Phase-5 Project:- CI/CD Deployment for Spring-boot Application.

Project Objective:-

As a Full Stack Developer, you have to build a CI/CD pipeline to demonstrate continuous deployment and host the application on AWS EC2 instance.

Sprint Planning:-

This project is completed in three sprints each of one week

1st Sprint:-

1. Created a Spring-Boot Application
2. Uploaded in Git Hub

2nd Sprint:-

1. Configured Maven
2. Generated a jar file using Maven clean install Command
3. Created an Windows AWS Instance
4. Configured Windows AWS Instance

3rd Sprint:

1. Installed Jenkins
2. Configured Jenkins to automate the Process
3. Hosted Spring-Boot application on Windows AWS Instance

Technologies Used:-

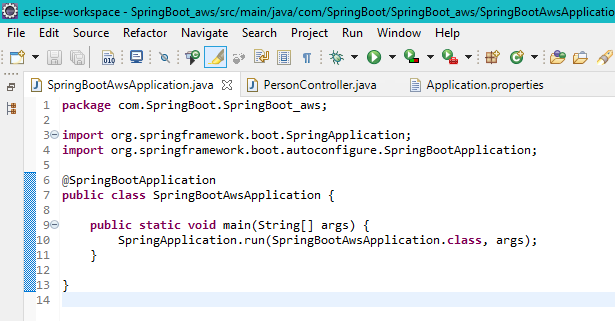
1. Spring-Boot framework
2. Apache Maven
3. AWS EC2 Instance
4. Jenkins

Project Features:-

Create a Spring-Boot Application. Create a Windows Amazon EC2 instance and install all the technologies required in the Windows server. Start configuration for Jenkins and automate the Spring-Boot application by providing the Git-Hub link along with provide Windows Batch Command. Start Jenkins Build and go to the configured port and check for the application status

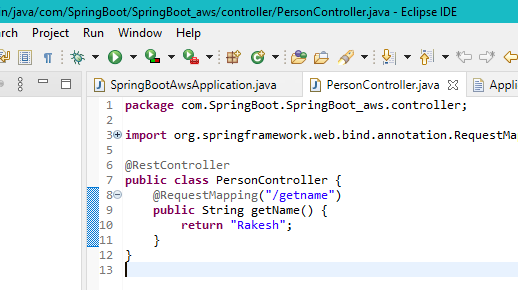
Refer below Screenshots for more understanding

Spring-Boot Application:-



Figure(1).Spring-Boot Application

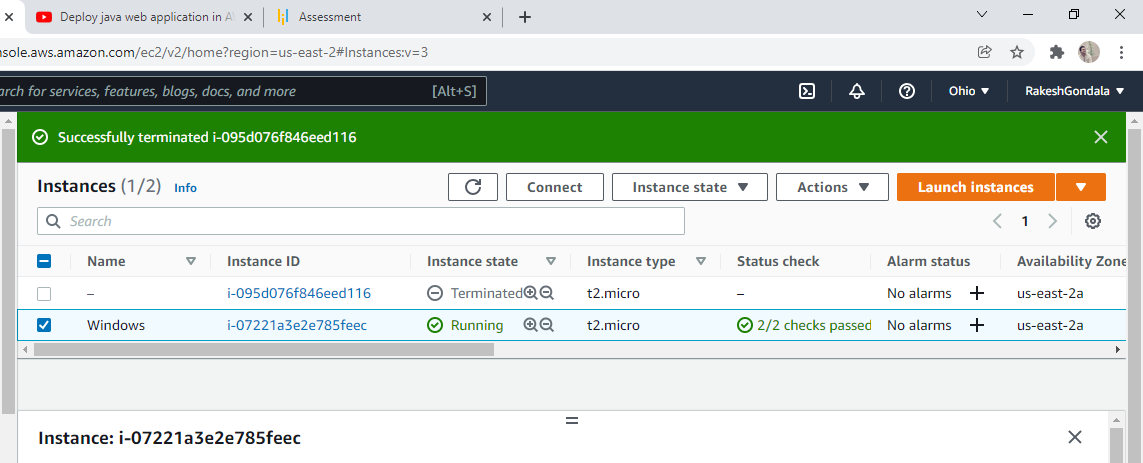
Create a Spring-Boot Application as given in above figure. By Using Spring Initializer website we can create a Spring-Boot Application with all required dependencies. It also comes with an in-built Tomcat Server where we can launch our Spring-Boot application. It runs on default port 8080.



Figure(2) Spring-Boot Application

The above Figure displays code for Controller of Spring-Boot application. We can access the Application by using @RequestMapping(“/getname”) from URL=”localhost:8080/getname”

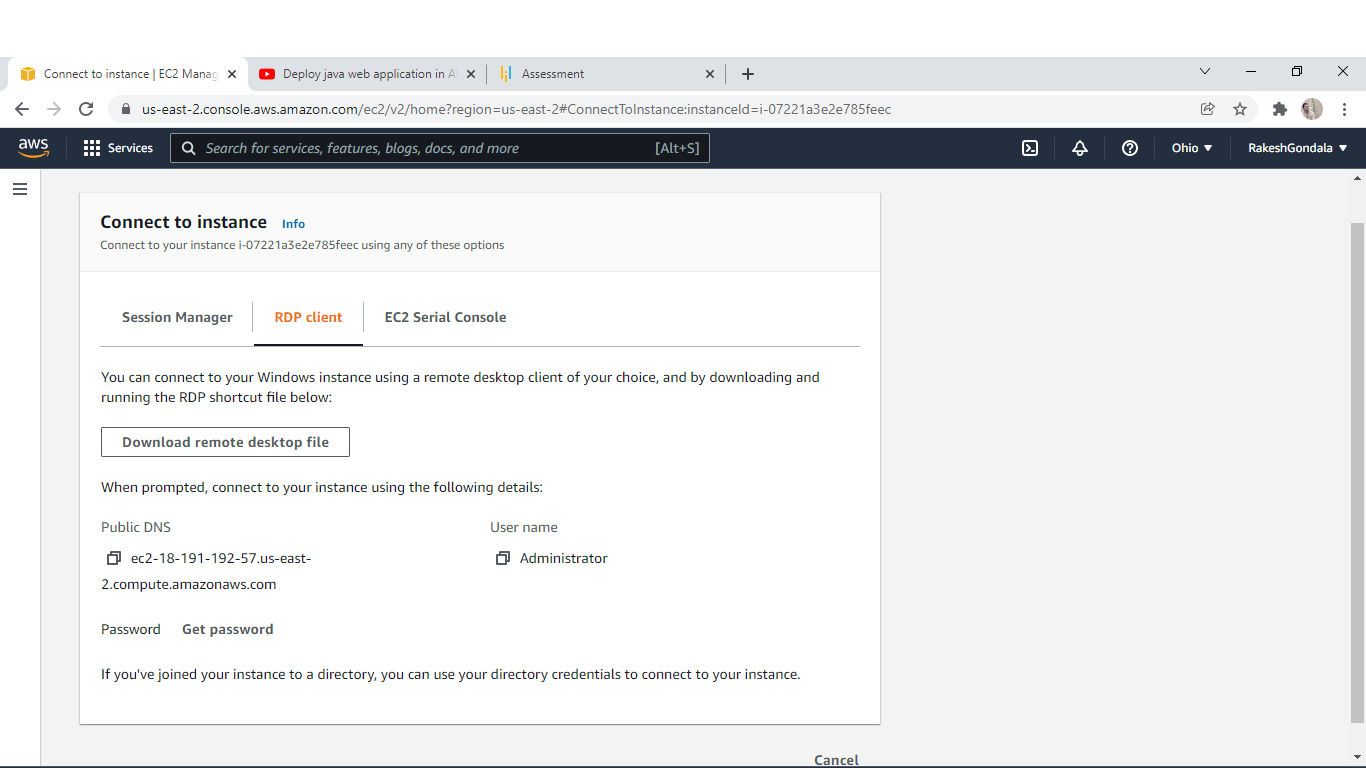
Amazon ec2 Instance:-



Figure(1) Amazon ec2 Instance

The above Figure shows the Windows aws ec2 instance configuration. We can connect to the instance by clicking the option “Connect” on the above dashboard. Next we can connect to the instance by Downloading Remote Desktop Connection file and connecting it by providing the credentials as given in aws ec2 dashboard upon connecting.

Figure(2) Amazon ec2 Instance



So as shown in above picture we can connect to the instance by downloading the remote desktop file. After launching remote desktop file the server looks as shown in below picture

**

Figure(3) Amazon ec2 Instance

Jenkins Configuration:-

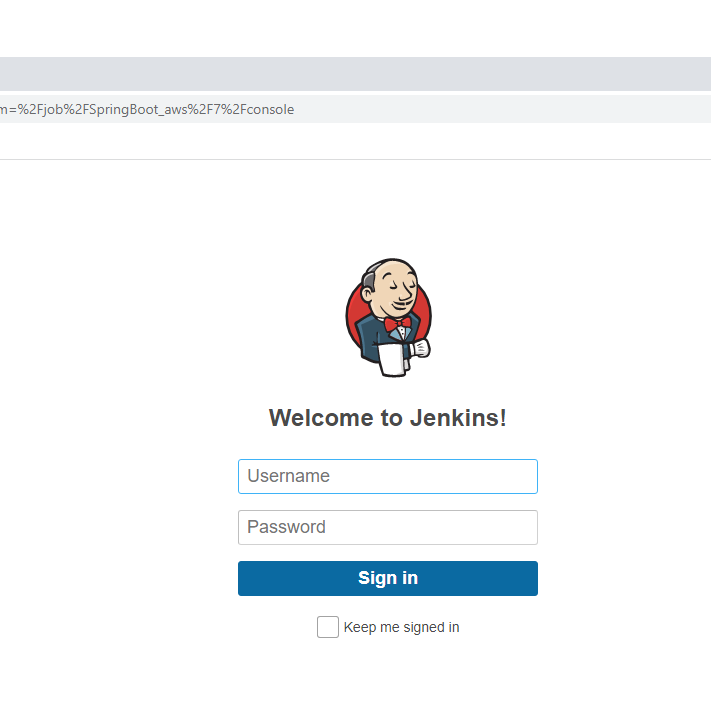
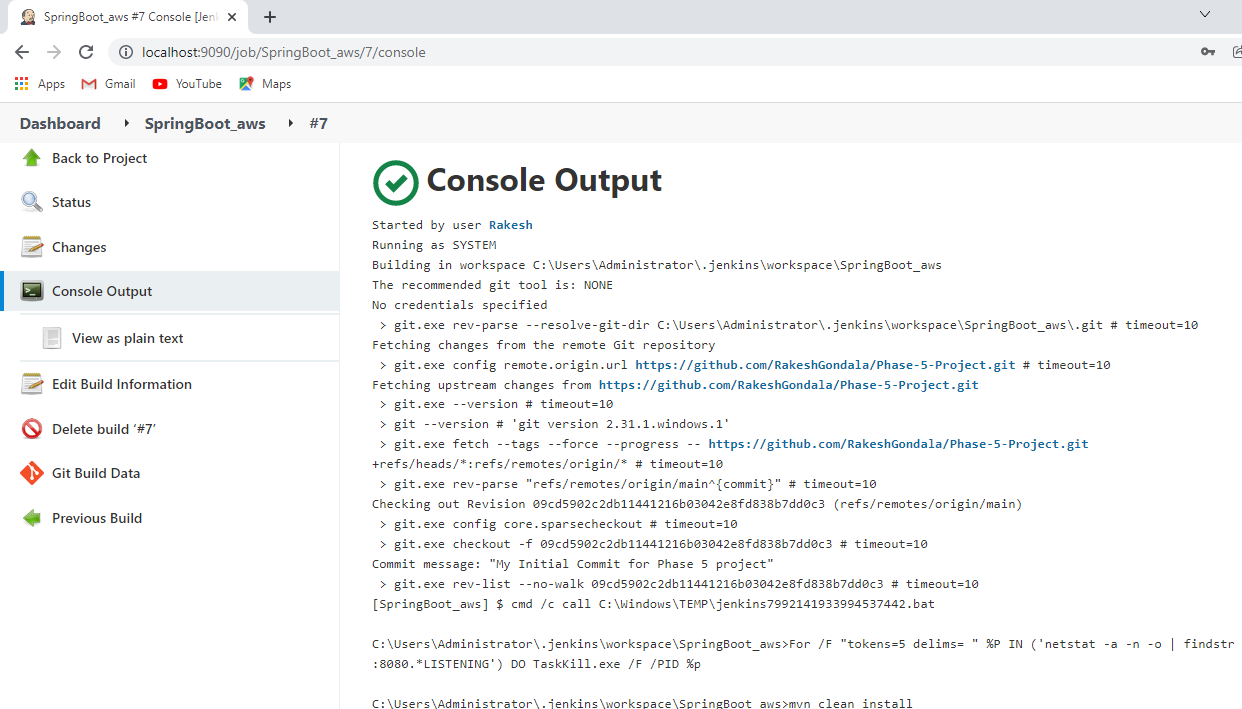
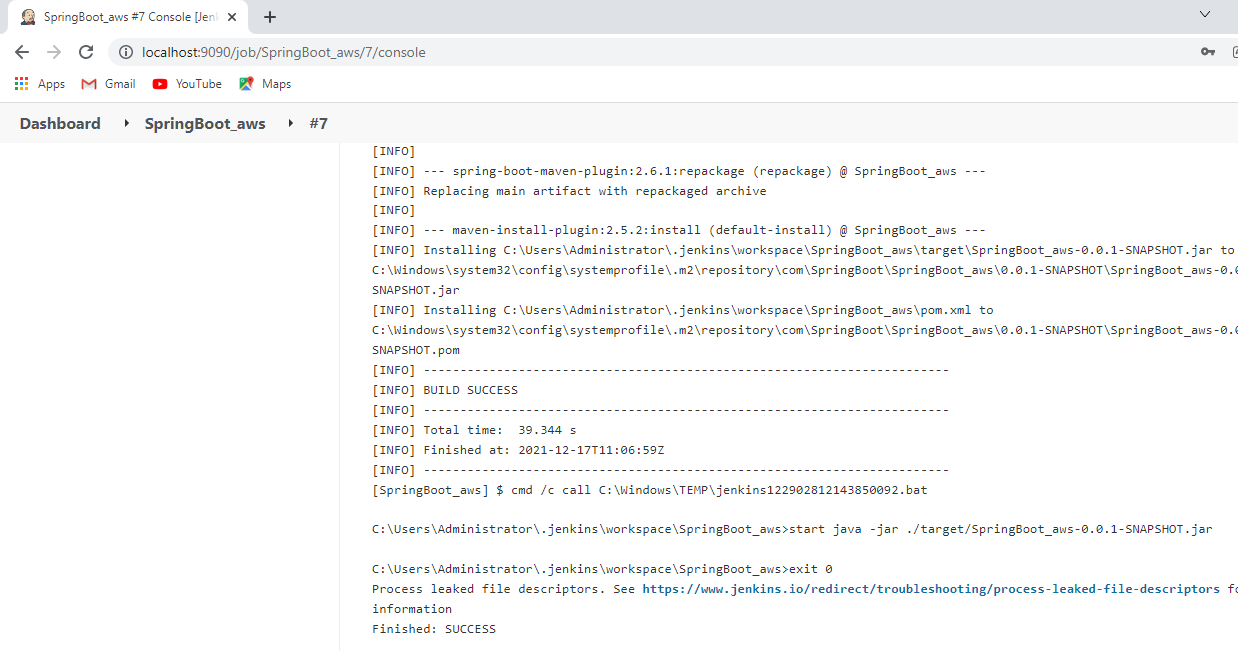


Figure (1) Jenkins login

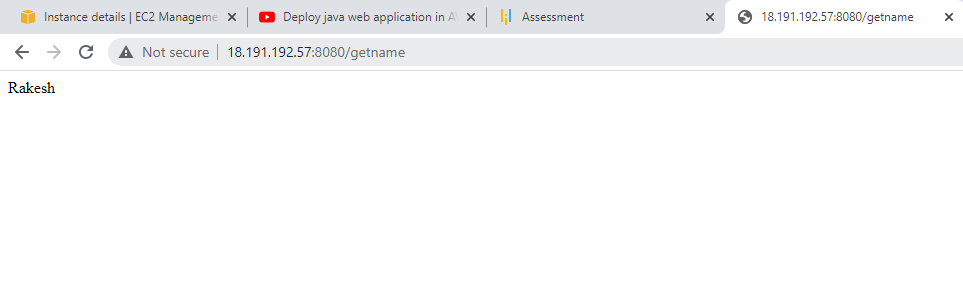
We can download Jenkins war file from Jenkins official website and execute “java -jar jenkins.war” file so that it can install Jenkins on our Windows Server Instance. After Installation we need to go to port 8080 this is the default port for Jenkins and we need to login

**

As you can see in the above picture Jenkins job was successfully completed and it is hosted on amazon ec2 Windows instance

**

Final Output :-



As you can see the output which is automated by Jenkins

Git-Hub Link:-

https://github.com/RakeshGondala/Phase-5-Project.git

CONCLUSION:

As this is a prototype application appearance may not be much awestruck. Please try to evaluate the application’s operations and send us the feedback.

THANK YOU

Rakesh Gondala.