INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

CS457
Devops Final Assignment-1

Submitted to:
Dr.Uma S

Submitted by:
Team 6
Pokala Dattatreya (18BCS067)
Rama Dundi Saketh(18BCS076)
S Sampath(18BCS087)

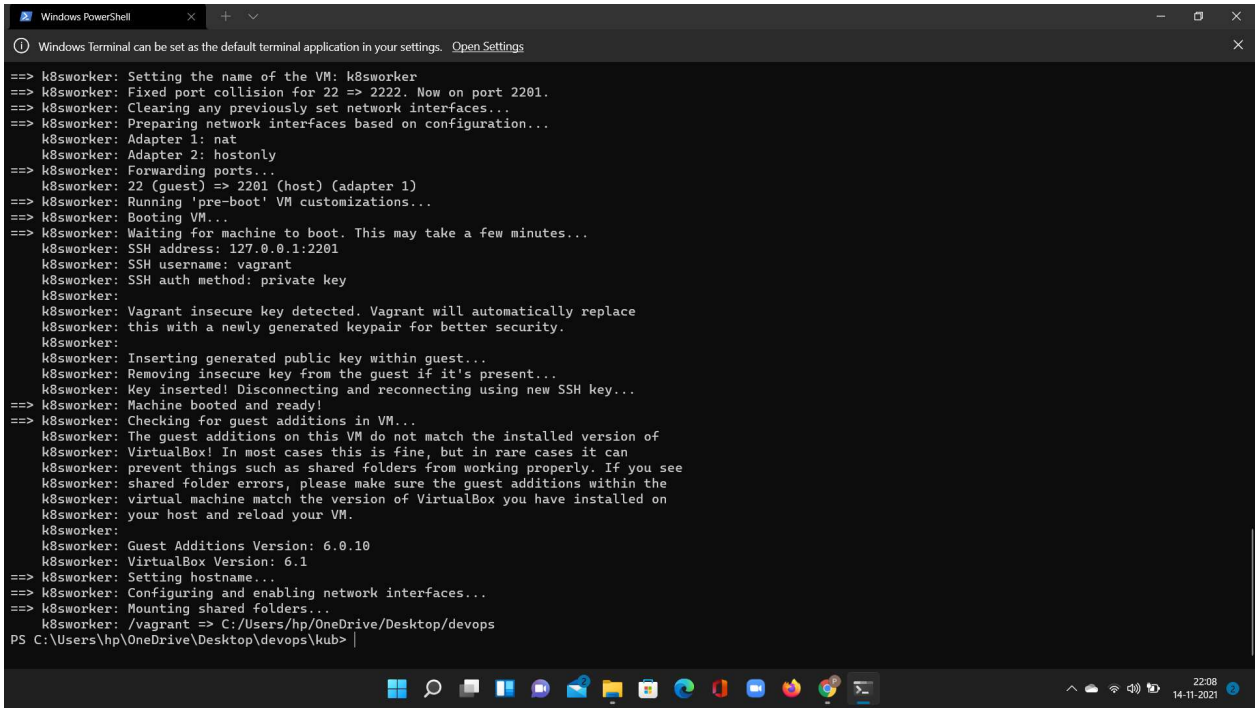
Set up complete CI/CD Jenkins pipeline for kubernetes

Tools and Technologies used:

->Virtual Box
->Docker
->Jenkins
->Kubernetes

Step1:

->Setting up 3 VM's using vagrant



Step2:

->Clone the kubespray git repo
link:https://github.com/kubernetes-sigs/kubespray.git

->Install the Kubespray Packages

->Copy inventory file to current users

->Prepare host.yml for kubespray

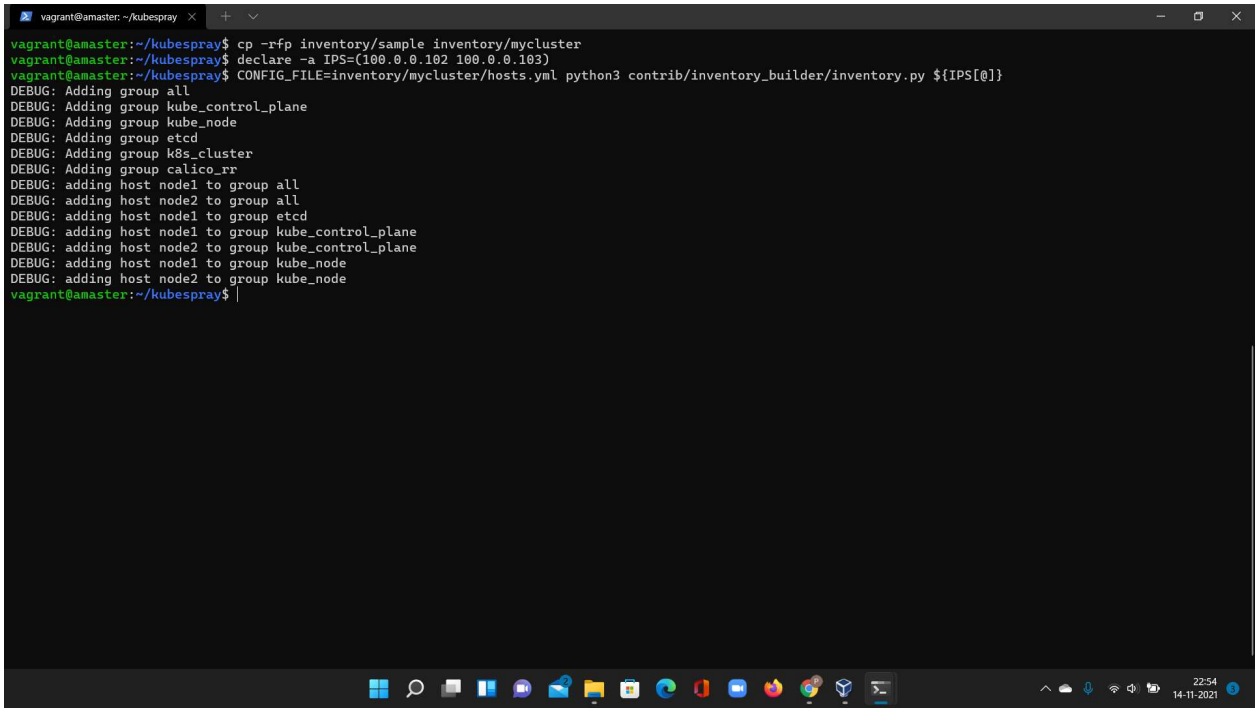
Commands to run:

-> sudo pip3 install -r requirements.txt

-> cp -rfp inventory/sample inventory/mycluster

-> declare -a IPS=(100.0.0.102 100.0.0.103)

-> CONFIG_FILE=inventory/mycluster/hosts.yml python3 contrib/inventory_builder/inventory.py ${IPS[@]}



->After running the above commands do verify the hosts.yml and it should be like below

```
vagrant@amaster:~/kubespray$ cat inventory/mycluster/hosts.yml
all:
  hosts:
    node1:
      ansible_host: 100.0.0.102
      ip: 100.0.0.102
      access_ip: 100.0.0.102
    node2:
      ansible_host: 100.0.0.103
      ip: 100.0.0.103
      access_ip: 100.0.0.103
  children:
    kube_control_plane:
      hosts:
        node1:
        node2:
    kube_node:
      hosts:
        node1:
        node2:
    etcd:
      hosts:
        node1:
    k8s_cluster:
      children:
        kube_control_plane:
        kube_node:
    calico_rr:
      hosts: {}
vagrant@amaster:~/kubespray$
```

->Run the ansible-playbook on ansible node

Commands to run:

-> ansible-playbook -i inventory/mycluster/hosts.yml --become --become-user=root cluster.yml

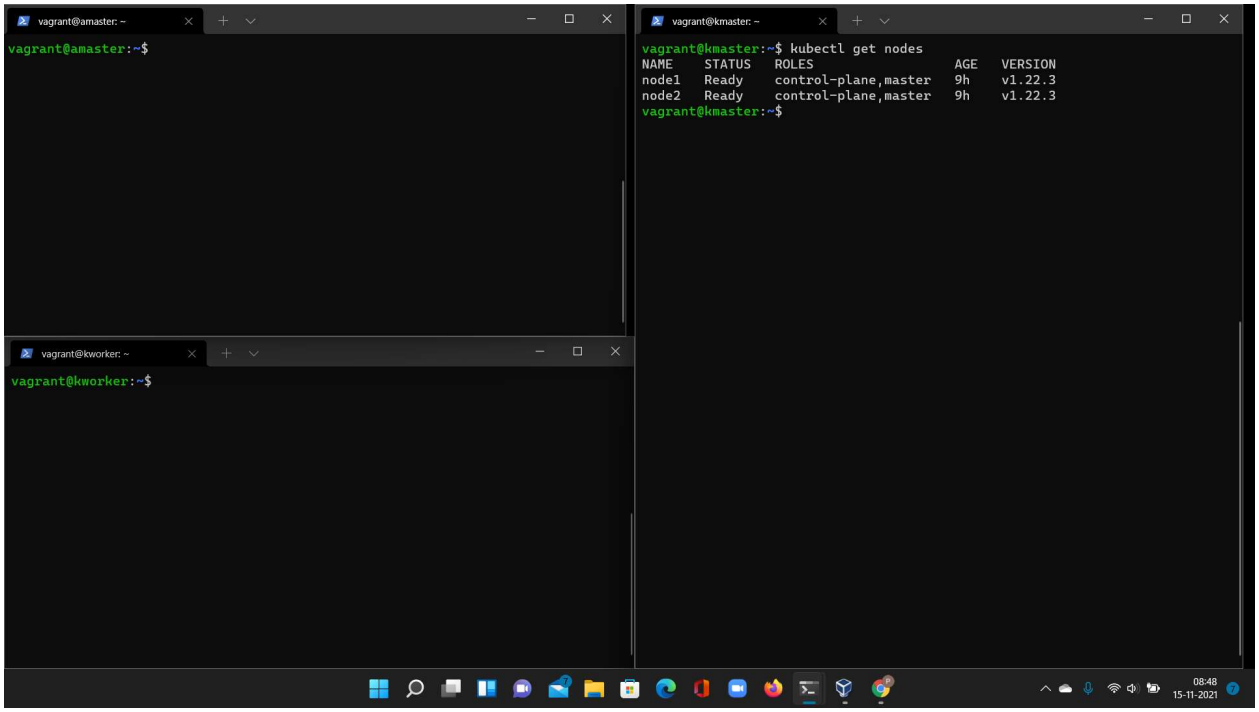
Step3:
->Install Kubectl on kubernetes master

Commands to run:

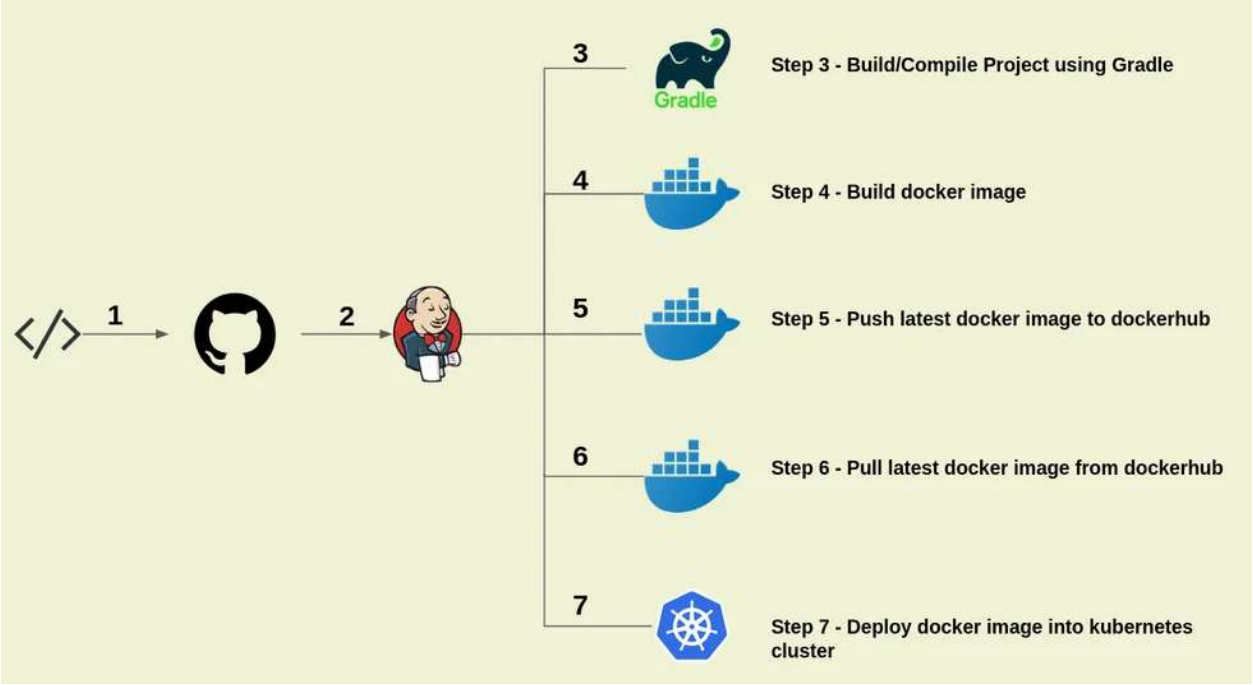
1) curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl

2)sudo cp /etc/kubernetes/admin.conf /home/vagrant/config
3)mv config .kube/
4)sudo chown $(id -u):$(id -g) $HOME/.kube/config

Now verify the kubernetes nodes:
Command:kubectl get nodes



Step4:
overview of our GitHub and DockerHub flow :



4.1)Push code to github
Github link: https://github.com/dattu24/devopsIQ

4.2)Install Java and Jenkins

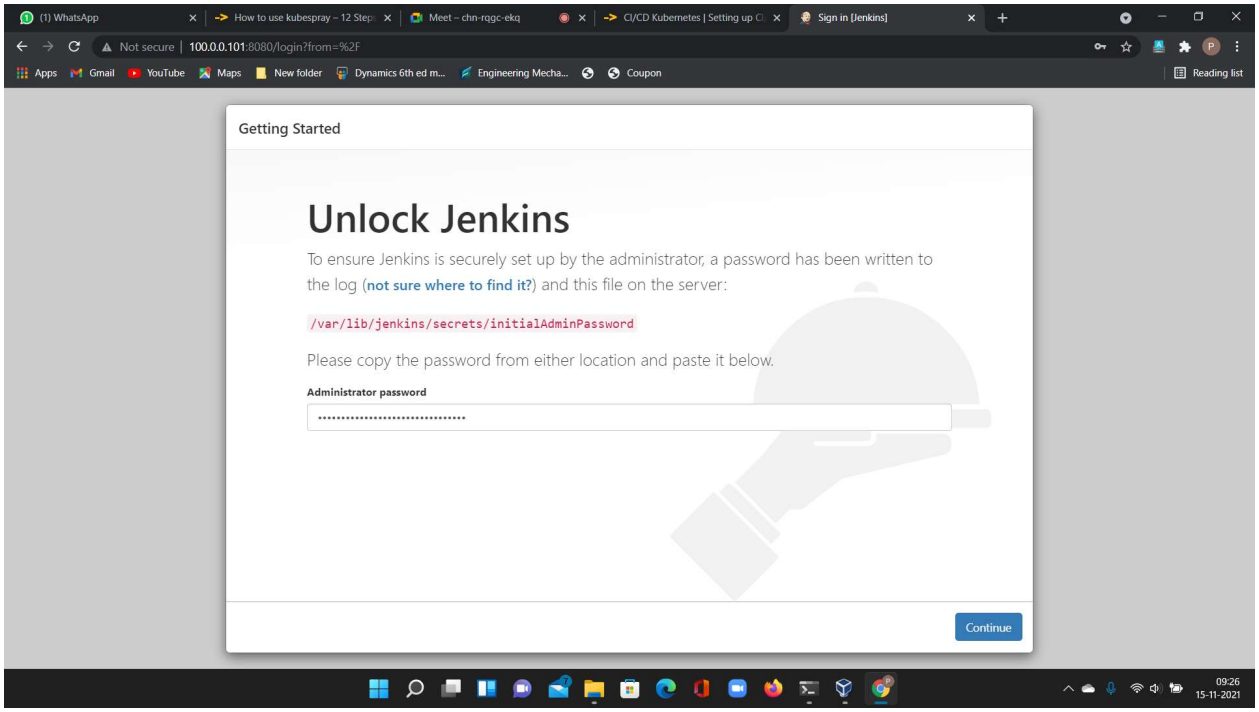
Commands to execute:

1)sudo apt install openjdk-11-jdk

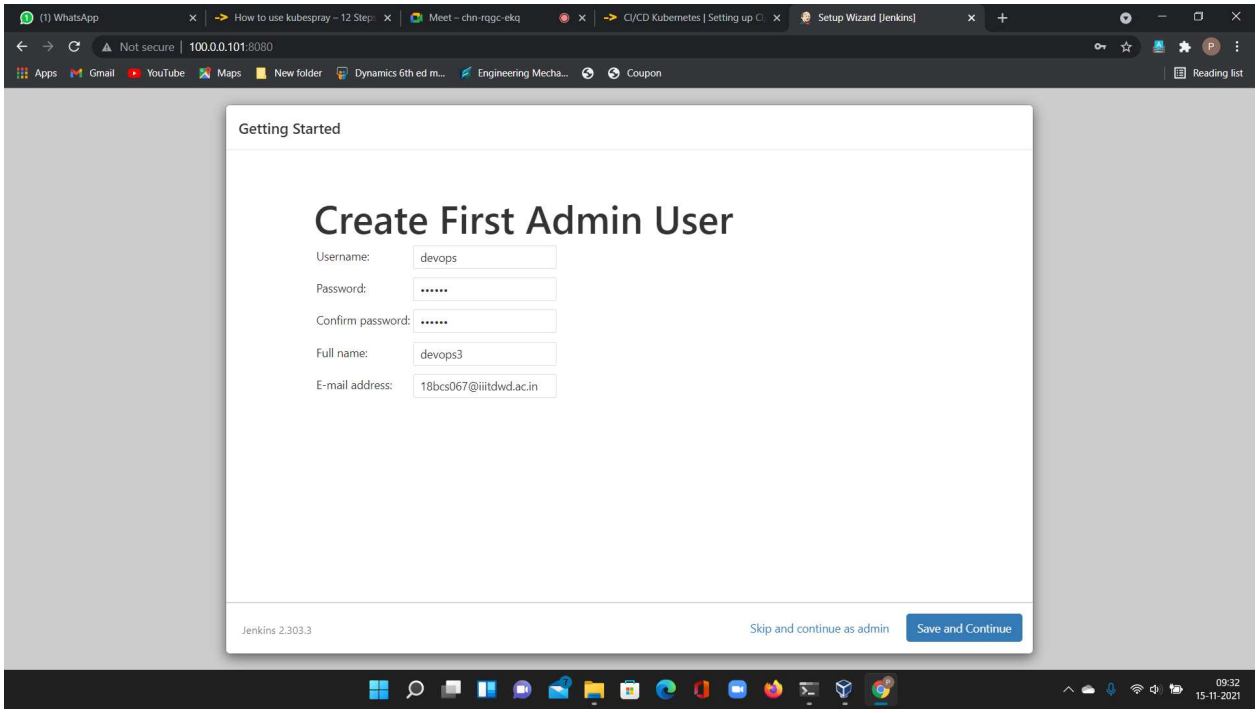
2)wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

3)sudo apt-get install jenkins

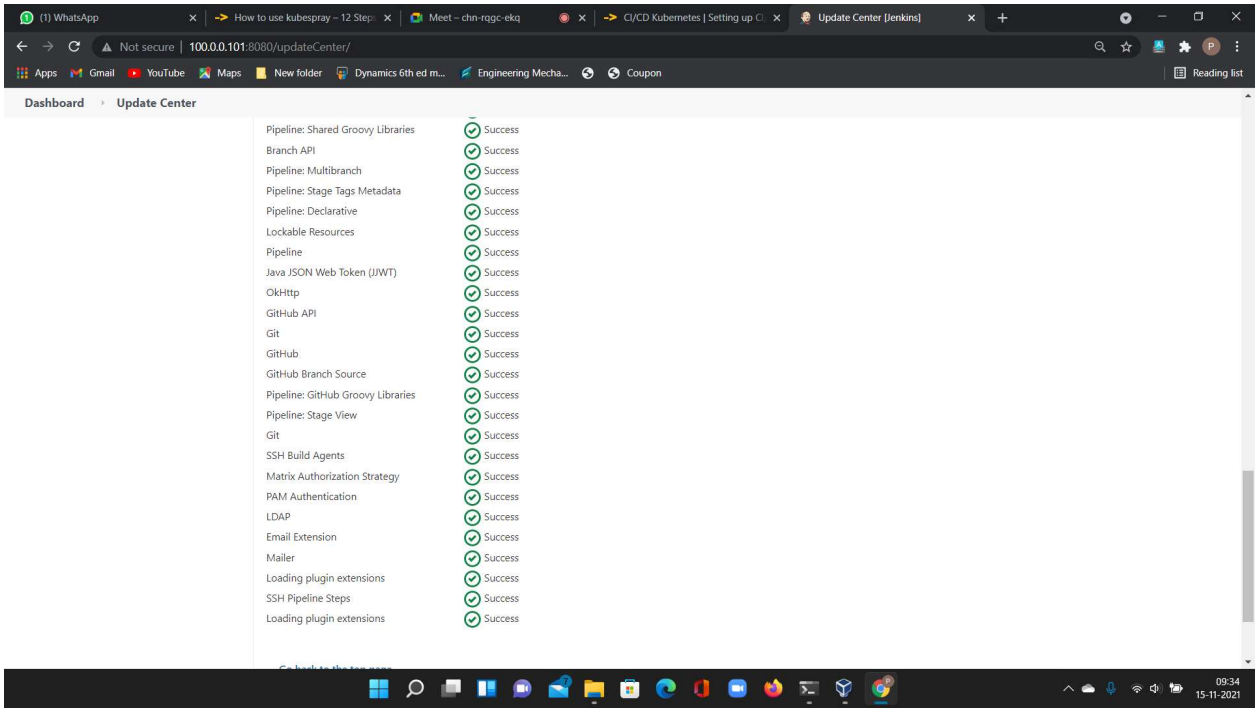
4.3)Now Verify Jenkins installation
Go to this link to verify http://100.0.0.101:8080/



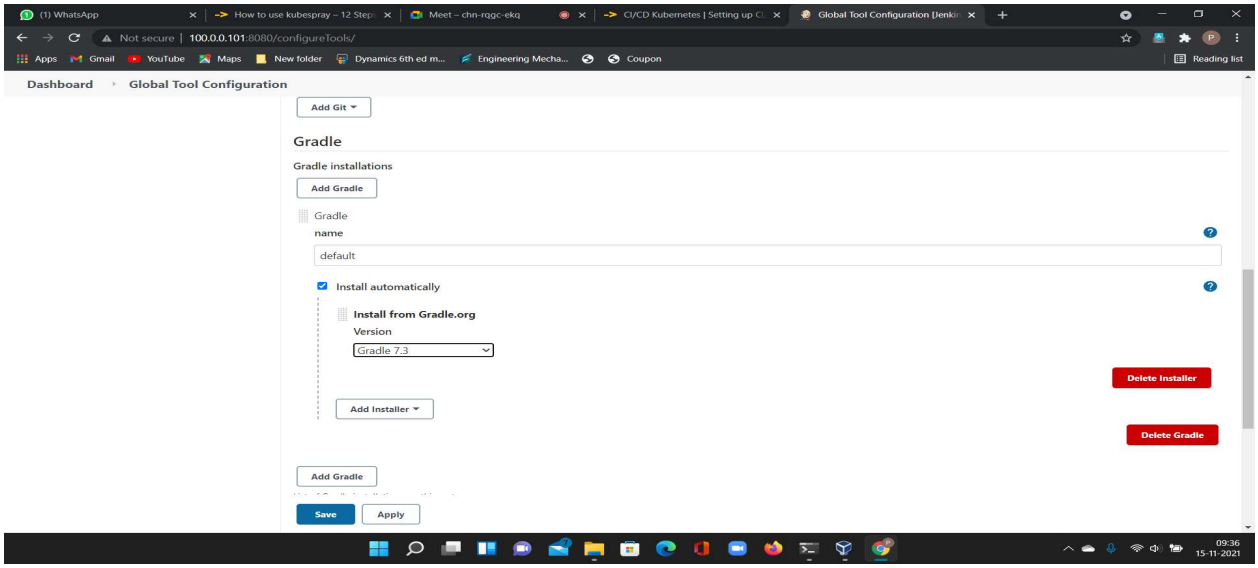
->Customize Jenkins and install suggested plugin
->Then setup username and password



4.4)Install one more plugin SSH Pipeline steps:
For installing plugin please goto - Manage Jenkins -> Manage Plugin -> Available then in the search box type SSH Pipeline Steps.



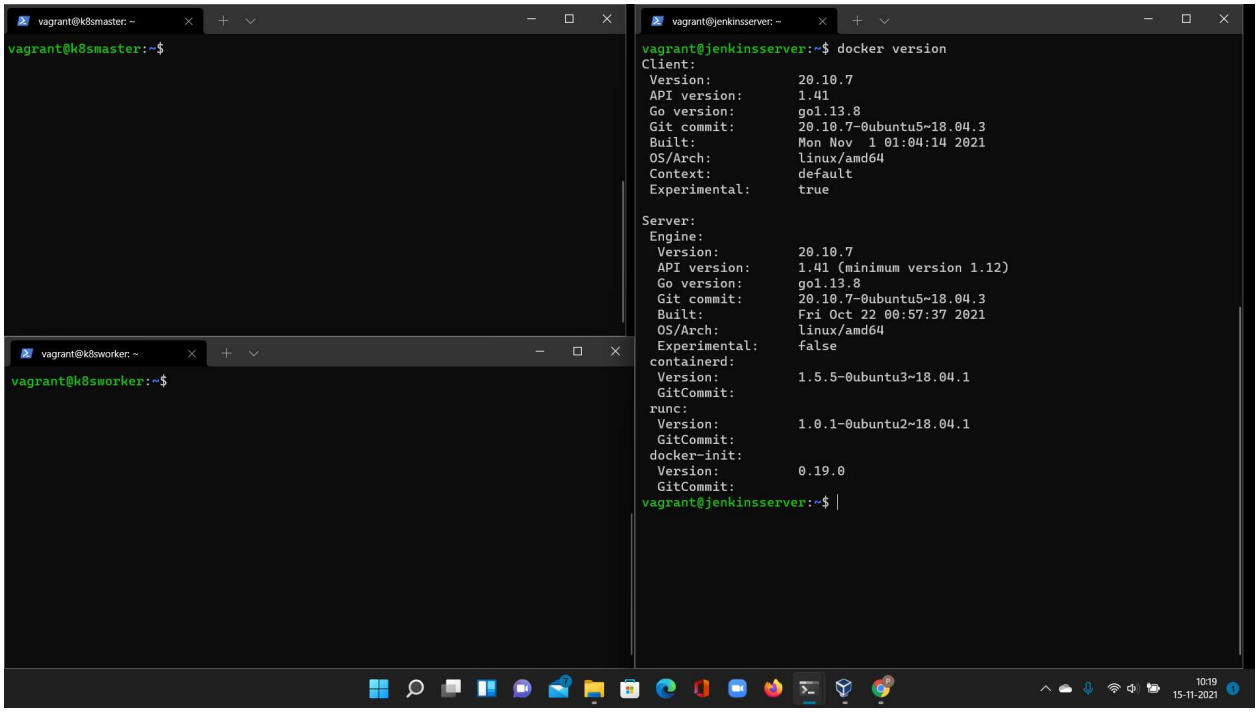
4.5)Setup gradle
To setup Gradle Goto - Manage Jenkins -> Global Tool Configuration -> Gradle



4.6)Install Jenkins on Docker on Jenkins Server

Commands to execute

1)vagrant ssh jenkinsserver
2)sudo apt install docker.io
Adding current user to docker
3)sudo usermod -aG docker $USER
4)sudo usermod -aG docker jenkins



Step5:

Here we are using spring boot applications.

Now, we should write Pipeline Script:

5.1)
1)Goto : Jenkins -> New Items
2)Enter an item name : item_name
3)Select Pipeline

4)Click Ok

5.2)
Clone the git repo

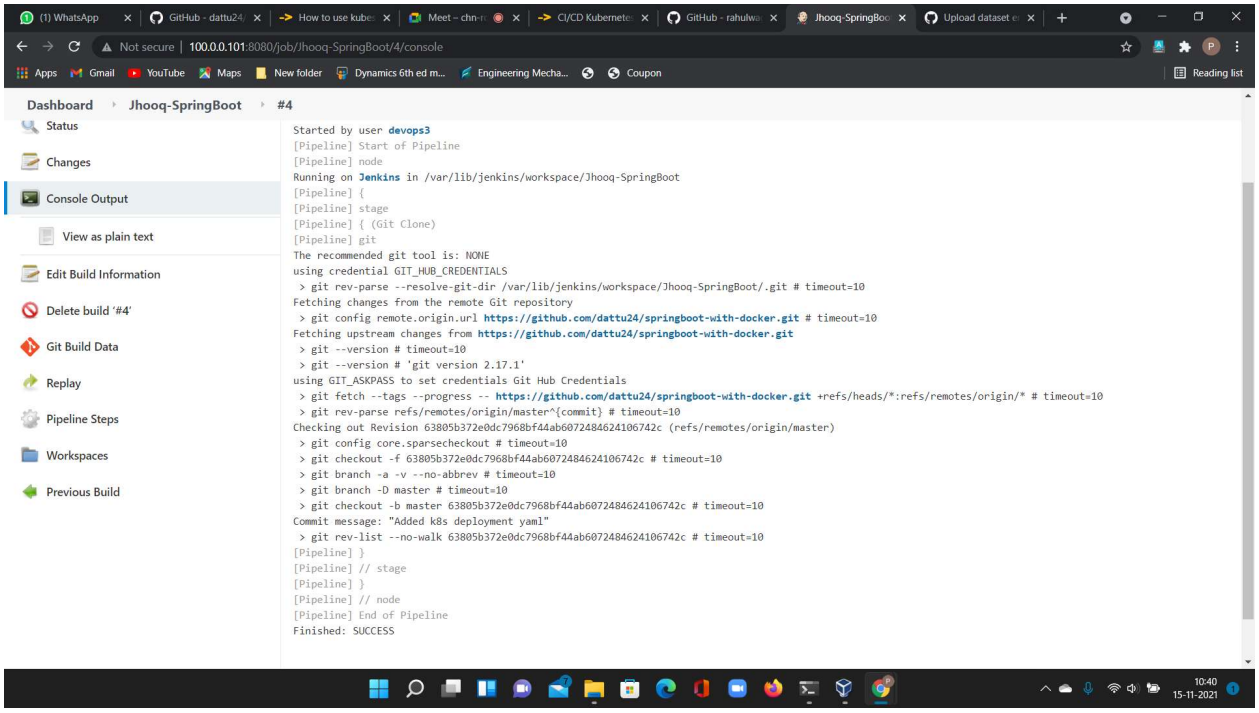
Code:
Adding a stage in pipeline(stage1):

```
stage("Git Clone"){
        git credentialsId: 'GIT_HUB_CREDENTIALS', url:
    'https://github.com/rahulwagh/spring-boot-docker.git'
        }
```

Jenkins store git Credentials:

Goto : Jenkins -> Manage Jenkins -> Manage Credentials

Logs of Build



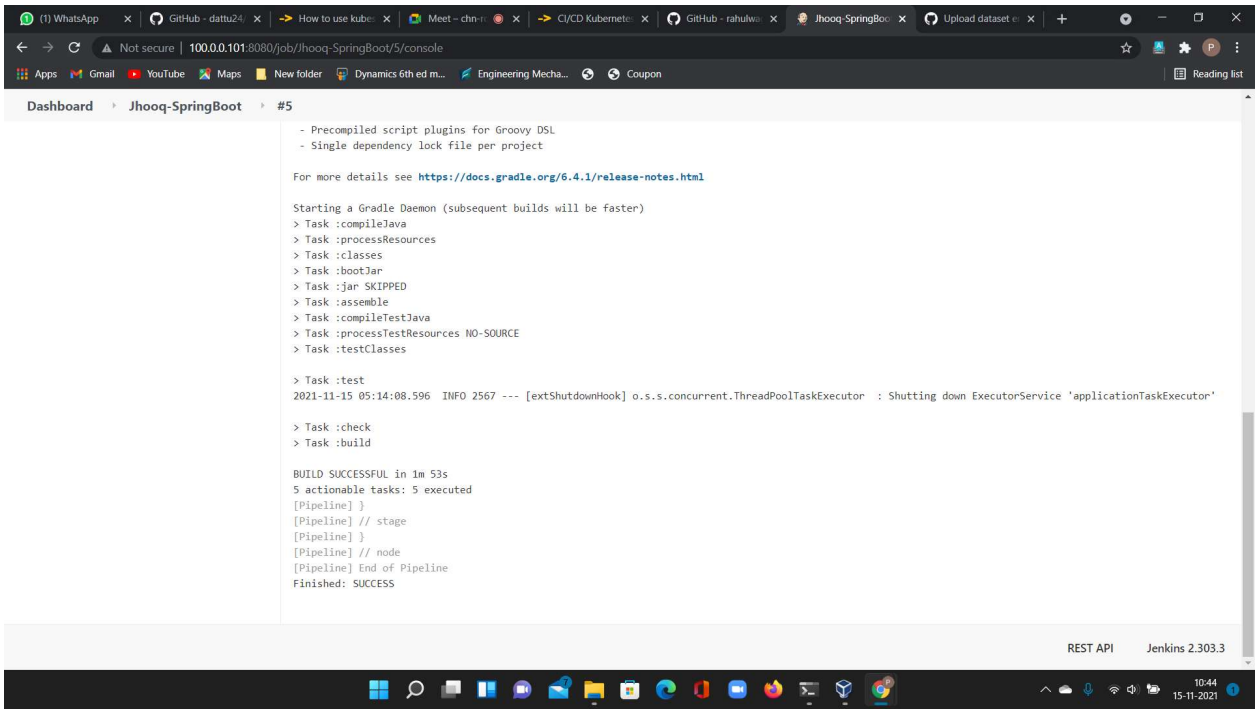
5.3)Build the spring boot application

Code:

Pipeline(stage2)

```
stage('Gradle Build') {
            sh './gradlew build'
            }
```

Build Log



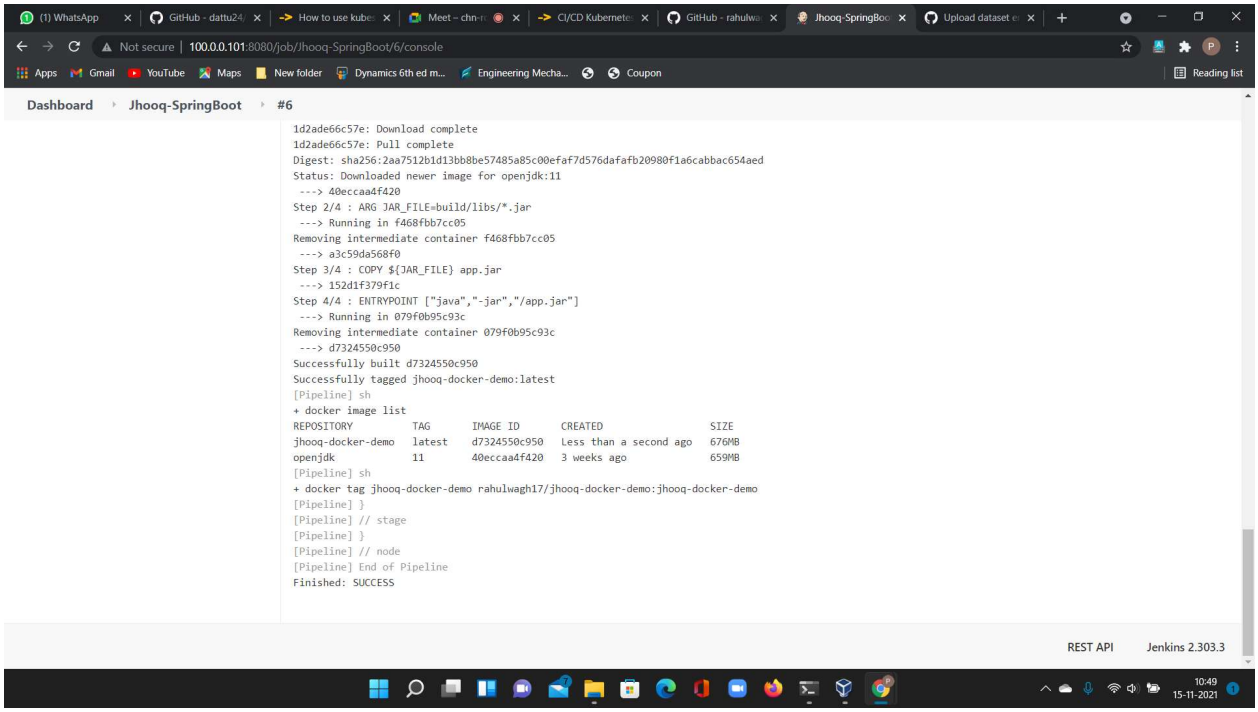
5.4)Build the docker image and tag it

Code:

Pipeline(stage3)

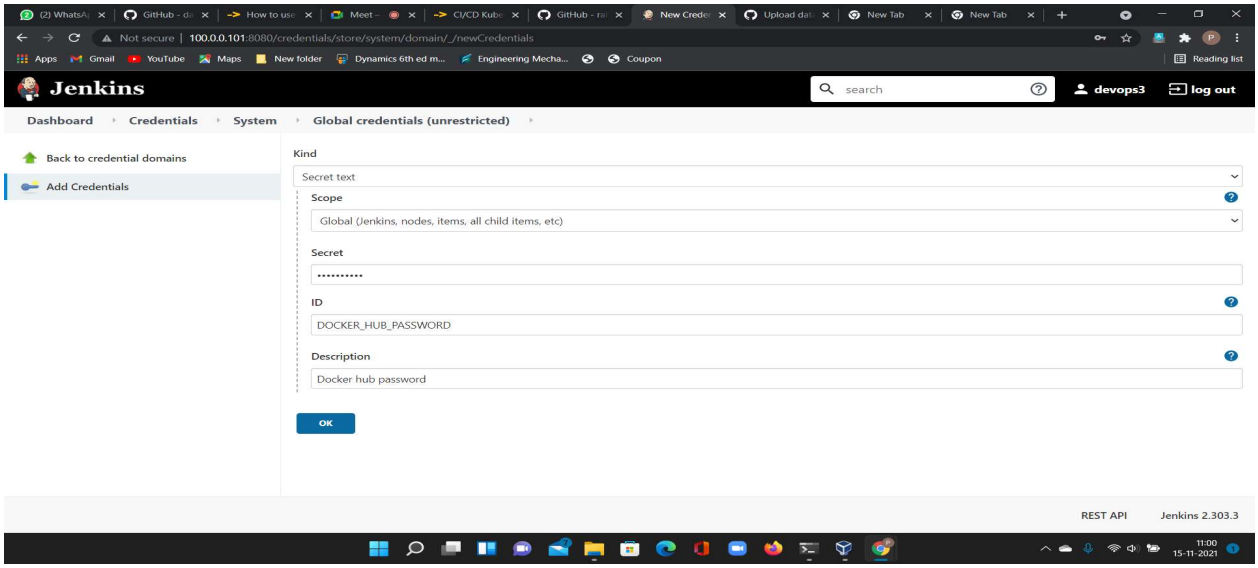
```
stage("Docker build"){
        sh 'docker version' sh 'docker build -t jhooq-docker-demo .' sh 'docker
        image list' sh 'docker tag jhooq-docker-demo
        dattu24/jhooq-docker-demo:jhooq-docker-demo'
         }
```

Build log



5.5)Jenkins store DockerHub credentials:

For storing DockerHub Credentials you need to GOTO: Jenkins -> Manage Jenkins -> Manage Credentials -> Stored scoped to jenkins -> global -> Add Credentials



5.6)Docker login via CLI:

Code:

pipeline(stage4):

```
stage("Docker Login"){
        withCredentials([string(credentialsId:
    'DOCKER_HUB_PASSWORD', variable: 'PASSWORD')])
        { sh 'docker login -u rahulwagh17 -p $PASSWORD' }
        }
```

Build log:

5.7)Push Docker hub in to Docker Image:

Code:
Pipeline(stage5):

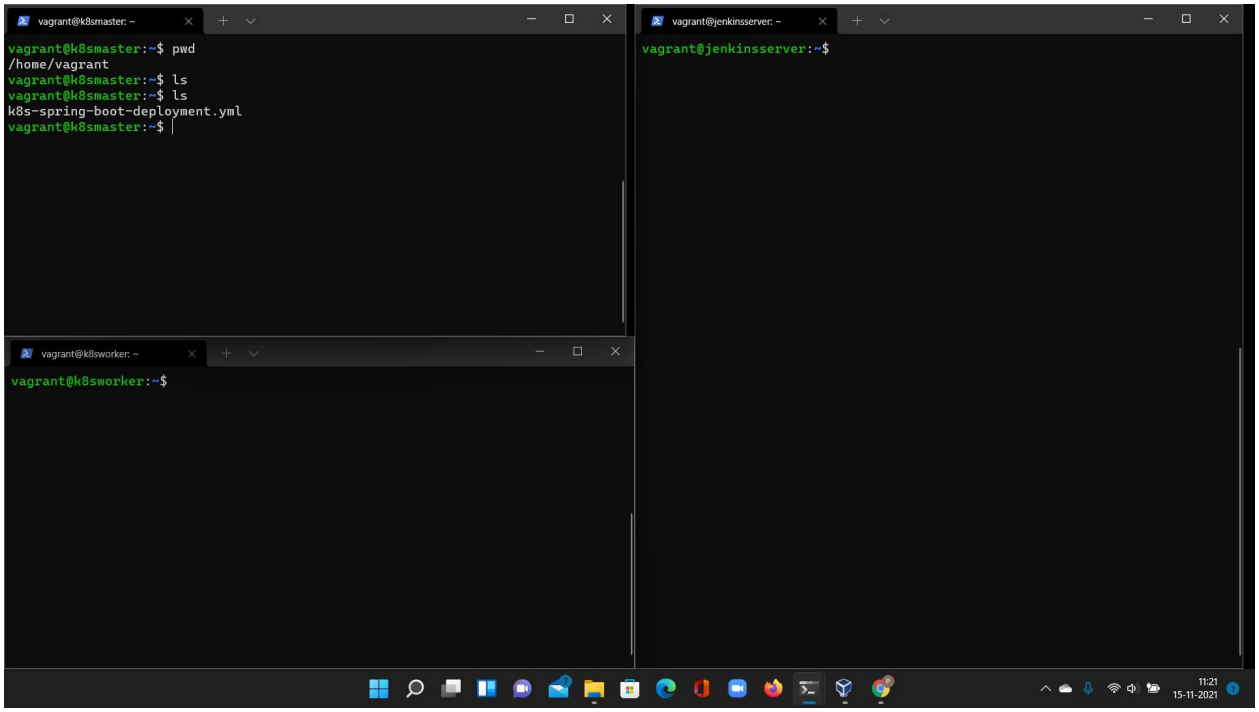
```
stage("Push Image to Docker Hub"){
        sh 'docker push  dattu24/jhooq-docker-demo:jhooq-docker-demo'
 }
```

5.8)SSH Into k8smaster server:
5.9)Copy k8s-spring-boot-deployment.yml to k8smaster server
5.10)Create kubernetes deployment and service

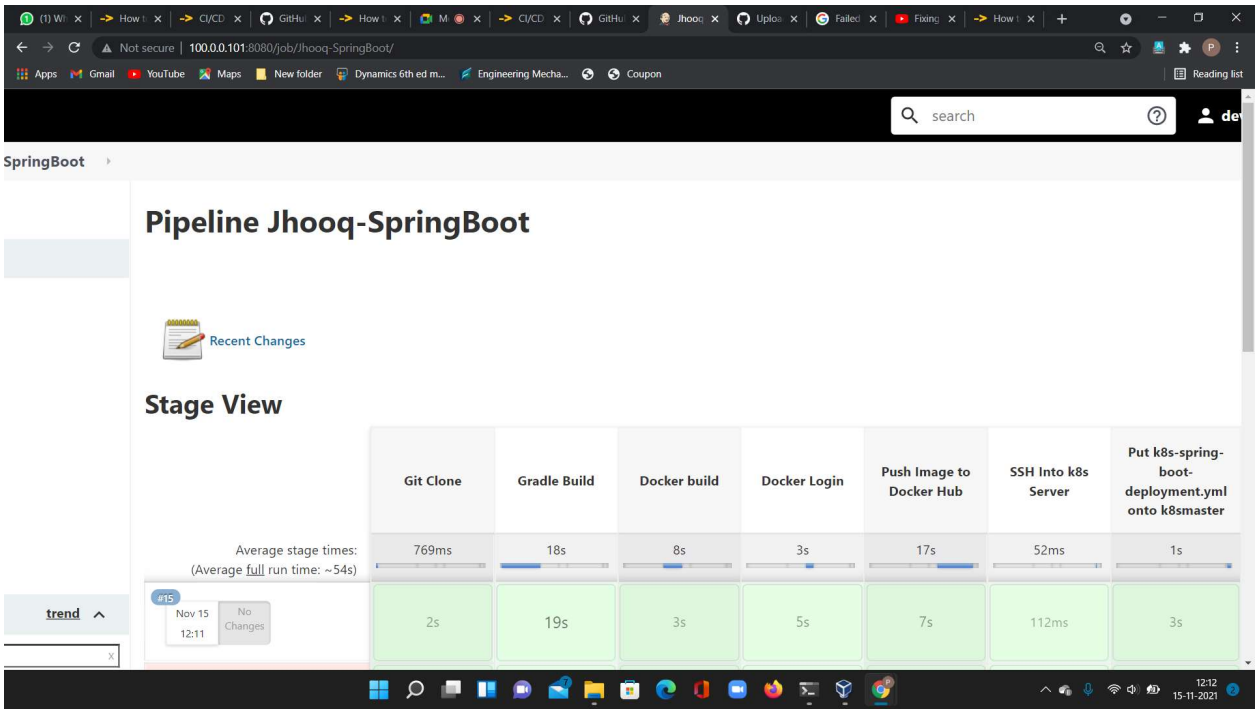
Code

Pipeline(stage6 with internal stages)

```
        stage("SSH Into k8s Server") { def remote = [:] remote.name = 'K8S master' remote.host = '100.0.0.2' remote.user = 'vagrant' remote.password = 'vagrant' remote.allowAnyHosts = true
        stage('Put k8s-spring-boot-deployment.yml onto k8smaster') { sshPut remote: remote, from: 'k8s-spring-boot-deployment.yml', into: '.' }
         stage('Deploy spring boot') { sshCommand remote: remote, command: "kubectl apply -f k8s-spring-boot-deployment.yml" } }
```


vagrant@k8smaster: ~
vagrant@k8smaster:~$ pwd
/home/vagrant
vagrant@k8smaster:~$ ls
vagrant@k8smaster:~$ ls
k8s-spring-boot-deployment.yml
vagrant@k8smaster:~$
vagrant@jenkinsserver: ~
vagrant@jenkinsserver:~$
vagrant@k8sworker: ~
vagrant@k8sworker:~$

Pipeline:


Not secure | 100.0.0.101:8080/job/Jhooq-SpringBoot/
search
SpringBoot
Pipeline Jhooq-SpringBoot
Recent Changes
Stage View
Git Clone
Gradle Build
Docker build
Docker Login
Push Image to Docker Hub
SSH Into k8s Server
Put k8s-spring-boot-deployment.yml onto k8smaster
Average stage times:
(Average full run time: ~54s)
769ms
18s
8s
3s
17s
52ms
1s
#15
Nov 15
12:11
No Changes
2s
19s
3s
5s
7s
112ms
3s
trend

After,Application is successfully deployed on to kubernetes through Jenkins CICD pipeline.