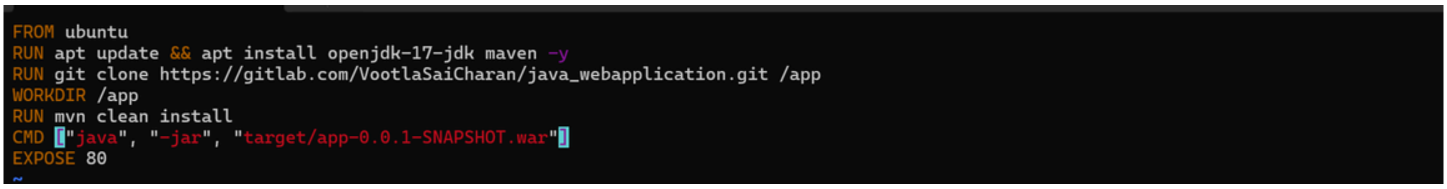
This documentation provides a step-by-step guide for deploying a Java web application using Docker on an AWS EC2 instance. The project demonstrates the setup of an EC2 Linux environment, installation of Docker, and deployment of a Java application using a Dockerfile.

Step 1: Setting Up the EC2 Instance

1. **Launch EC2 Instance:**
   - Navigate to the EC2 dashboard on AWS.
   - Select "Launch Instance" and choose an Amazon Linux AMI.
   - Configure the instance type, security group, and storage.

Step 2: Installing Docker

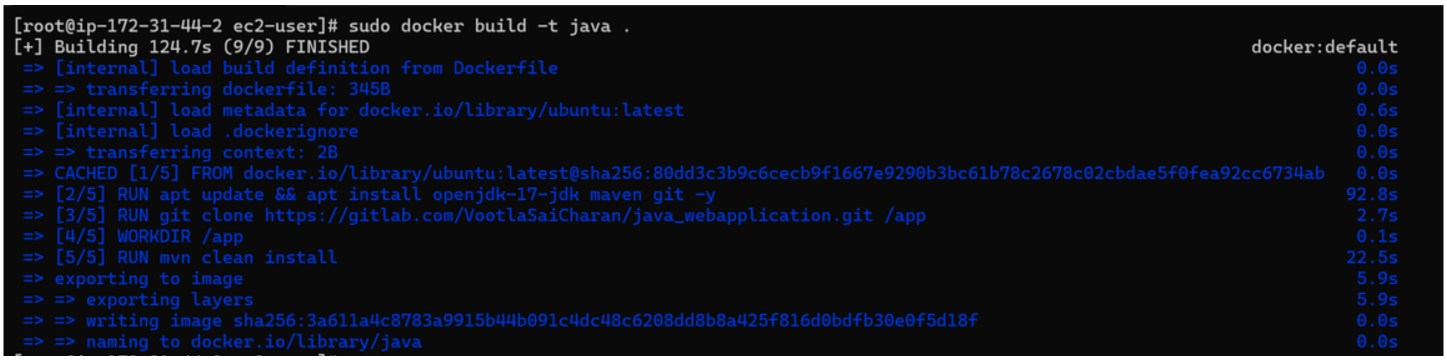
1. sudo yum install docker -y
2. sudo service docker start
3. vi Dockerfile



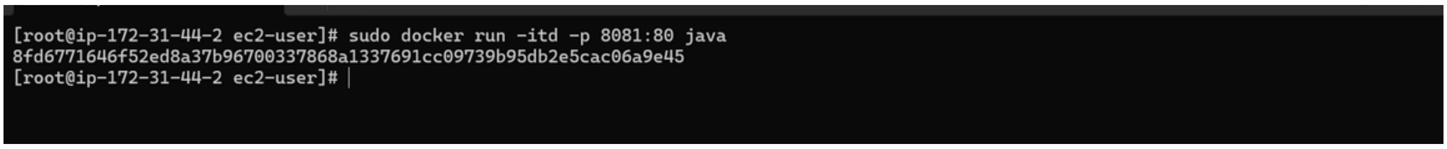
```
FROM ubuntu
RUN apt update && apt install openjdk-17-jdk maven -y
RUN git clone https://gitlab.com/VootlaSaiCharan/java_webapplication.git /app
WORKDIR /app
RUN mvn clean install
CMD ["java", "-jar", "target/app-0.0.1-SNAPSHOT.war"]
EXPOSE 80
~
```

Building and Running the Docker Image

1. sudo docker build -t java .
2. sudo docker run -idt -p 8081:80 java



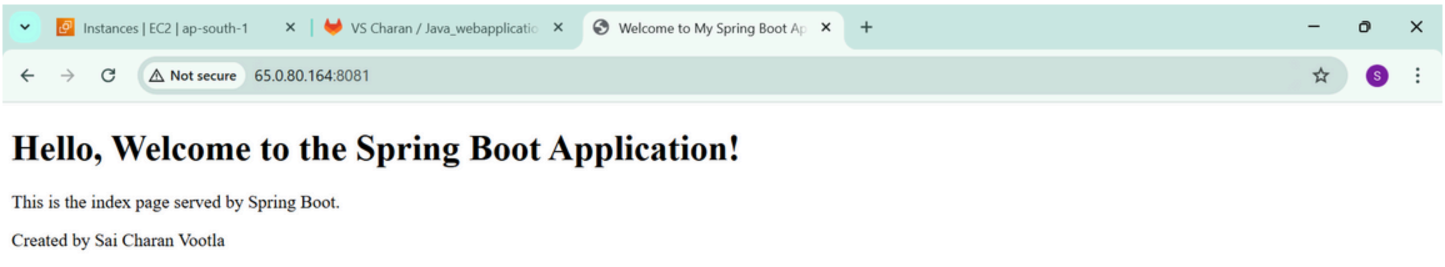
```
[root@ip-172-31-44-2 ec2-user]# sudo docker build -t java .
[+] Building 124.7s (9/9) FINISHED                                                                          docker:default
 => [internal] load build definition from Dockerfile                                                                  0.0s
 => => transferring dockerfile: 345B                                                                                  0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                                      0.6s
 => [internal] load .dockerignore                                                                                     0.0s
 => => transferring context: 2B                                                                                       0.0s
 => CACHED [1/5] FROM docker.io/library/ubuntu:latest@sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab  0.0s
 => [2/5] RUN apt update && apt install openjdk-17-jdk maven git -y                                                  92.8s
 => [3/5] RUN git clone https://gitlab.com/VootlaSaiCharan/java_webapplication.git /app                               2.7s
 => [4/5] WORKDIR /app                                                                                                0.1s
 => [5/5] RUN mvn clean install                                                                                      22.5s
 => exporting to image                                                                                                5.9s
 => => exporting layers                                                                                               5.9s
 => => writing image sha256:3a611a4c8783a9915b44b091c4dc48c6208dd8b8a425f816d0bdfb30e0f5d18f                          0.0s
 => => naming to docker.io/library/java                                                                               0.0s
```



```
[root@ip-172-31-44-2 ec2-user]# sudo docker run -itd -p 8081:80 java
8fd6771646f52ed8a37b96700337868a1337691cc09739b95db2e5cac06a9e45
[root@ip-172-31-44-2 ec2-user]# |
```

Testing the Application

1. **Access the Application:**
   - Open a browser and navigate to http://<public-ip>. Ensure that port 80 is open in the security group.
2. **Verify Functionality:**
   - Confirm that the application is running and accessible.


Instances | EC2 | ap-south-1
VS Charan / Java_webapplicatio
Welcome to My Spring Boot Ap
Not secure
65.0.80.164:8081

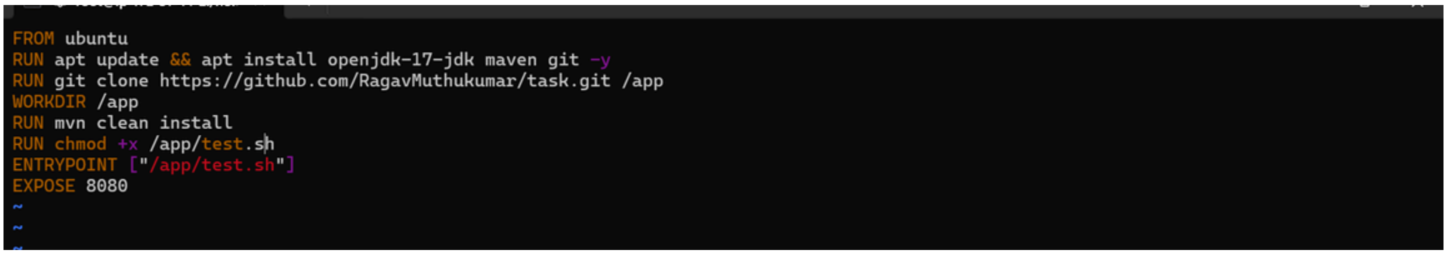
Hello, Welcome to the Spring Boot Application!

This is the index page served by Spring Boot.

Created by Sai Charan Vootla

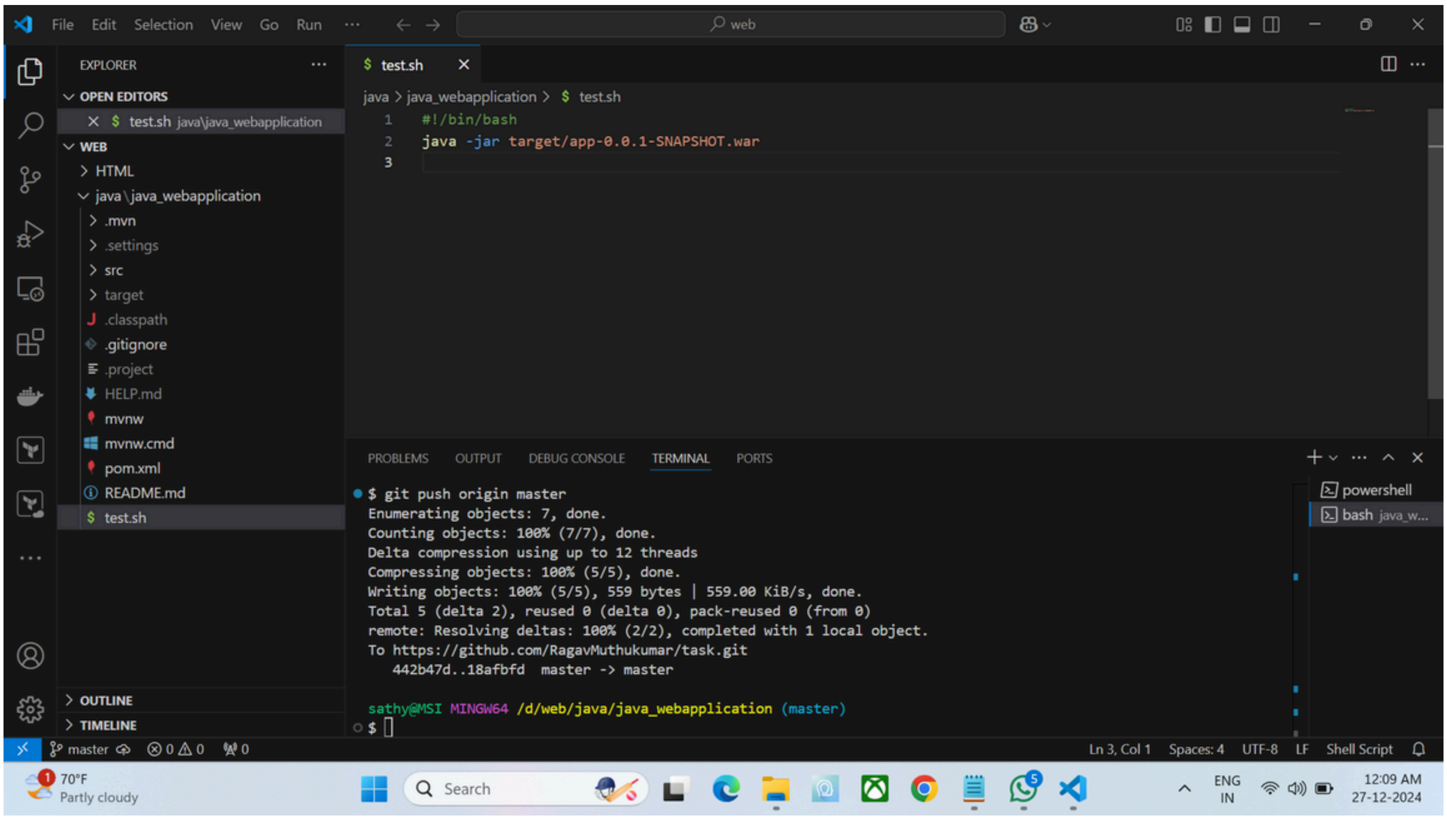
Deploying a Java Web Application with Custom Entry Point Using Docker on EC2

Same steps but you need to add a .sh file in github repo



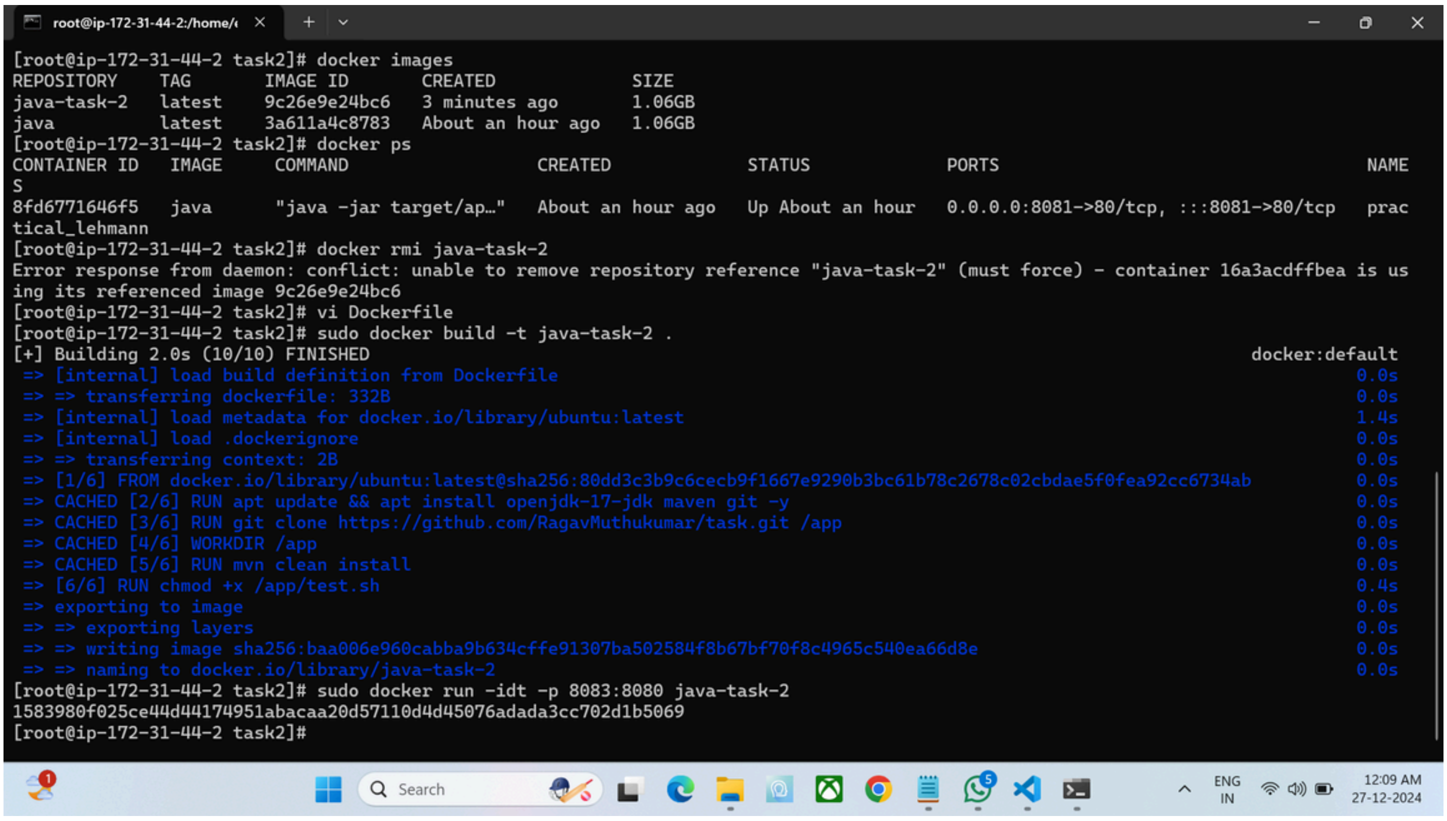
```
FROM ubuntu
RUN apt update && apt install openjdk-17-jdk maven git -y
RUN git clone https://github.com/RagavMuthukumar/task.git /app
WORKDIR /app
RUN mvn clean install
RUN chmod +x /app/test.sh
ENTRYPOINT ["/app/test.sh"]
EXPOSE 8080
~
~
```

test.sh

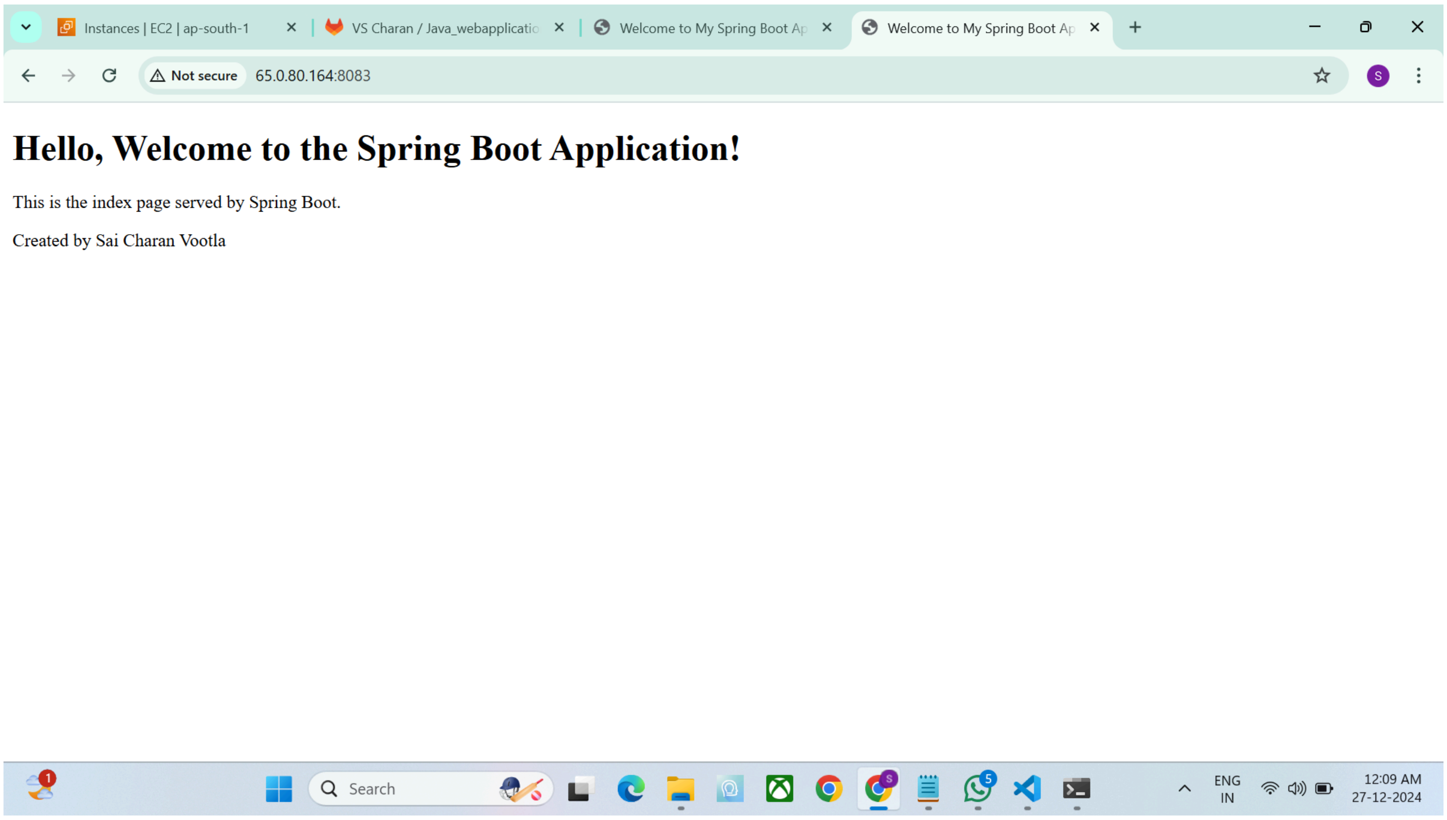

test.sh
java > java_webapplication > $ test.sh
EXPLORER
OPEN EDITORS
test.sh java\java_webapplication
WEB
HTML
java \ java_webapplication
.mvn
.settings
src
target
.classpath
.gitignore
.project
HELP.md
mvnw
mvnw.cmd
pom.xml
README.md
test.sh
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
bash java_w...
master
Ln 3, Col 1 Spaces: 4 UTF-8 LF Shell Script

```
#!/bin/bash
java -jar target/app-0.0.1-SNAPSHOT.war

```

```
$ git push origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 559 bytes | 559.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/RagavMuthukumar/task.git
   442b47d..18afbfd  master -> master

sathy@MSI MINGW64 /d/web/java/java_webapplication (master)
$ 
```



```
[root@ip-172-31-44-2 task2]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
java-task-2   latest    9c26e9e24bc6   3 minutes ago    1.06GB
java          latest    3a611a4c8783   About an hour ago   1.06GB
[root@ip-172-31-44-2 task2]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED             STATUS              PORTS                                     NAME
S
8fd6771646f5   java      "java -jar target/ap…"   About an hour ago   Up About an hour    0.0.0.0:8081->80/tcp, :::8081->80/tcp     prac
tical_lehmann
[root@ip-172-31-44-2 task2]# docker rmi java-task-2
Error response from daemon: conflict: unable to remove repository reference "java-task-2" (must force) - container 16a3acdffbea is us
ing its referenced image 9c26e9e24bc6
[root@ip-172-31-44-2 task2]# vi Dockerfile
[root@ip-172-31-44-2 task2]# sudo docker build -t java-task-2 .
[+] Building 2.0s (10/10) FINISHED                                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                                          0.0s
 => => transferring dockerfile: 332B                                                                                          0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                                              1.4s
 => [internal] load .dockerignore                                                                                             0.0s
 => => transferring context: 2B                                                                                               0.0s
 => [1/6] FROM docker.io/library/ubuntu:latest@sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab         0.0s
 => CACHED [2/6] RUN apt update && apt install openjdk-17-jdk maven git -y                                                    0.0s
 => CACHED [3/6] RUN git clone https://github.com/RagavMuthukumar/task.git /app                                               0.0s
 => CACHED [4/6] WORKDIR /app                                                                                                 0.0s
 => CACHED [5/6] RUN mvn clean install                                                                                        0.0s
 => [6/6] RUN chmod +x /app/test.sh                                                                                           0.4s
 => exporting to image                                                                                                        0.0s
 => => exporting layers                                                                                                       0.0s
 => => writing image sha256:baa006e960cabba9b634cffe91307ba502584f8b67bf70f8c4965c540ea66d8e                                  0.0s
 => => naming to docker.io/library/java-task-2                                                                                0.0s
[root@ip-172-31-44-2 task2]# sudo docker run -idt -p 8083:8080 java-task-2
1583980f025ce44d44174951abacaa20d57110d4d45076adada3cc702d1b5069
[root@ip-172-31-44-2 task2]#
```


Hello, Welcome to the Spring Boot Application!

This is the index page served by Spring Boot.

Created by Sai Charan Vootla

Instances | EC2 | ap-south-1
VS Charan / Java_webapplicatio
Welcome to My Spring Boot Ap
Welcome to My Spring Boot Ap
Not secure
65.0.80.164:8083
Search
ENG IN
12:09 AM
27-12-2024

if you want to save a storage use this method:

Dockerfile:

```
FROM maven AS build

WORKDIR /app

COPY . /app

RUN mvn clean install



FROM openjdk:17-alpine

WORKDIR /test

COPY --from=build /app/target/*.war /test

CMD ["java", "-jar", "app-0.0.1-SNAPSHOT.war"]

EXPOSE 8080
```