**Final Specification Document**

**PROJECT AND DEVELOPER DETAILS:**

PROJECT :**CI/CD DEPLOYMENT FOR SPRINGBOOT APPLICATION.**

DEVELOPER : SALAMBASHA ALLABAHAS

FULLSTACK DEVELOPER

( MS FSD FEB Cohert1)

PROJECT DESCRIPTION :

This project concentrating more on  automate the integration and deployment of the web application. A Spring boot application is hosted on Amazon Web Server Ec2 lab instance. The deployed website can be accessible to everyone.

TECHNOLOGIES USED

* Eclipse
* GitHub
* Jenkins
* AWS EC2/ Virtual machine

**STEPS FOLLOWED TO DEPLOY A SPRINGBOOT APPLICATION ON EC2 SERVER**

1. https://github.com/callicoder/spring-boot-websocket-chat-demo spring boot application is chosen to deploy.
2. First spring boot application deployed on the local server.
3. Repository cloned using the command:

**git clone** [**https://github.com/callicoder/spring-boot-websocket-chat-demo**](https://github.com/callicoder/spring-boot-websocket-chat-demo)

1. Get inside to the project and used the command

**mvn clean package**

1. Then created the Image using the command

**docker build -t springbootimage .**

1. Container created using the below command

**docker run -p 8085:8080 --name springbootcontainer springbootimage**

1. Now the spring boot application deployed on the local server.

**USING AWS EC2 SERVICE:**

1)Created a EC2 instance using EC2 console

2) Security configuration added with **All traffic and anywhere**

3)downloaded the pem file.

4)cmd opened in the folder where the pem file is existing.

5)EC2 server accessed using the ssh command

**ssh -i "projectKeypair.pem"** [**ec2-user@ec2-52-205-93-225.compute-1.amazonaws.com**](mailto:ec2-user@ec2-52-205-93-225.compute-1.amazonaws.com)

6) After being inside the EC2 instance to change ourselves to root command

**sudo – i**

7)Updated using **yum update**

8) git installed using **yum install git –y**

9)maven installed using **yum install maven -y**

9) Then repository cloned by using **git clone**

10)**ls** command will the list of directories inside the folder

11) Get inside the project folder using the **cd** command.

12)Repository tested using the command **mvn clean package**

13) Now docker has been installed using the command

**yum install docker -y**

12)Docker started by using the command **systemctl start docker**

13) Now in the ec2 insatance we can build the image using the command

**docker build -t springbootimage .**

14) Using the image Container can be created

**docker run -p 8085:8080 --name springbootcontainer springbootimage**

15) Now the application is deployed on the amazon EC2 server http://52.205.93.225:8085/

**JENKINS**

* 1. Jenkins installed by using the commands present in the Jenkins web site
  2. <http://52.205.93.225:8080/> Jenkins shows its dashboard in this page
  3. Docker plug in installed
  4. Maven configuration completed on Global tool configuration
  5. The same website which hosted on the aws ec2 server has been created as a job and tested using Jenkins

The above steps are captured as a screen shot and pasted on the output images.docx file.

NOTE: **Output Images File Attached Separately.**

**Filename: SALAM BASHA A\_PHASE\_5\_NIKUNJ SHAH.DOCX**

===================================