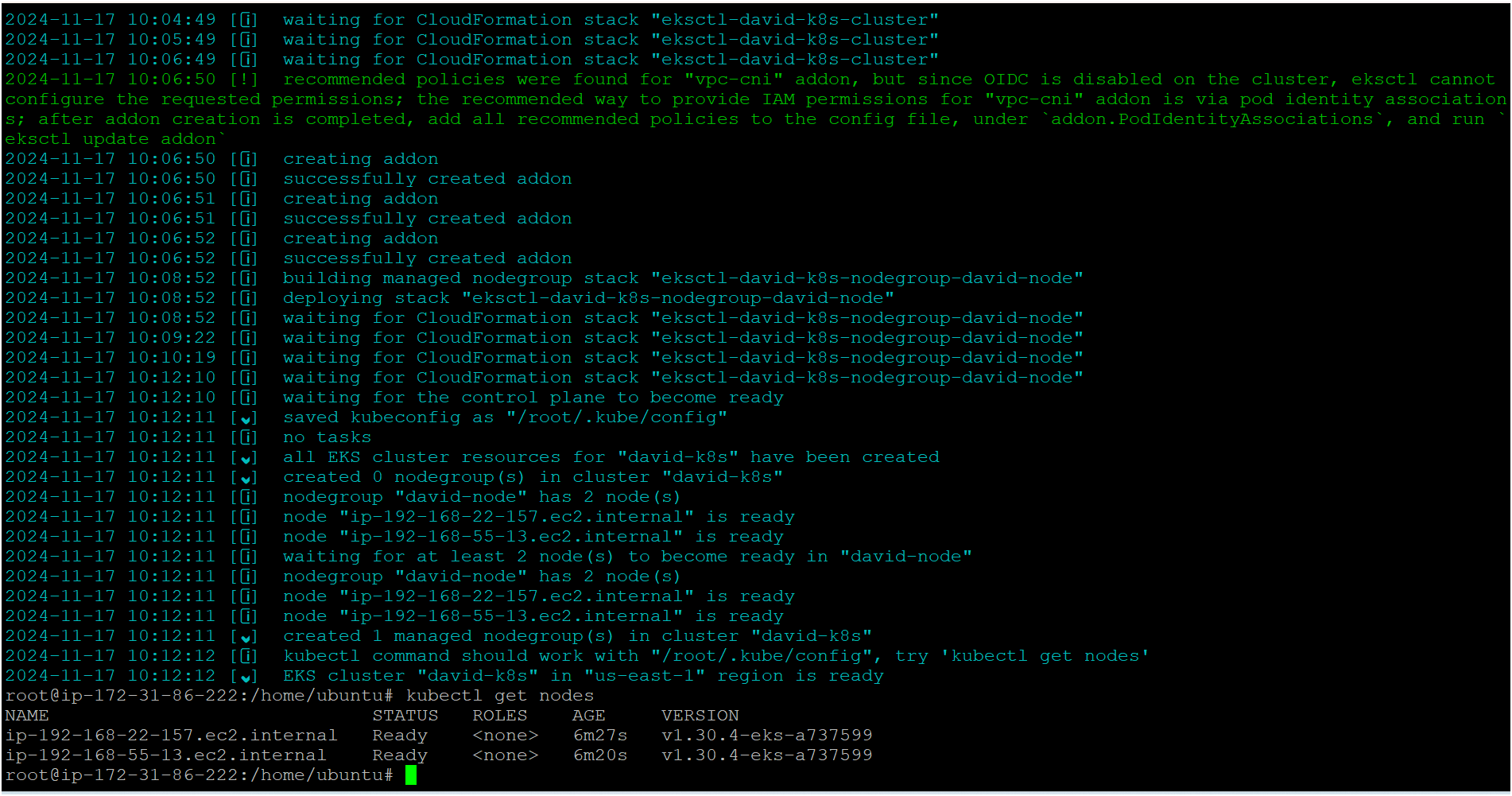
# **How to connect eks cluster to Jenkins**

**Eks cluster**

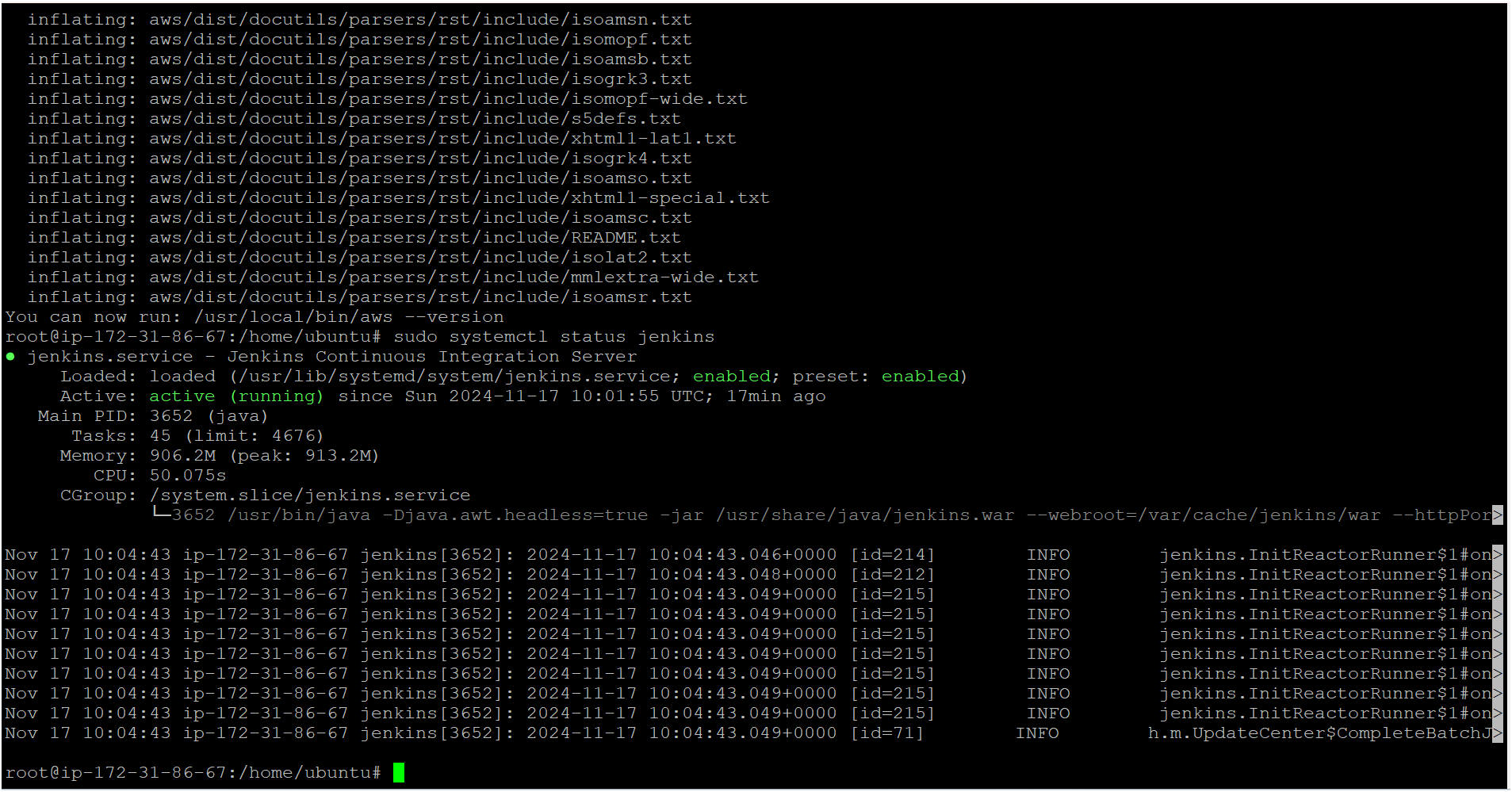
* Firstly, we have to set up k8s cluster as shown below image 👇

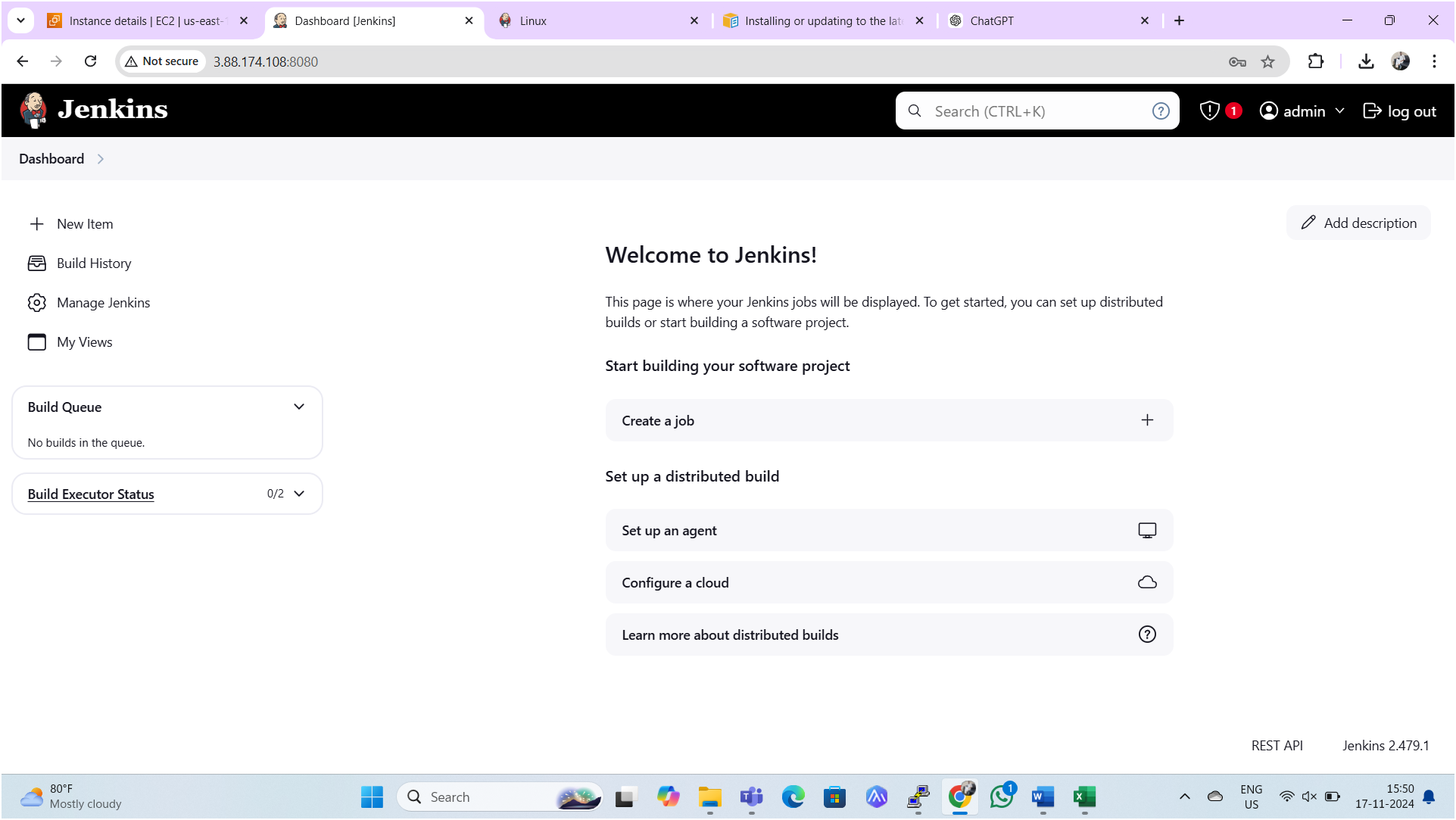


* Now my cluster is ready as shown above image

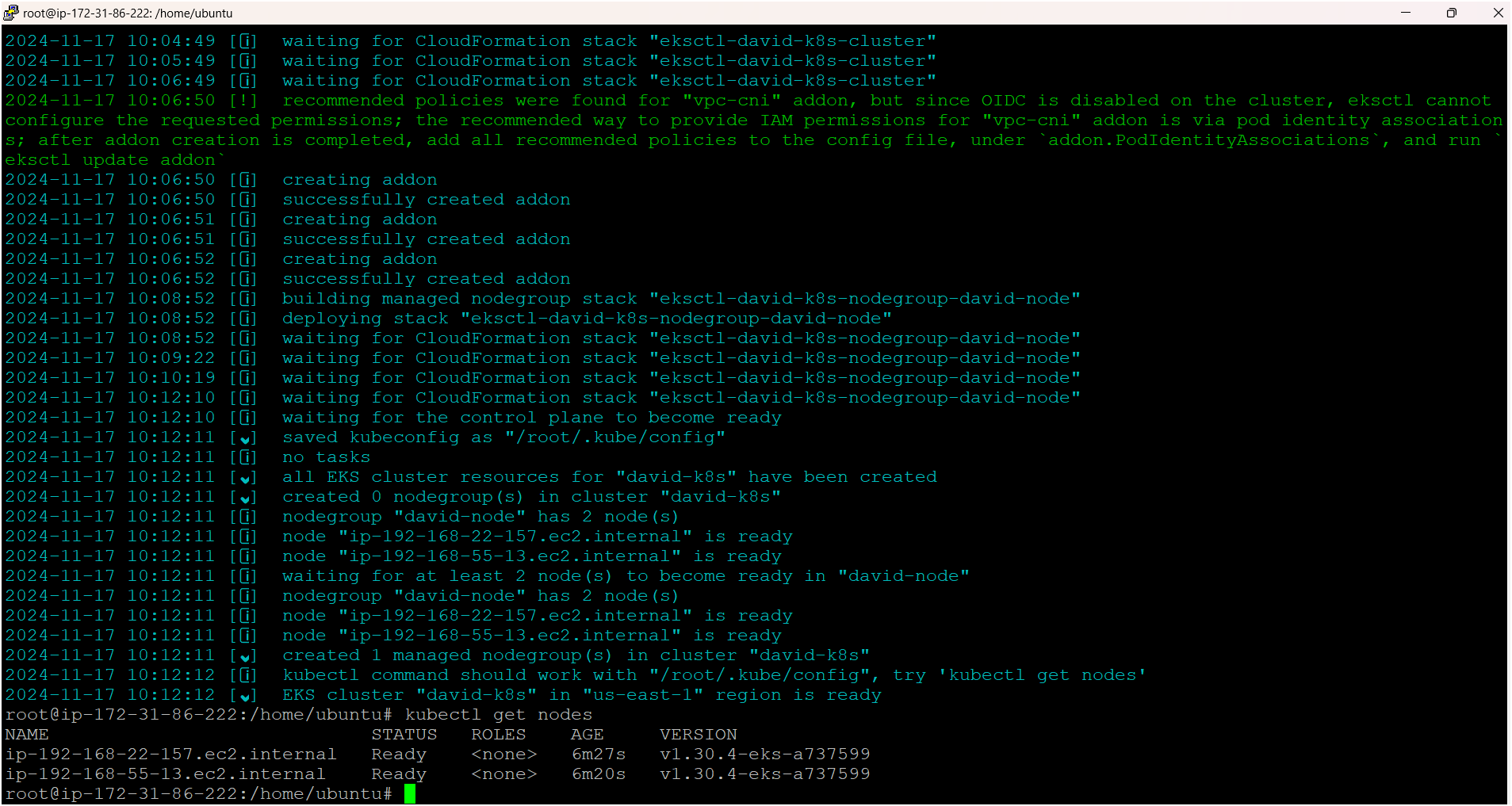
**Jenkins**

* Now we have to set up Jenkins as shown below image 👇
* Please make sure you have to install kubectl and aws cli on Jenkins server

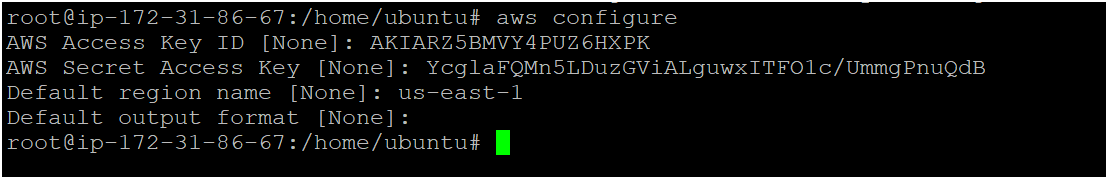




* Next, we need to copy Kube config file into the Jenkins server from k8s master node as shown below image 👇

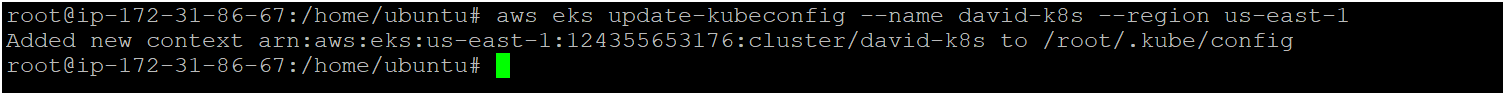


* Firstly, we have to provide aws access keys on Jenkins server



* Execute following command on Jenkins server

aws eks update-kubeconfig --name <cluster-name> --region <region>



* As seen above image Kube-config file has copied to Jenkins serve from master node i.e. cluster.
* Now execute following commands in Jenkins server one by one

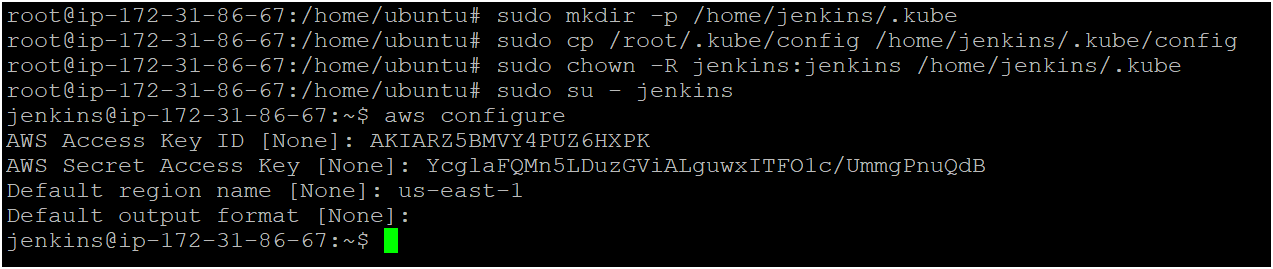
sudo mkdir -p /home/jenkins/.kube

sudo cp /root/.kube/config /home/jenkins/.kube/config

sudo chown -R jenkins:jenkins /home/jenkins/.kube

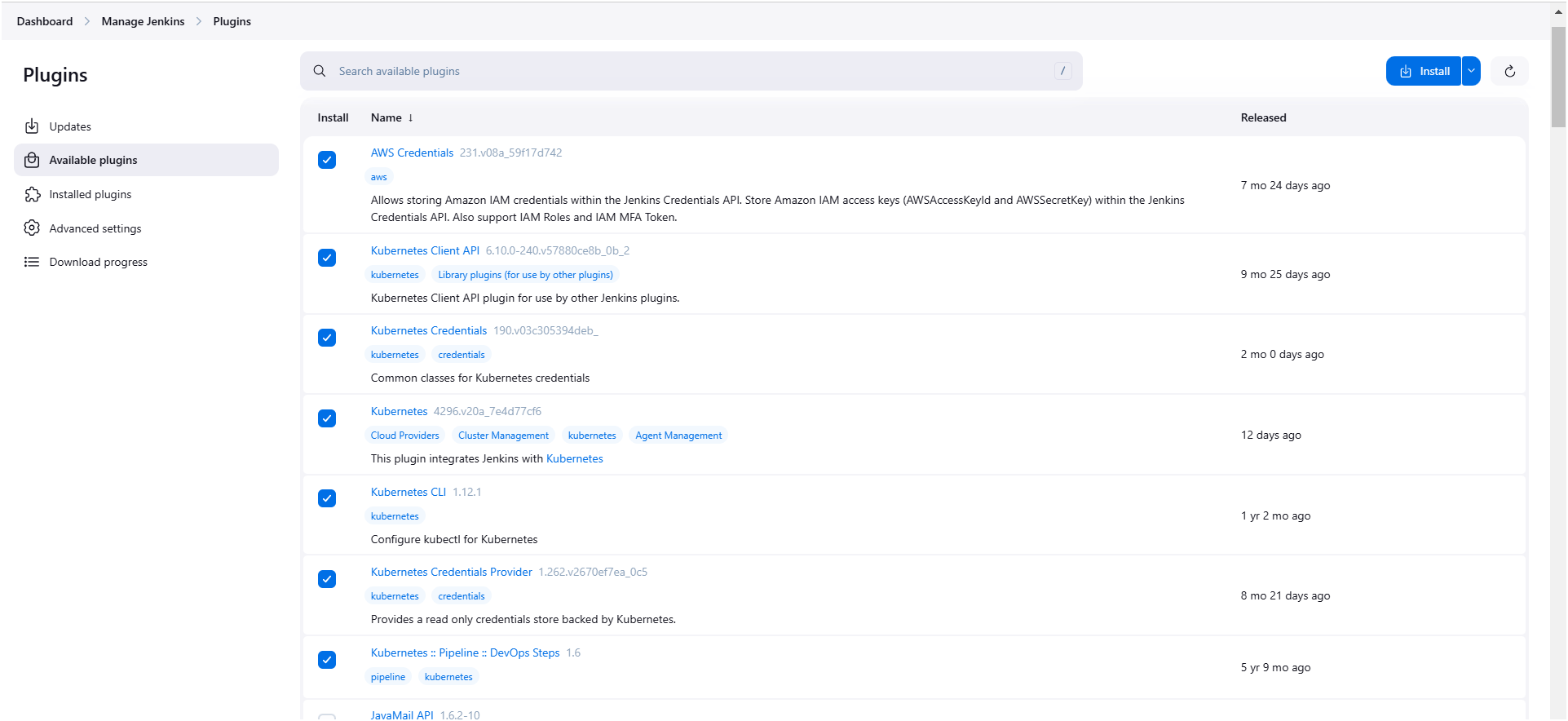
sudo su - jenkins

aws configure (provide access keys on Jenkins server)



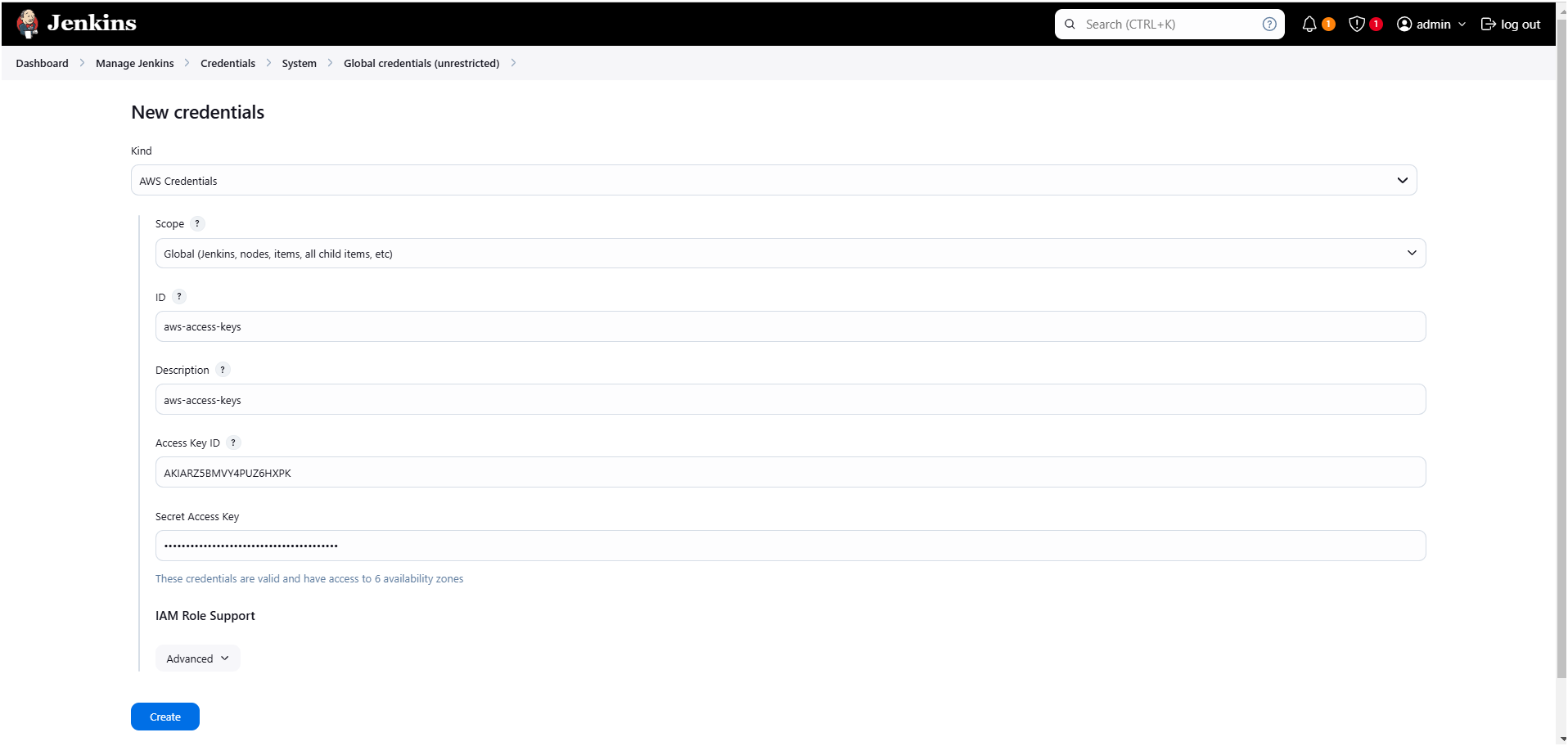
* Install following plugins on Jenkins

1. [AWS Credentials](https://plugins.jenkins.io/aws-credentials)
2. [Kubernetes Client API](https://plugins.jenkins.io/kubernetes-client-api)
3. Kubernetes CredentialsVersion
4. [Kubernetes](https://plugins.jenkins.io/kubernetes)
5. [Kubernetes CLI](https://plugins.jenkins.io/kubernetes-cli)
6. [Kubernetes Credentials Provider](https://plugins.jenkins.io/kubernetes-credentials-provider)
7. [Kubernetes :: Pipeline :: DevOps Steps](https://plugins.jenkins.io/kubernetes-pipeline-devops-steps)



* We have to provide aws access in Jenkins in credentials

Go to manage Jenkins 🡪 go to credentials 🡪 system 🡪 global credentials



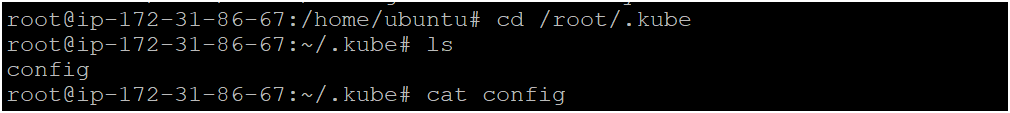
* Now we have to provide Kube-config file in Jenkins web server
* Firstly, we need to copy and save it in a document or file in your local as shown below image
* Execute below two commands on Jenkins server for copying config file

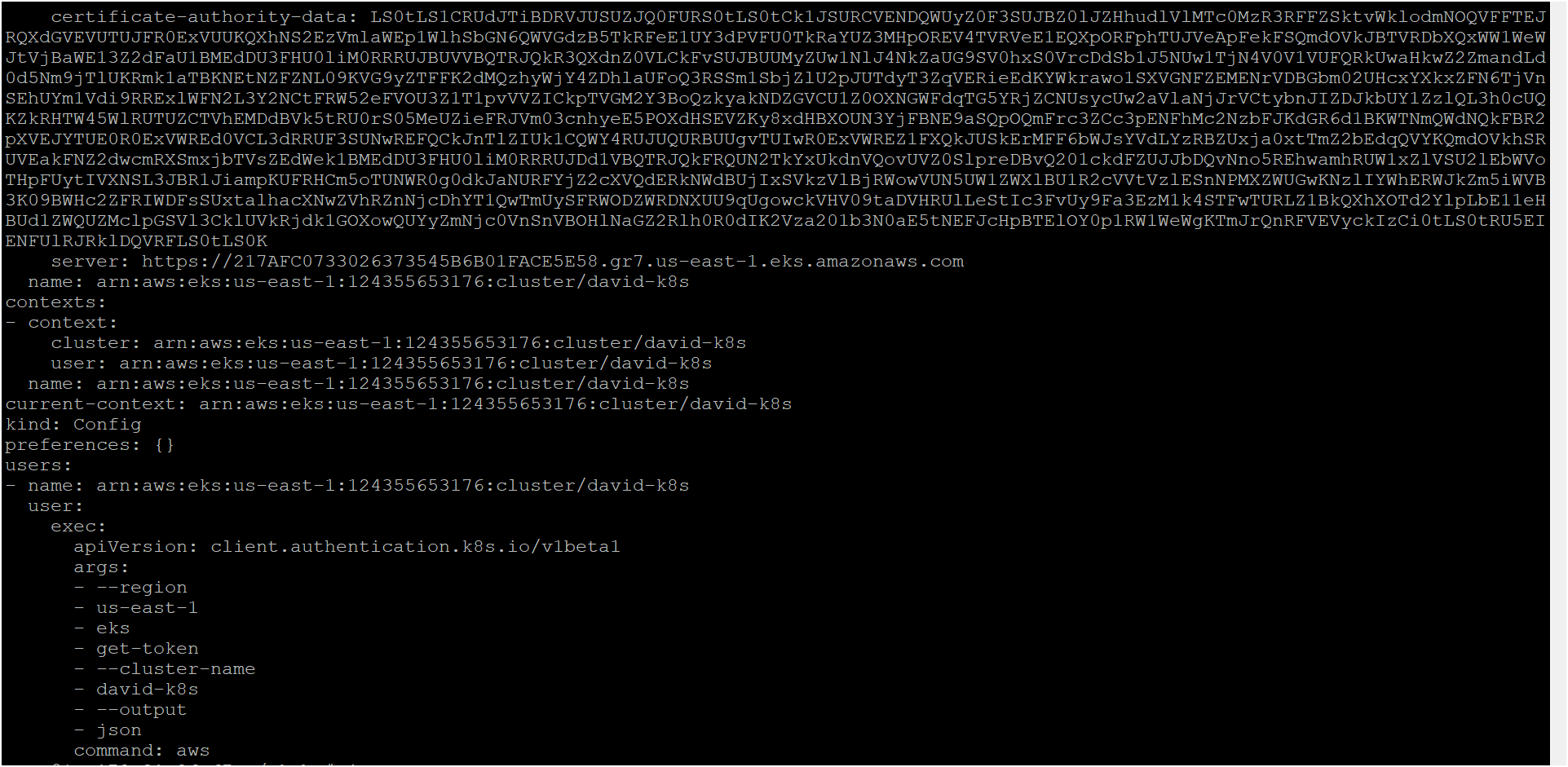
cd /root/.kube

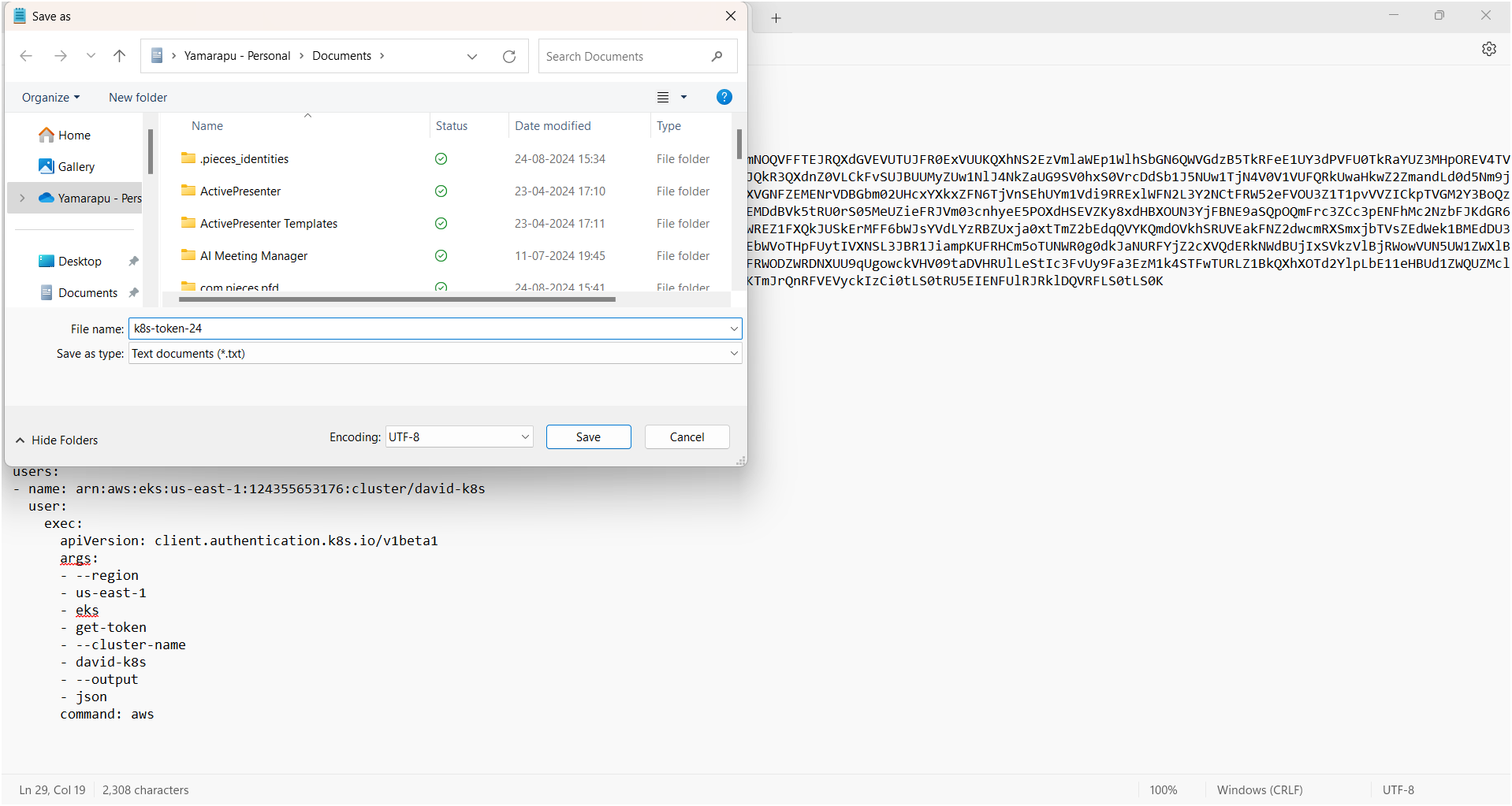
ls

cat config (now copy the entire token and save it in a document or file)

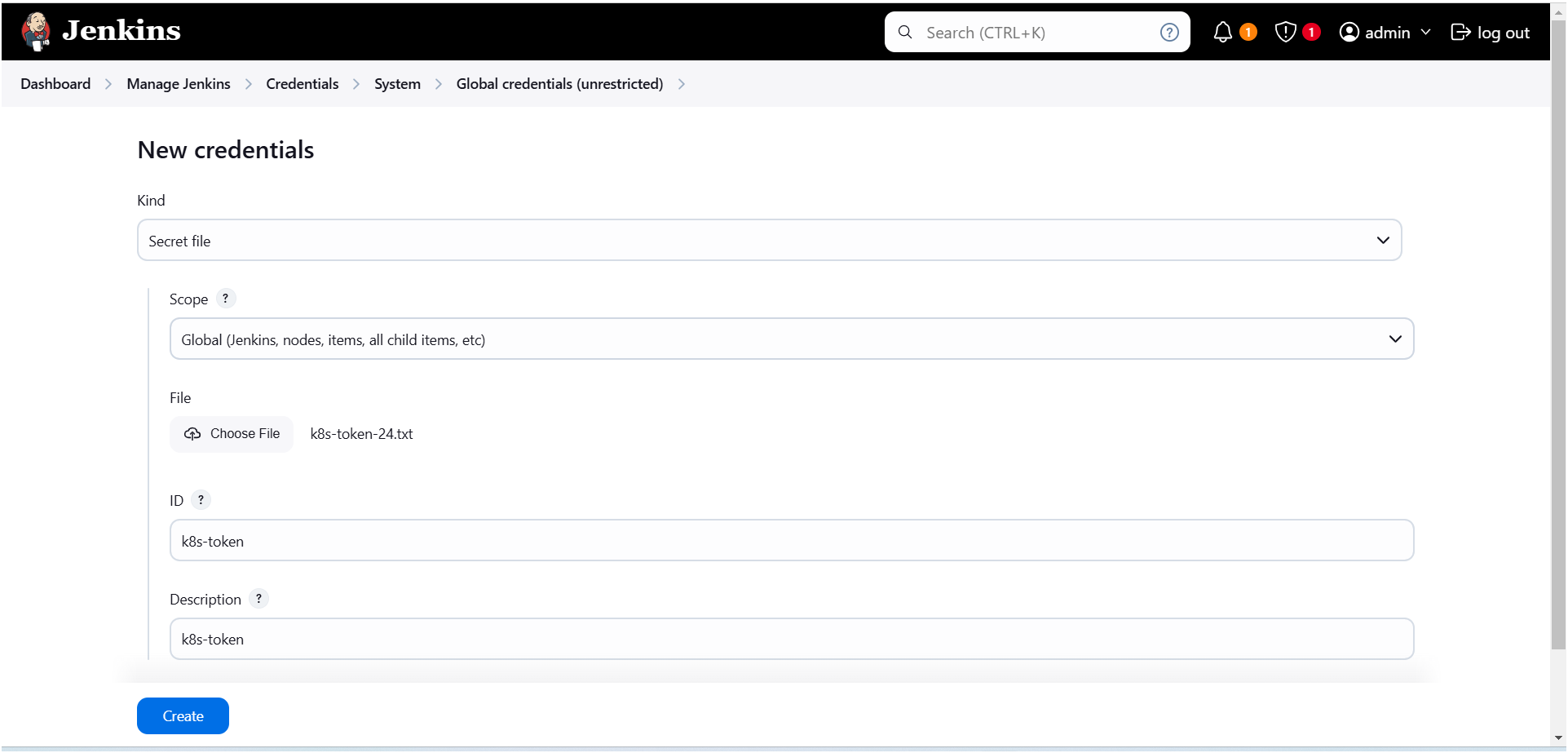
* Please make sure you have to copy this config file from Jenkins server only not from cluster.



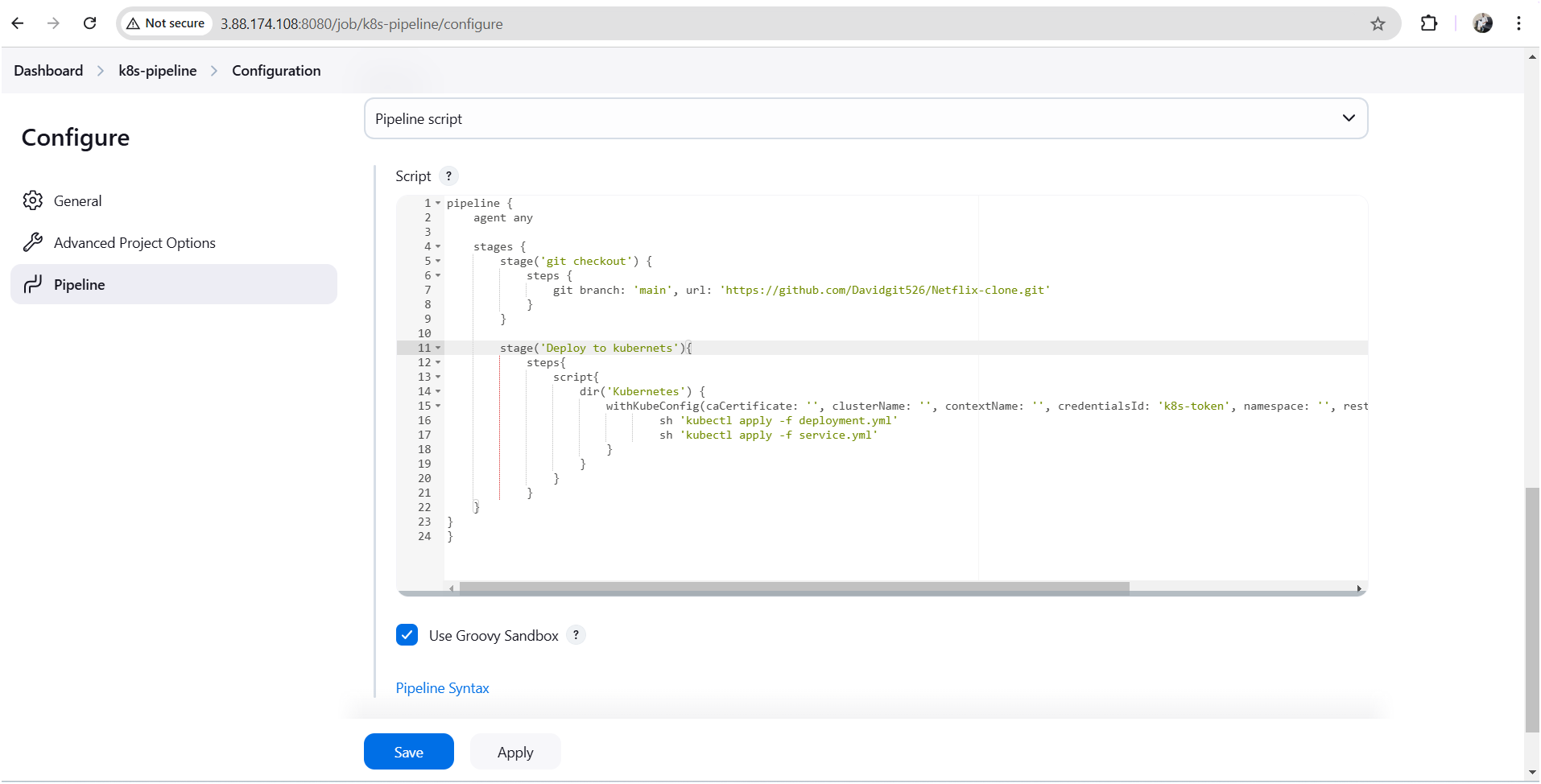




* Now we have to provide this file in Jenkins web server in credentials as follows



* Now create a pipeline and check whether eks cluster integrated with Jenkins or not



**Pipeline Script:**

pipeline {

agent any

stages {

stage('git checkout') {

steps {

git branch: 'main', url: 'https://github.com/Msocial123/Netflix-clone.git'

}

}

stage('Docker build & push') {

steps {

script {

withDockerRegistry(credentialsId: 'docker-cred') {

sh 'docker build -t netflix-2025 .'

sh 'docker tag netflix-2025:latest muralisocial123/netflix-2025:latest'

sh 'docker push muralisocial123/netflix-2025:latest'

}

}

}

}

stage('deploy to container'){

steps {

sh 'docker run -d --name netflix-1 -p 81:80 netflix-2025'

}

}

stage('deploy to kubernetes') {

steps {

script {

dir('Kubernetes') {

withKubeConfig(caCertificate: '', clusterName: '', contextName: '', credentialsId: 'k8s-config', namespace: '', restrictKubeConfigAccess: false, serverUrl: '') {

sh 'kubectl apply -f deployment.yml'

sh 'kubectl apply -f service.yml'

}

}

}

}

}

}

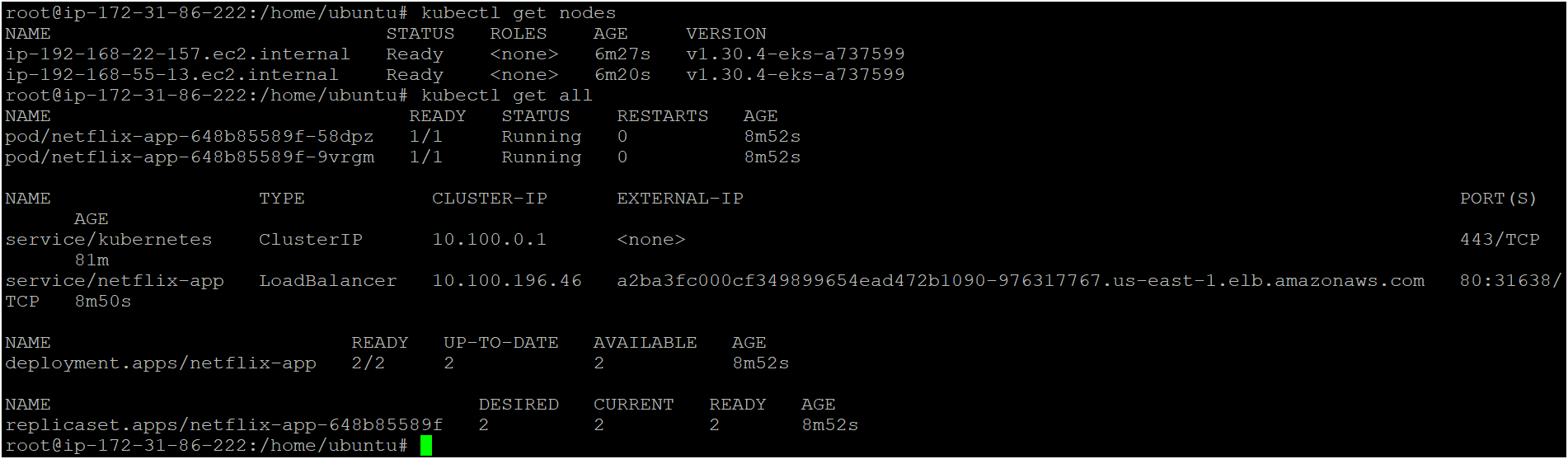
}

If you get docker login failure or permission denied error provide the the following commands on your Jenkins server

* sudo usermod -aG docker Jenkins
* sudo systemctl restart Jenkins
* chmod 777 /var/run/docker.sock



* My Netflix application successfully has deployed on eks cluster



* Am able to access my application

