INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

Dharwad

ज्ञानेन विकासः

Devops Jenkins Assignment

Submitted to:

Dr.Uma S

Submitted by:

Advay Aggarwal (18BCS002)
Perumalla Tushar(18BCS065)
Meghana Hadimani(18BCS052)
Samana B S(18BCS088)
Rahul Pryadarshini(18BCS074)

Setting up CI/CD Jenkins pipeline for kubernetes

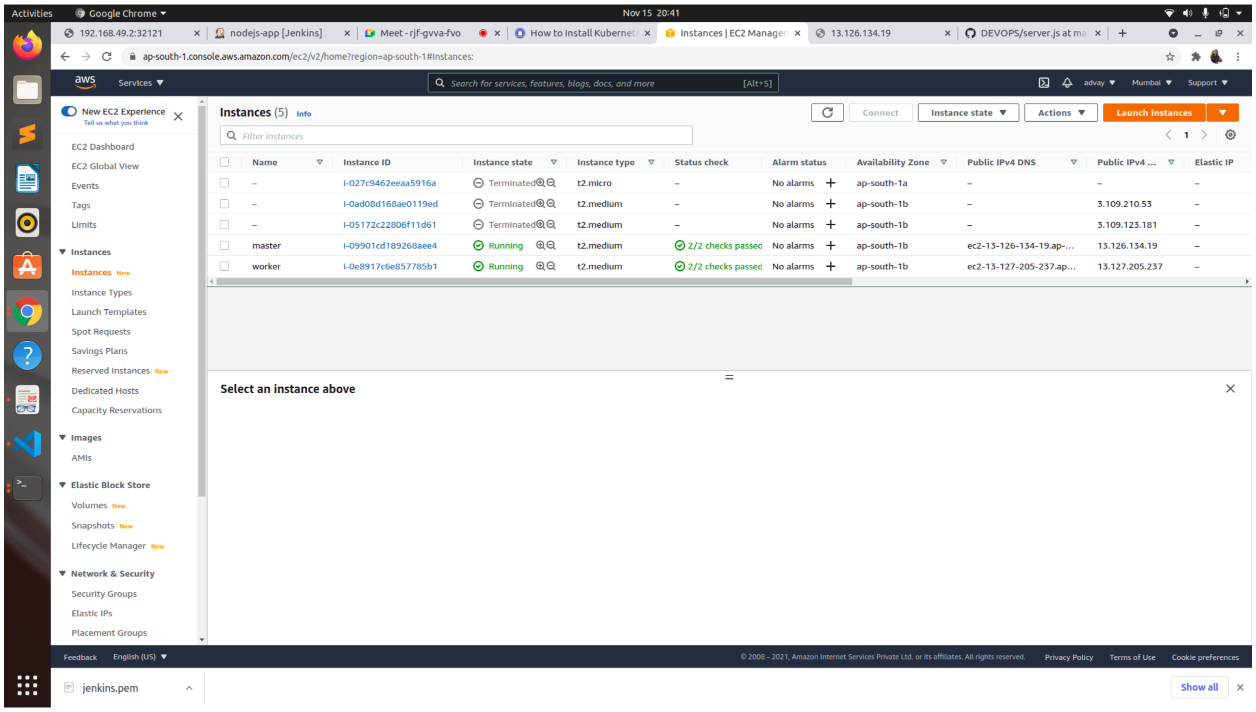
Tools and Technologies used :

- Github
- Docker and Docker hub
- Jenkins
- Kubernetes Cluster

Prerequisites:

- NodeJS v8+
- 2 AWS Ec2 Ubuntu instances of size t2.medium and 15 GB of volumes attached.
- Install Docker and kubernetes on AWS Instances

AWS Instance:



Instances (5)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
–	i-027c9462eeaa5916a	Terminated	t2.micro	–	No alarms	ap-south-1a	–	–	–
–	i-0ad08d168ae0119ed	Terminated	t2.medium	–	No alarms	ap-south-1b	–	3.109.210.53	–
–	i-05172c22806f11d61	Terminated	t2.medium	–	No alarms	ap-south-1b	–	3.109.123.181	–
master	i-09901cd189268aee4	Running	t2.medium	2/2 checks passed	No alarms	ap-south-1b	ec2-13-126-134-19.ap-...	13.126.134.19	–
worker	i-0e8917c6e857785b1	Running	t2.medium	2/2 checks passed	No alarms	ap-south-1b	ec2-13-127-205-237.ap...	13.127.205.237	–

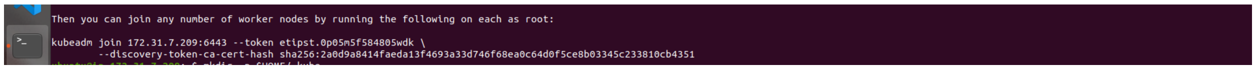
Select an instance above

Step - 1 : Setting Up kubernetes Cluster

We have used the kubeadm tool to set up the kubernetes cluster.

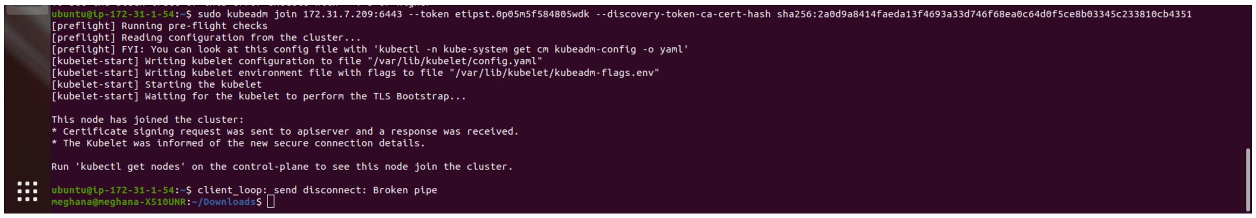
Setting up a kubernetes cluster containing 2 nodes master and worker.

1. Install docker and add docker daemon after that enable and start the docker on both the nodes
2. Install kubernetes(Kubeadm,Kubelet and Kubectl) on both the nodes
3. Initialize the master node using kubeadm. Output of this command would be a key , through which worker nodes can join the kubernetes cluster. Copy the token and save it somewhere.



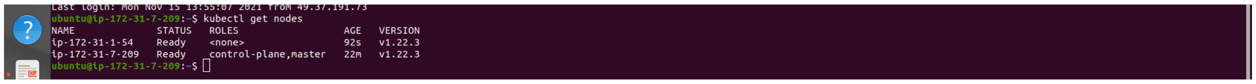
```
Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.7.209:6443 --token etipst.0p05m5f584805wdk \
        --discovery-token-ca-cert-hash sha256:2a0d9a8414faeda13f4693a33d746f68ea0c64d0f5ce8b03345c233810cb4351
```

4. Using the token copied earlier run this command on worker node : `sudo kubeadm join 172.31.7.219:6443 --token etipst.0p05m5f584805wdk --discovery-token-ca-cert-hashsha256:2a0d9a8414faeda13f4693a33d746f68ea0c64d0f5ce8b03345c233810cb4351`
   to get following output



```
ubuntu@ip-172-31-1-54:~$ sudo kubeadm join 172.31.7.209:6443 --token etipst.0p05m5f584805wdk --discovery-token-ca-cert-hash sha256:2a0d9a8414faeda13f4693a33d746f68ea0c64d0f5ce8b03345c233810cb4351
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@ip-172-31-1-54:~$ client_loop: send disconnect: Broken pipe
meghana@meghana-X510UNR:~/Downloads$
```

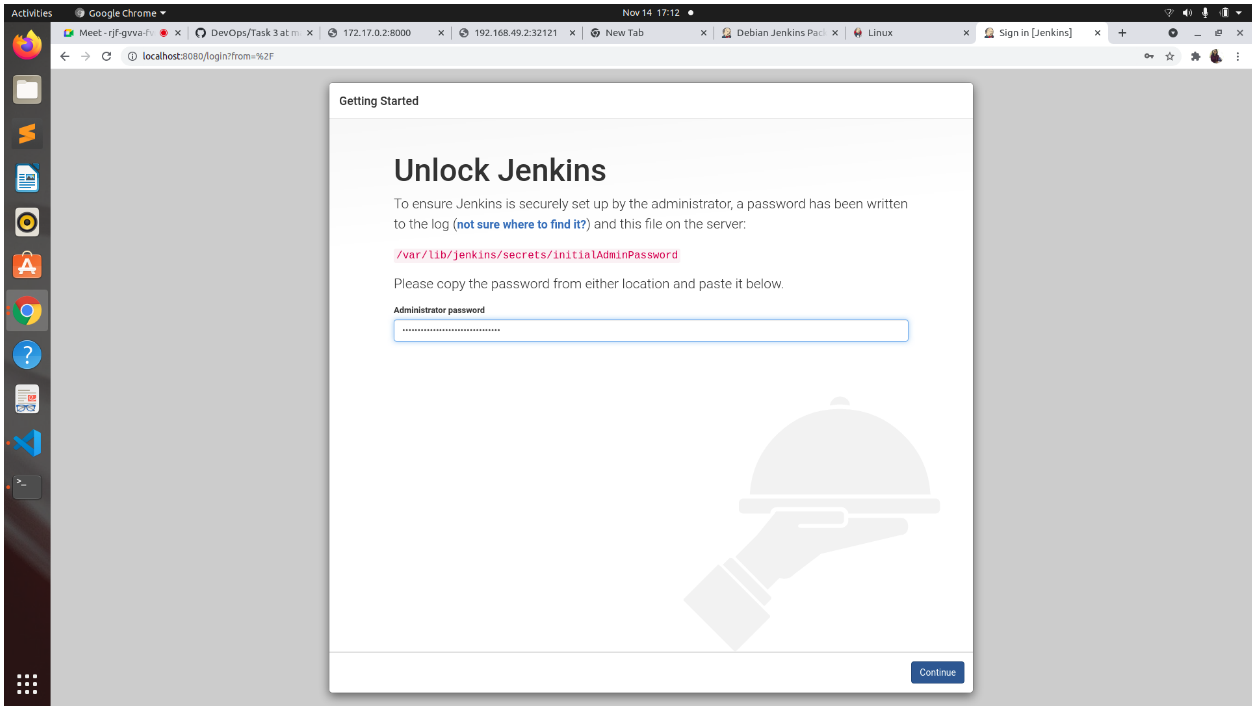
5. When we run `kubectl get nodes` on master node Now there would be 2 nodes one master and one newly joined worker node newly joined node’s role name would be <none> to label it as worker Using the token copied earlier



```
Last login: Mon Nov 15 13:55:07 2021 from 49.37.191.73
ubuntu@ip-172-31-7-209:~$ kubectl get nodes
NAME              STATUS   ROLES                  AGE   VERSION
ip-172-31-1-54    Ready    <none>                 92s   v1.22.3
ip-172-31-7-209   Ready    control-plane,master   22m   v1.22.3
ubuntu@ip-172-31-7-209:~$
```

Step - 2 : Setting up jenkins On aws ec2 ubuntu

1. Install Java and Jenkins. Run the jenkins file, the first time it asks for an admin password. Run `cat /var/lib/jenkins/secrets/initialAdminPassword` and copy the password and paste it.


Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

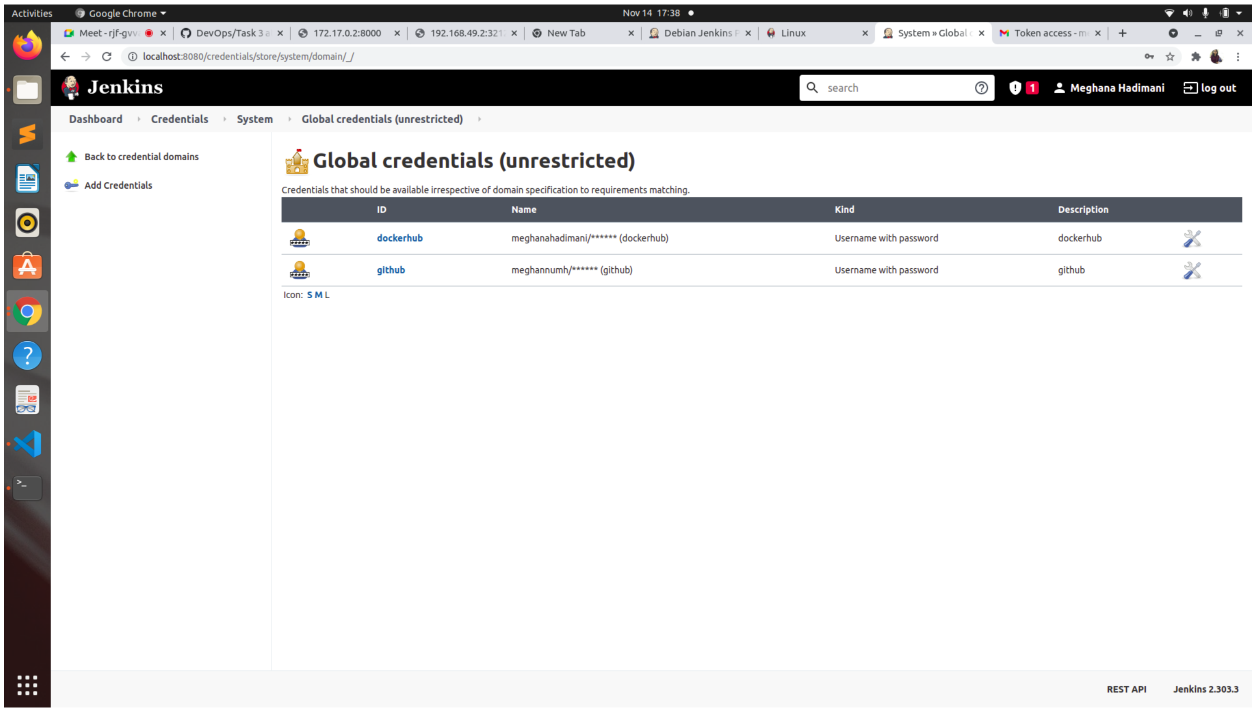
Please copy the password from either location and paste it below.

Administrator password

Next install the suggested plugins After installing the dashboard opens up Go to Manage Jenkins -> Manage Plugins -> Available and install plugins for Nodejs, Docker, Kubernetes, Github, Kubernetes CLI.

2. Configure Docker Hub and github credentials. Go to Manage Jenkins -> Manage credentials and add a credential for docker hub with your username and password. Create a credential id (which will be used later) and description. Similarly add github credentials, i.e. username

and Personal Access Token.



Dashboard › Credentials › System › Global credentials (unrestricted) ›

Back to credential domains
Add Credentials

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

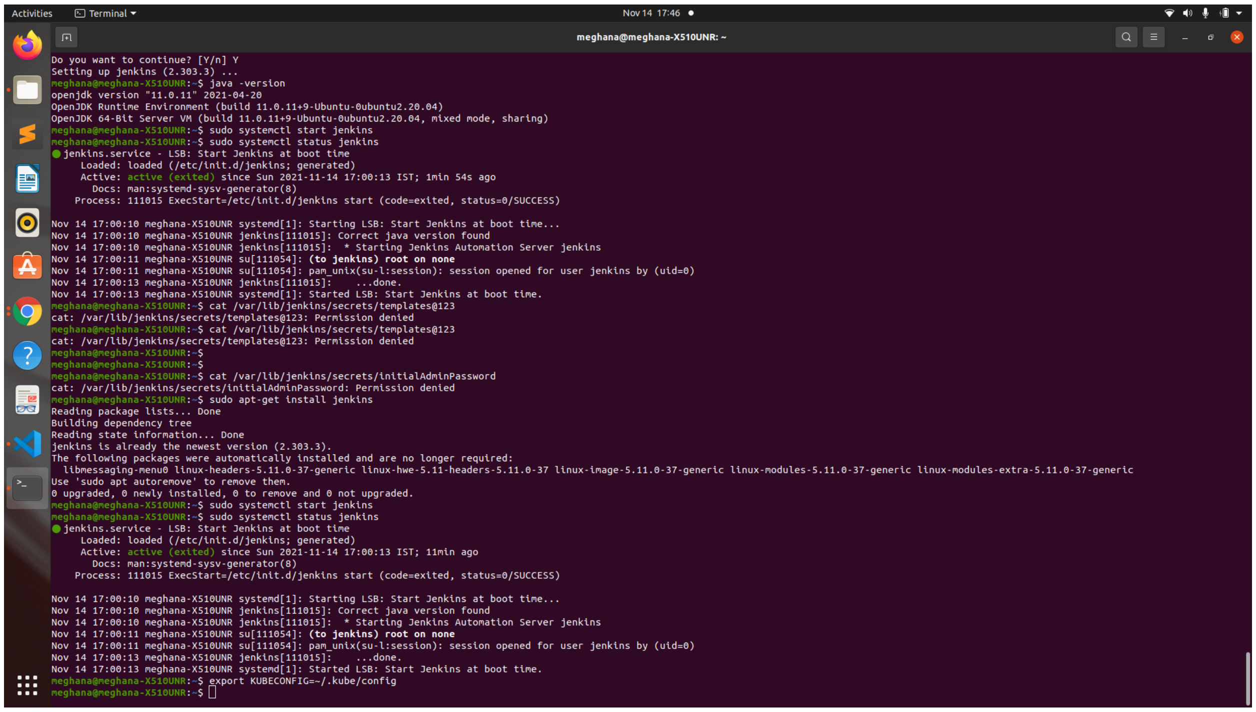
	ID	Name	Kind	Description
	dockerhub	meghanahadimani/****** (dockerhub)	Username with password	dockerhub
	github	meghannumh/****** (github)	Username with password	github

Icon: S M L

REST API   Jenkins 2.303.3

3. Run the following command to create an environment variable named KUBECONFIG and provide the .kube/config path. Jenkins goes to this path to execute kubectl commands.

```
export KUBECONFIG=~/.kube/config
```



```
Do you want to continue? [Y/n] Y
Setting up jenkins (2.303.3) ...
meghana@meghana-X510UNR:~$ java -version
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed mode, sharing)
meghana@meghana-X510UNR:~$ sudo systemctl start jenkins
meghana@meghana-X510UNR:~$ sudo systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
     Loaded: loaded (/etc/init.d/jenkins; generated)
     Active: active (exited) since Sun 2021-11-14 17:00:13 IST; 1min 54s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 111015 ExecStart=/etc/init.d/jenkins start (code=exited, status=0/SUCCESS)

Nov 14 17:00:10 meghana-X510UNR systemd[1]: Starting LSB: Start Jenkins at boot time...
Nov 14 17:00:10 meghana-X510UNR jenkins[111015]: Correct java version found
Nov 14 17:00:10 meghana-X510UNR jenkins[111015]:  * Starting Jenkins Automation Server jenkins
Nov 14 17:00:11 meghana-X510UNR su[111054]: (to jenkins) root on none
Nov 14 17:00:11 meghana-X510UNR su[111054]: pam_unix(su-l:session): session opened for user jenkins by (uid=0)
Nov 14 17:00:13 meghana-X510UNR jenkins[111015]:    ...done.
Nov 14 17:00:13 meghana-X510UNR systemd[1]: Started LSB: Start Jenkins at boot time.
meghana@meghana-X510UNR:~$ cat /var/lib/jenkins/secrets/templates@123
cat: /var/lib/jenkins/secrets/templates@123: Permission denied
meghana@meghana-X510UNR:~$ cat /var/lib/jenkins/secrets/templates@123
cat: /var/lib/jenkins/secrets/templates@123: Permission denied
meghana@meghana-X510UNR:~$
meghana@meghana-X510UNR:~$
meghana@meghana-X510UNR:~$ cat /var/lib/jenkins/secrets/initialAdminPassword
cat: /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
meghana@meghana-X510UNR:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
jenkins is already the newest version (2.303.3).
The following packages were automatically installed and are no longer required:
  libmessaging-menu0 linux-headers-5.11.0-37-generic linux-hwe-5.11-headers-5.11.0-37 linux-image-5.11.0-37-generic linux-modules-5.11.0-37-generic linux-modules-extra-5.11.0-37-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
meghana@meghana-X510UNR:~$ sudo systemctl start jenkins
meghana@meghana-X510UNR:~$ sudo systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
     Loaded: loaded (/etc/init.d/jenkins; generated)
     Active: active (exited) since Sun 2021-11-14 17:00:13 IST; 11min ago
       Docs: man:systemd-sysv-generator(8)
    Process: 111015 ExecStart=/etc/init.d/jenkins start (code=exited, status=0/SUCCESS)

Nov 14 17:00:10 meghana-X510UNR systemd[1]: Starting LSB: Start Jenkins at boot time...
Nov 14 17:00:10 meghana-X510UNR jenkins[111015]: Correct java version found
Nov 14 17:00:10 meghana-X510UNR jenkins[111015]:  * Starting Jenkins Automation Server jenkins
Nov 14 17:00:11 meghana-X510UNR su[111054]: (to jenkins) root on none
Nov 14 17:00:11 meghana-X510UNR su[111054]: pam_unix(su-l:session): session opened for user jenkins by (uid=0)
Nov 14 17:00:13 meghana-X510UNR jenkins[111015]:    ...done.
Nov 14 17:00:13 meghana-X510UNR systemd[1]: Started LSB: Start Jenkins at boot time.
meghana@meghana-X510UNR:~$ export KUBECONFIG=~/.kube/config
meghana@meghana-X510UNR:~$
```

Install docker on jenkins server and add current ubuntu usr and jenkins to docker group

Step - 3: Setting up code

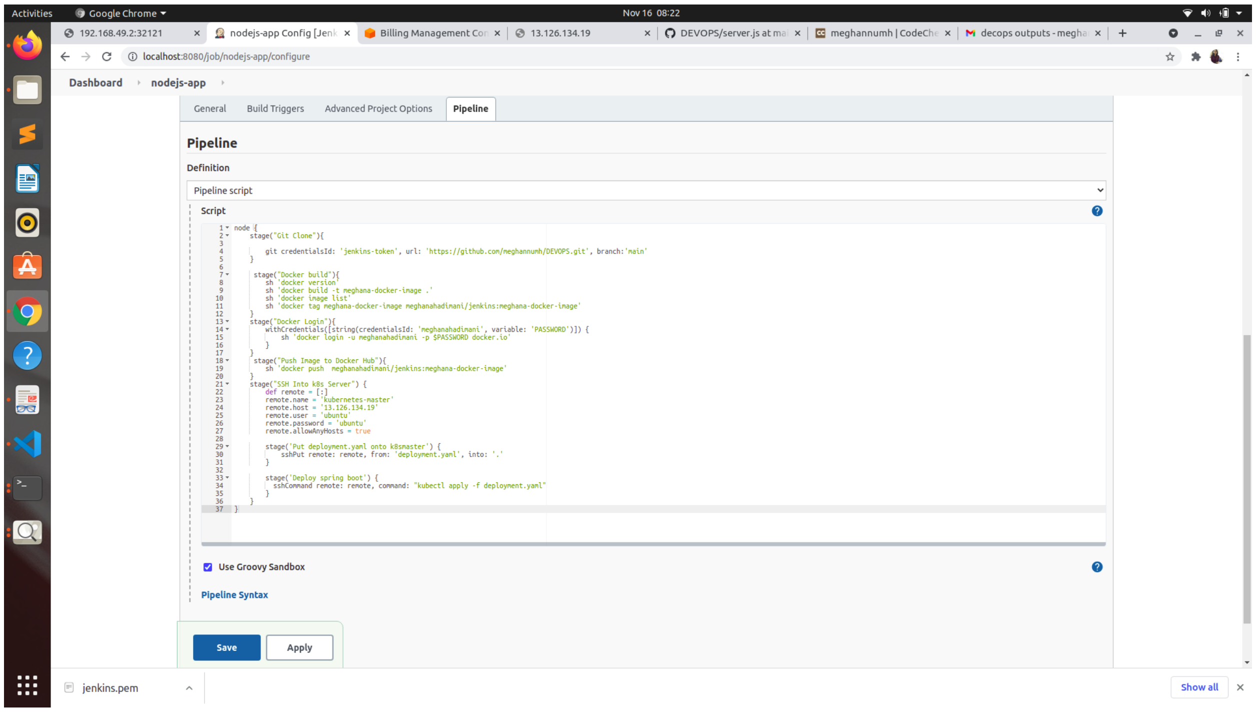
We made a simple nodejs application, code is on github. Code link

Deployment file link

Step - 4: Building the pipeline

1. Create a jenkins pipeline. Add this code to create stages which include:
    A. Git clone
    B. Docker Build
    C. Docker login
    D. Push image to docker hub
    E. Adding ssh into k8 server
    F. Copy Deployment.yaml file to kubernetes master
    G. Create the deployment and service on kubernetes

Pipeline code:



Pipeline

Definition

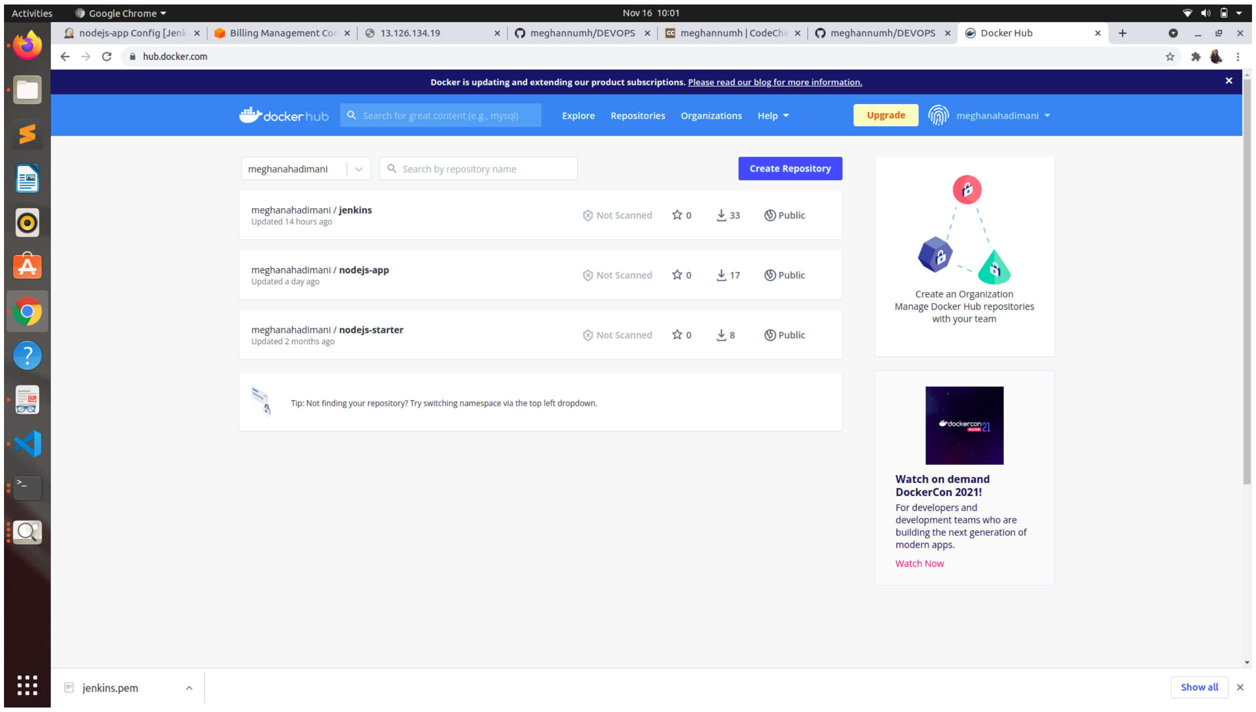
Pipeline script

Script

```
node {
    stage("Git Clone"){

        git credentialsId: 'jenkins-token', url: 'https://github.com/meghannumh/DEVOPS.git', branch:'main'
    }

     stage("Docker build"){
        sh 'docker version'
        sh 'docker build -t meghana-docker-image .'
        sh 'docker image list'
        sh 'docker tag meghana-docker-image meghanahadimani/jenkins:meghana-docker-image'
    }
    stage("Docker Login"){
        withCredentials([string(credentialsId: 'meghanahadimani', variable: 'PASSWORD')]) {
            sh 'docker login -u meghanahadimani -p $PASSWORD docker.io'
        }
    }
     stage("Push Image to Docker Hub"){
        sh 'docker push  meghanahadimani/jenkins:meghana-docker-image'
    }
    stage("SSH Into k8s Server") {
        def remote = [:]
        remote.name = 'kubernetes-master'
        remote.host = '13.126.134.19'
        remote.user = 'ubuntu'
        remote.password = 'ubuntu'
        remote.allowAnyHosts = true

        stage('Put deployment.yaml onto k8smaster') {
            sshPut remote: remote, from: 'deployment.yaml', into: '.'
        }

        stage('Deploy spring boot') {
          sshCommand remote: remote, command: "kubectl apply -f deployment.yaml"
        }
    }
}
```

☑ Use Groovy Sandbox

Pushing image to docker hub:



Docker is updating and extending our product subscriptions. Please read our blog for more information.

meghanahadimani / jenkins
Updated 14 hours ago
Not Scanned 0 33 Public

meghanahadimani / nodejs-app
Updated a day ago
Not Scanned 0 17 Public

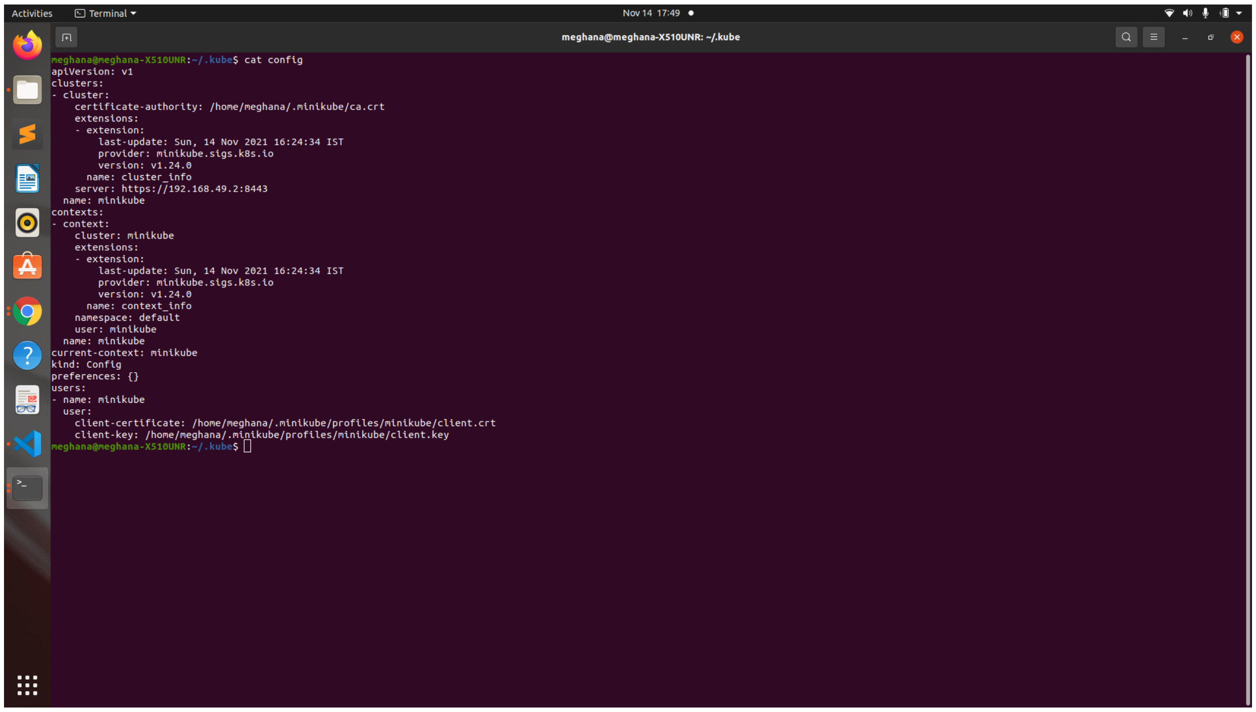
meghanahadimani / nodejs-starter
Updated 2 months ago
Not Scanned 0 8 Public

Tip: Not finding your repository? Try switching namespace via the top left dropdown.

Create an Organization
Manage Docker Hub repositories with your team

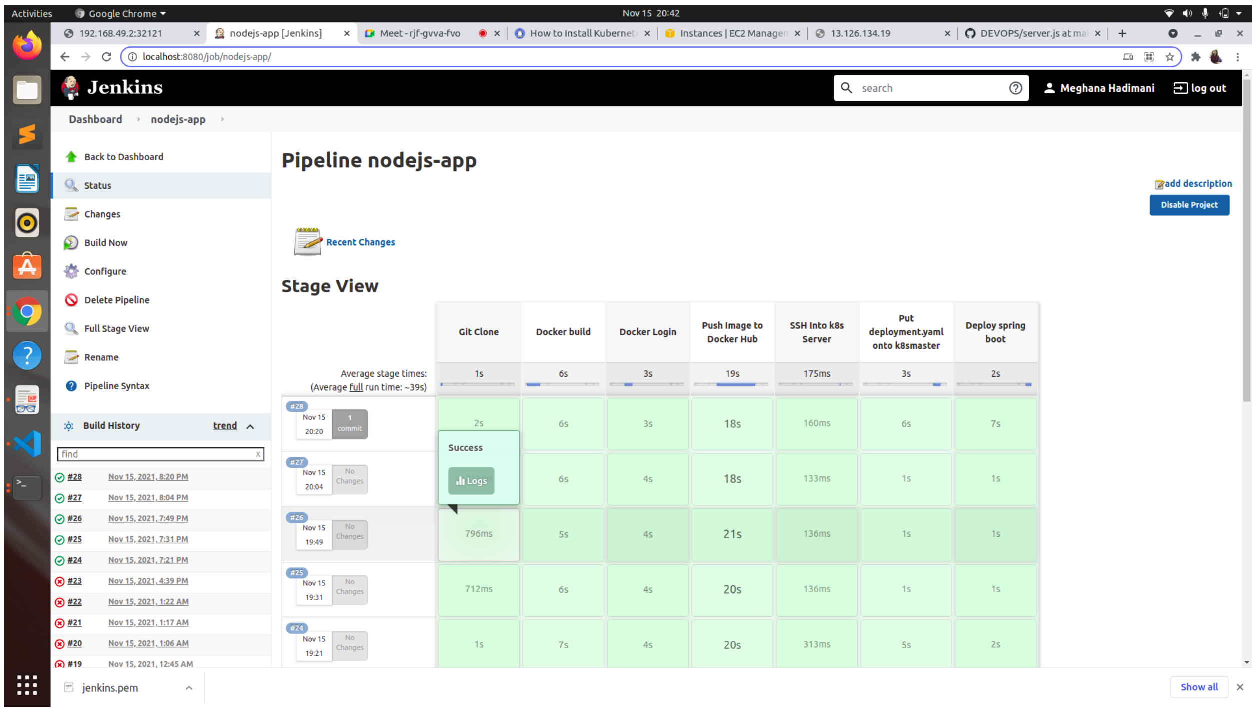
Watch on demand DockerCon 2021!
For developers and development teams who are building the next generation of modern apps.
Watch Now

Go to the ~/ .kube directory and type the command `cat config` to get all the information about the cluster to fill inside the withKubeConfig method in the pipeline.



```
meghana@meghana-X510UNR:~/.kube$ cat config
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /home/meghana/.minikube/ca.crt
    extensions:
    - extension:
        last-update: Sun, 14 Nov 2021 16:24:34 IST
        provider: minikube.sigs.k8s.io
        version: v1.24.0
      name: cluster_info
    server: https://192.168.49.2:8443
  name: minikube
contexts:
- context:
    cluster: minikube
    extensions:
    - extension:
        last-update: Sun, 14 Nov 2021 16:24:34 IST
        provider: minikube.sigs.k8s.io
        version: v1.24.0
      name: context_info
    namespace: default
    user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
    client-certificate: /home/meghana/.minikube/profiles/minikube/client.crt
    client-key: /home/meghana/.minikube/profiles/minikube/client.key
meghana@meghana-X510UNR:~/.kube$
```

Logs of Deployment on Jenkins:



Pipeline nodejs-app

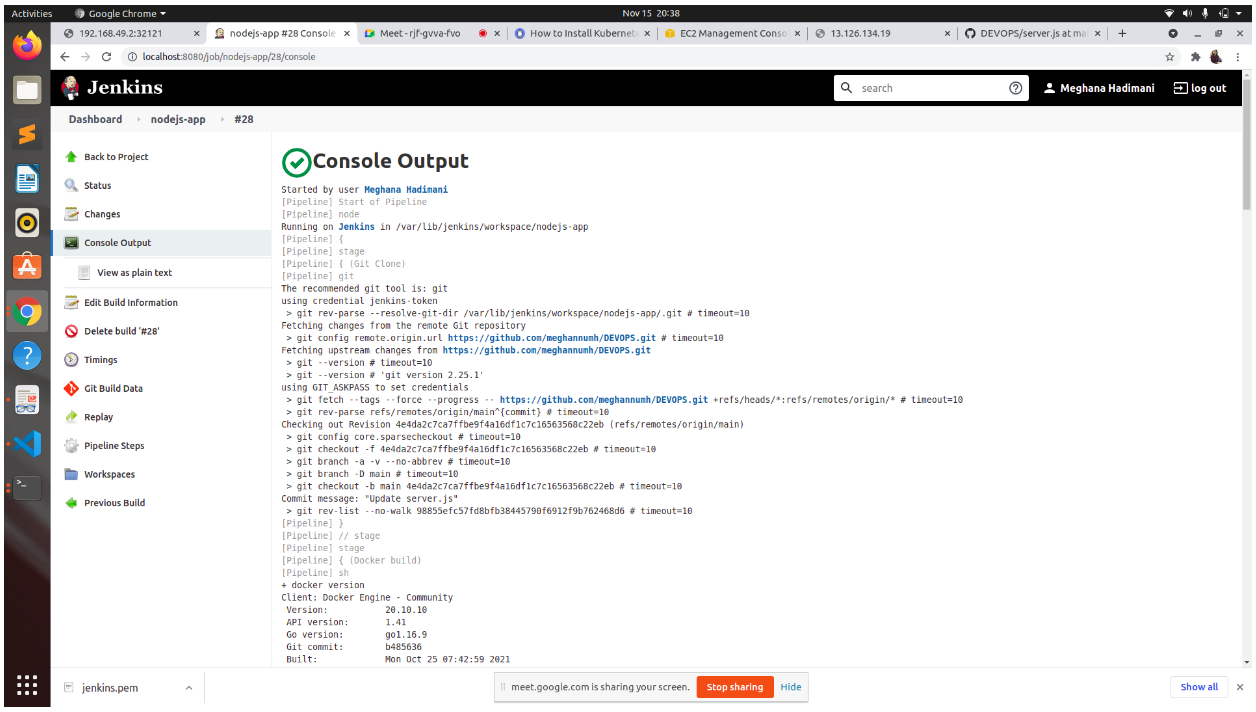
Stage View

	Git Clone	Docker build	Docker Login	Push Image to Docker Hub	SSH Into k8s Server	Put deployment.yaml onto k8smaster	Deploy spring boot
Average stage times: (Average full run time: ~39s)	1s	6s	3s	19s	175ms	3s	2s
#28 Nov 15 20:20 1 commit	2s	6s	3s	18s	160ms	6s	7s
#27 Nov 15 20:04 No Changes	Success Logs	6s	4s	18s	133ms	1s	1s
#26 Nov 15 19:49 No Changes	796ms	5s	4s	21s	136ms	1s	1s
#25 Nov 15 19:31 No Changes	712ms	6s	4s	20s	136ms	1s	1s
#24 Nov 15 19:21 No Changes	1s	7s	4s	20s	313ms	5s	2s

Build History

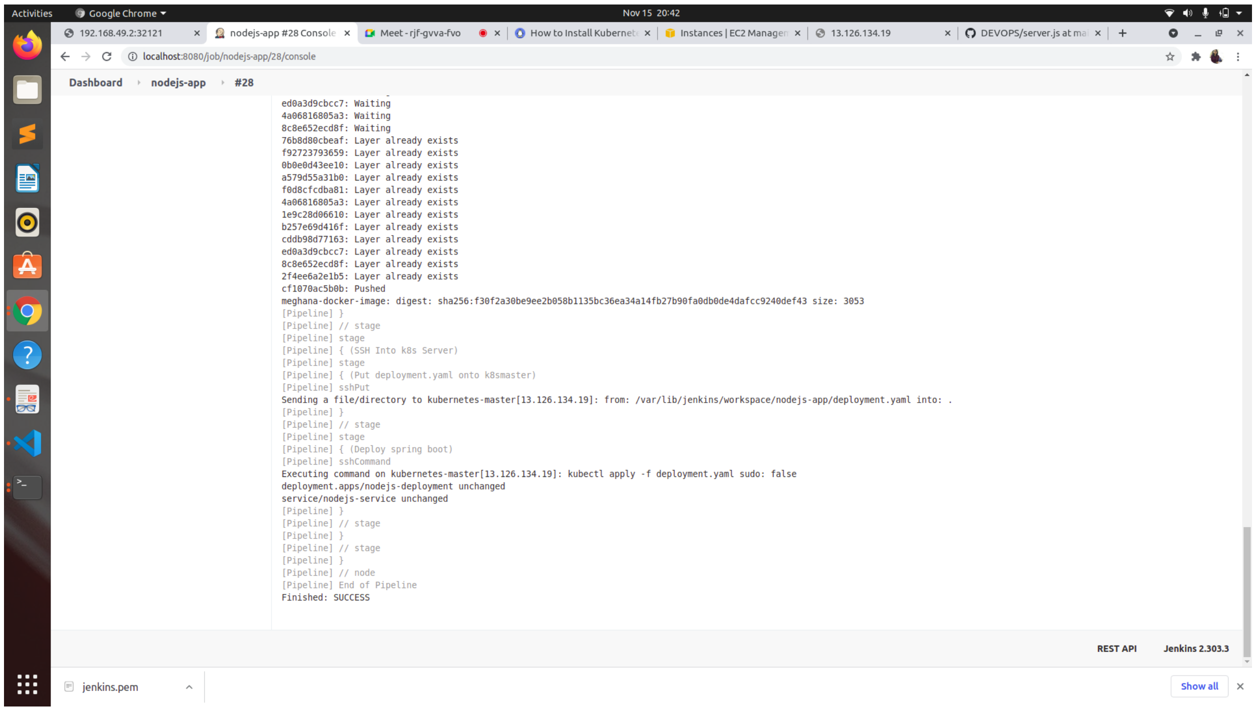
#28	Nov 15, 2021, 8:20 PM
#27	Nov 15, 2021, 8:04 PM
#26	Nov 15, 2021, 7:49 PM
#25	Nov 15, 2021, 7:31 PM
#24	Nov 15, 2021, 7:21 PM
#23	Nov 15, 2021, 4:39 PM
#22	Nov 15, 2021, 1:22 AM
#21	Nov 15, 2021, 1:17 AM
#20	Nov 15, 2021, 1:06 AM
#19	Nov 15, 2021, 12:45 AM

Output:



Console Output

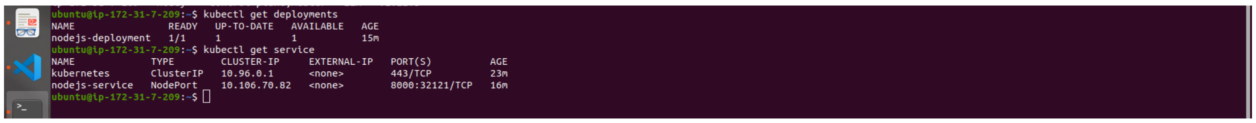
```
Started by user Meghana Hadimani
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/nodejs-app
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Git Clone)
[Pipeline] git
The recommended git tool is: git
using credential jenkins-token
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/nodejs-app/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/meghannumh/DEVOPS.git # timeout=10
Fetching upstream changes from https://github.com/meghannumh/DEVOPS.git
 > git --version # timeout=10
 > git --version # 'git version 2.25.1'
using GIT_ASKPASS to set credentials
 > git fetch --tags --force --progress -- https://github.com/meghannumh/DEVOPS.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 4e4da2c7ca7ffbe9f4a16df1c7c16563568c22eb (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 4e4da2c7ca7ffbe9f4a16df1c7c16563568c22eb # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D main # timeout=10
 > git checkout -b main 4e4da2c7ca7ffbe9f4a16df1c7c16563568c22eb # timeout=10
Commit message: "Update server.js"
 > git rev-list --no-walk 98855efc57fd8bfb38445790f6912f9b762468d6 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Docker build)
[Pipeline] sh
+ docker version
Client: Docker Engine - Community
 Version:           20.10.10
 API version:       1.41
 Go version:        go1.16.9
 Git commit:        b485636
 Built:             Mon Oct 25 07:42:59 2021
```



```
ed0a3d9cbcc7: Waiting
4a06816805a3: Waiting
8c8e652ecd8f: Waiting
76b8d80cbeaf: Layer already exists
f92723793659: Layer already exists
0b0e0d43ee10: Layer already exists
a579d55a31b0: Layer already exists
f0d8cfcdba81: Layer already exists
4a06816805a3: Layer already exists
1e9c28d06610: Layer already exists
b257e69d416f: Layer already exists
cddb98d77163: Layer already exists
ed0a3d9cbcc7: Layer already exists
8c8e652ecd8f: Layer already exists
2f4ee6a2e1b5: Layer already exists
cf1070ac5b0b: Pushed
meghana-docker-image: digest: sha256:f30f2a30be9ee2b058b1135bc36ea34a14fb27b90fa0db0de4dafcc9240def43 size: 3053
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (SSH Into k8s Server)
[Pipeline] stage
[Pipeline] { (Put deployment.yaml onto k8smaster)
[Pipeline] sshPut
Sending a file/directory to kubernetes-master[13.126.134.19]: from: /var/lib/jenkins/workspace/nodejs-app/deployment.yaml into: .
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy spring boot)
[Pipeline] sshCommand
Executing command on kubernetes-master[13.126.134.19]: kubectl apply -f deployment.yaml sudo: false
deployment.apps/nodejs-deployment unchanged
service/nodejs-service unchanged
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Checking the deployments on kubernetes server

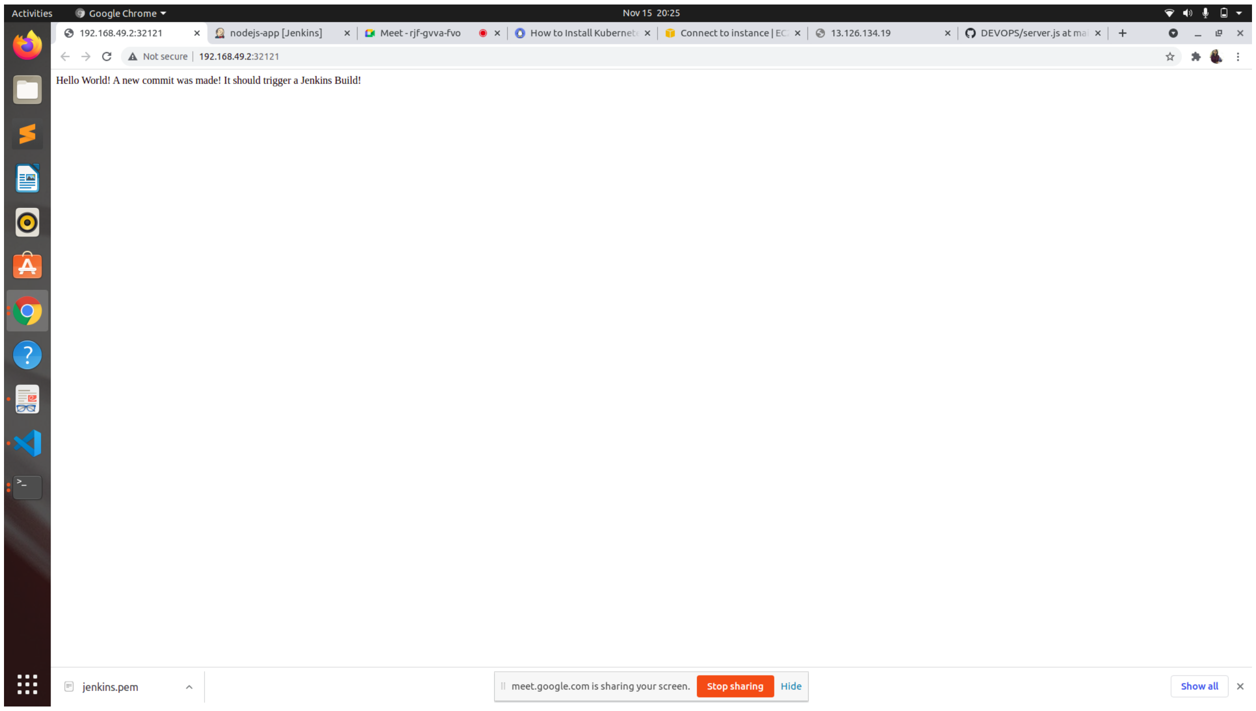
1. To get list of deployments on server run kubectl get deployments
2. To get list of services on kubernetes server run kubectl get services



```
ubuntu@ip-172-31-7-209:~$ kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
nodejs-deployment   1/1     1            1           15m
ubuntu@ip-172-31-7-209:~$ kubectl get service
NAME             TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
kubernetes       ClusterIP   10.96.0.1      <none>        443/TCP          23m
nodejs-service   NodePort    10.106.70.82   <none>        8000:32121/TCP   16m
ubuntu@ip-172-31-7-209:~$
```

node-js service in running on Nodeport, so Deployment is Successful

From the terminal output, node-js-service is running on port number 32121, so open tcp custom 32121 on kubernetes service instance



```
Hello World! A new commit was made! It should trigger a Jenkins Build!
```

Application is successfully deployed in kubernetes through Jenkins CICD pipeline.