**Deployed a Python Web Application on Kubernetes Cluster Capstone Project no 1**

**Process:**

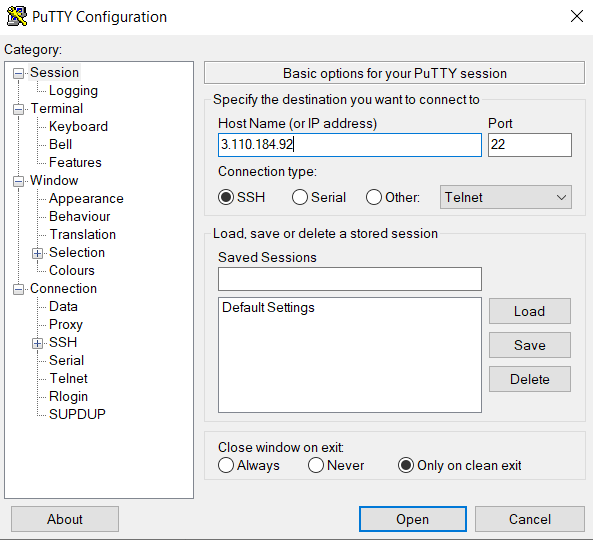
**Set Up AWS Resources**

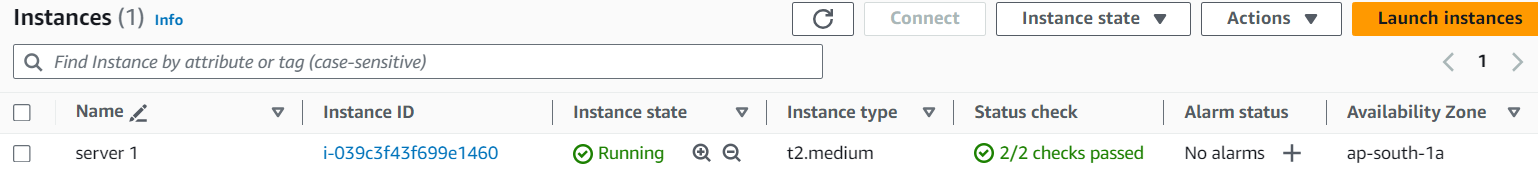
Logged in to AWS Management Console.  
• Launched EC2 Instance: (virtual machine - Ubuntu).

And security groups to allowed HTTP (port 80) and SSH (port 22) traffic with instance type - T2.medium and 15Gb disk size and created PPK file.

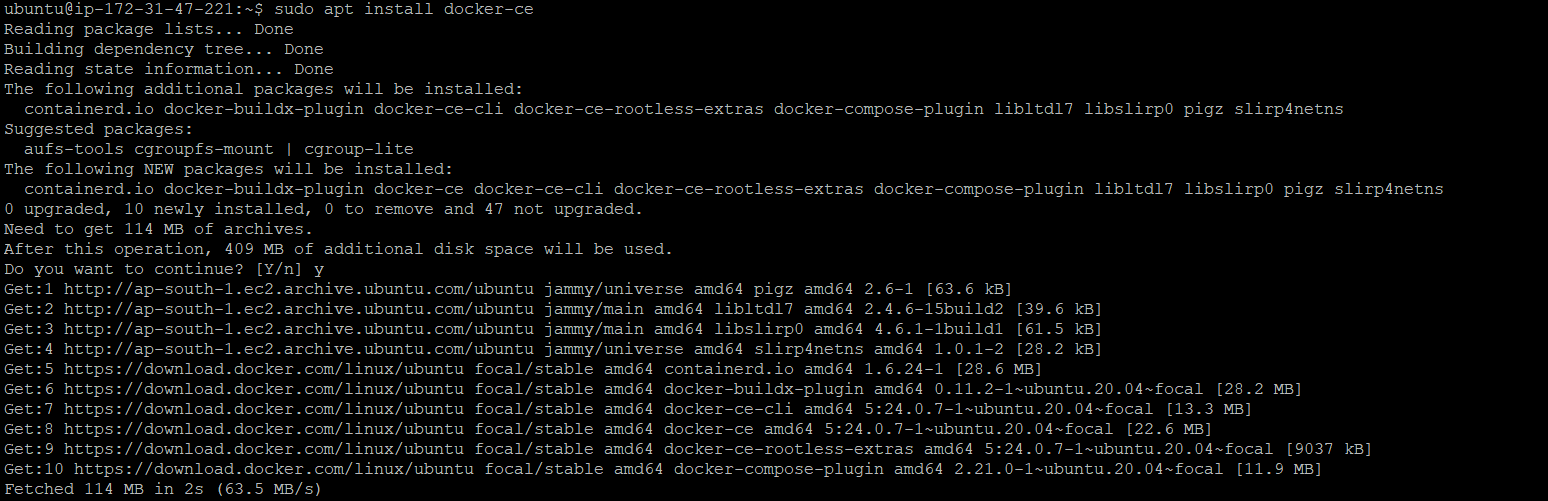
**Connect to Your EC2 Instance**

•    Used SSH to connect to EC2 instance using the PPK file via Putty as below.

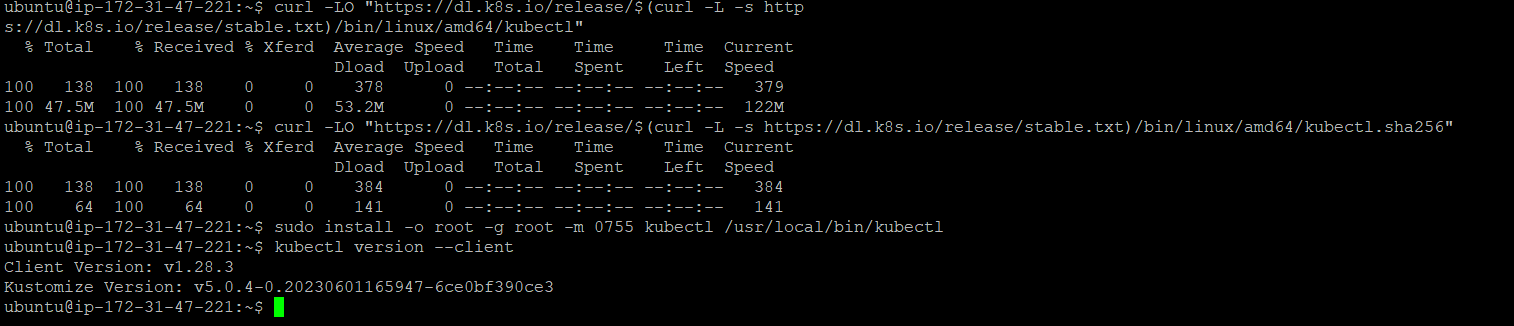




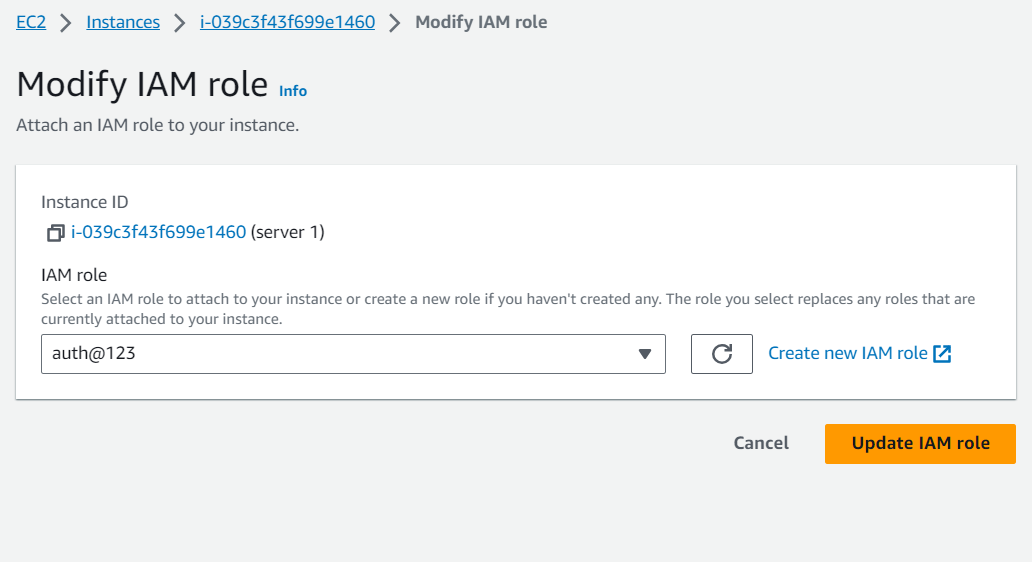
1. **Installed docker on to machine as below:**

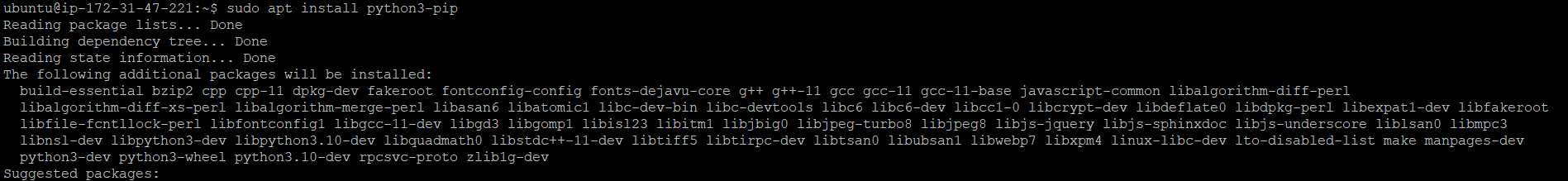


**2.Installed kubectl on to machine as below:**

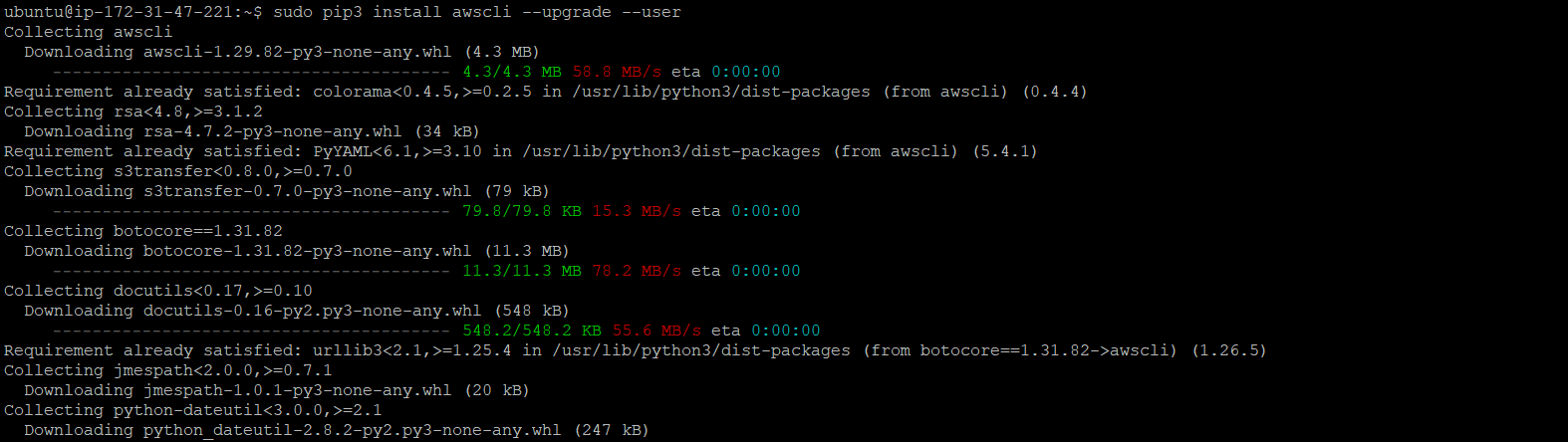


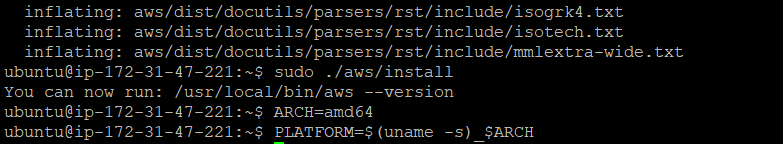
**3.Attached IAM role to the machine for the granting admin access.**



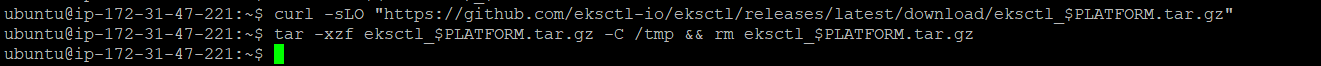
**4.Application is based on python so installed python on to the machine .** 

**5.Installed AWS CLI on to machine as below:**





**6.Installed EKSCTL on to machine as below for creating cluster as below:**

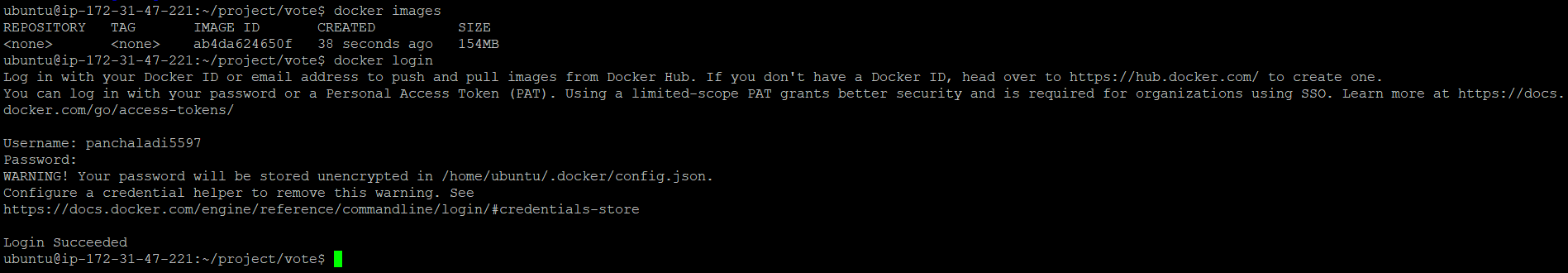


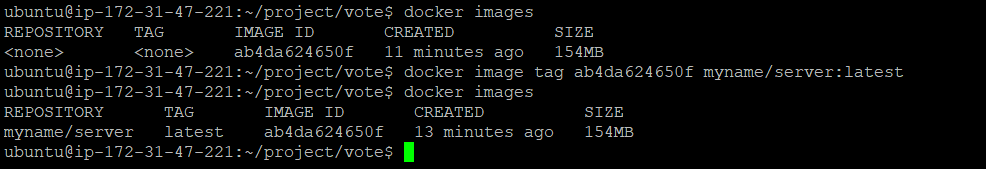
**7.build the docker file and got the image from it.**

dockerfile : <https://github.com/dockersamples/example-voting-app/blob/main/vote/Dockerfile>

Git Hub Repo : <https://github.com/dockersamples/example-voting-app>

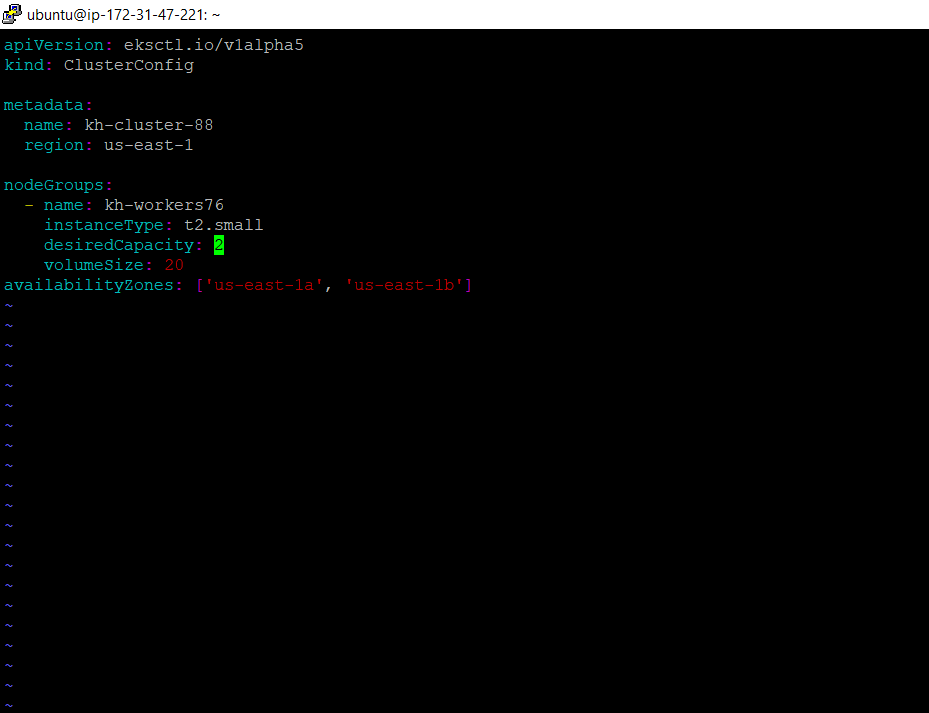




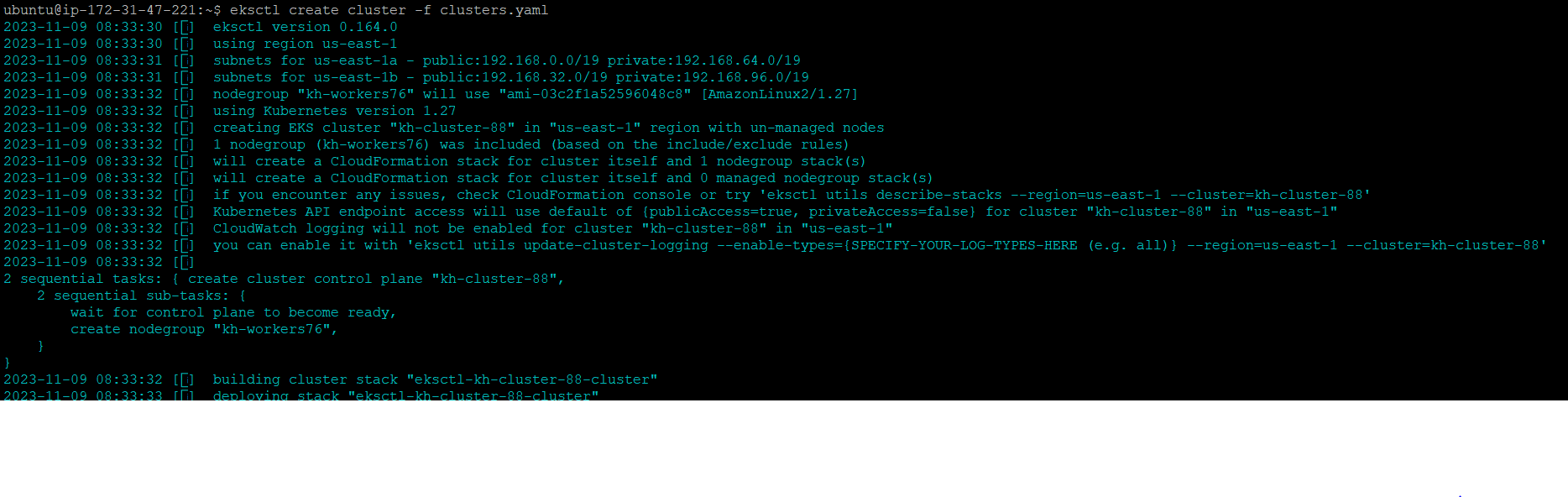


**8.Now created yaml file for cluster creation and launching it**

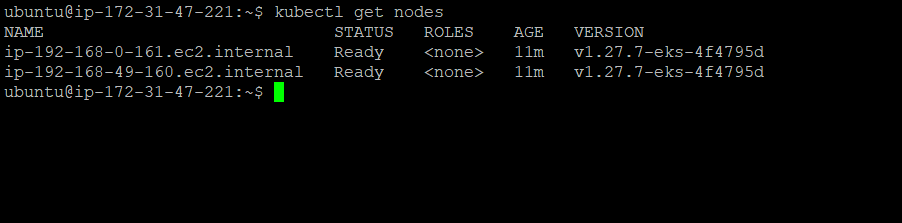
**As below:**



**9.Created custom cluster using clusters.yaml file as using below command:**

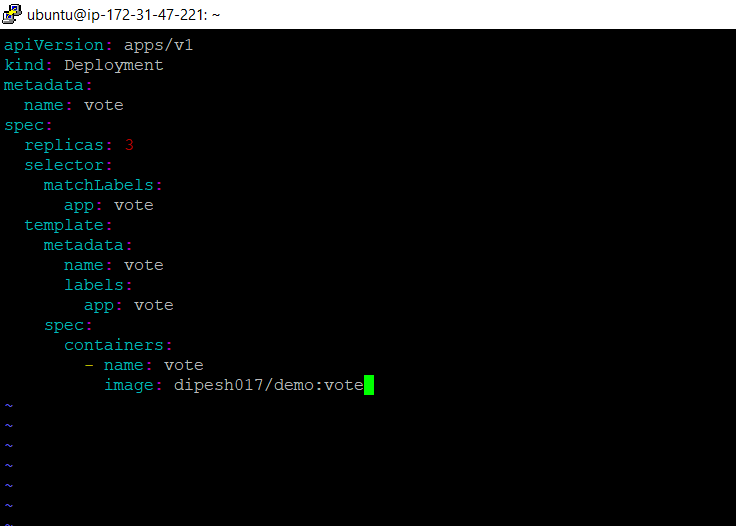


**10.After successfully launching the cluster two machine nodes are up as below:**



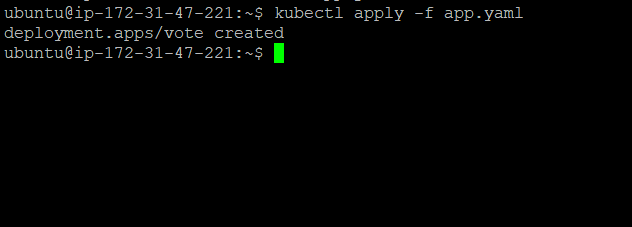
**11.Now created deployment file for the deploy the app**

**as below:**

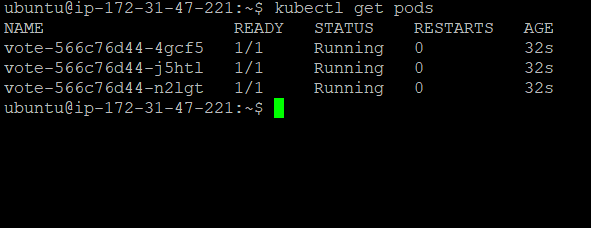


**12.And using the below command applied the changes**

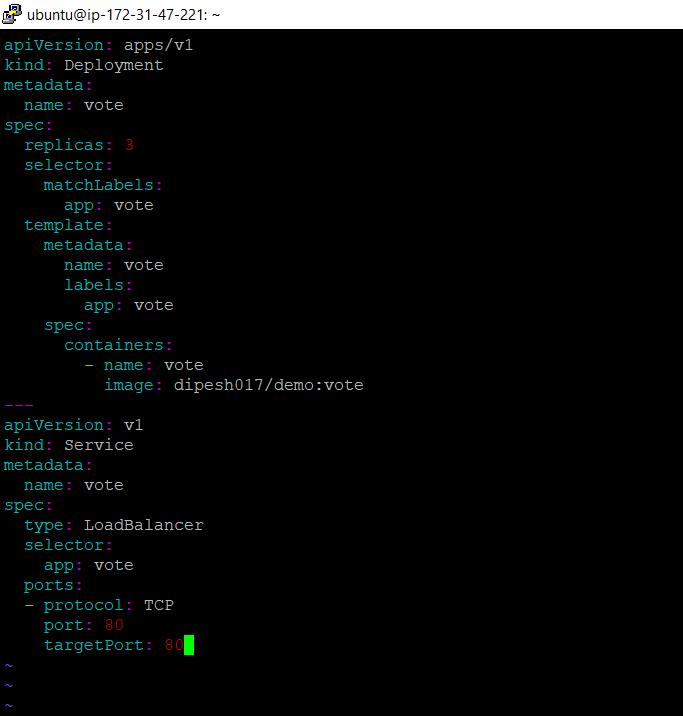
= Kubectl apply -f app.yaml



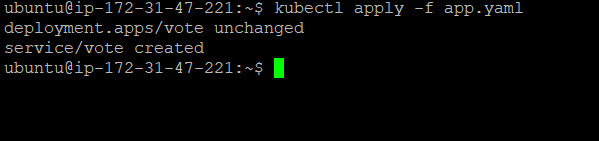
**12.As we need three pods/replicas that is in running state as below ss:**



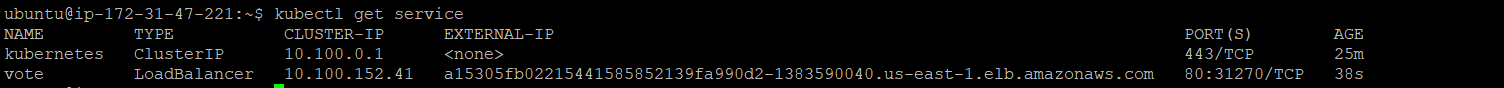
**13. for accessing app in browser and balancing load among pods we require service components file so created as below**

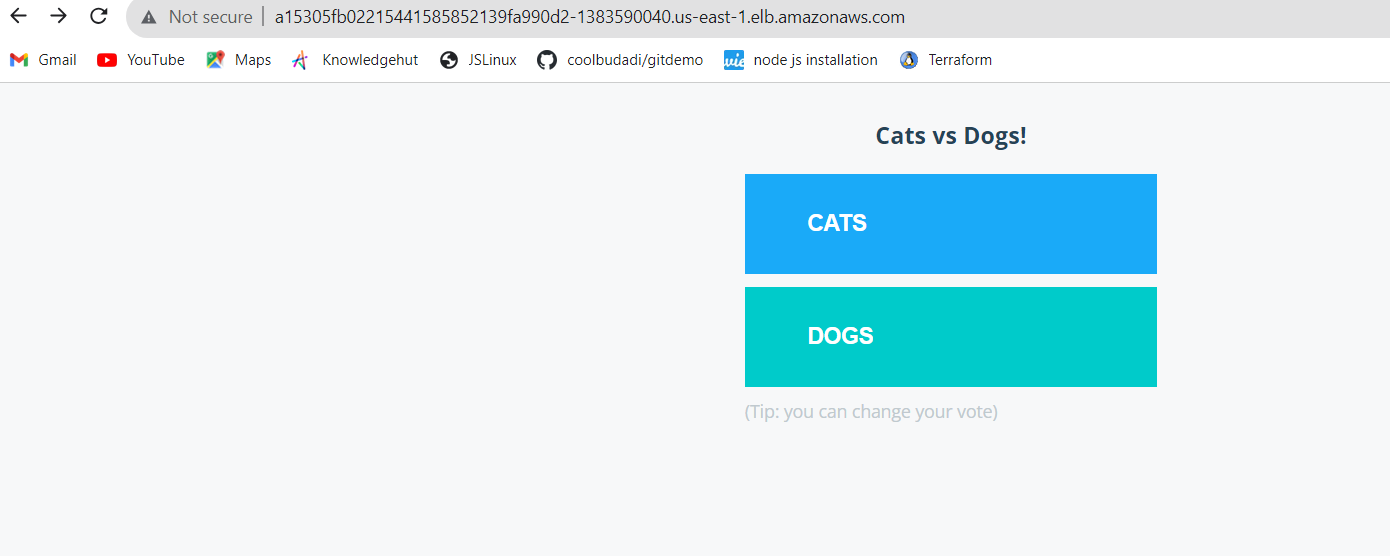


**14.And using the below command applied the changes**

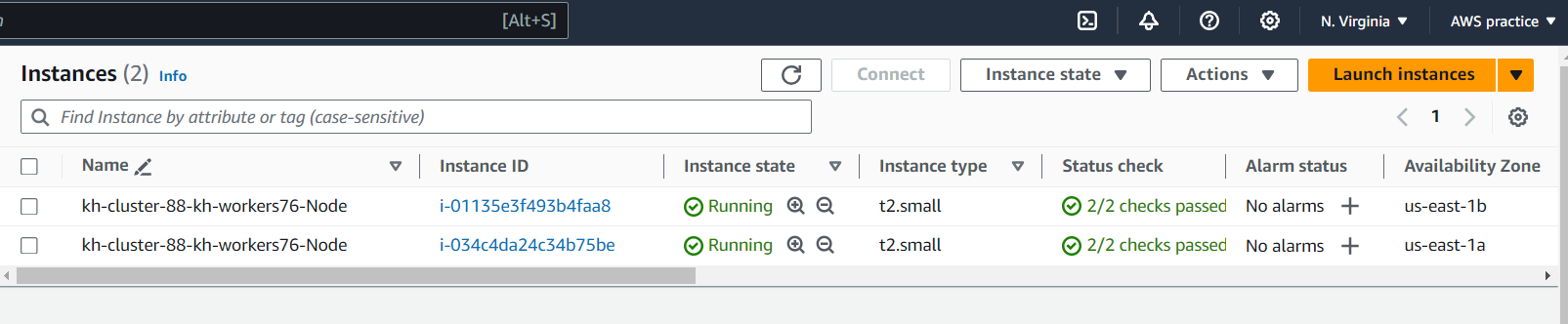


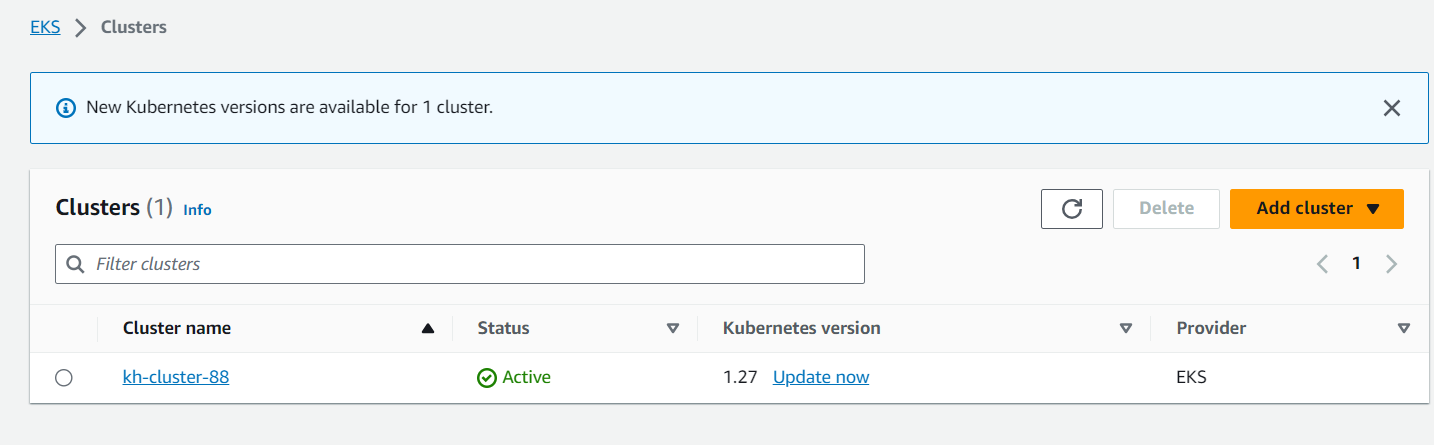
**15.And we got the url-External ip for viewing the website**





**12.And when the cluster got successfully launched and we got the two machines and EKS cluster as below:**

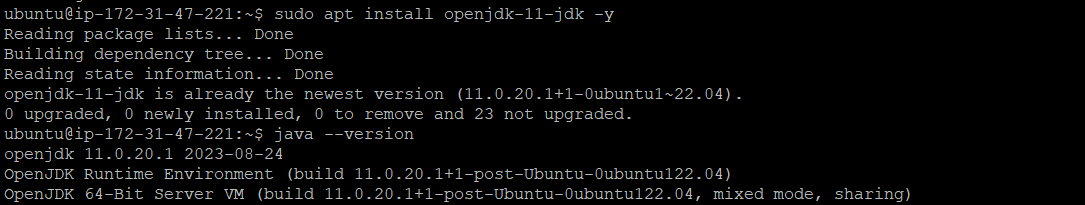


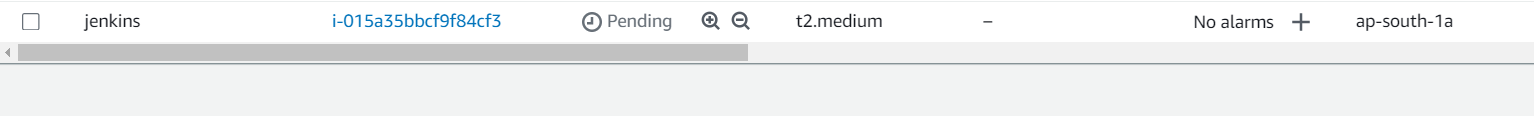


**Now we have launched and deployed Eks cluster now we need**

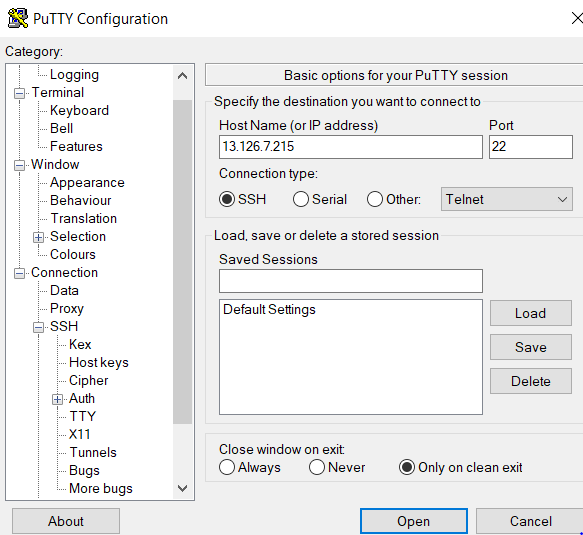
**Integrate CICD On it so we need to install Jenkins and Java at**

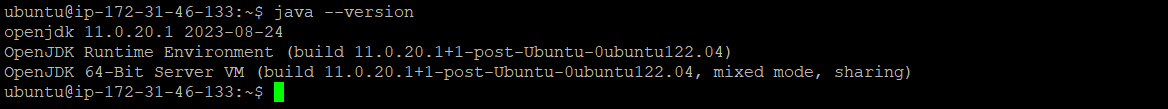
**One machine and on to the second main machine we need to install only Java**



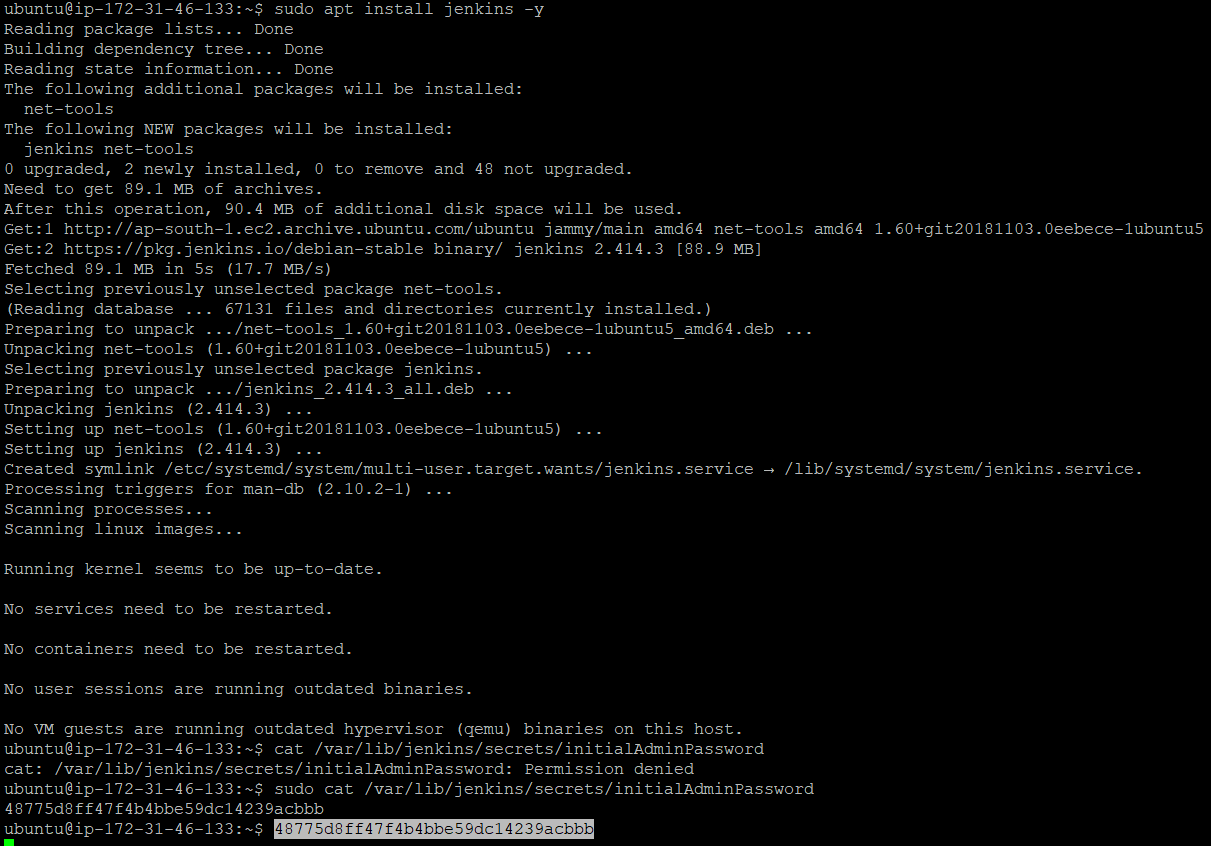


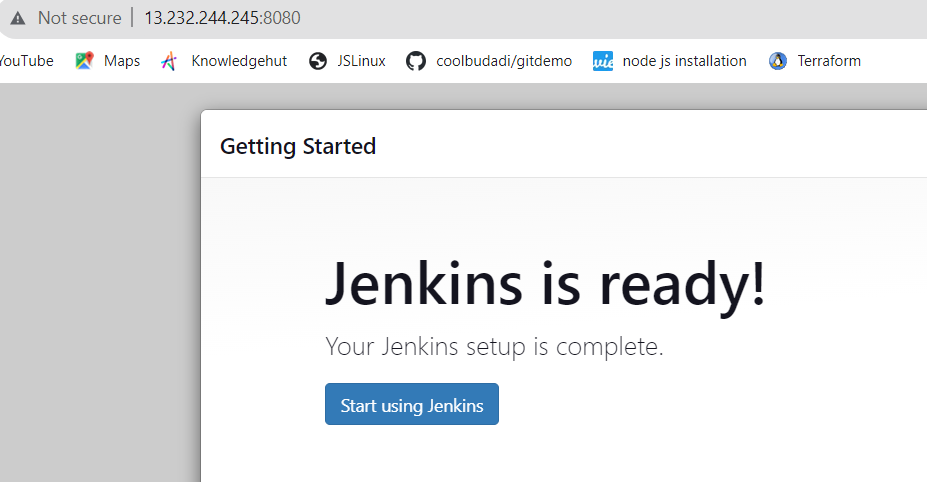
**Now need to SSH into Jenkins machine and install Jenkins,Java on it as below:**



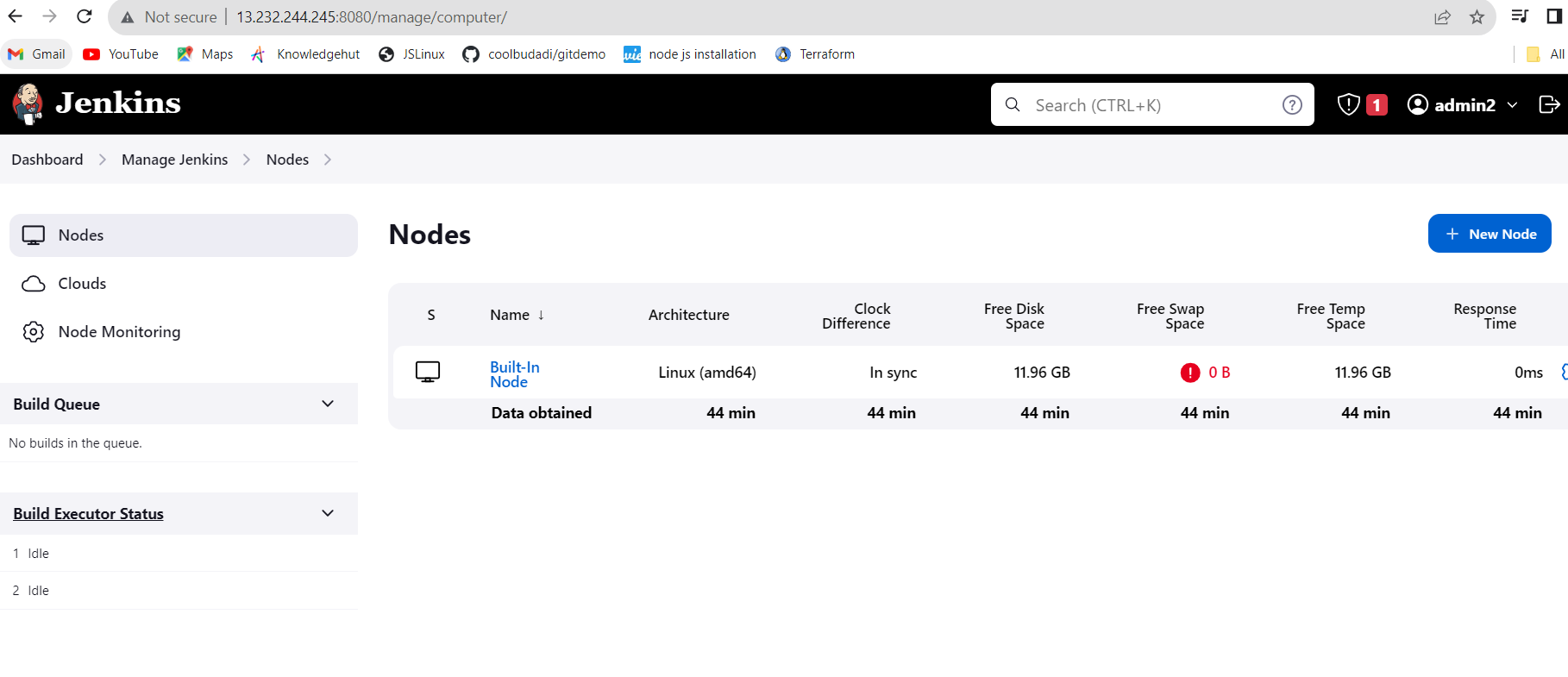


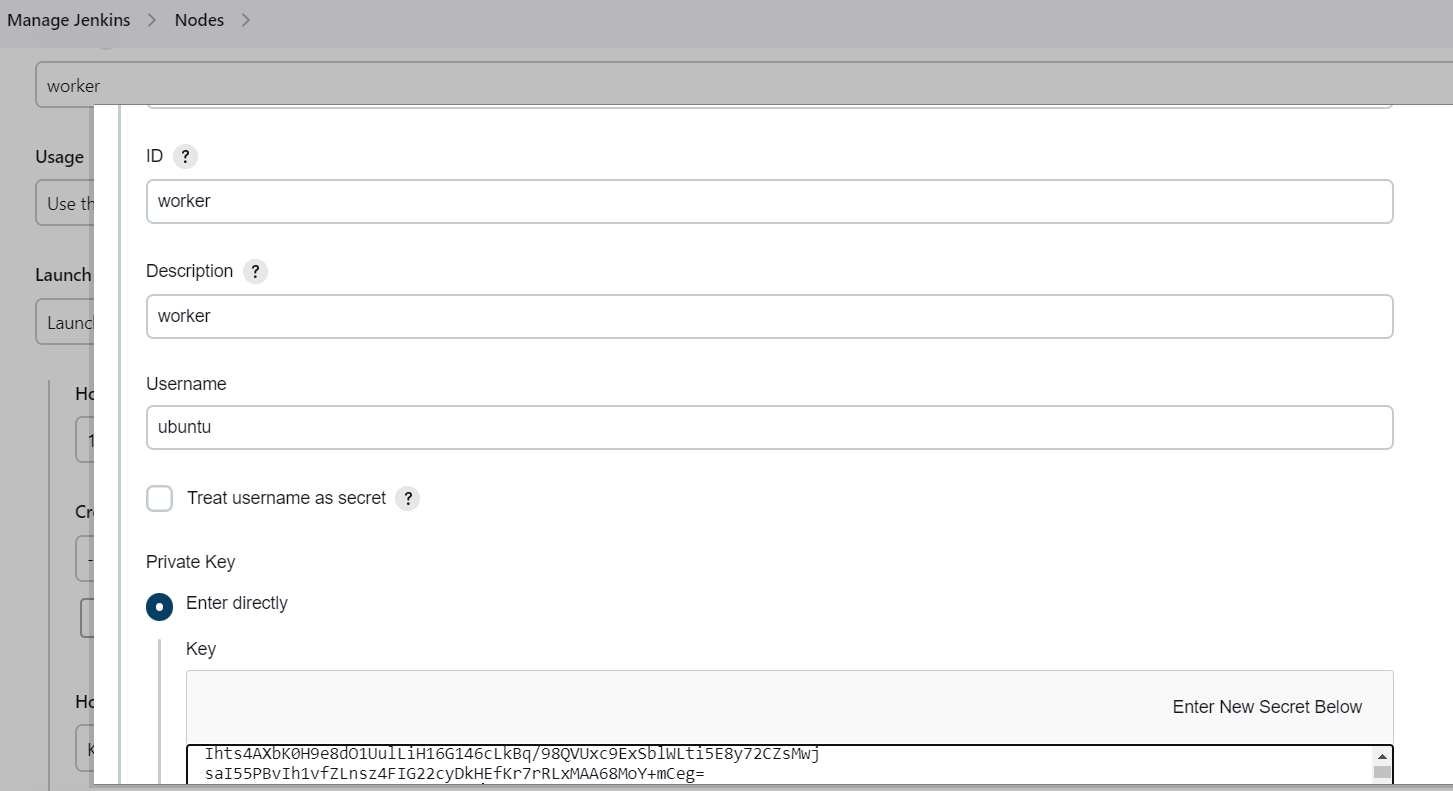
**Installation of Jenkins on second machine**





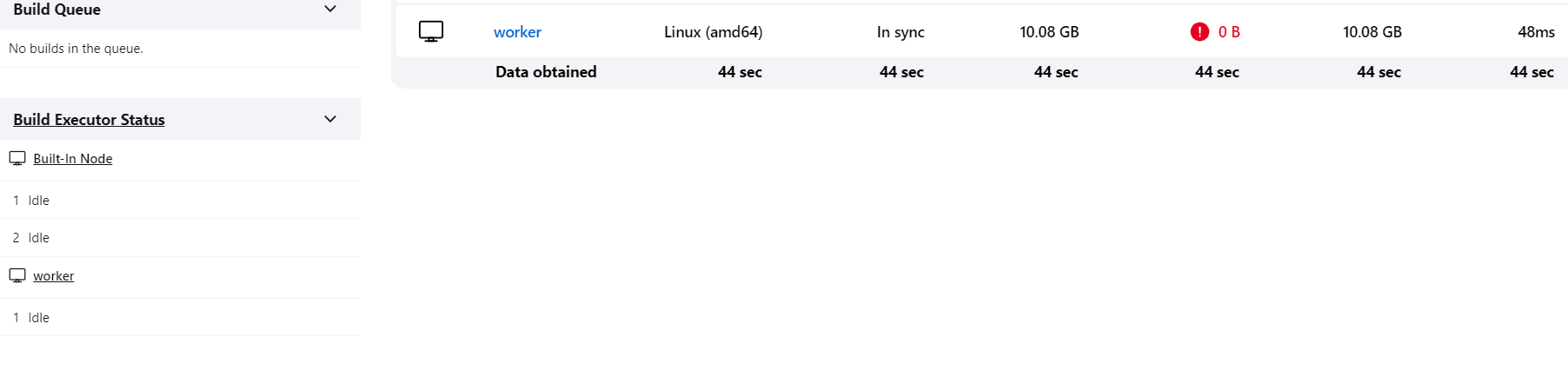
**For integration with Jenkins and application we need to follow below process:**

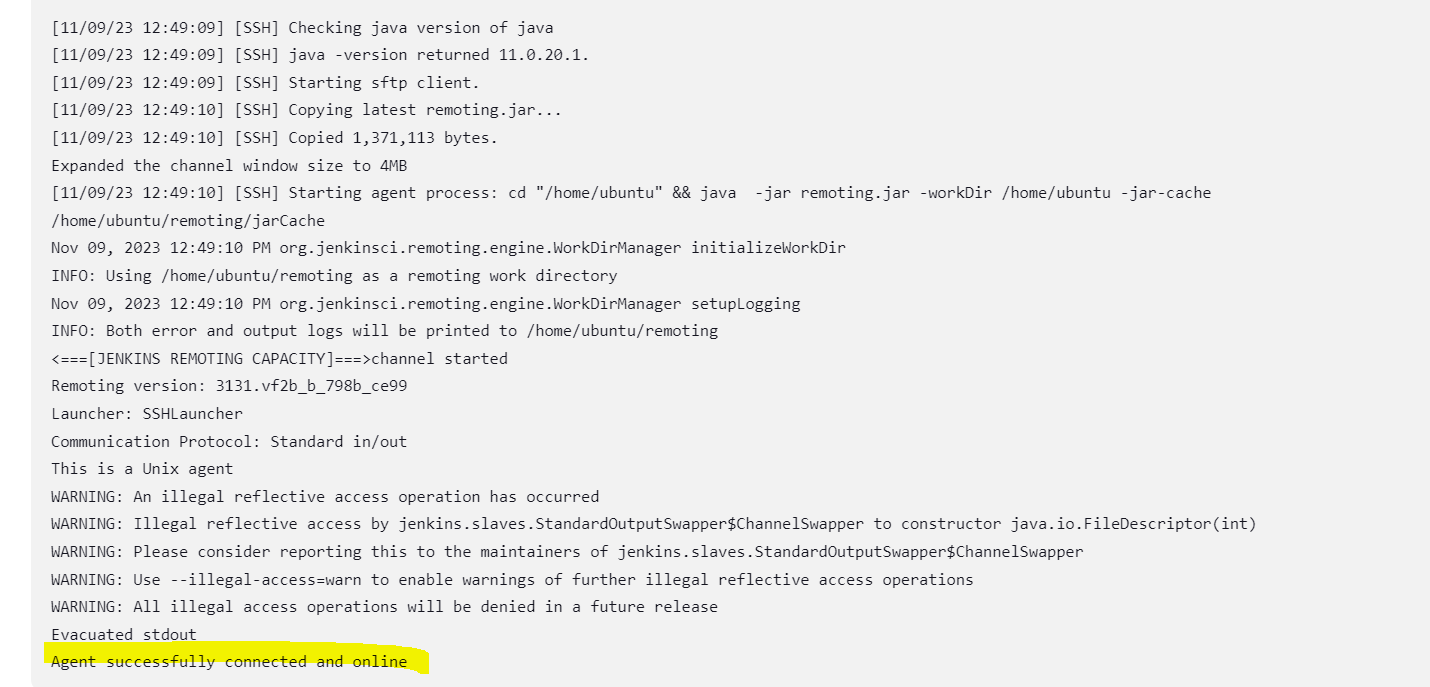


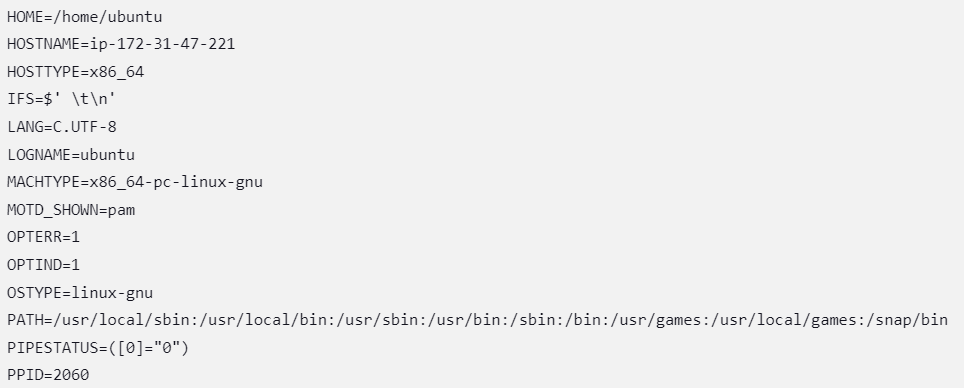


**Manage Jenkins > New node > save it and build it .**

**Output we got result worker node is successfully connected as below:**





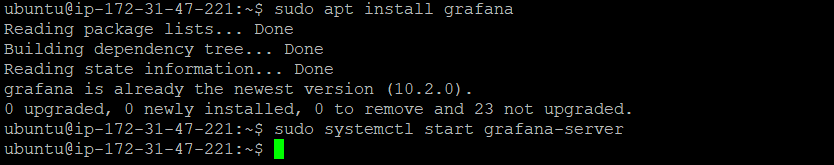


Worker Node : <http://13.232.244.245:8080/computer/worker/log>

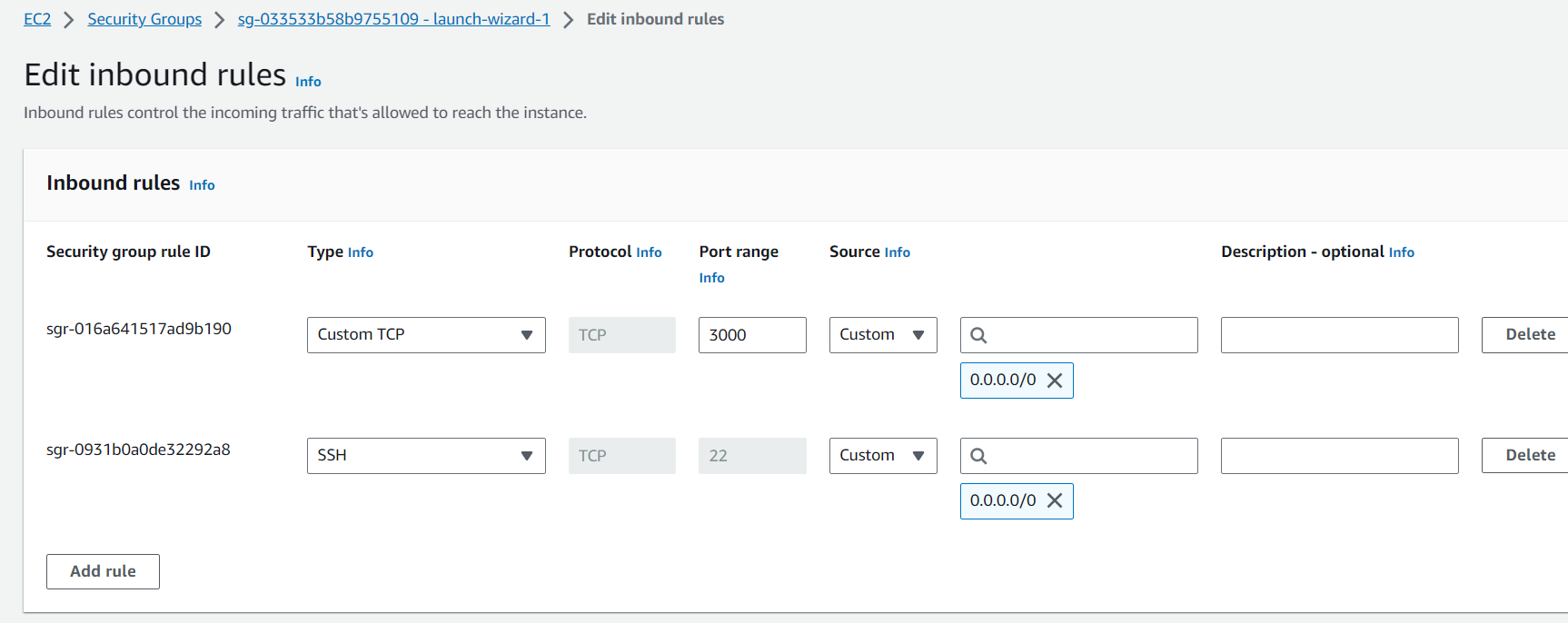
<http://13.232.244.245:8080/computer/worker/builds>

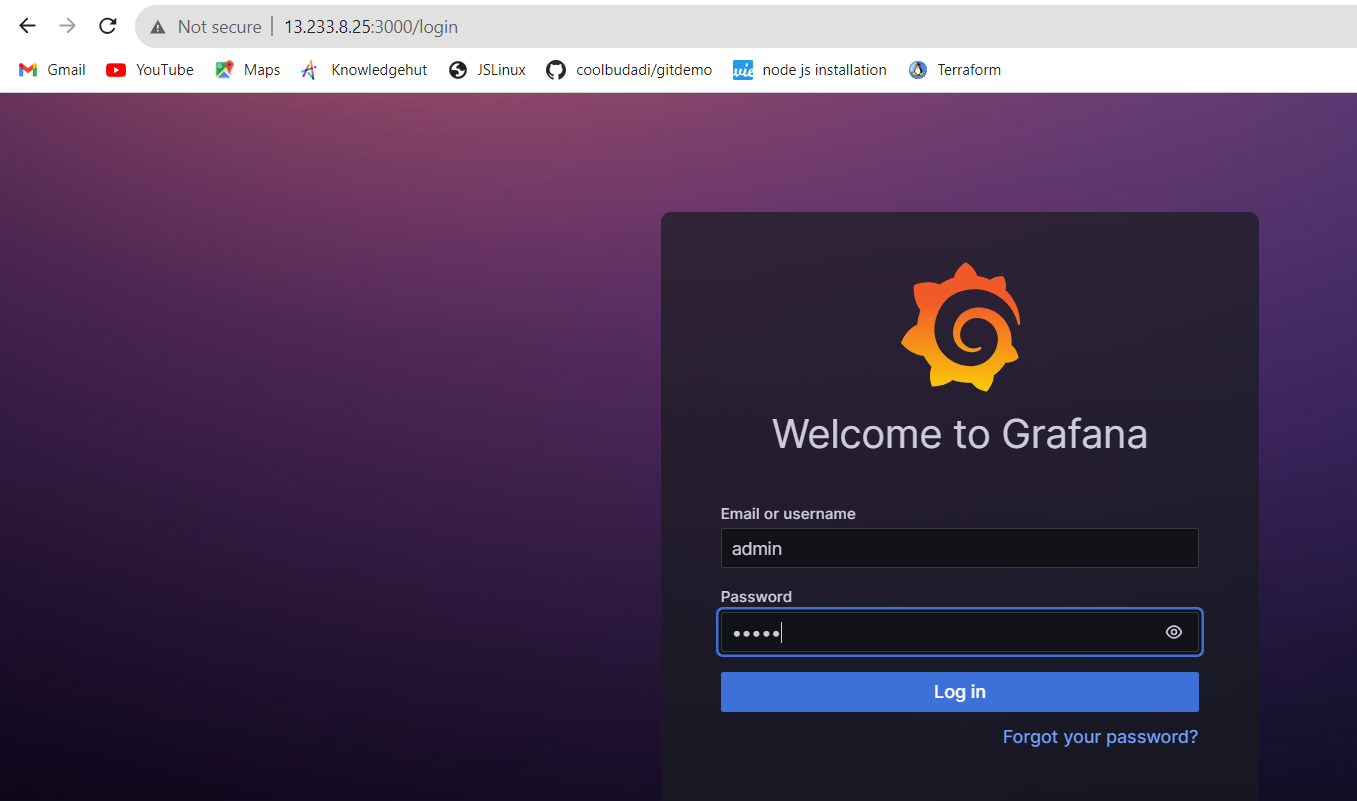
**Monitoring(Grafana + Cloudwatch)**

**Installation of Grafana to the main machine.**



**Added 3000 port entry to SG as below:**





**Created Dashboard for monitoring alerts.**

