Objective:

1. Create and download a Spring Boot application with Spring initializer.
2. Import the application into Eclipse.
3. Create controller class with endpoint.
4. Run and verify the application.
5. Write a Docker file to containerize the application
6. Build the docker image.
7. Push the docker image to Docker Hub
8. Run on application as container in detached mode and on system port 8089

Solution:-

Step 1: Installing docker and on server

Installation Source links :

Docker - https://docs.docker.com/engine/install/ubuntu/

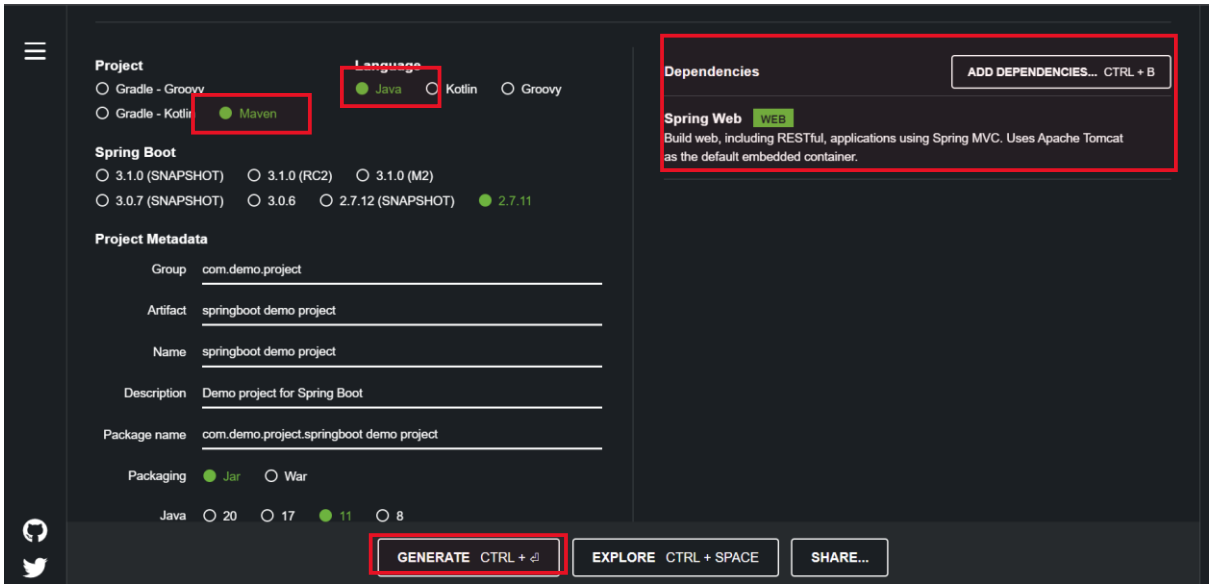
Maven: apt install maven

Java : apt install openjdk-11-jdk

Generating a spring boot project from spring initializer site

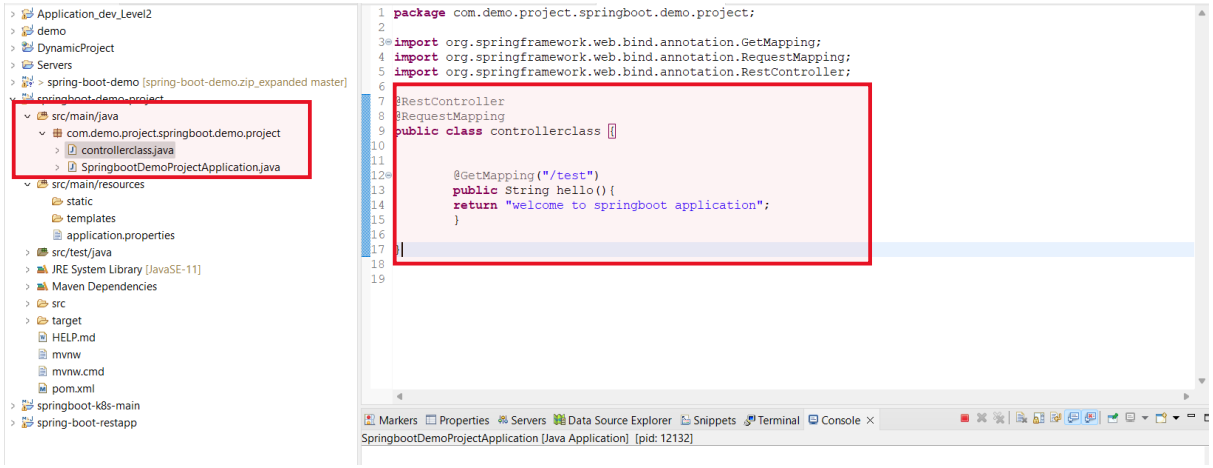
https://start.spring.io/

Add the **spring web** dependency.



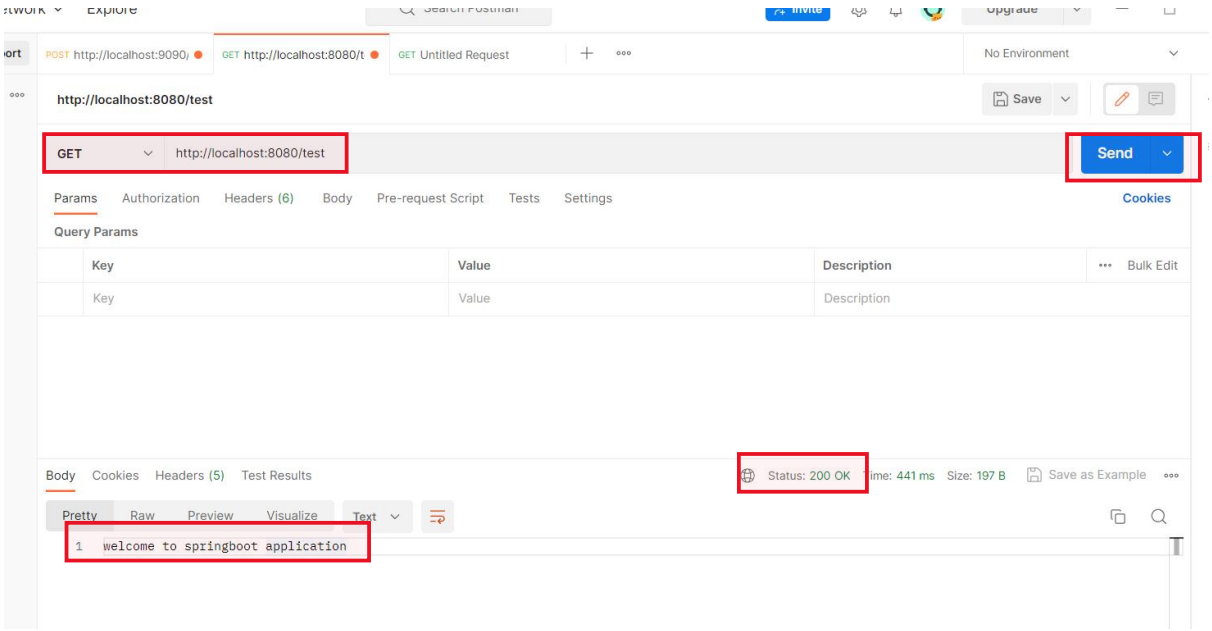
Importing into eclipse IDE

And creating a controller class which acts as a rest application



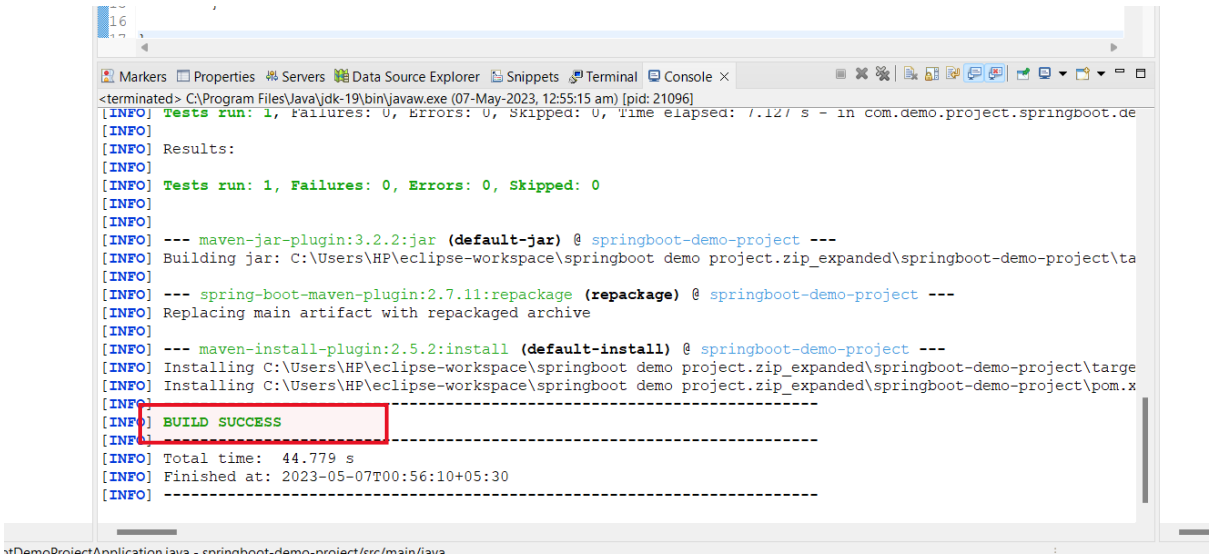
Running the application on local machine by using postman

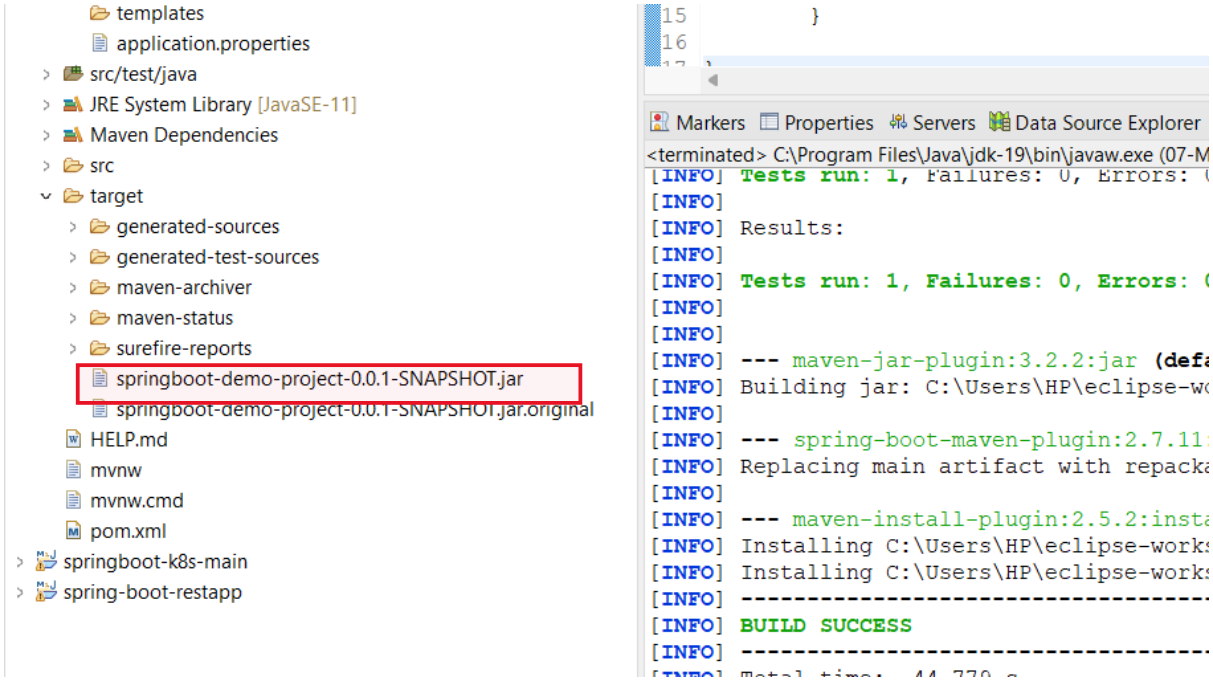
http://localhost:8080/test



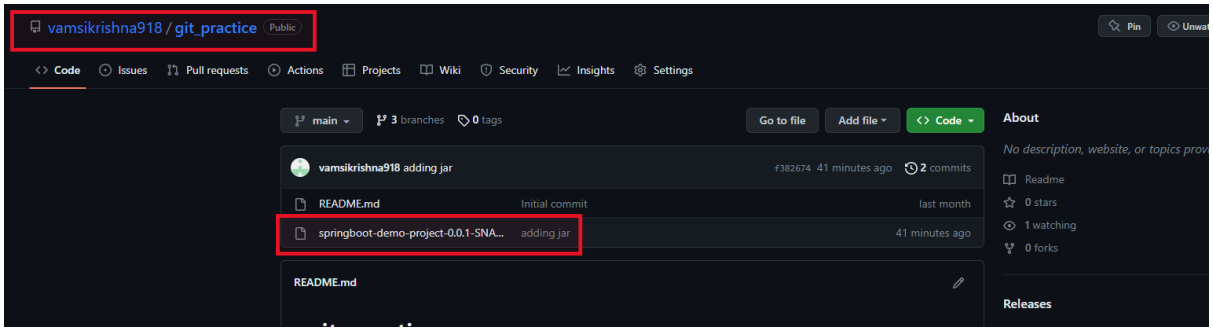
Building the application with **mvn clean insatll**

With which we can get a jar file



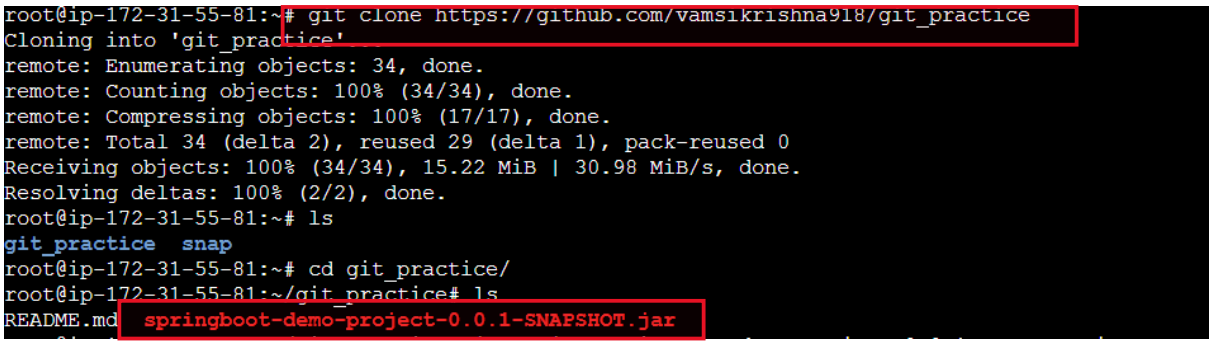


I have imported the jar file into **git hub** and **clone the repo in ubuntu machine**



Cloning the repository to get the jar

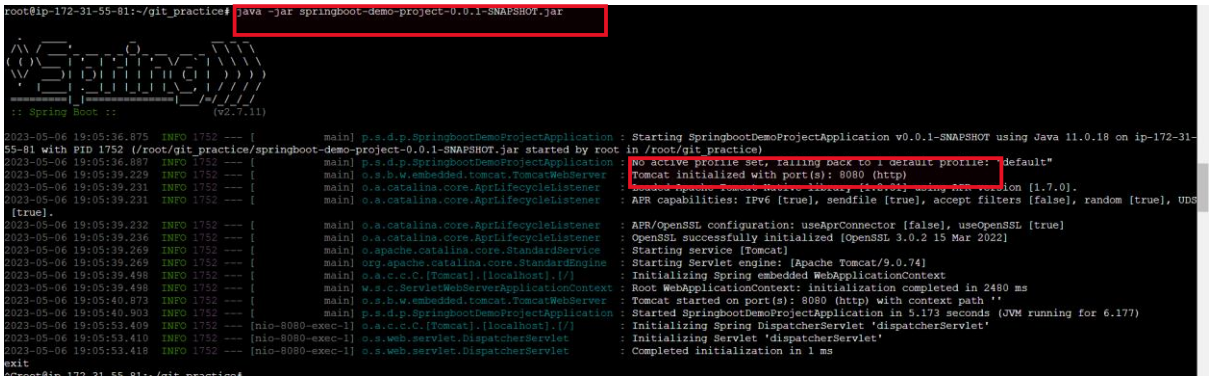
Git clone https://github.com/vamsikrishna918/git_practice



Running the jar

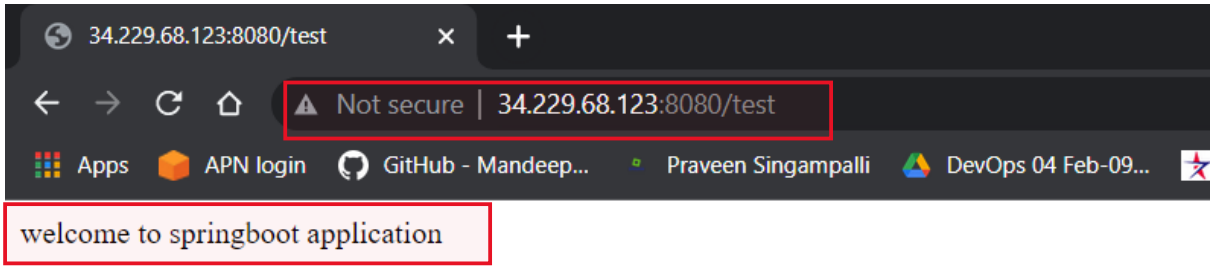
Command :

Java -jar springboot-demo-project.jar



Accessing the application

http://Public ip:8080



Creating the Docker file:

Command :

Vi Dockerfile

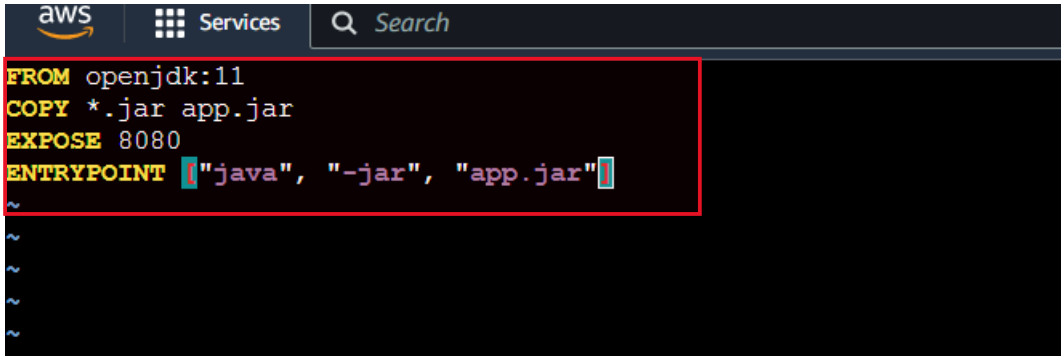
Insert below:

FROM openjdk:11

COPY *.jar app.jar

EXPOSE 8080

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



Building the docker image;

Command :

docker build -t springboot-demo .

Verifying the docker images

Command :

Docker images

```
root@ip-172-31-55-81:~/git practice# docker images
REPOSITORY                 TAG       IMAGE ID       CREATED          SIZE
springboot-demo            latest    9583f6ab5103   46 seconds ago   672MB
vamsi12358/addressbook     1.0       0d7bce3f708c   3 hours ago      491MB
root@ip-172-31-55-81:~/git_practice# docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
```

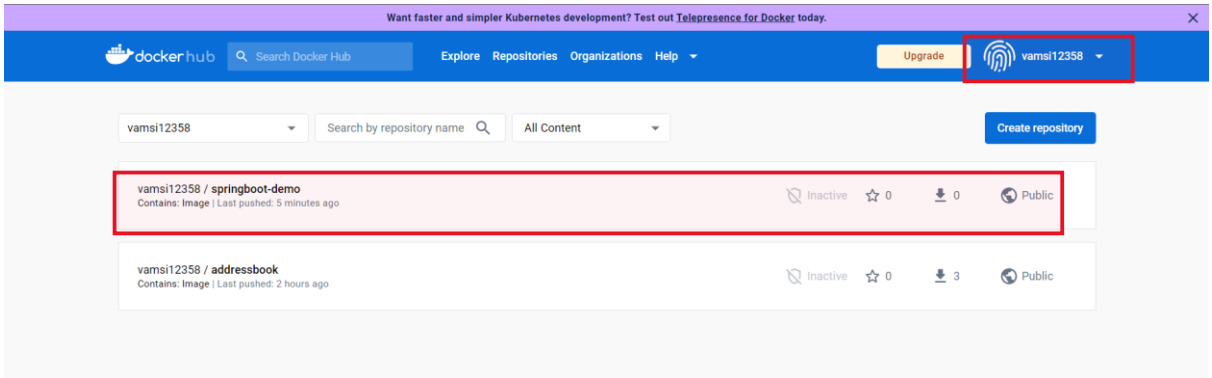
Login in to docker hub

```
root@ip-172-31-55-81:~/git_practice# docker login
Authenticating with existing credentials...
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-172-31-55-81:~/git_practice# docker push springboot-demo
Using default tag: latest
```

Pushing the docker images into docker hub;

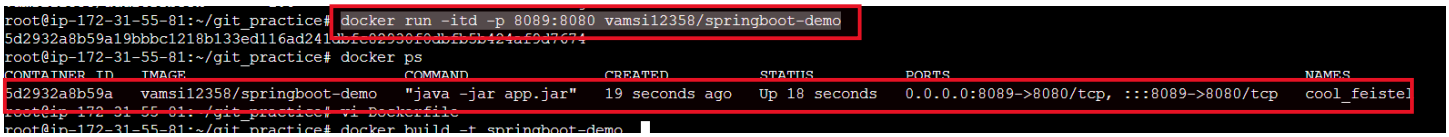
```
root@ip-172-31-55-81:~/git_practice# docker tag  springboot-demo vamsi12358/springboot-demo
root@ip-172-31-55-81:~/git_practice# docker push vamsi12358/springboot-demo
Using default tag: latest
The push refers to repository [docker.io/vamsi12358/springboot-demo]
d56d81d8976d: Pushed
7b7f3078e1db: Pushed
826c3ddbb29c: Pushed
b626401ef603: Pushed
9b55156abf26: Pushed
293d5db30c9f: Pushed
03127cdb479b: Pushed
9c742cd6c7a5: Pushed
latest: digest: sha256:1b018096dd5307d193c4bf5bdd6f1eb7e8f4a5f58d5a3edfabca62ff2c2bd23a size: 2007
```



Running the spring boot image with **port binding 8089**

Command :

docker run -itd -p 8089:8080 vamsi12358/springboot-demo



Verifing the spring boot application on port 8089

http://Public ip:8089/test