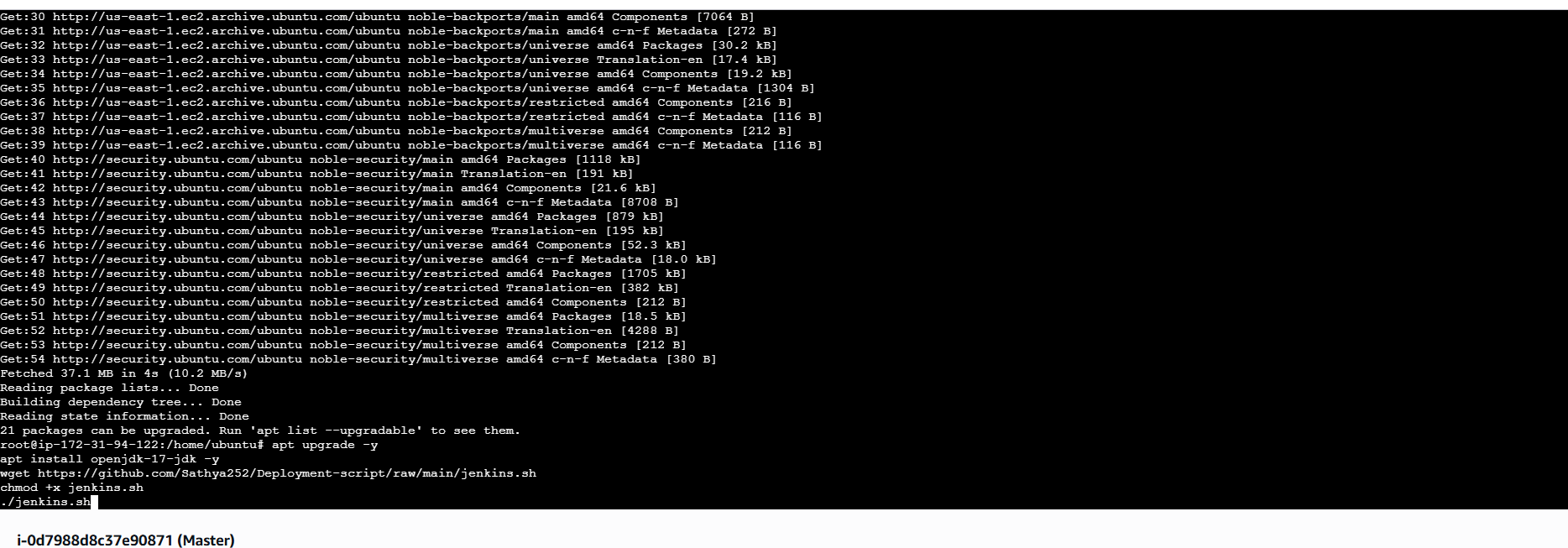


sudo su

apt update



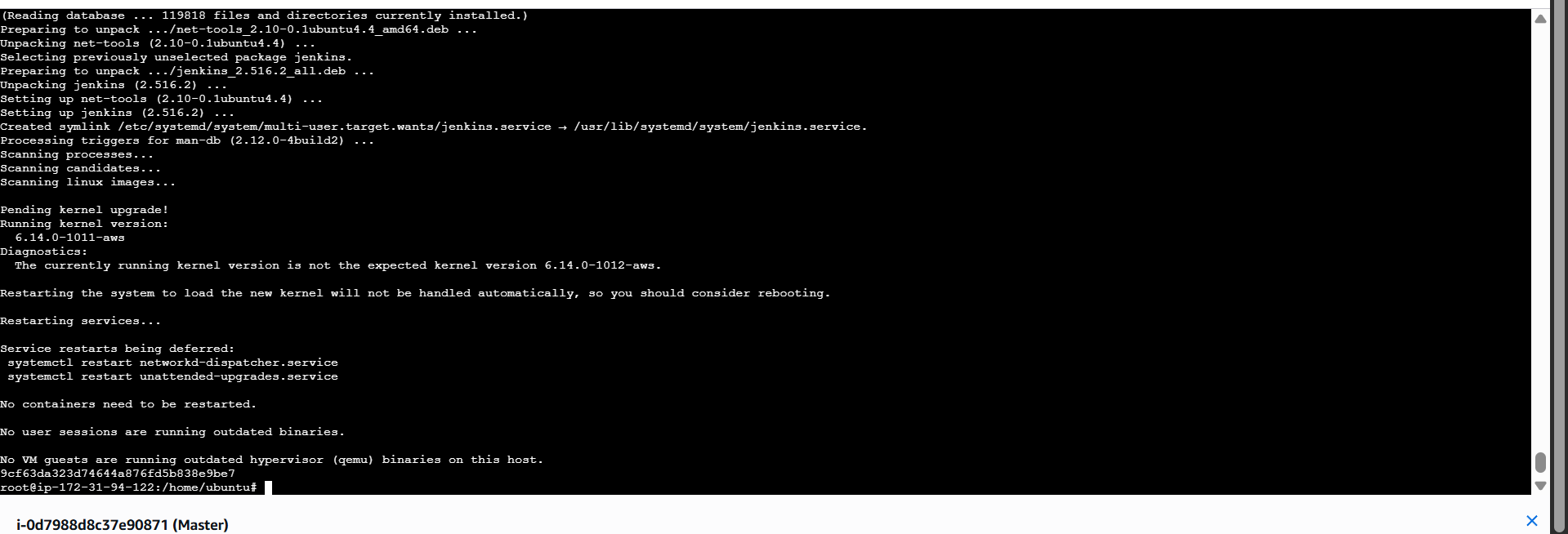
apt upgrade -y

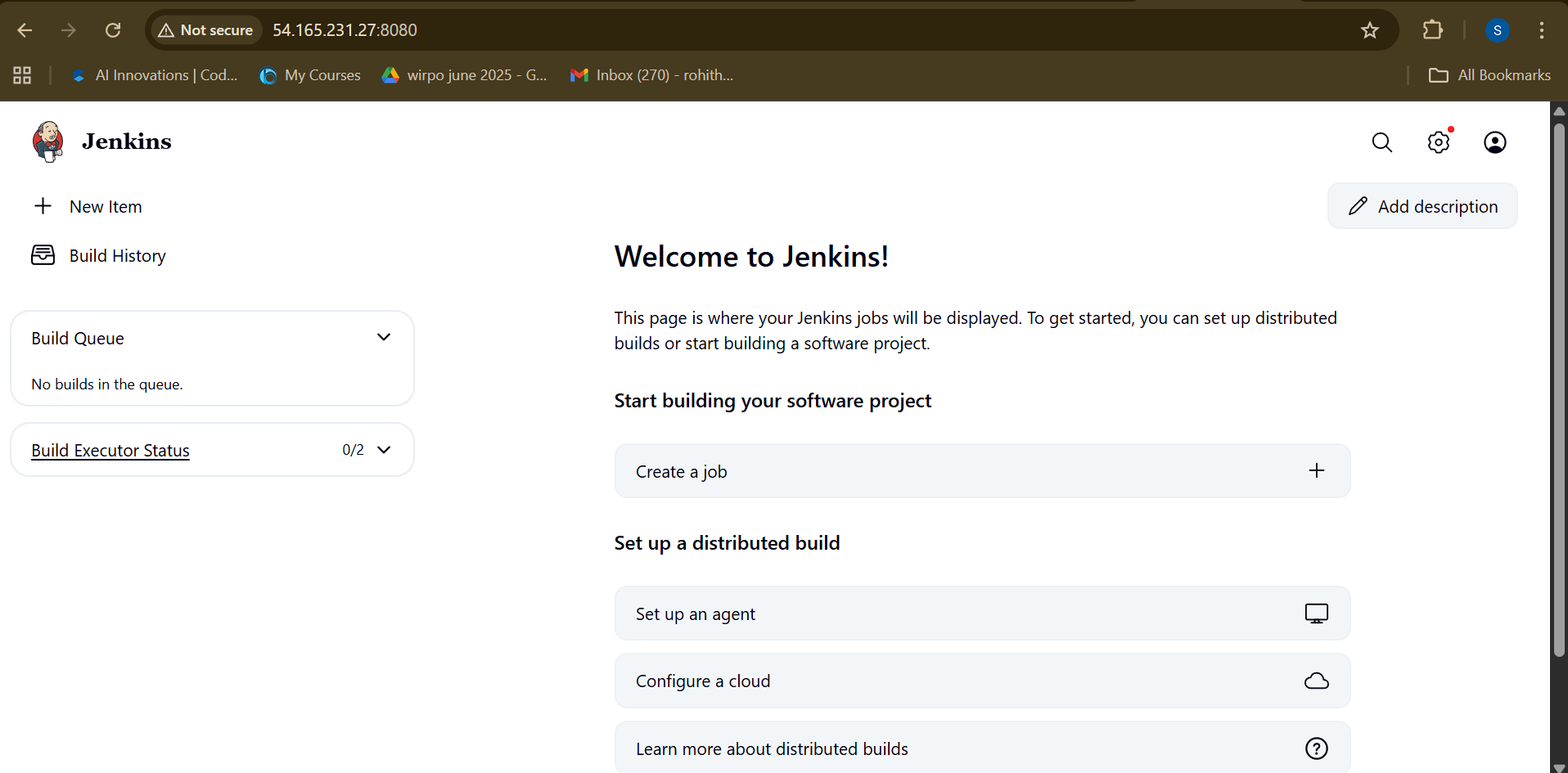
apt install openjdk-17-jdk -y

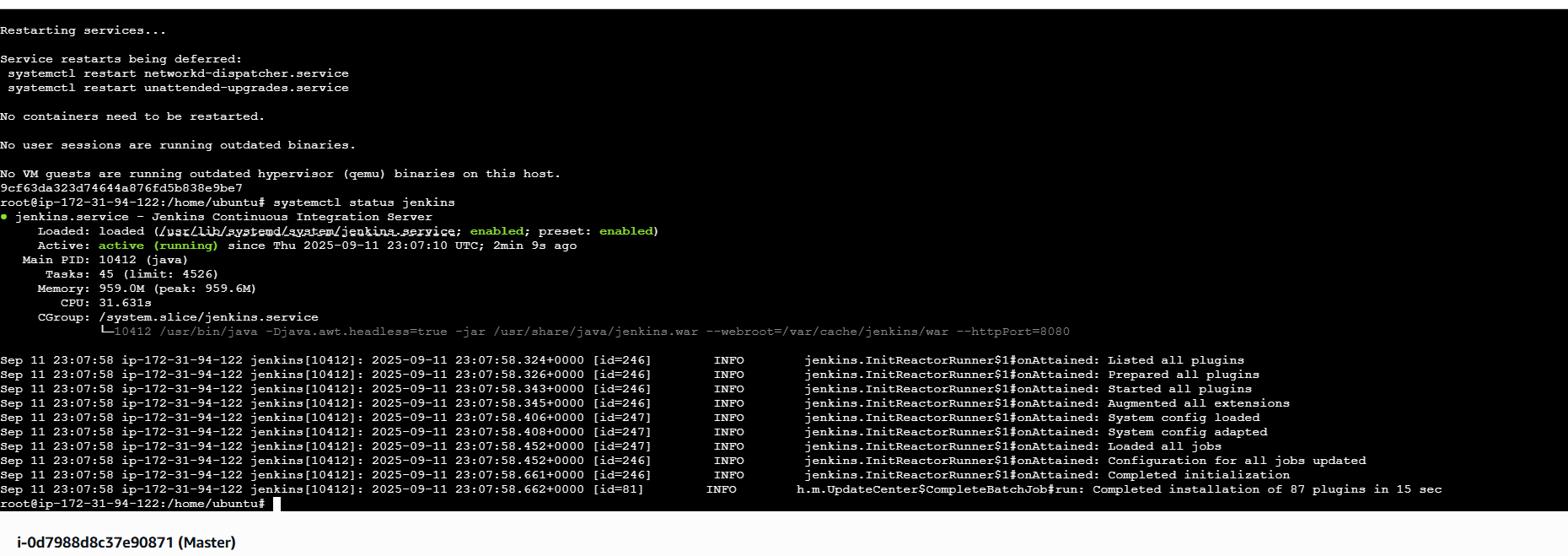
wget https://github.com/Sathya252/Deployment-script/raw/main/jenkins.sh

chmod +x jenkins.sh

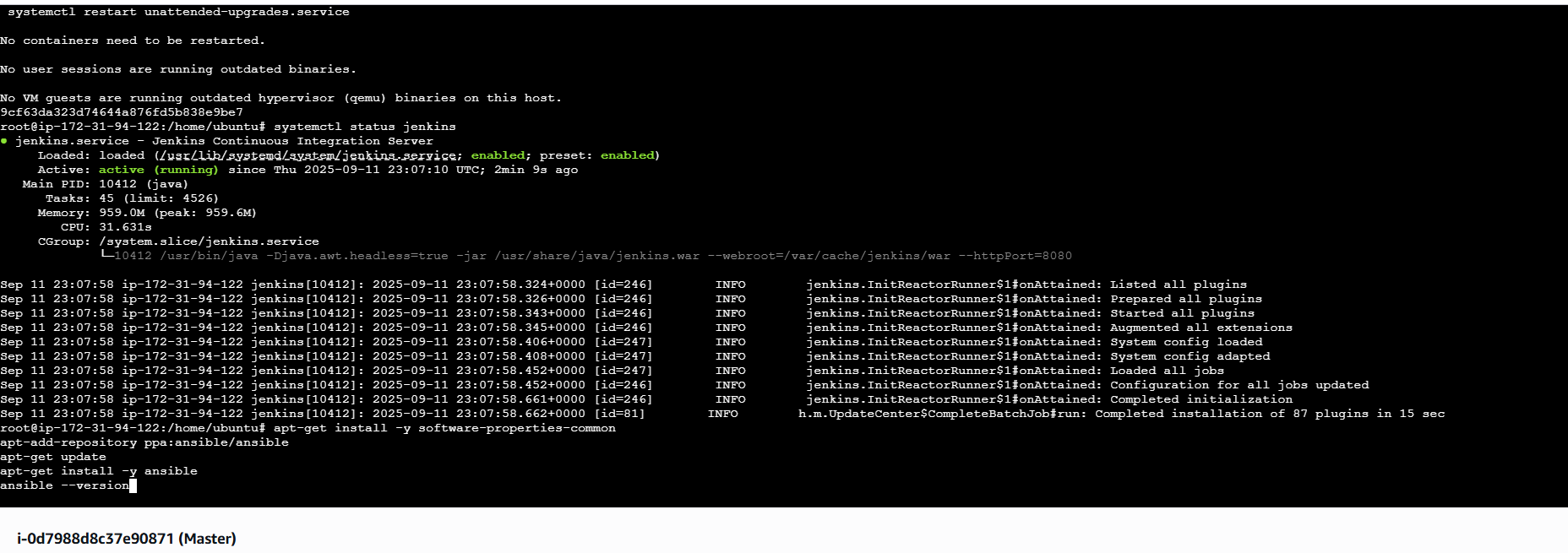
./[jenkins.sh](http://jenkins.sh)







systemctl status jenkins



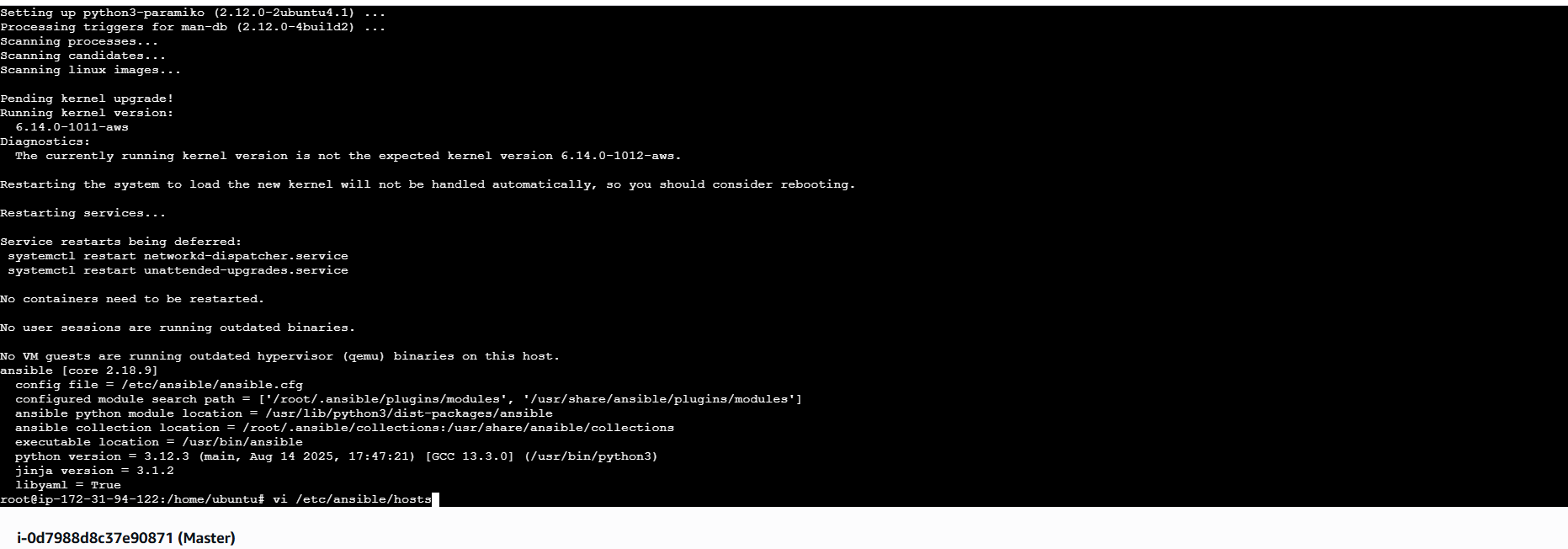
apt-get install -y software-properties-common

apt-add-repository ppa:ansible/ansible

apt-get update

apt-get install -y ansible

ansible --version



vi /etc/ansible/hosts





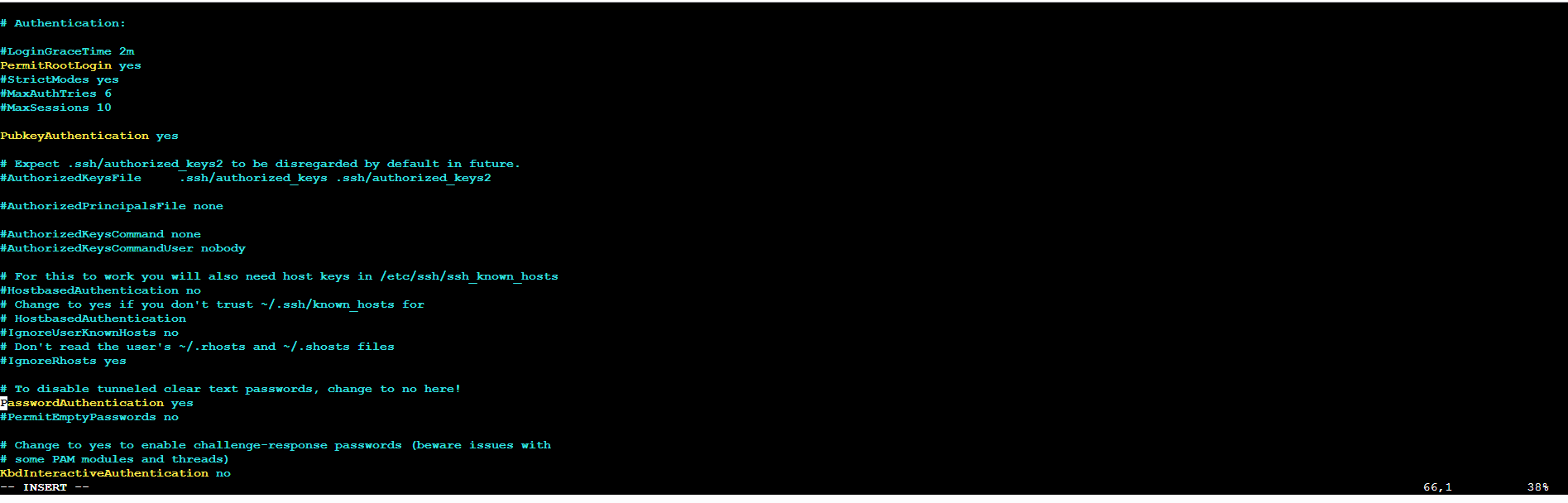
adduser devops

(put password as devops)

And press enter three times and press y

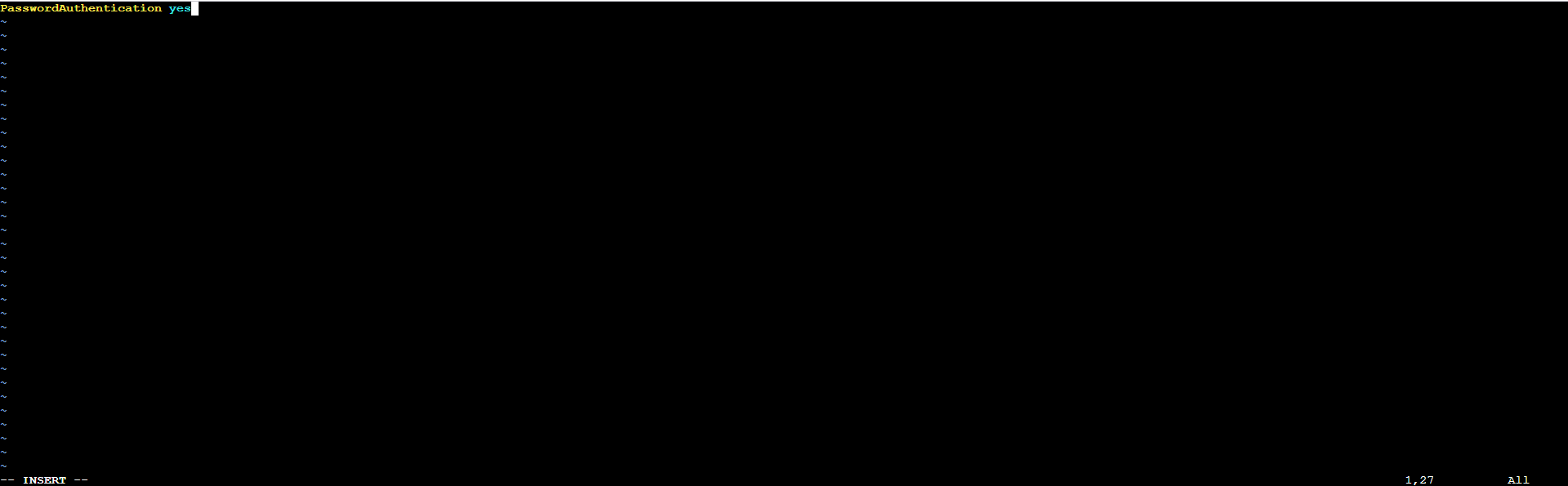
vi /etc/ssh/sshd\_config







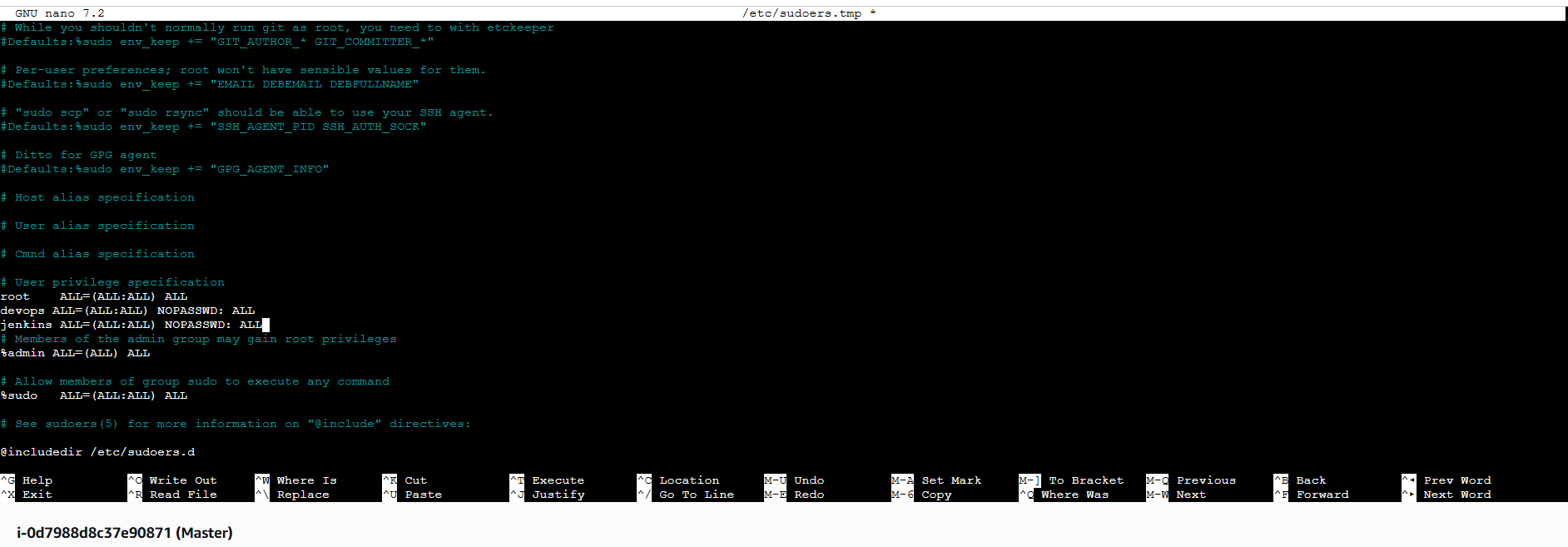
**vi /etc/ssh/sshd\_config.d/60-cloudimg-settings.conf**

****

****

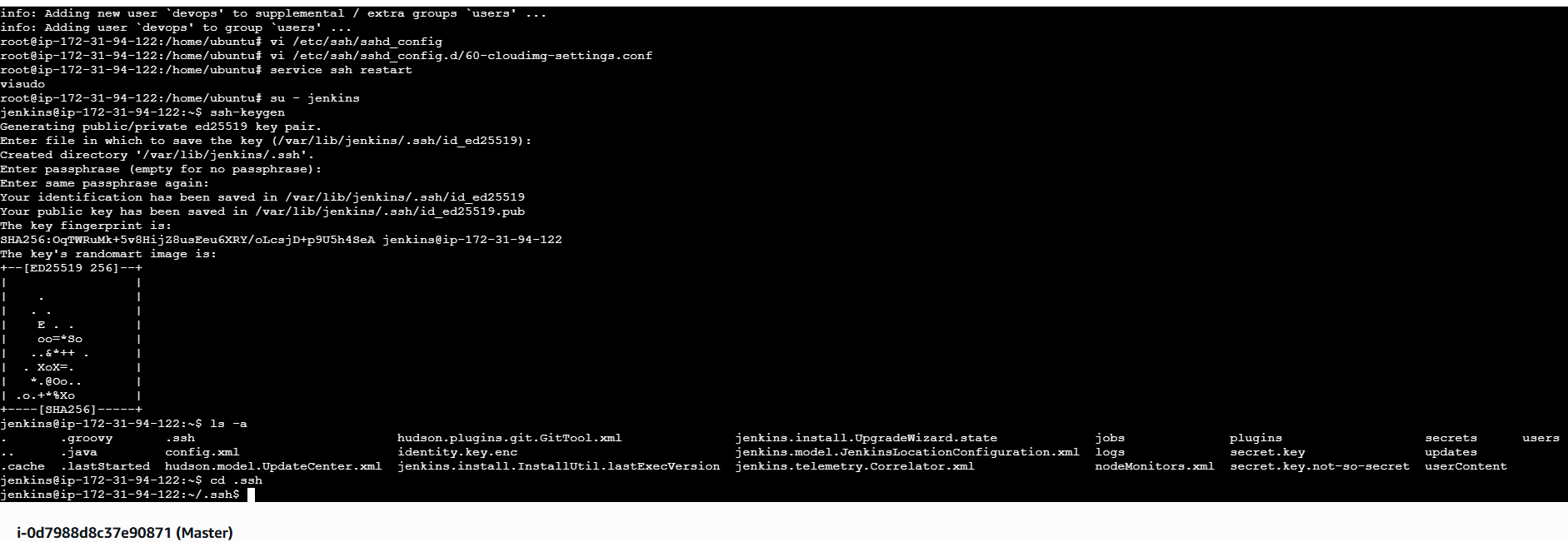
service ssh restart

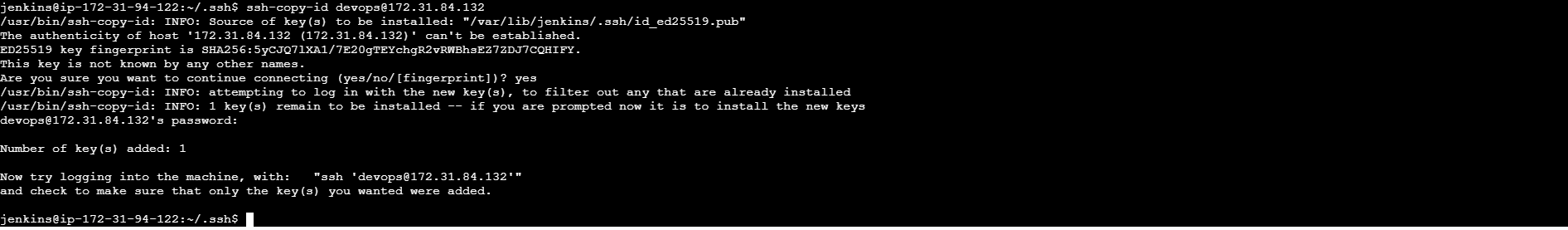
visudo



devops ALL=(ALL:ALL) NOPASSWD: ALL

jenkins ALL=(ALL:ALL) NOPASSWD: ALL





su – jenkins

ssh-keygen

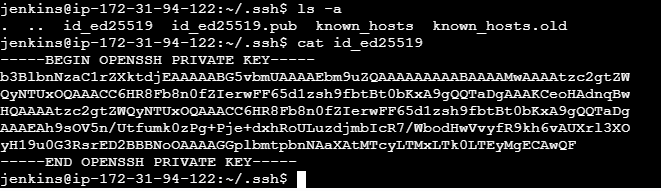
(above command generate the key in master machine)

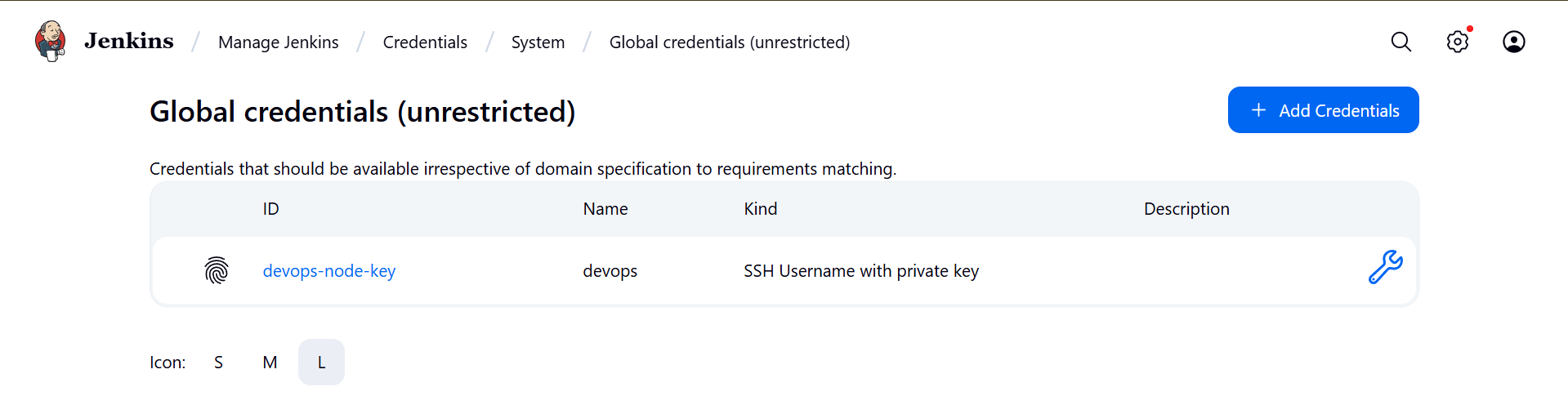
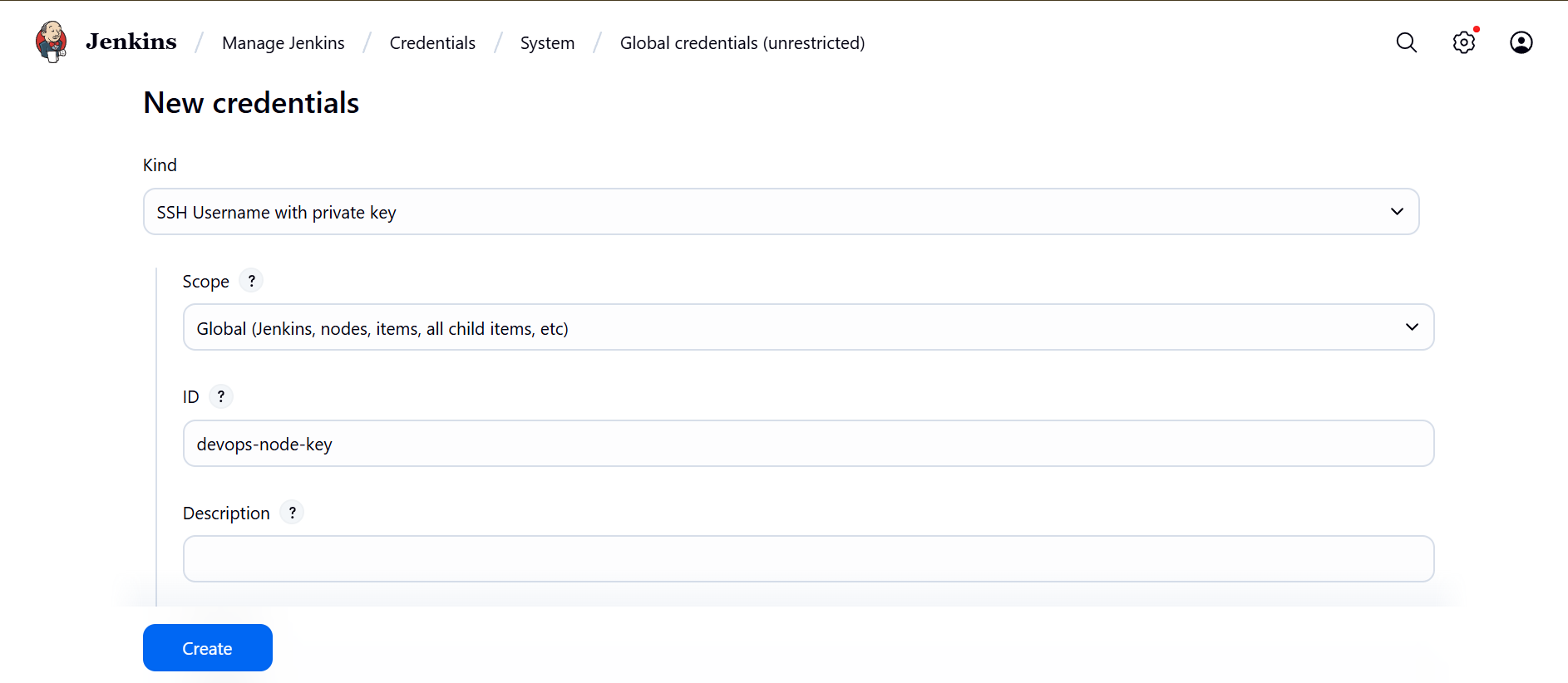
(press enter three times)

ls -a

cd .ssh

ssh-copy-id devops@privateipofnode1





ID : devops-node-key

Username : devops



mkdir apachewebsite

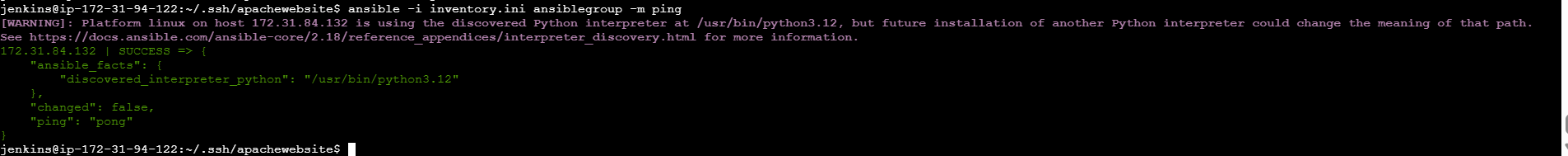
cd apachewebsite

**Inventory file content:-** nodes private ip is placed in this file

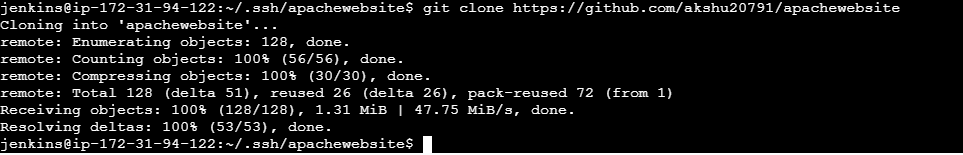
vi inventory.ini  
[ansiblegroup]

172.31.89.150 ansible\_user=devops

ansible -i inventory.ini ansiblegroup -m ping



**git clone https://github.com/Sathya252/apachewebsite**

****

vi install\_apache.yml

****

**- hosts: all**

**become: yes**

**tasks:**

**- name: Install Apache**

**apt:**

**name: apache2**

**state: present**

**update\_cache: yes**

**- name: Start and Enable Apache**

**service:**

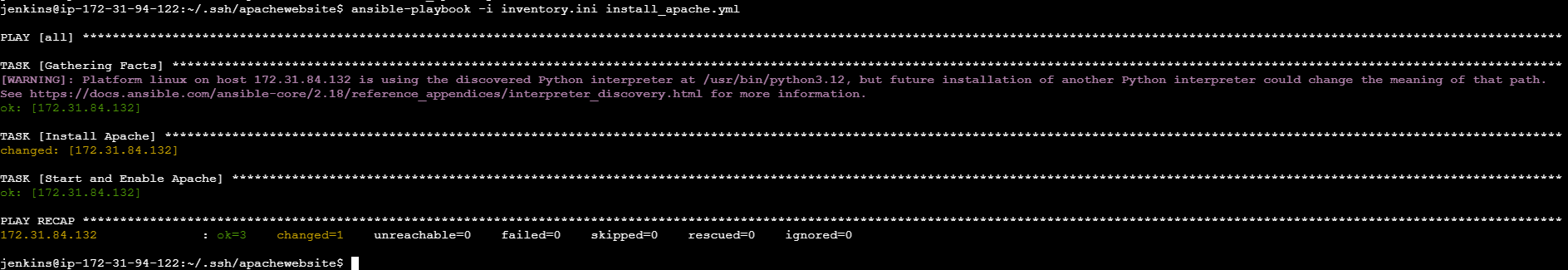
**name: apache2**

**state: started**

**enabled: yes**

****

ansible-playbook -i inventory.ini install\_apache.yml

****

**Install docker**

cd ~

sudo apt install docker.io -y

sudo systemctl enable docker

sudo systemctl start docker

sudo usermod -aG docker jenkins



# Apply Docker group to current shell

newgrp docker

# Confirm access

docker ps

# Work on Docker project

cd ~/.ssh/apachewebsite/apachewebsite

docker login -u rohith252

docker build -t rohith252/apachewebsite:latest .

docker push rohith252/apachewebsite:latest





**vi deployment.yml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: apache-website-deployment

labels:

app: apache-website

spec:

replicas: 2

selector:

matchLabels:

app: apache-website

template:

metadata:

labels:

app: apache-website

spec:

containers:

- name: apache-website

image: rohith252/apachewebsite:latest

ports:

- containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

name: apache-website-service

spec:

selector:

app: apache-website

ports:

- protocol: TCP

port: 80

targetPort: 80

type: NodePort



**vi service.yml**

apiVersion: v1

kind: Service

metadata:

name: apache-service

spec:

selector:

app: apache-website

ports:

- protocol: TCP

port: 80

targetPort: 80

type: LoadBalancer



cd ~

sudo snap install kubectl --classic

curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" | tar xz -C /tmp

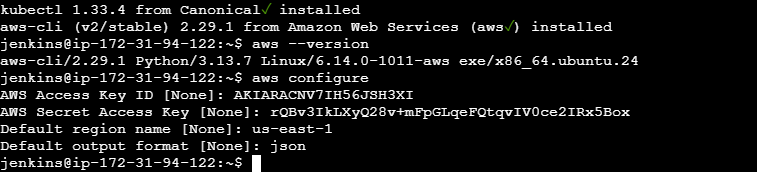
sudo mv /tmp/eksctl /usr/local/bin

sudo snap install aws-cli --classic

aws --version

aws configure

**to set your AWS Access Key, Secret Key, region, and output format.**



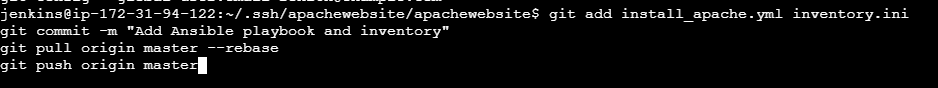
cd .ssh/apachewebsite/apachewebsite

mv ../install\_apache.yml .

mv ../inventory.ini .  


git config --global user.name "Your Name"

git config --global user.email "you@example.com"

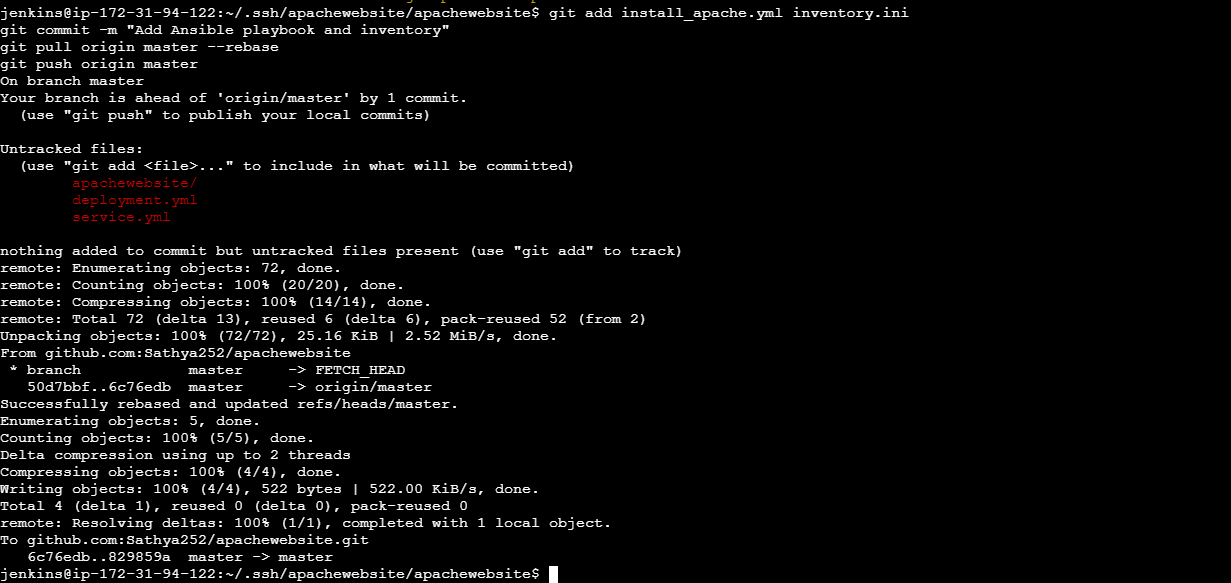


git add install\_apache.yml inventory.ini

git commit -m "Add Ansible playbook and inventory"

git pull origin master --rebase

git push origin master





cd ~  
eksctl create cluster \

--name apache-cluster-3 \ # change this each time (must be unique)

--version 1.30 \ # usually keep same unless testing different K8s version

--region us-east-1 \ # change only if you want cluster in another region

--nodegroup-name apache-nodes-3 \ # also change if reusing same name causes conflict

--node-type c7i-flex.large \ # can change instance type if you want different size

--nodes 2 # change number of nodes if you need more/less

cd ~

eksctl create cluster \

--name apache-cluster-3 \

--version 1.30 \

--region us-east-1 \

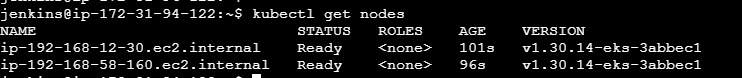
--nodegroup-name apache-nodes-3 \

--node-type c7i-flex.large \

--nodes 2

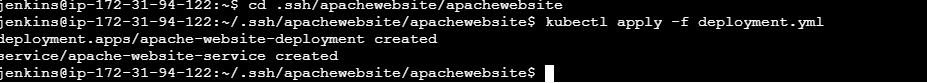


kubectl get nodes



cd .ssh/apachewebsite/apachewebsite

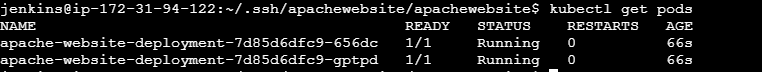
kubectl apply -f deployment.yml



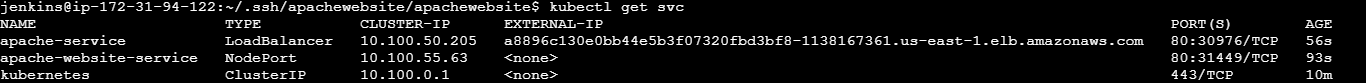
kubectl apply -f service.yml



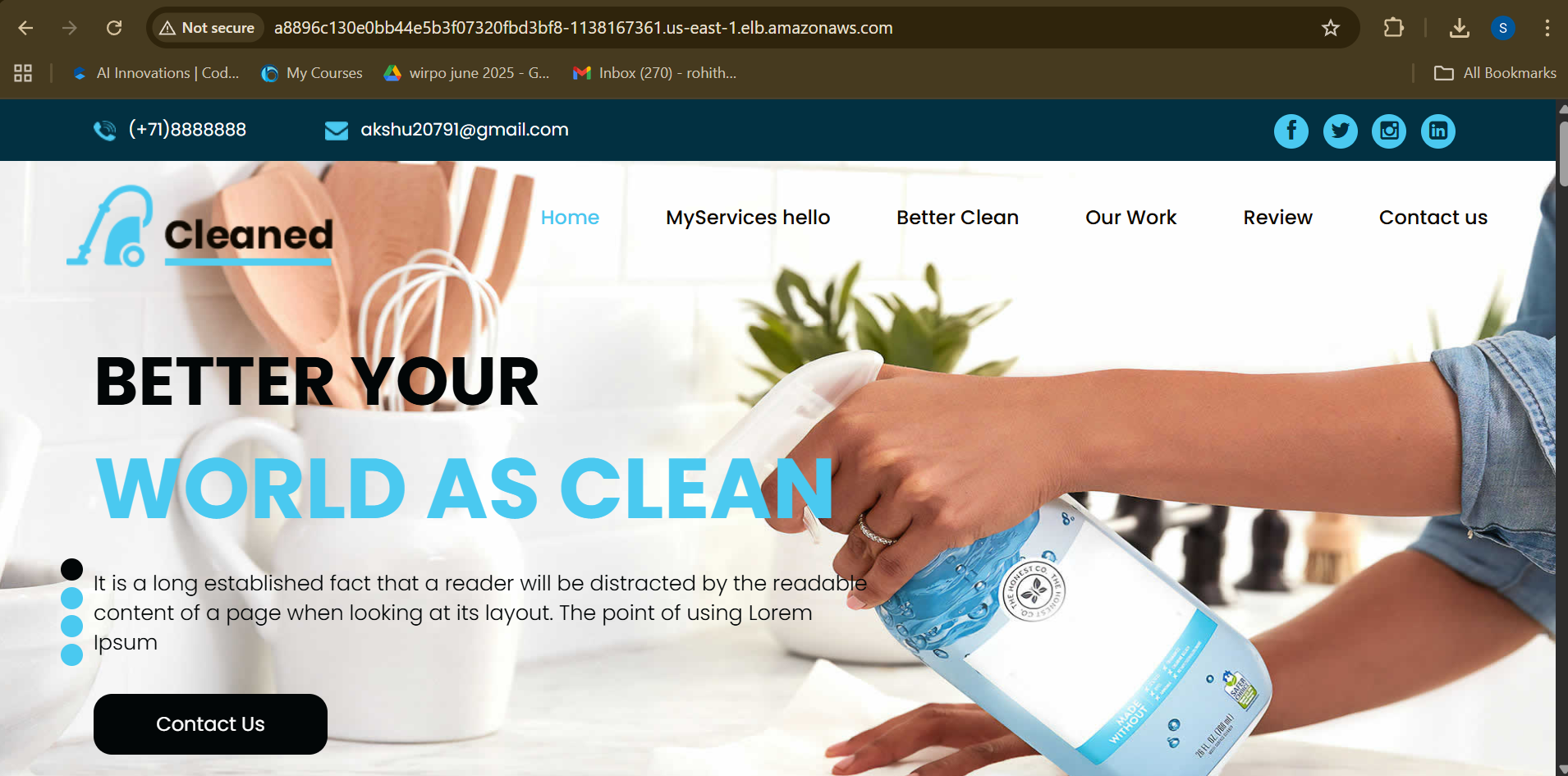
kubectl get pods



kubectl get svc



http://a8896c130e0bb44e5b3f07320fbd3bf8-1138167361.us-east-1.elb.amazonaws.com

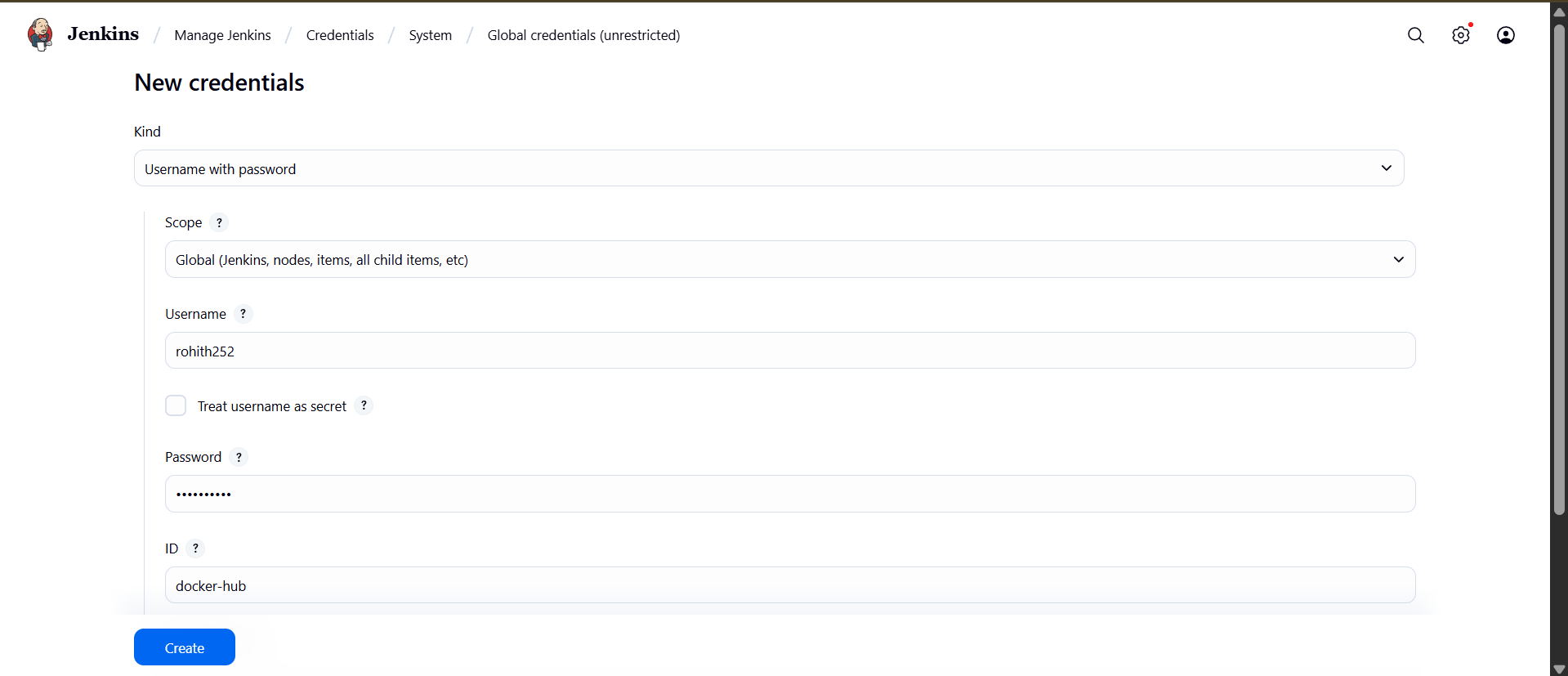


**Add Docker credentials in jenkins**

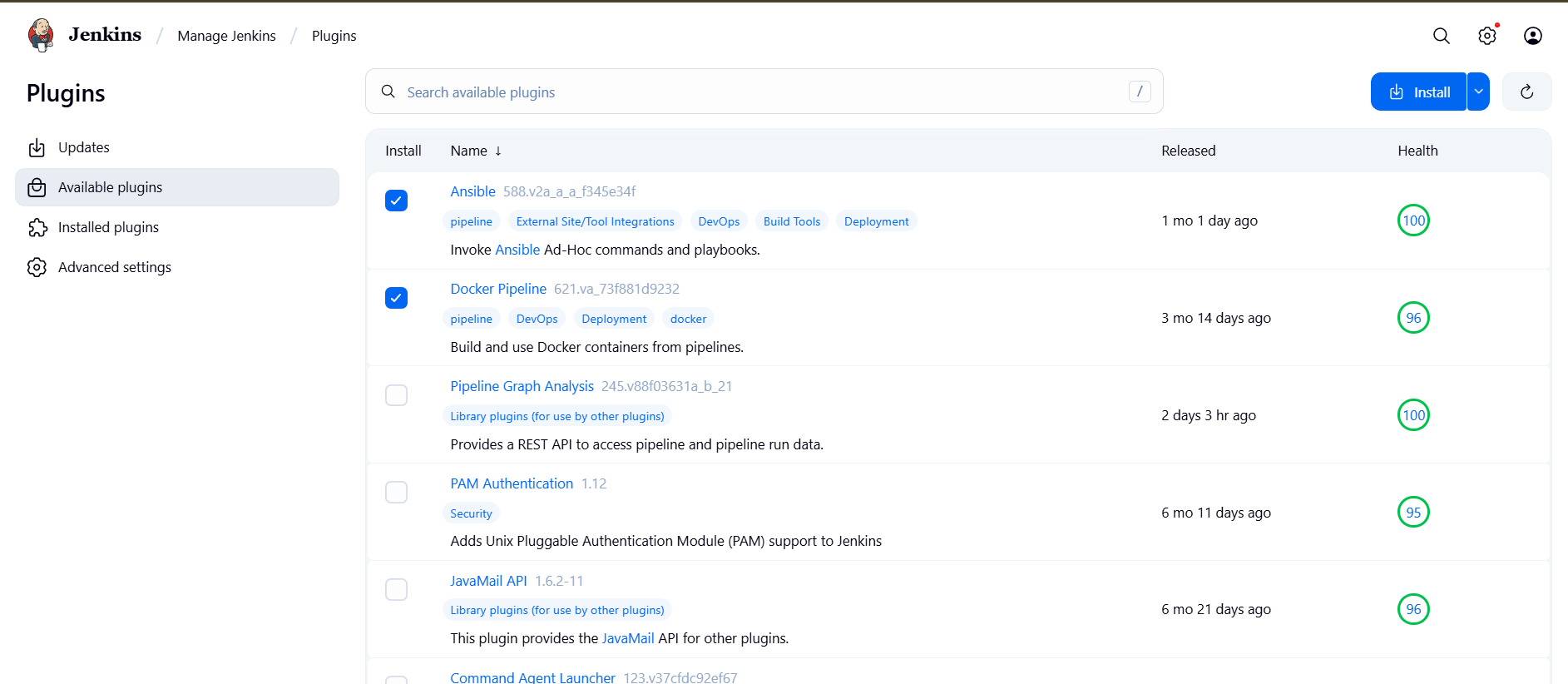
ID : docker-hub

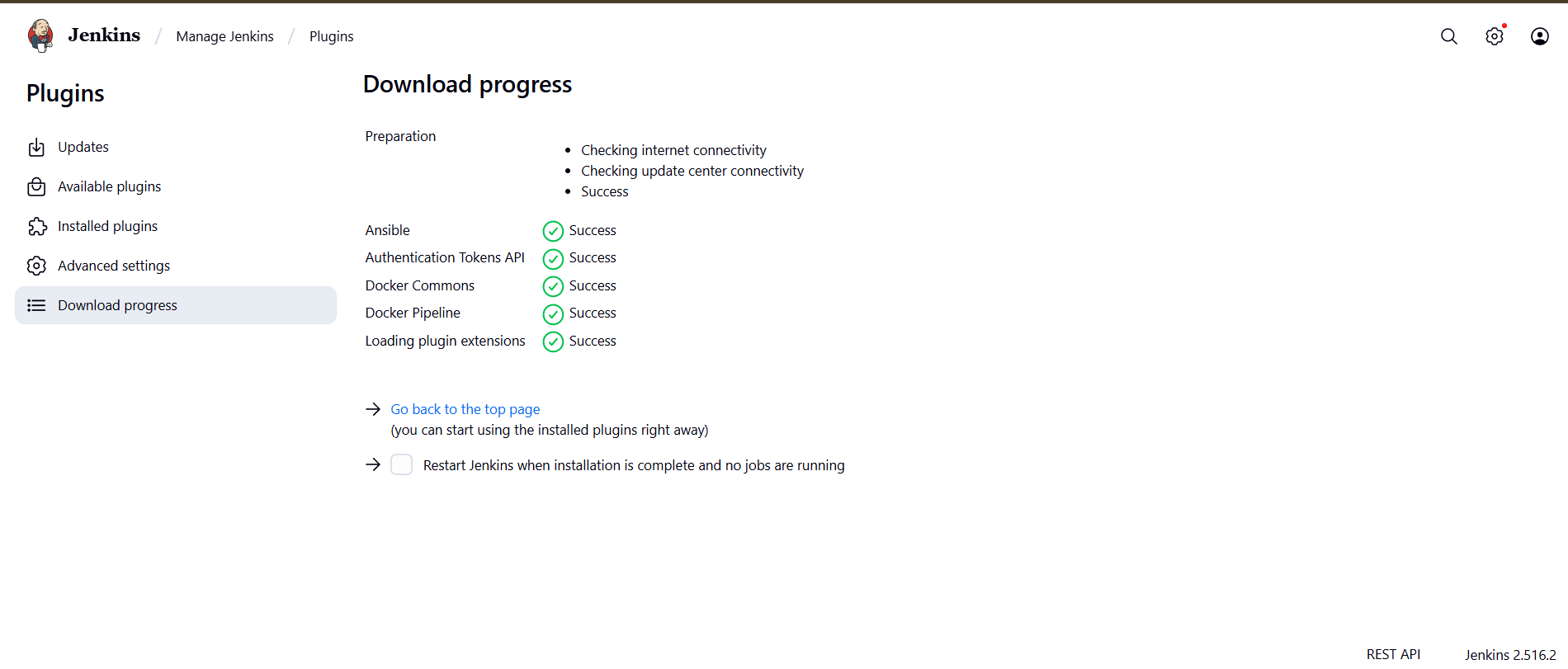
Username: <docker username>

passwd: <docker passwd>

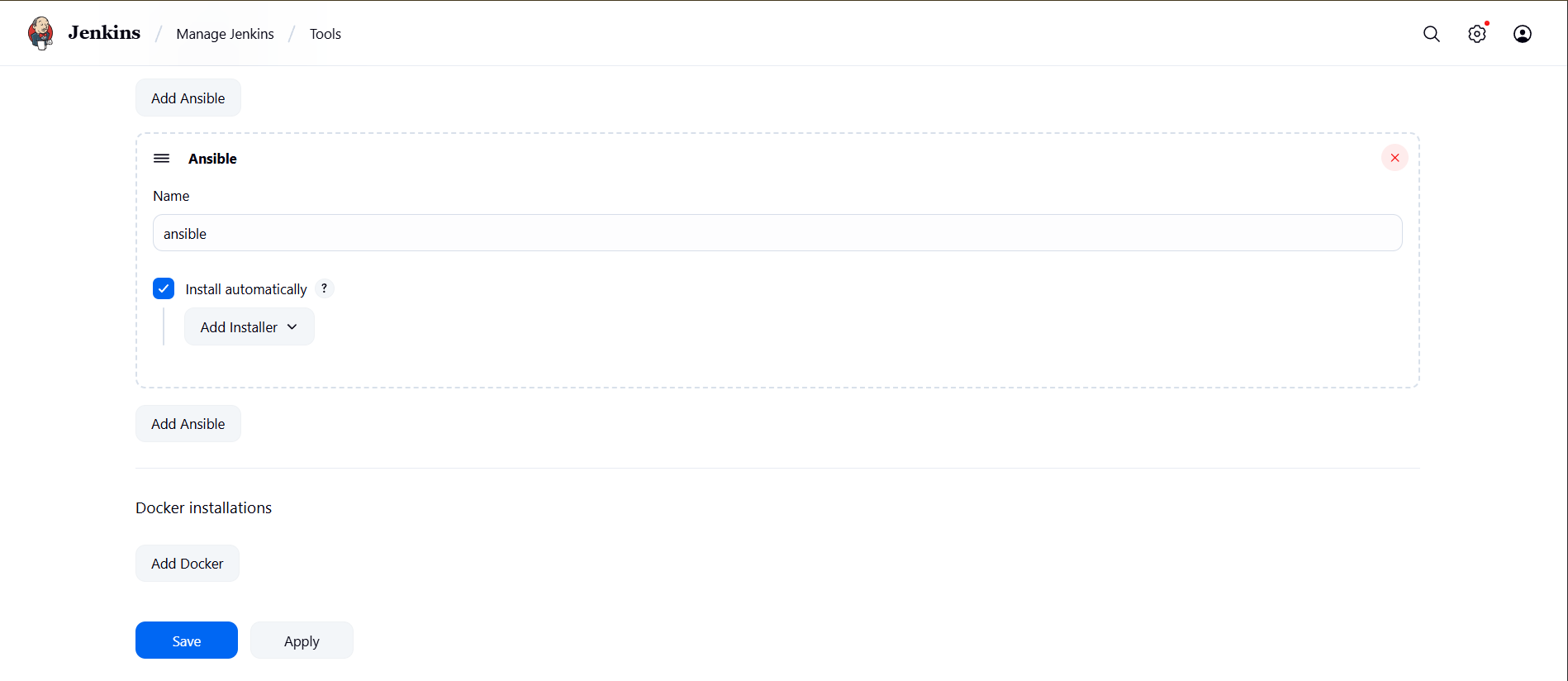


Install Ansible & Docker pipeline Plugins in jenkins

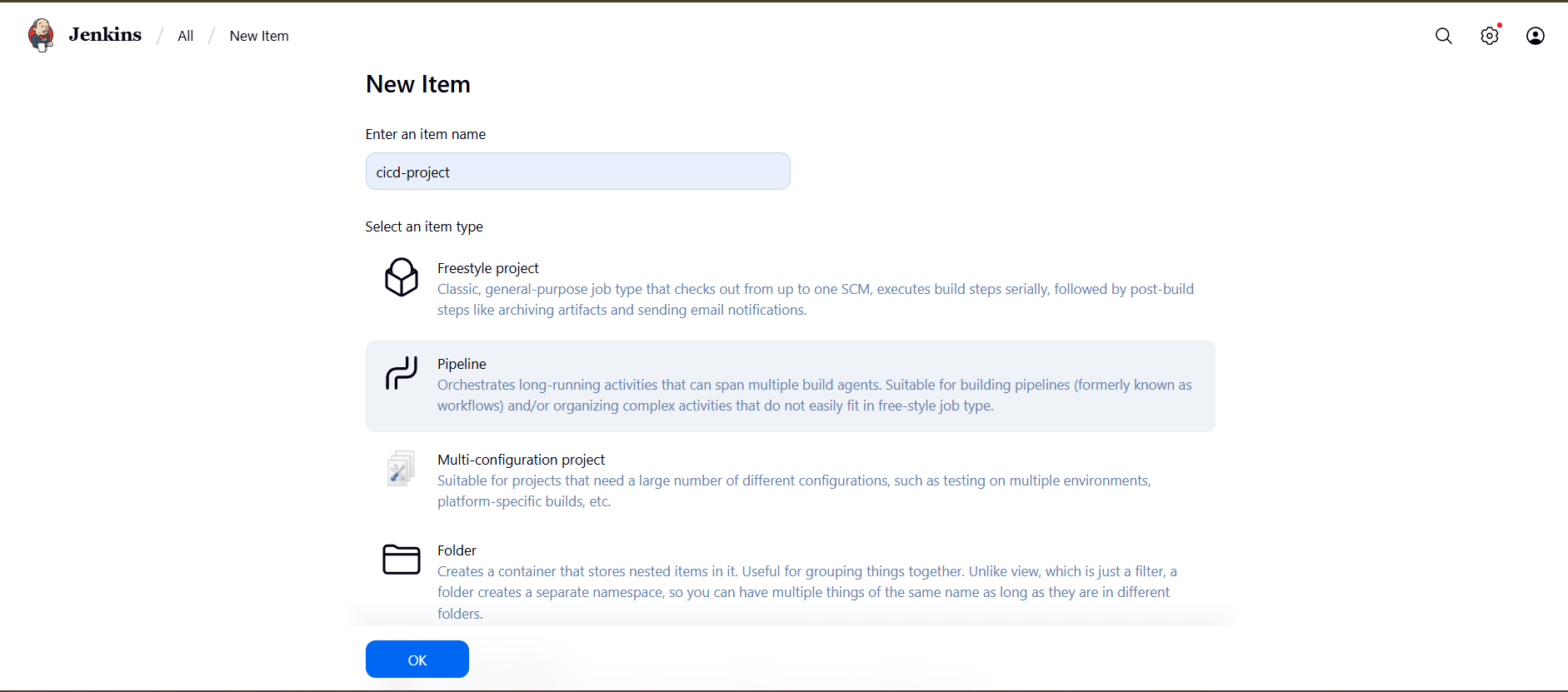




Add ansible tools in jenkins



Add new job pipeline in jenkins





pipeline {

agent any

environment {

REPO\_PATH = '/var/lib/jenkins/.ssh/apachewebsite/apachewebsite'

INVENTORY = '/var/lib/jenkins/.ssh/apachewebsite/inventory.ini'

DOCKER\_IMAGE = 'rohith252/apachewebsite:latest'

DOCKERHUB\_CREDENTIALS = 'docker-hub'

ANSIBLE\_CREDENTIALS = 'devops-node-key'

}

stages {

stage('Checkout GitHub Repo') {

steps {

git branch: 'master', url: 'https://github.com/Sathya252/apachewebsite.git'

}

}

stage('Install Apache via Ansible') {

steps {

withCredentials([sshUserPrivateKey(credentialsId: "${ANSIBLE\_CREDENTIALS}", keyFileVariable: 'SSH\_KEY', usernameVariable: 'SSH\_USER')]) {

sh """

ansible-playbook -i ${INVENTORY} ${REPO\_PATH}/install\_apache.yml

"""

}

}

}

stage('Build Docker Image') {

steps {

withCredentials([usernamePassword(credentialsId: "${DOCKERHUB\_CREDENTIALS}", usernameVariable: 'DOCKER\_USER', passwordVariable: 'DOCKER\_PASS')]) {

sh """

echo "$DOCKER\_PASS" | docker login -u "$DOCKER\_USER" --password-stdin

docker build -t ${DOCKER\_IMAGE} ${REPO\_PATH}

"""

}

}

}

stage('Push Docker Image to DockerHub') {

steps {

withCredentials([usernamePassword(credentialsId: "${DOCKERHUB\_CREDENTIALS}", usernameVariable: 'DOCKER\_USER', passwordVariable: 'DOCKER\_PASS')]) {

sh """

echo "$DOCKER\_PASS" | docker login -u "$DOCKER\_USER" --password-stdin

docker push ${DOCKER\_IMAGE}

"""

}

}

}

stage('Deploy to Kubernetes') {

steps {

sh """

kubectl apply -f ${REPO\_PATH}/deployment.yml

kubectl apply -f ${REPO\_PATH}/service.yml

"""

}

}

stage('Verify Deployment') {

steps {

sh """

kubectl get pods

kubectl get svc

"""

}

}

}

post {

success {

echo 'Pipeline completed successfully!'

}

failure {

echo 'Pipeline failed. Check the console output for errors.'

}

}

}

**Push this Modified Jenkinsfile to your repo:**

git add Jenkinsfile

git commit -m "Final Jenkinsfile ready for master branch"

git push origin master

