**CI/CD Deployment for Springboot Application**

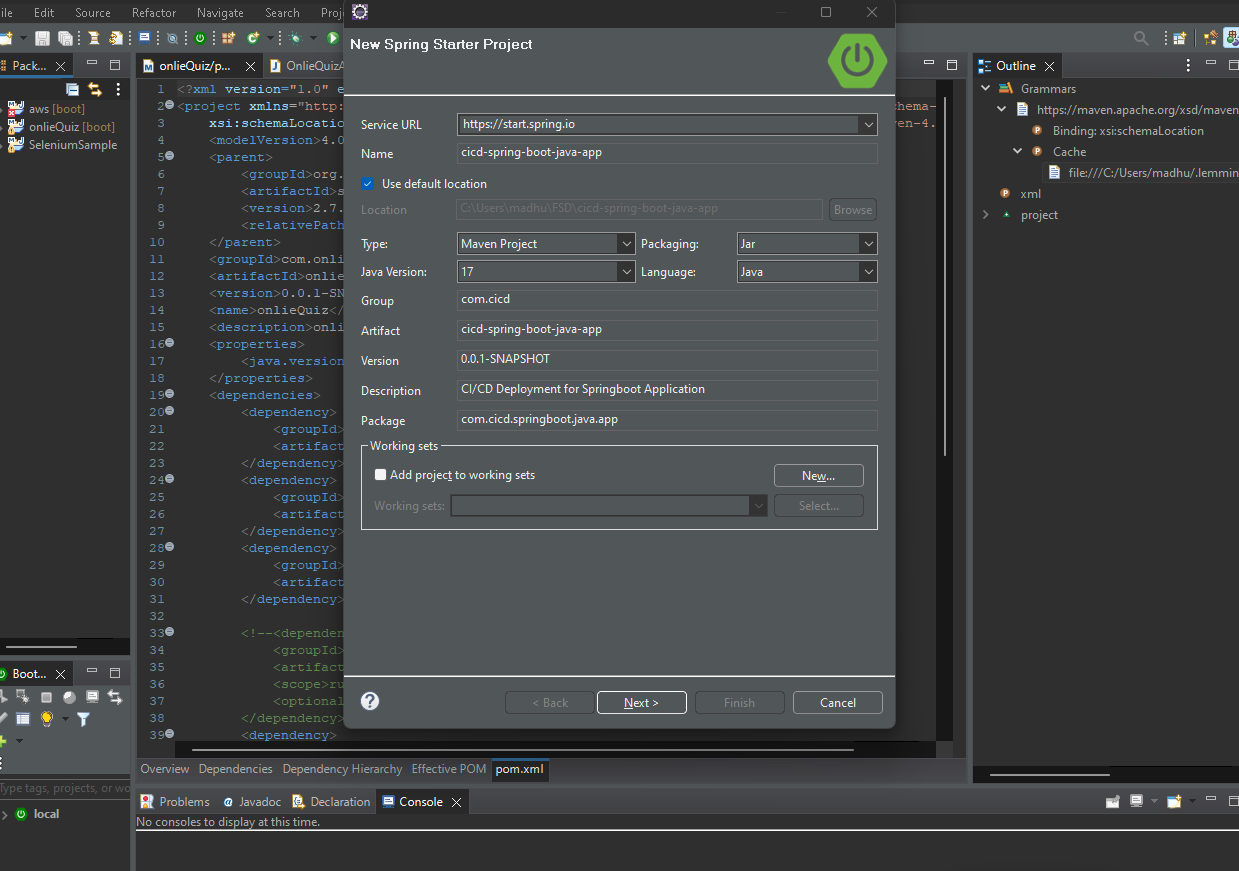
**Description:**

I have to build a CI/CD pipeline to demonstrate continuous deployment and host the application on AWS EC2 instance.

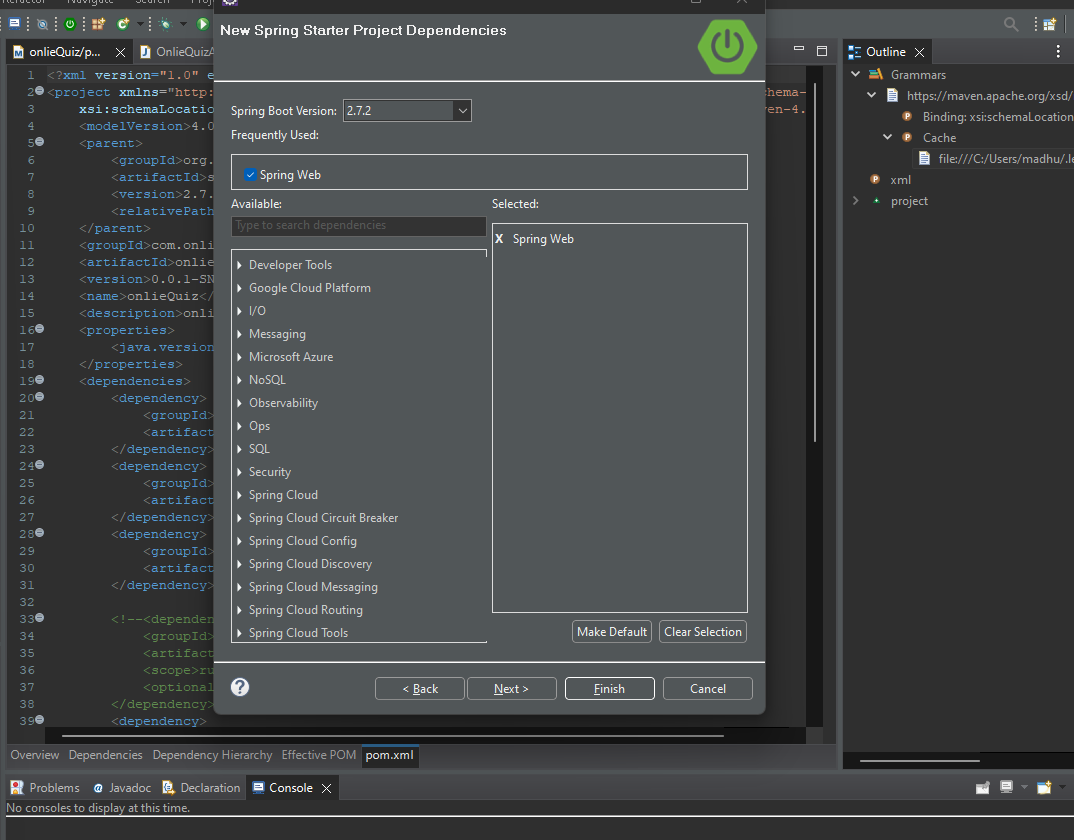
My approach is: **Spring boot Java App => AWS S3 =>AWS EC2=> Deployment**

Step 1: Creating Spring boot java app using Spring Starter Project

**File -> New -> Spring Starter Project**

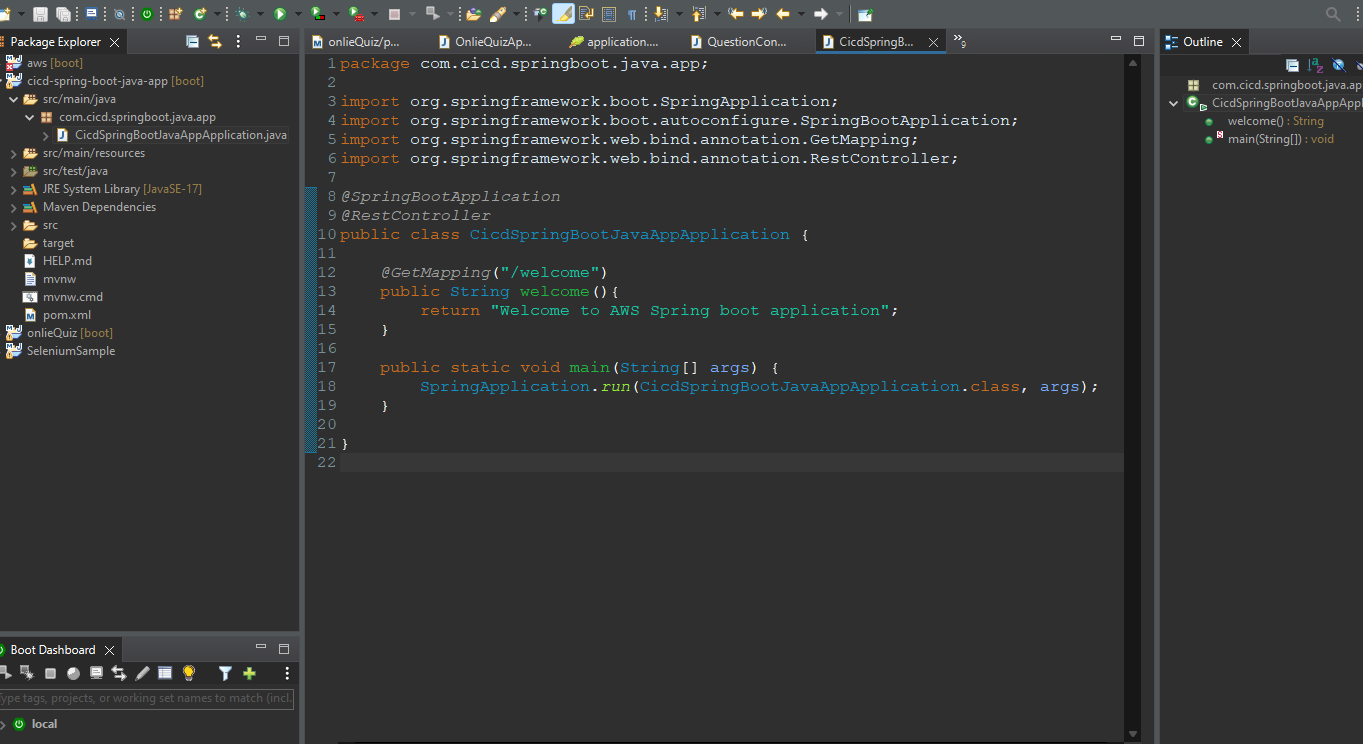


Click **Next,** and select **spring Web** then **finish.**



After creating spring boot java app, open **CicdSpriongBootJavaApplication.java** file

Here we add simple basic endpoint:

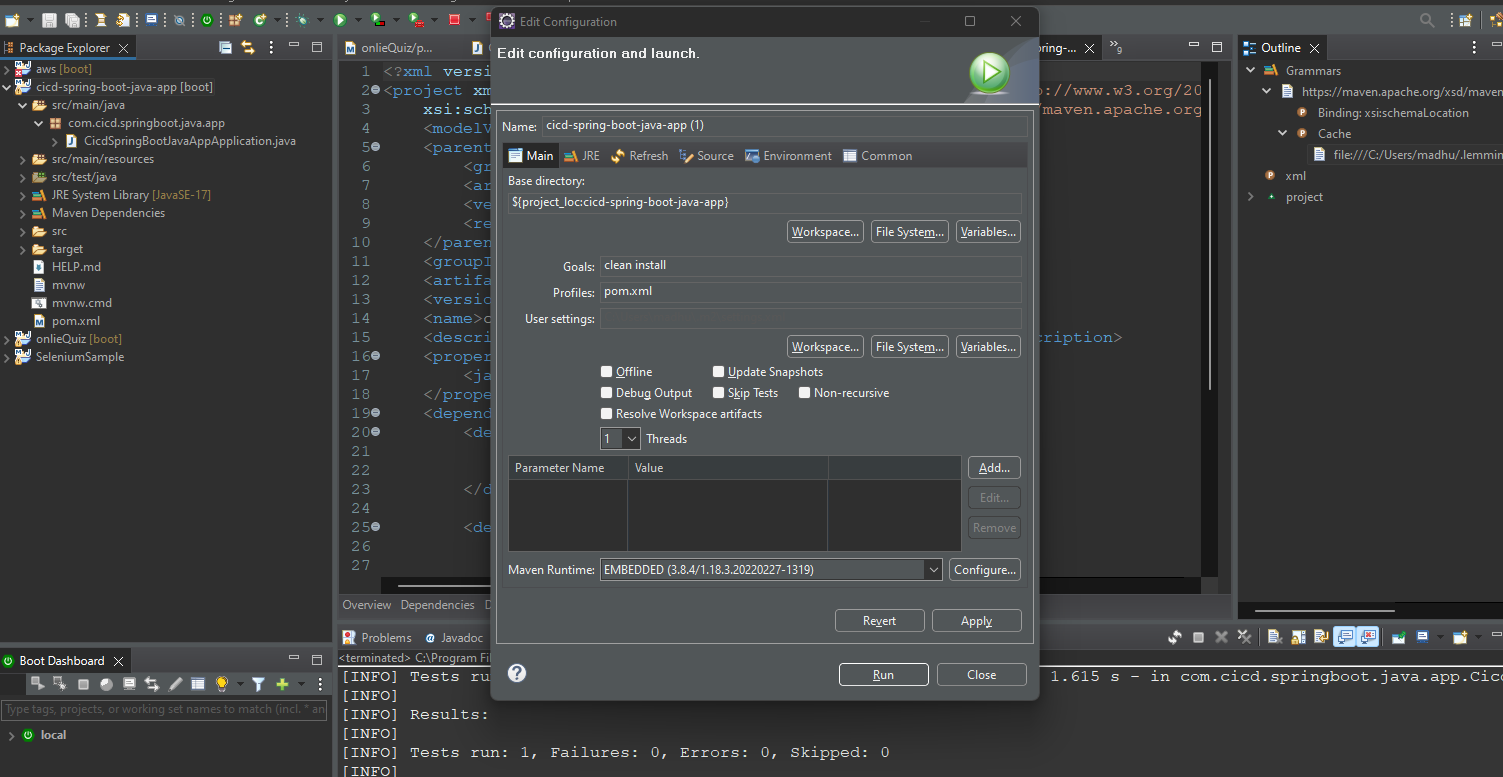


Now Run Spring boot application on 8080 port <http://localhost:8080/welcome>

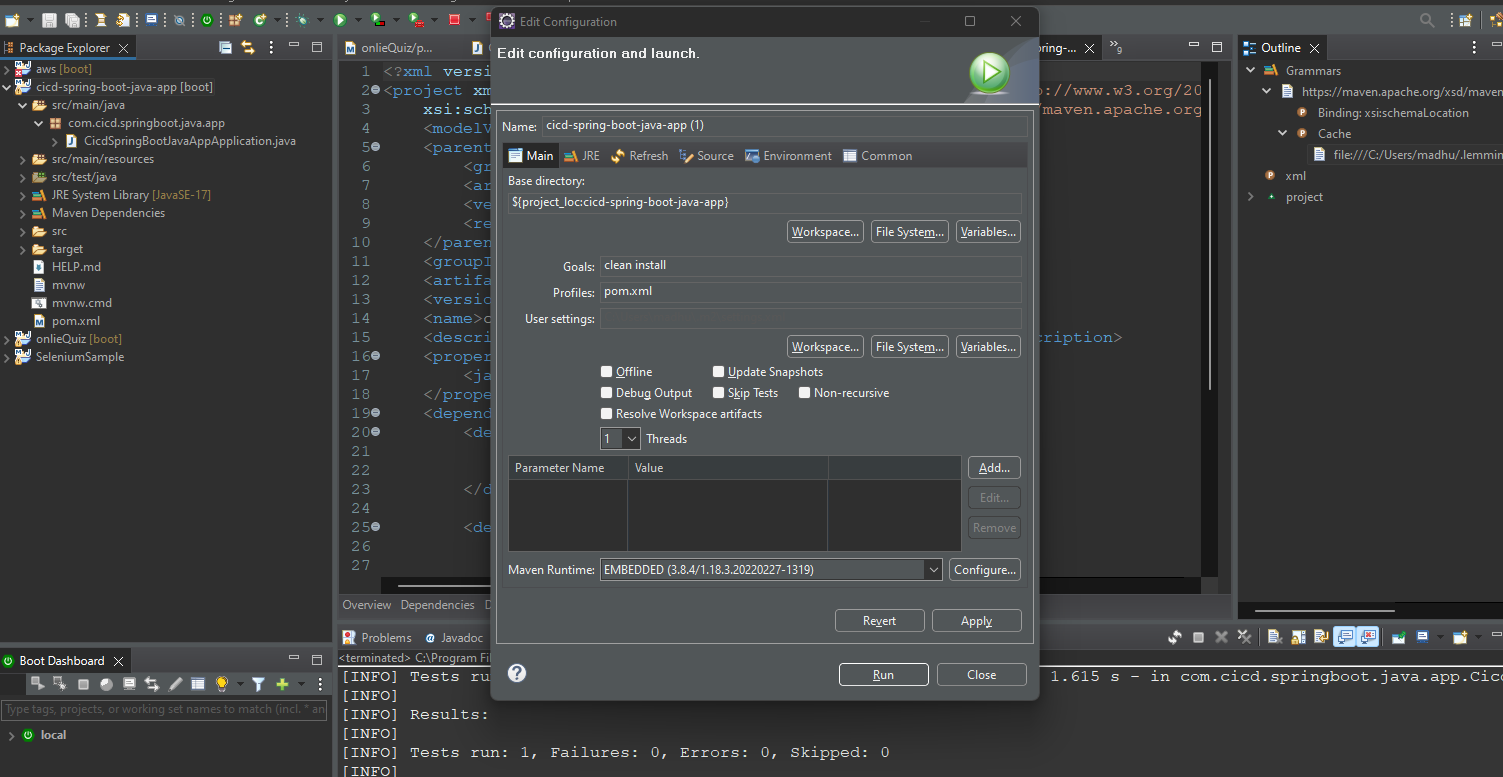


My spring boot project running successfully.

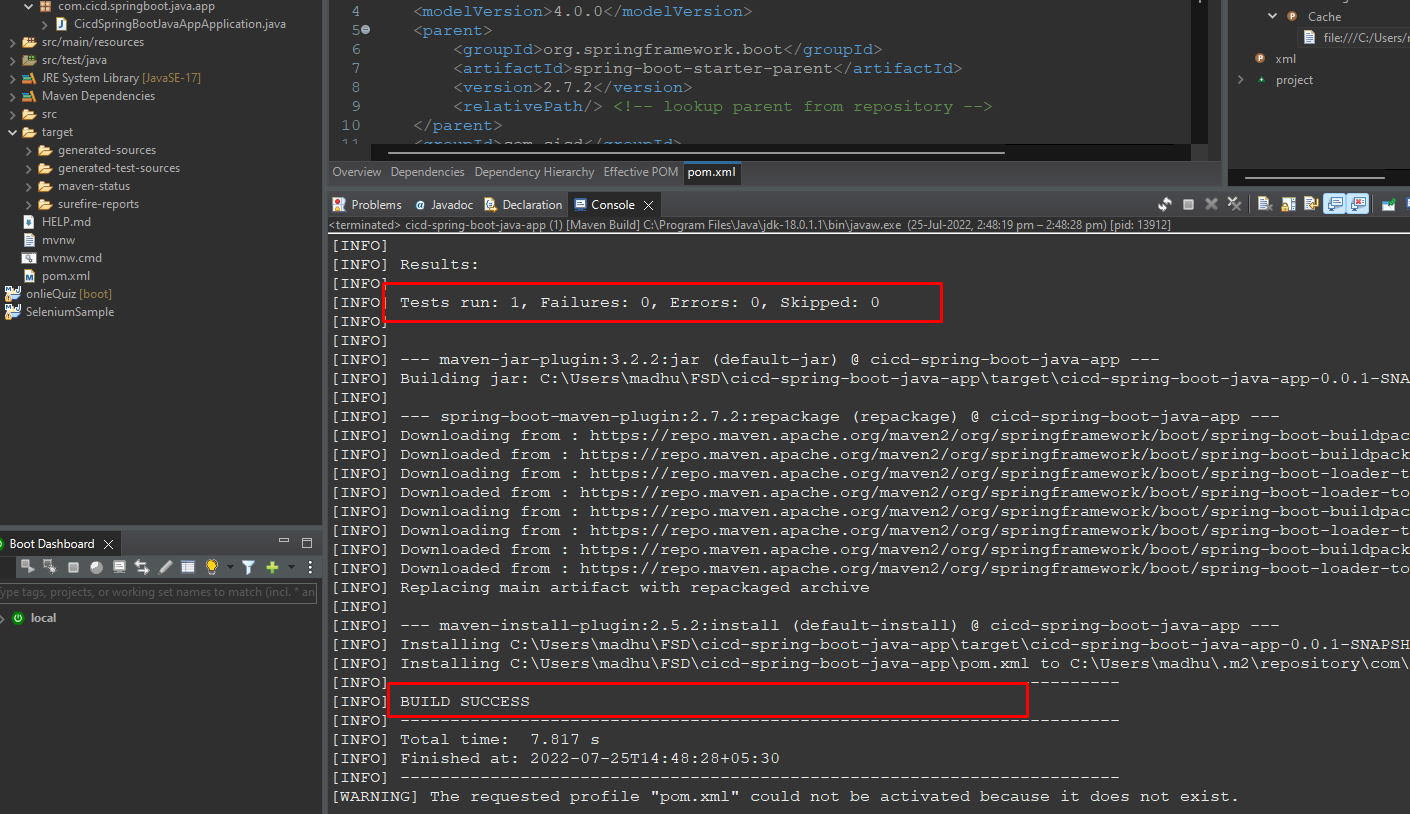
Go to project -> Right Click –> Run As -> Maven Build



Set as goals: **clean install** then apply and run

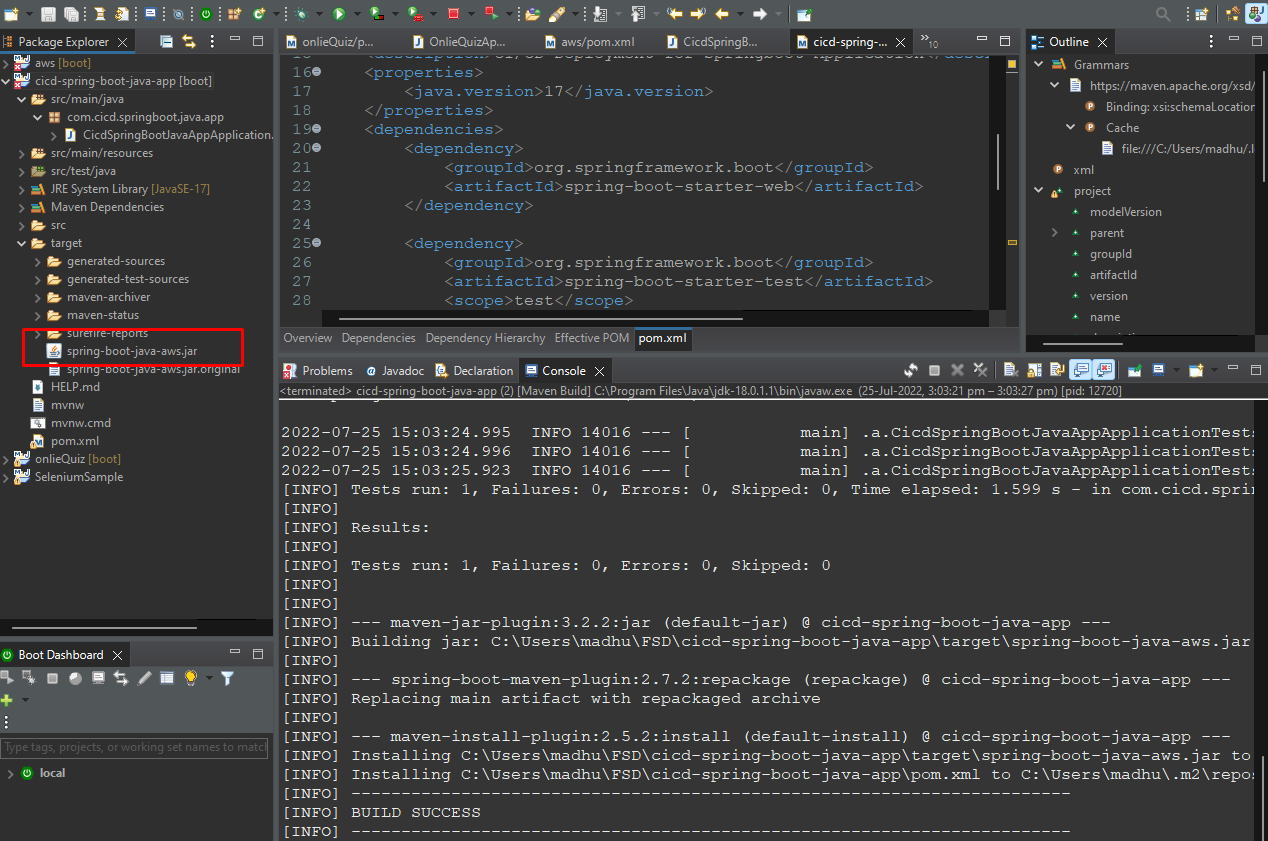


After run, the project will be build, run and execute all test cases.



Build status: **Success**

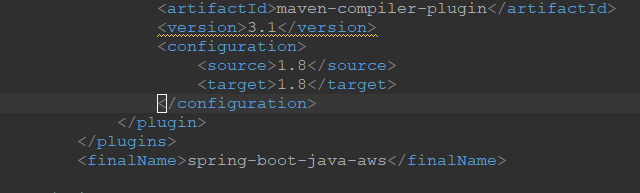
After completing build refresh project then it will appear one target folder.



We will get one jar file.

That is I gave it in pom.xml as

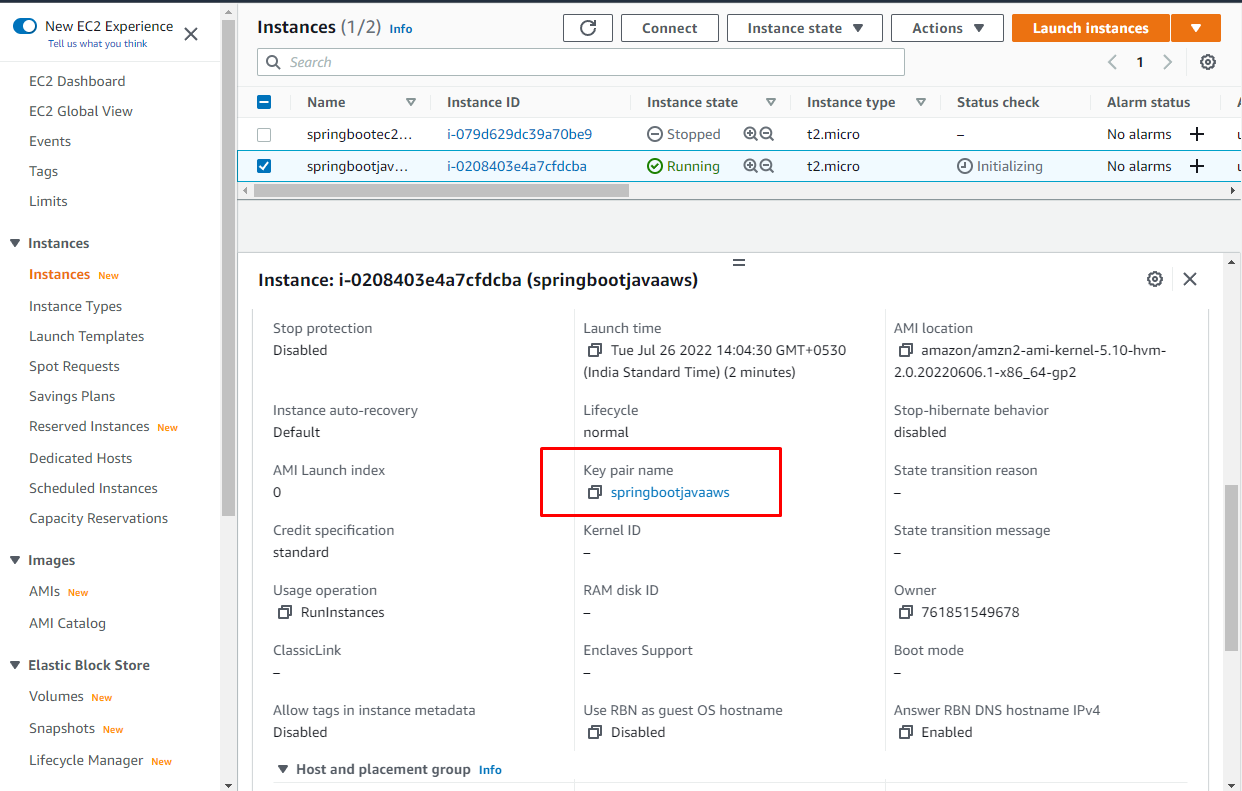
**<finalName>spring-boot-java-aws</finalName>**



Now my spring boot application is working successfully on port 8080.

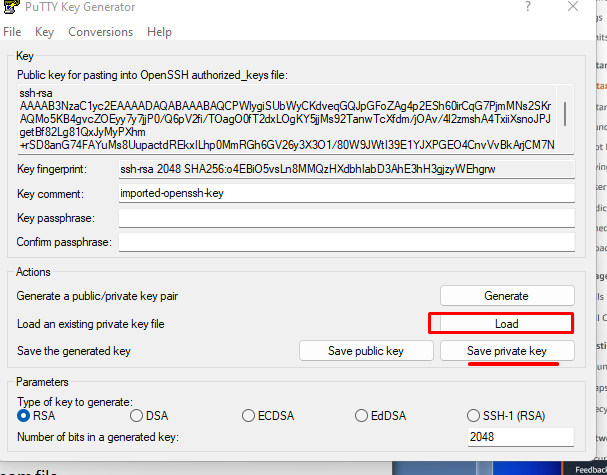
Step 2: Creating AWS EC2 instance

I created one Ec2 instance

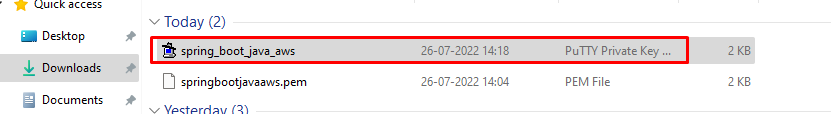


I downloaded key pair **springbootjavaaws.pem** file.

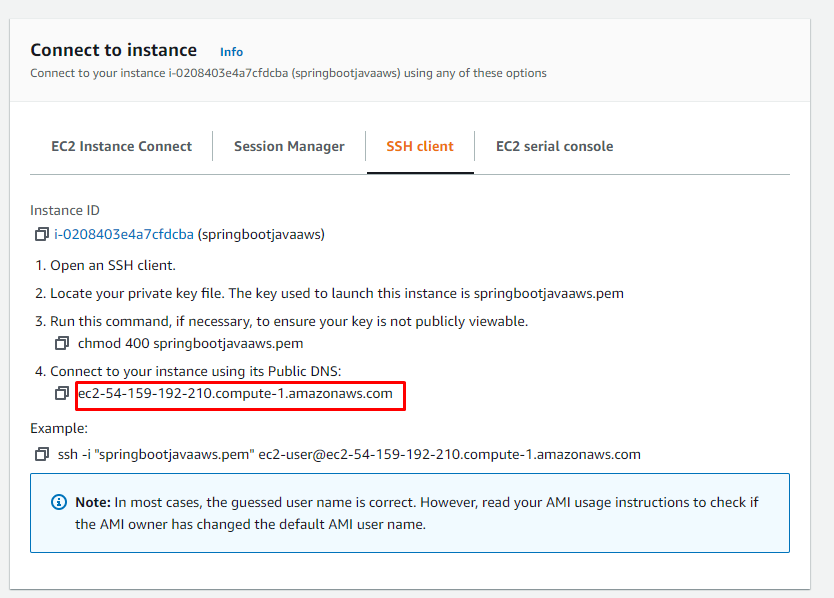
Now we should open PuTTYGen Software.

Go to PuttyGen -> Click Load ->upload downloaded key pair file and save private key 

Now we generated PuTTY private key file.



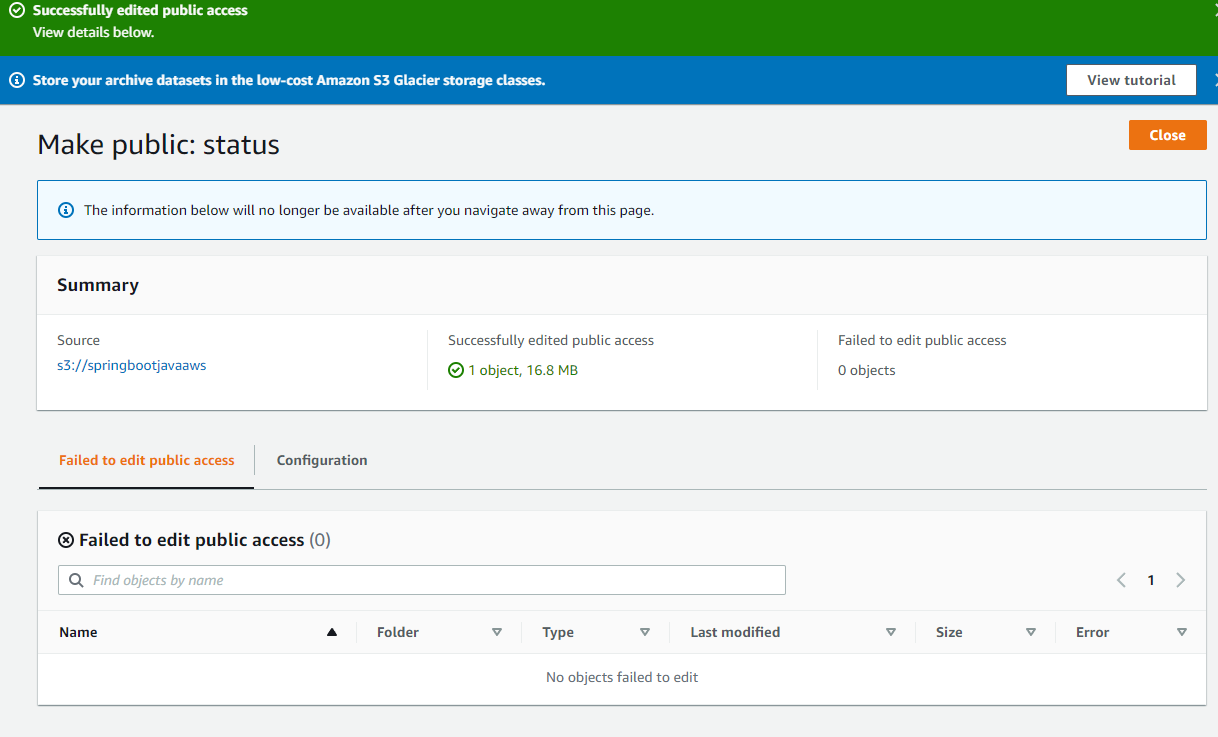
Now to we need to connect EC2 instance through putty.



Above are my username and IP address.

Step 3: Creating AWS S3 Bucket

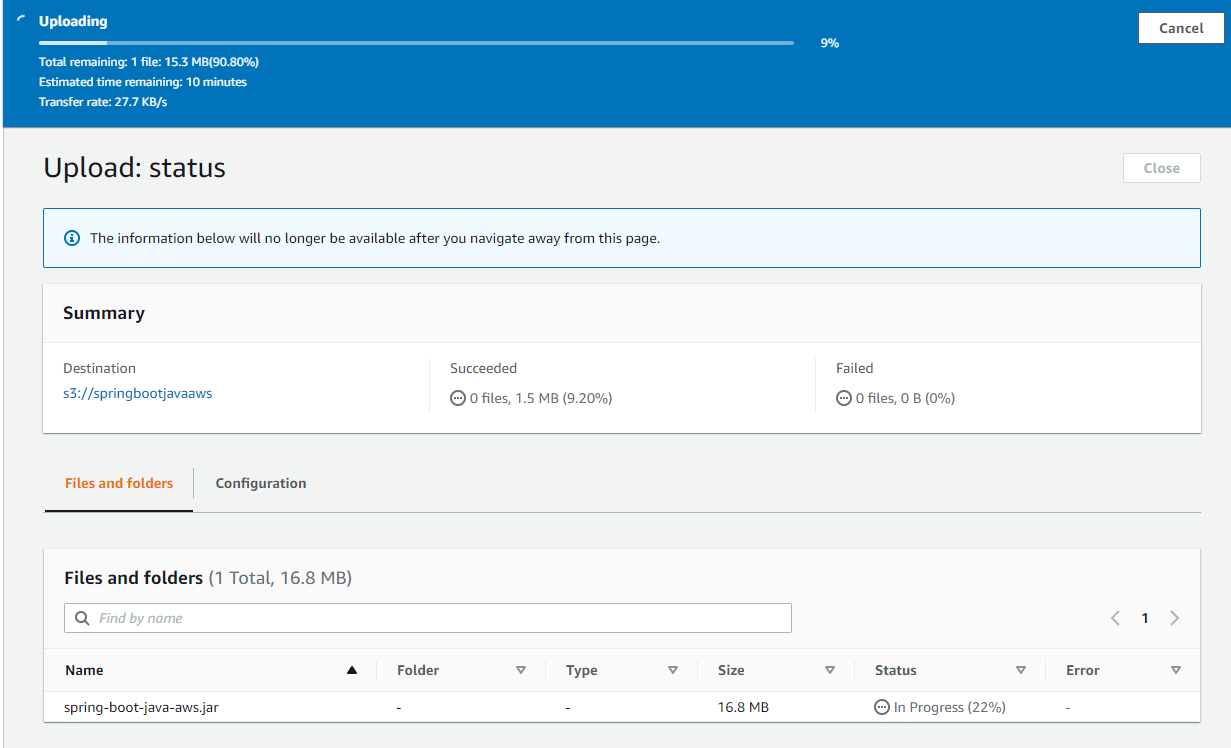
1. Go to AWS S3 -> create bucket-> give bucket name
2. Go to object ownership click on ACLs enabled.
3. We need to give public access to bucket



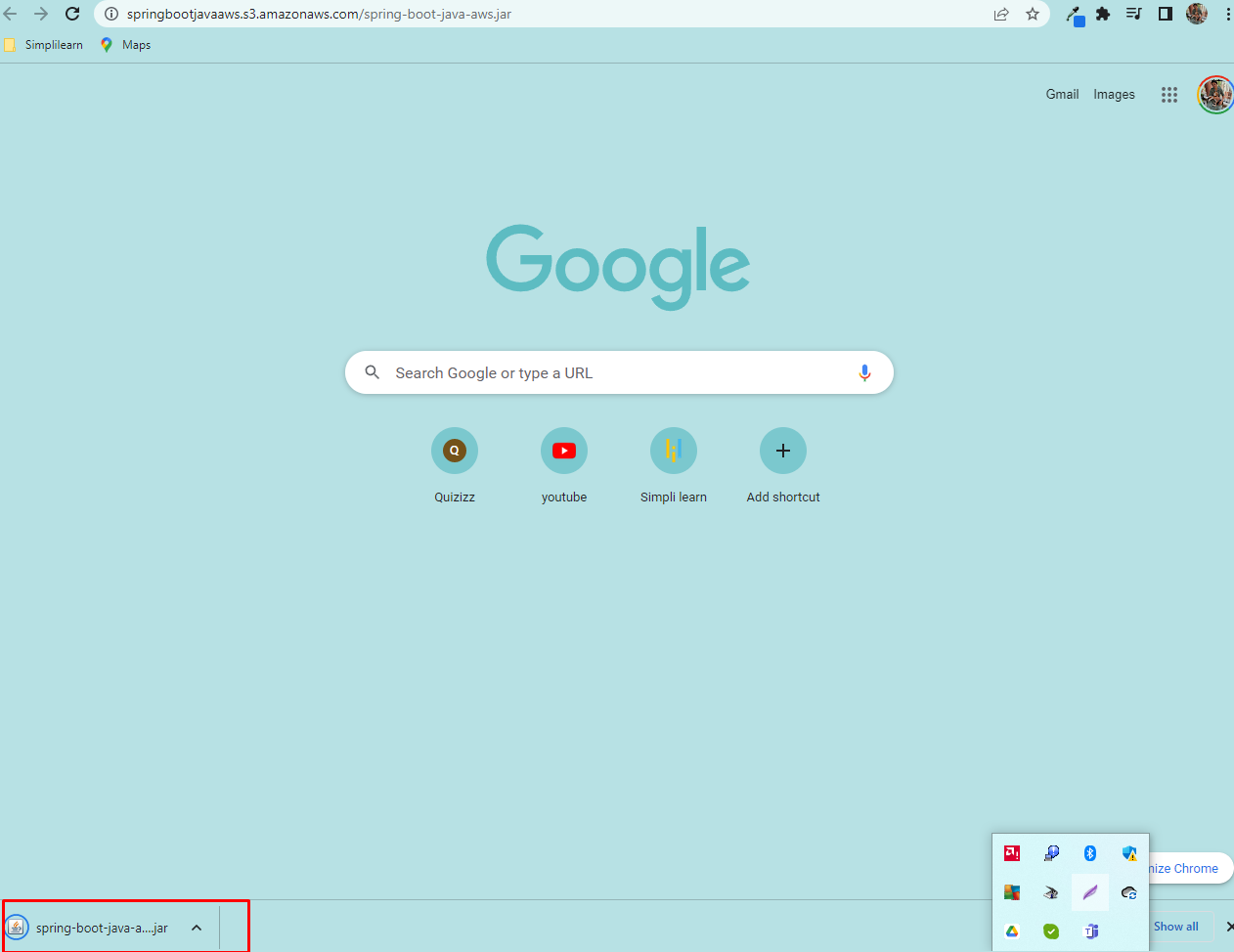
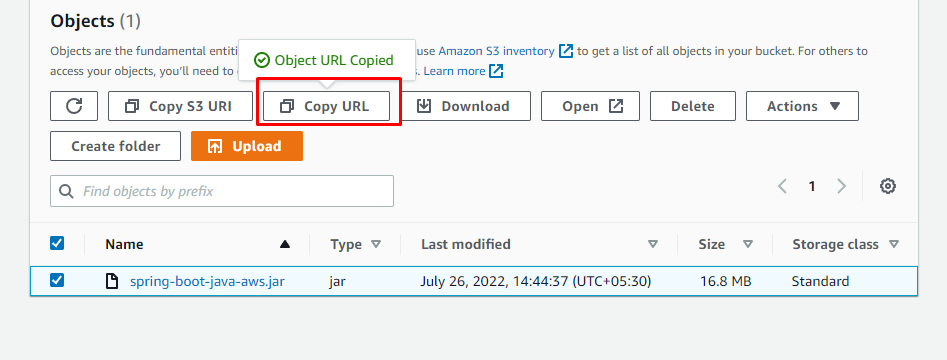
1. Click on create bucket.

After creating bucket click on Bucket.

And uploaded my spring boot jar file in bucket.



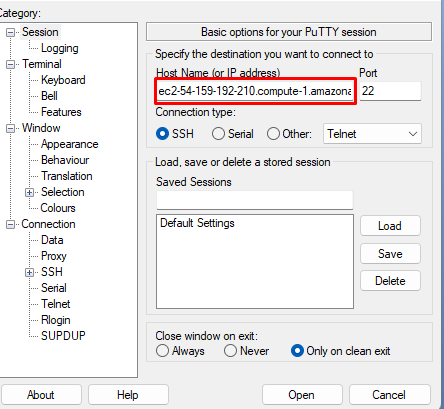
This step is for to make my bucket is public **Go to bucket -> action -> Make public using ACL.** Copy URL and paste it in new window we can able to download our jar file .



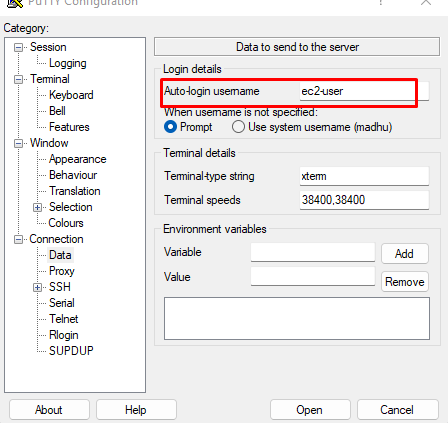
Successfully created S3 Bucket.

Here we are connecting EC2 instance through PuTTY Software.

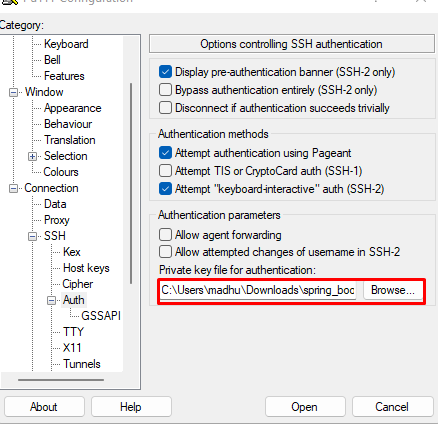
Next, go to putty software We need to enter ***Host Name***



1.Click on Data enter **username** is **ec2-user**

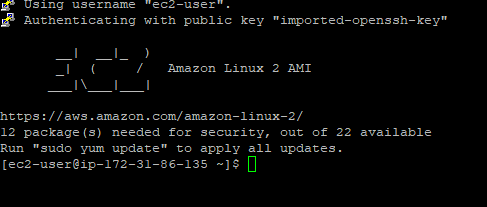


1. Expand SSH and go to AUTH



We need browse downloaded putty private key and open.

Now Putty command form will open.



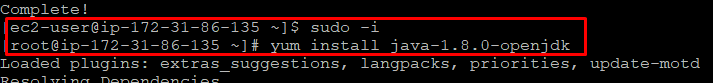
Step4: Connect to EC2

1. ls -> List of files are in the directory

2.pwd -> Shows absolute path

3. sudo yum update ->updates all the presently installed packages to their latest version .

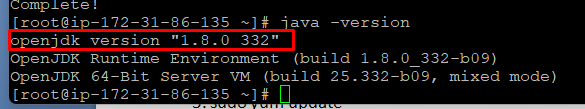
4. sudo –I ->run commands in our own user account with root privileges



Changed to root .

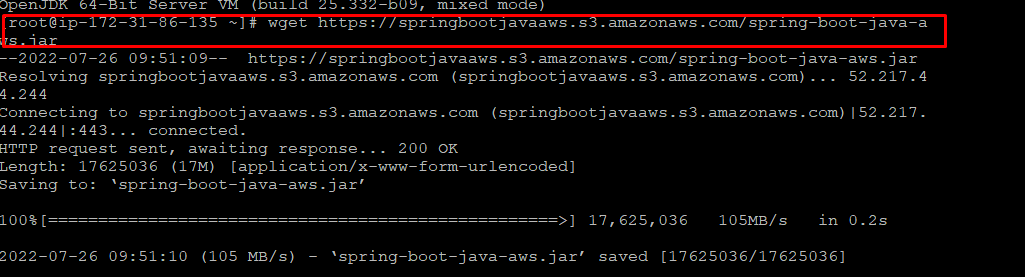
5. yum install java-1.8.0-openjdk -> installing java

6. java –version ->checking java version



7. We need get spring boot jar file. To run this command

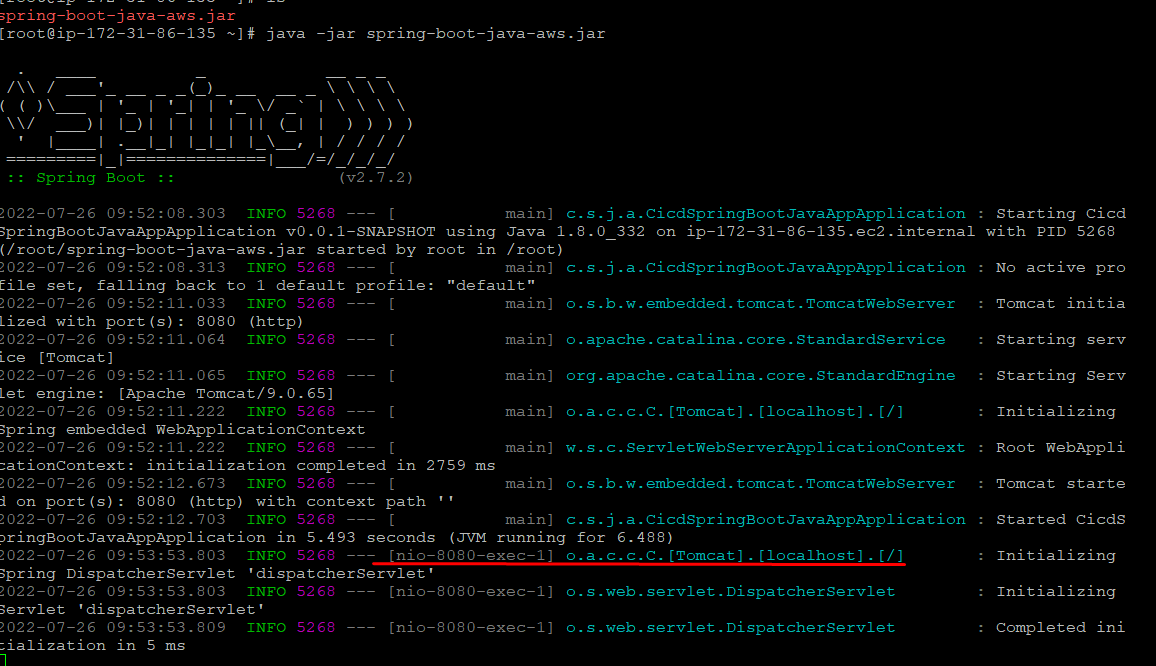
wget https://springbootjavaaws.s3.amazonaws.com/spring-boot-java-a ws.jar



8. ls -> checking list of files .

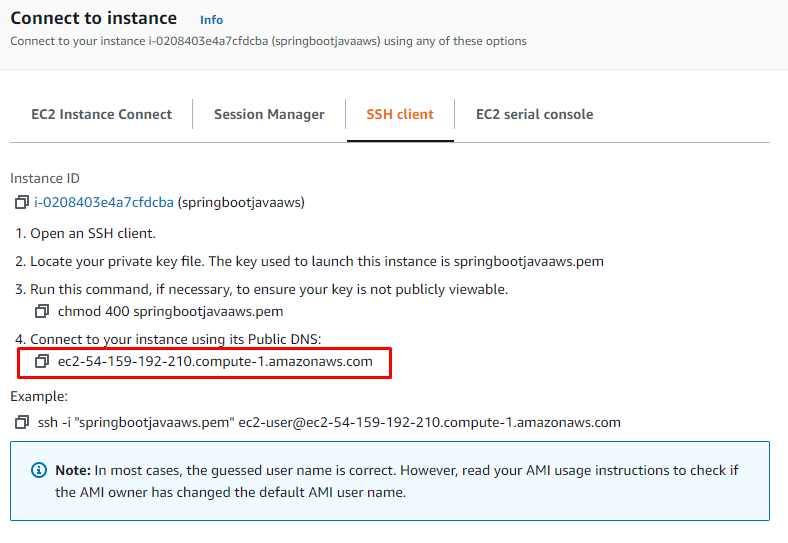


9. java -jar spring-boot-java-aws.jar -> going to run our spring boot program.



10. Once the program starting, now I am going to check in web

11. Go to EC2 instance -> connect -> copy public DNS



12. Open new window and paste IP Address followed by port number

http://ec2-54-159-192-210.compute-1.amazonaws.com:8080/welcome

Here

Amazon Public DNS: ec2-54-159-192-210.compute-1.amazonaws.com

My tomcat port number: 8080

My end point: /welcome



Successfully deployed my spring boot project in AWS using EC2 and S3