INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

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

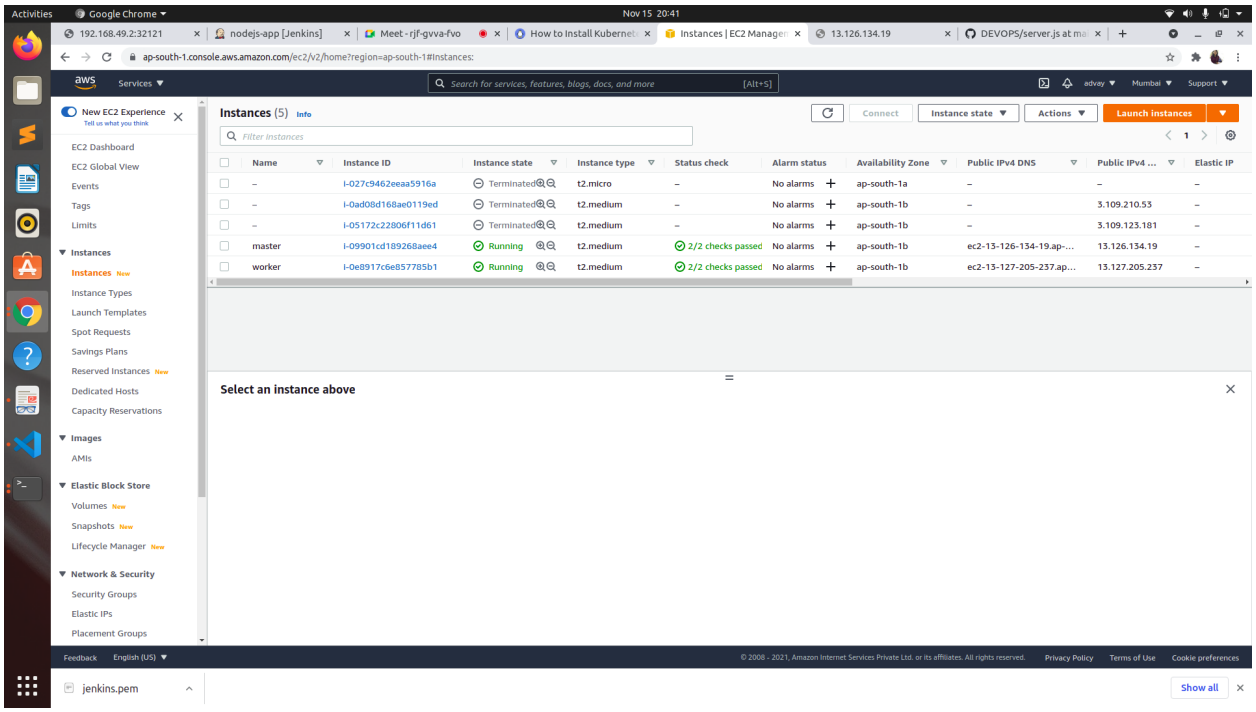
Tools and Technologies used :

- 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:

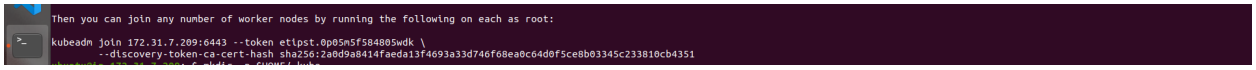


Step - 1 : Setting Up kubernetes Cluster

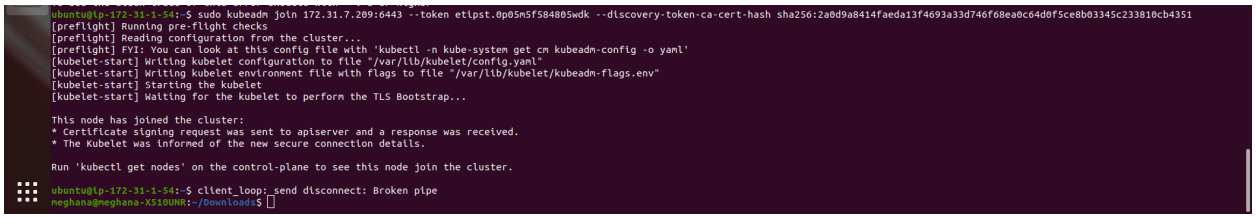
We have used the kubeadm tool to set up the kubernetes cluster.

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.



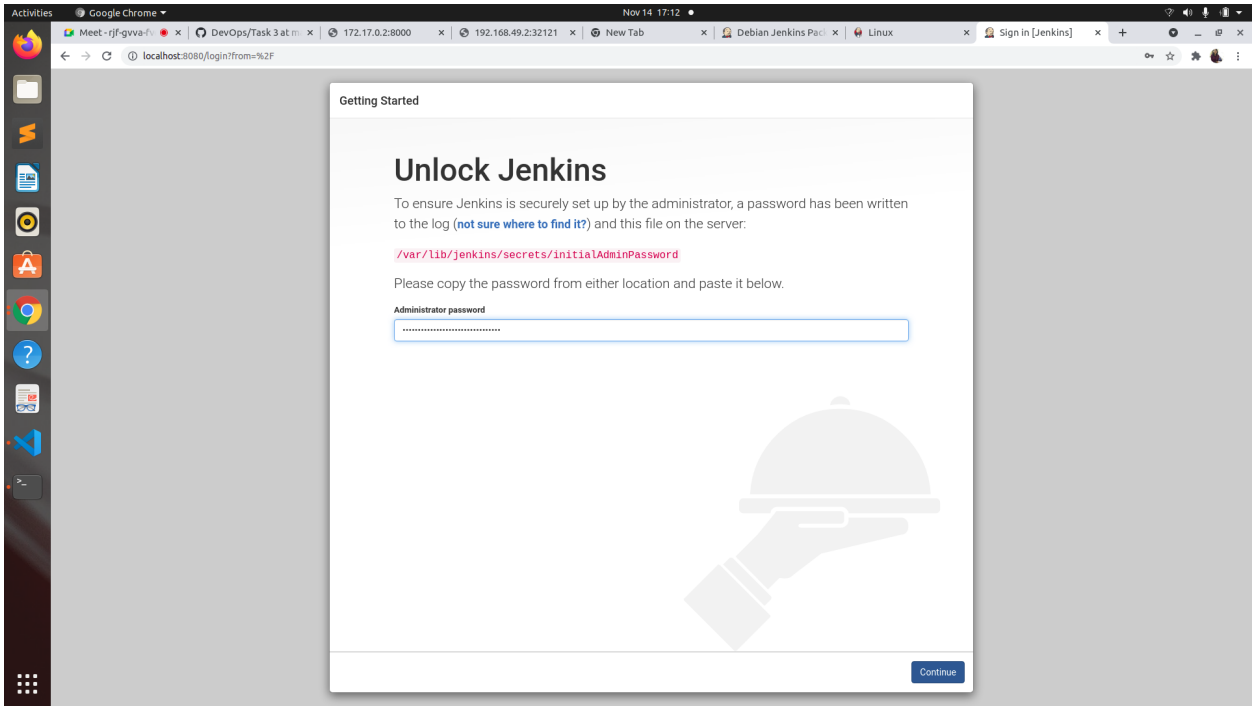
4. Using the token copied earlier run this command on worker node : `sudo kubeadm join 172.31.7.219:6443 --token etipst.0p05m5f584805wdk --discovery-token-ca-cert-hashsha256:2a0d9a8414faeda13f4693a33d746f68ea0c64d0f5ce8b03345c233810cb4351` to get following output



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

Step - 2 : Setting up jenkins locally

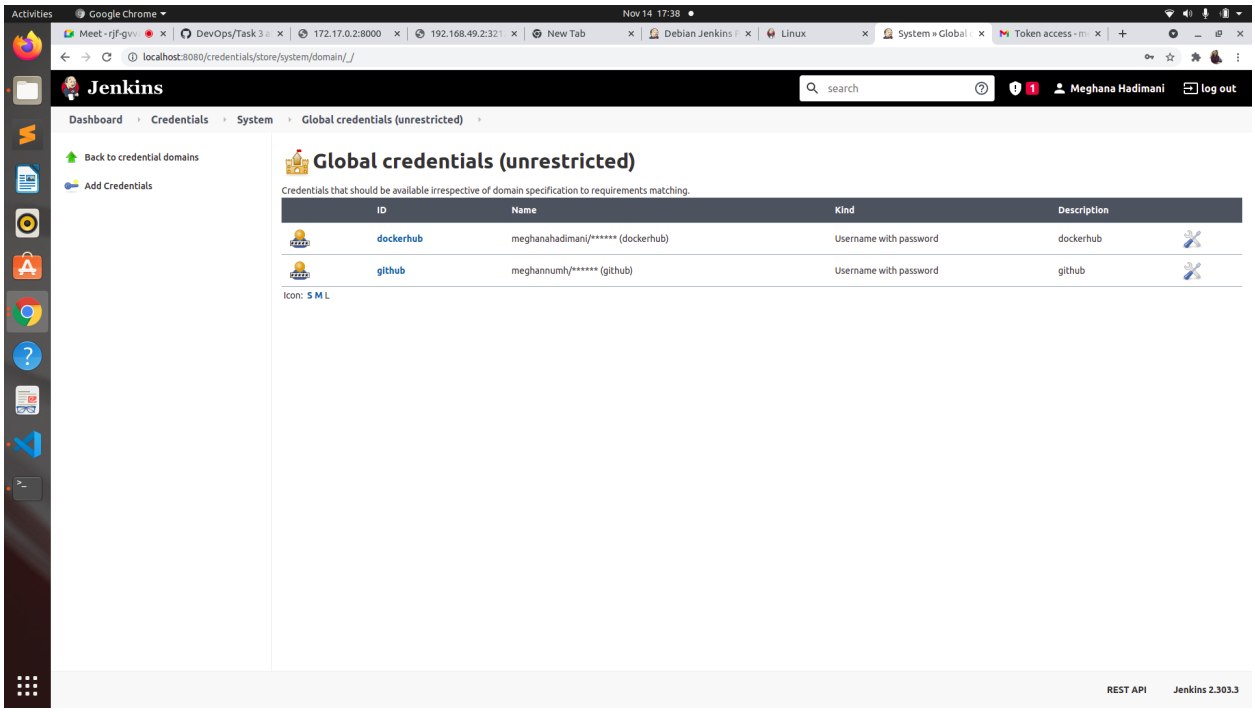
1. Install Java and Jenkins. Run the jenkins file, the first time it asks for an admin password. Run `cat /var/lib/jenkins/secrets/initialAdminPassword` and copy the password and paste it.



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.

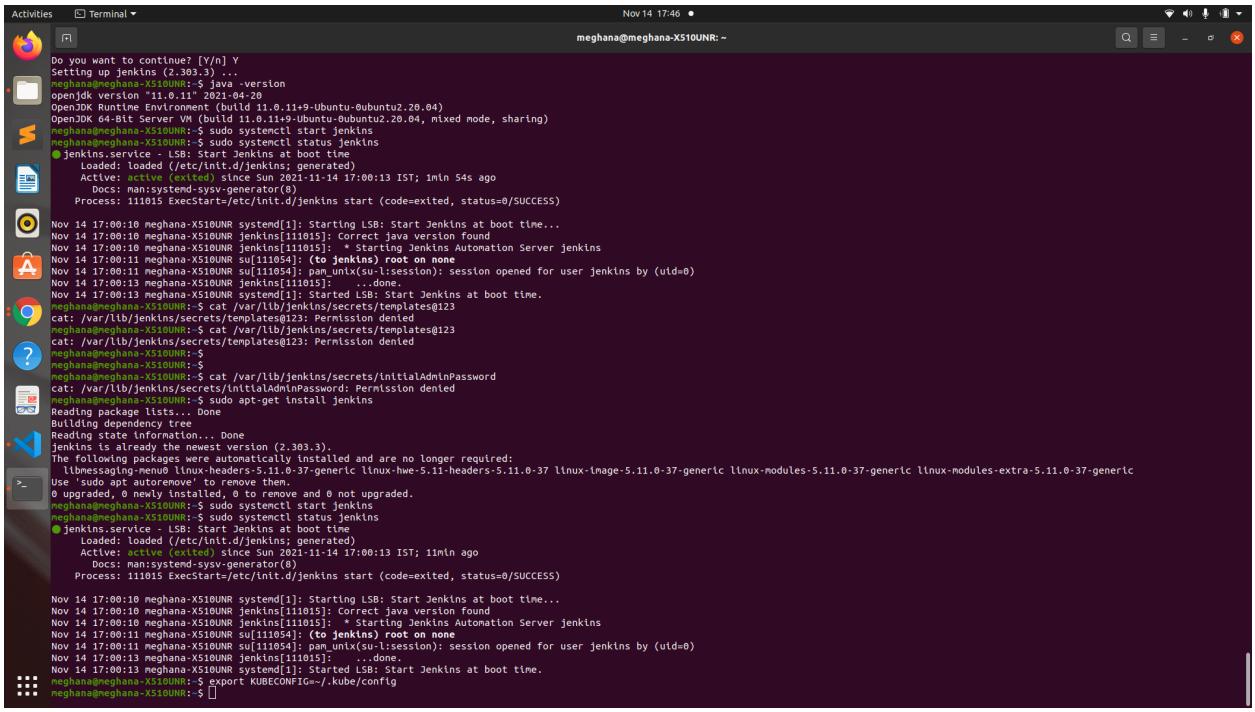
2. Configure Docker Hub and github credentials. Go to Manage Jenkins -> Manage credentials and add a credential for docker hub with your username and password. Create a credential id (which will be used later) and description. Similarly add github credentials, i.e. username

and Personal Access Token.



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
```



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

Step - 3: Setting up code

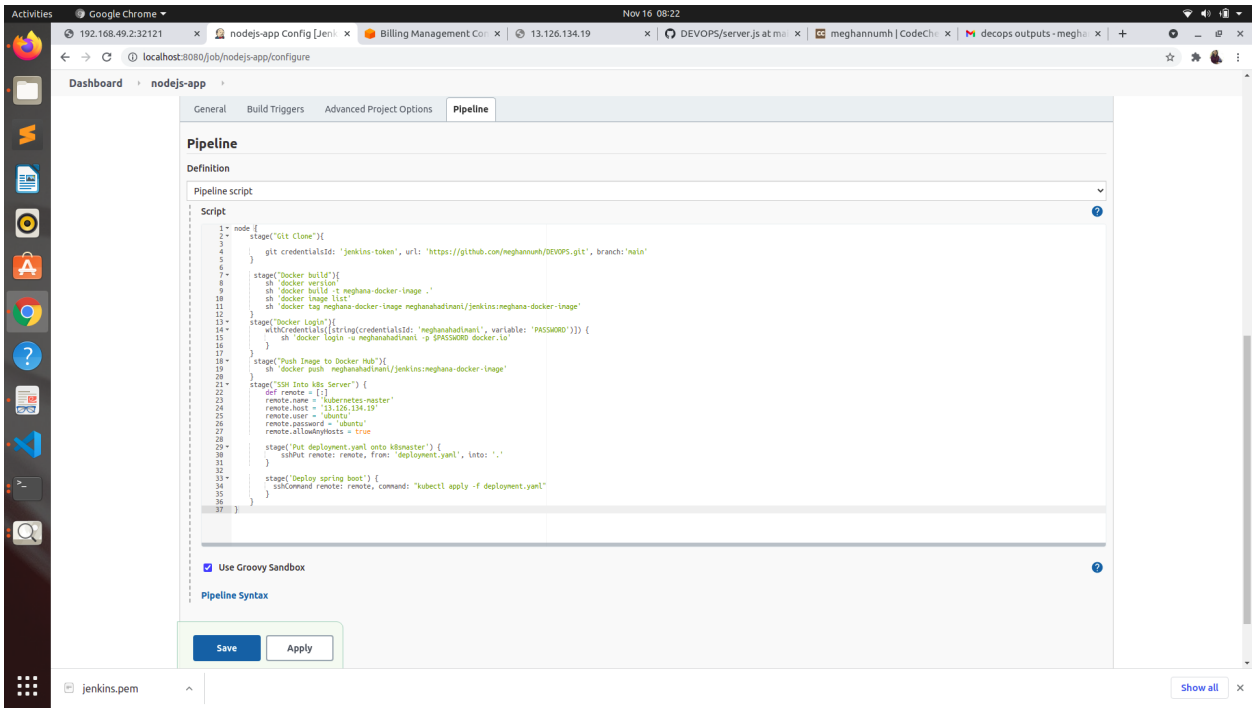
We made a simple node.js application, code is on github. Code link

Deployment file link

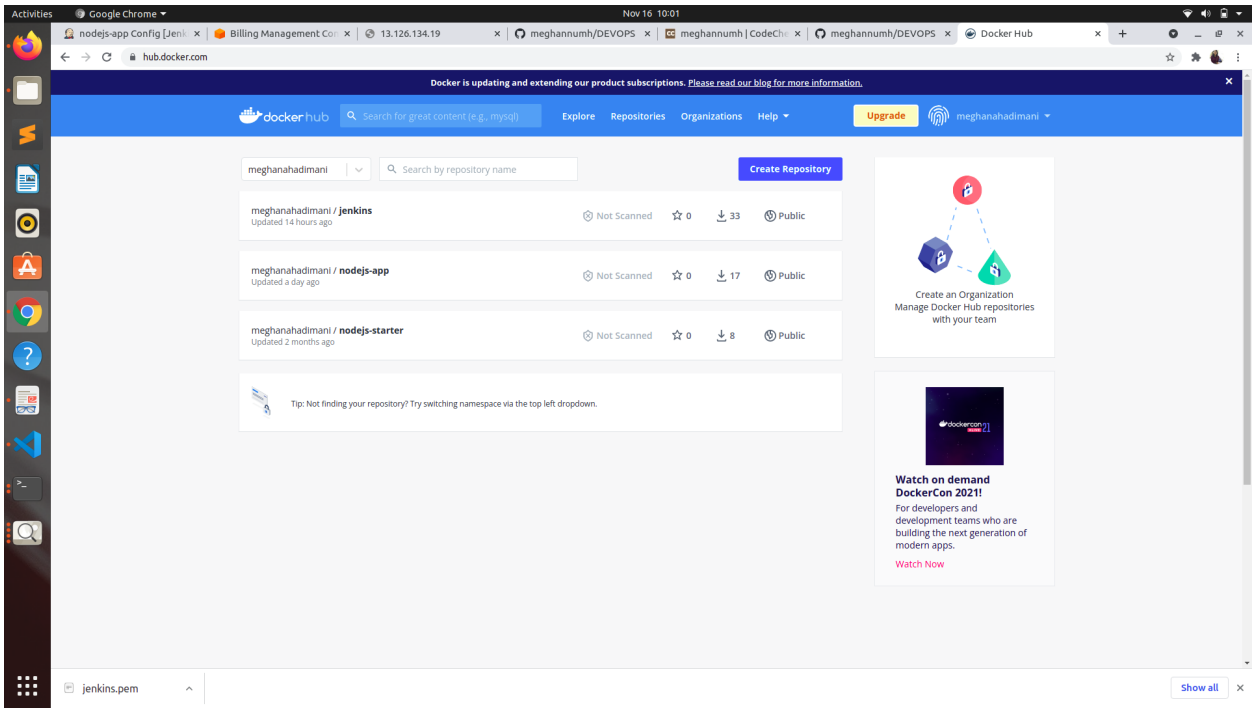
Step - 4: Building the pipeline

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

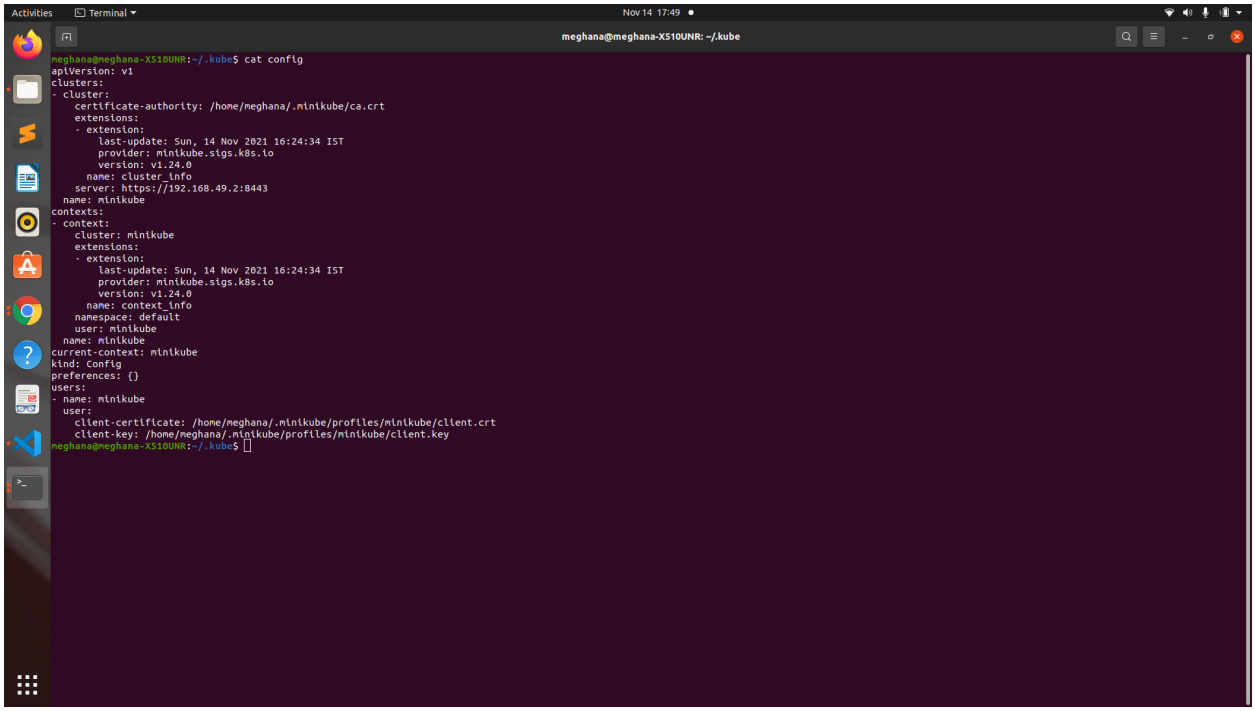
Pipeline code:



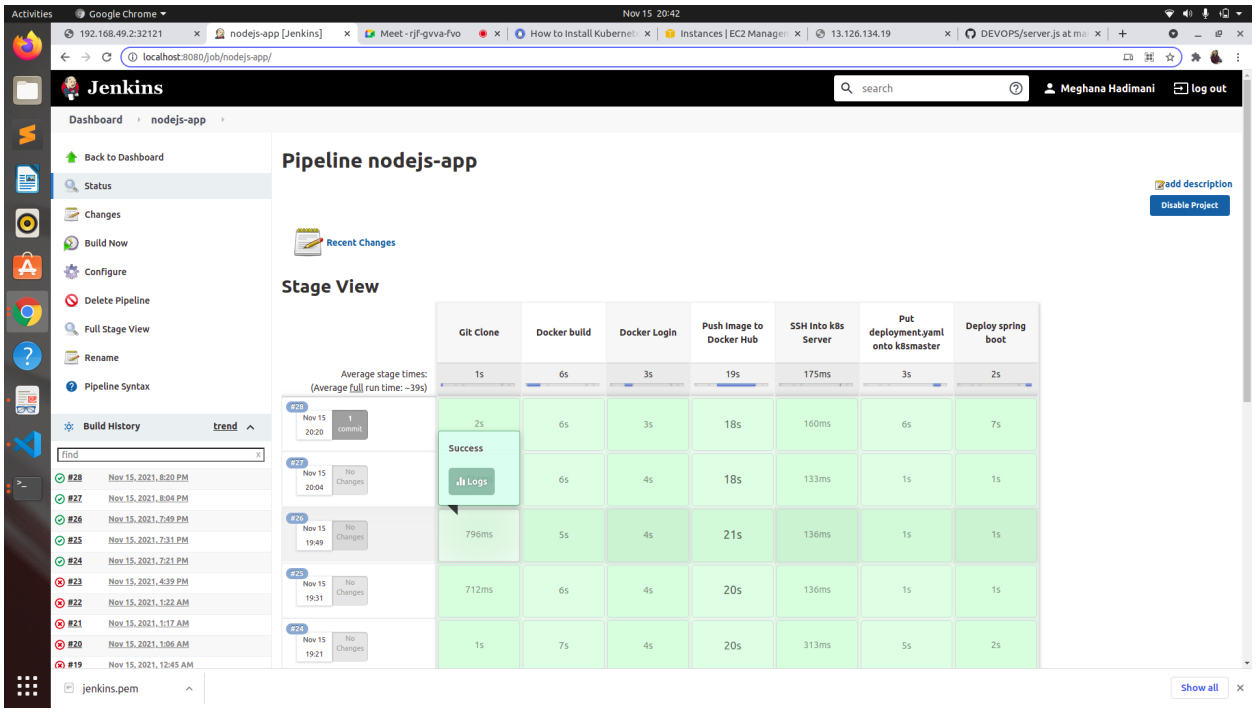
Pushing image to docker hub:



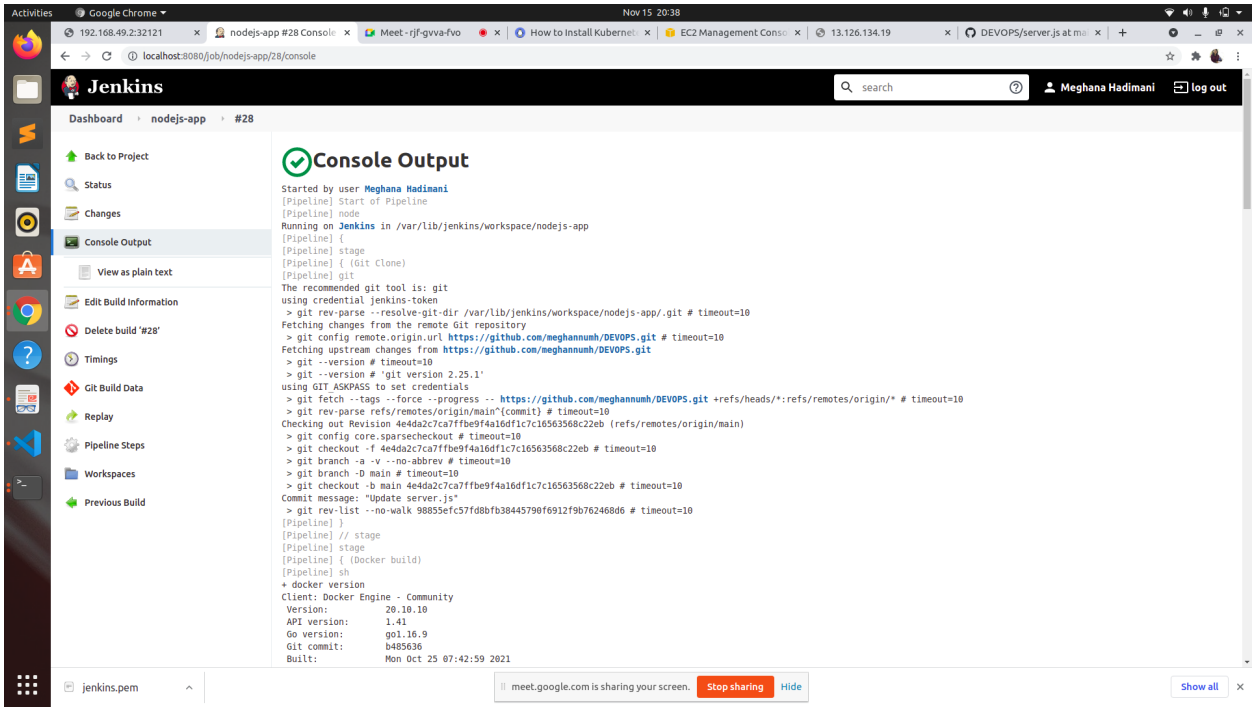
Go to the ~/ .kube directory and type the command `cat config` to get all the information about the cluster to fill inside the withKubeConfig method in the pipeline.

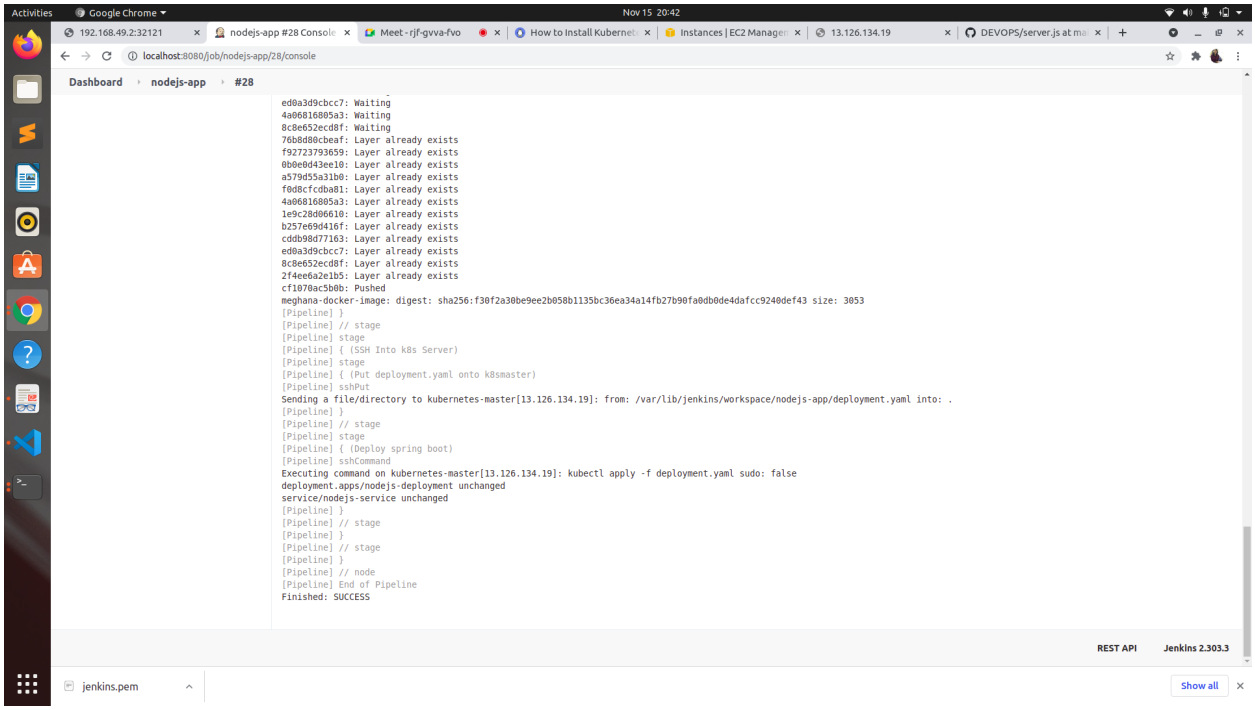


Logs of Deployment on Jenkins:



Output:



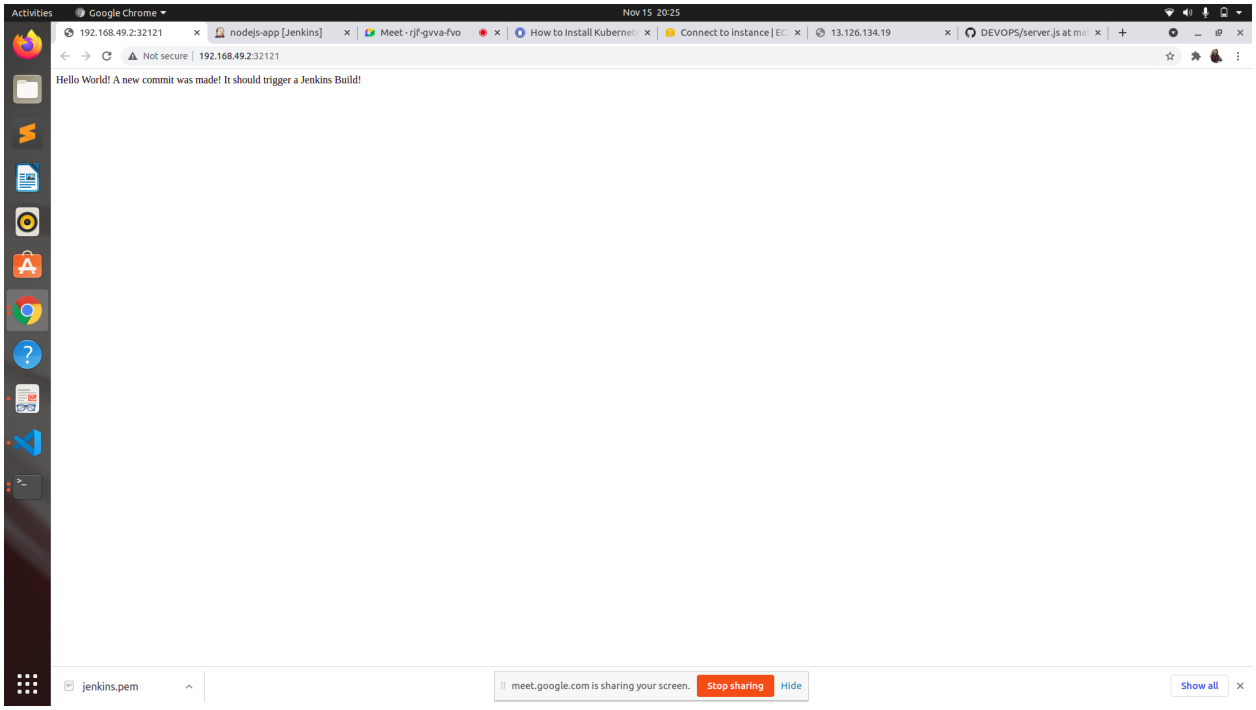


Checking the deployments on kubernetes server

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

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



Application is successfully deployed in kubernetes through Jenkins CICD pipeline.