**Jenkins Pipeline to Create EKS cluster Using Terraform**

**Pre-requisites:**

* Install Java
* Install GIT
* Install Jenkins
* Install Terraform
* Allow SUDO permissions for Jenkins User
* Make PasswordAuthentication to yes
* Create IAM Role to Access AWS Services
* Install Kubectl
* Install aws-iam-authenticator

Install Java:

yum install java-1.8.0-openjdk -y

Install GIT:

yum install git -y

Install Jenkins:

sudo wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo

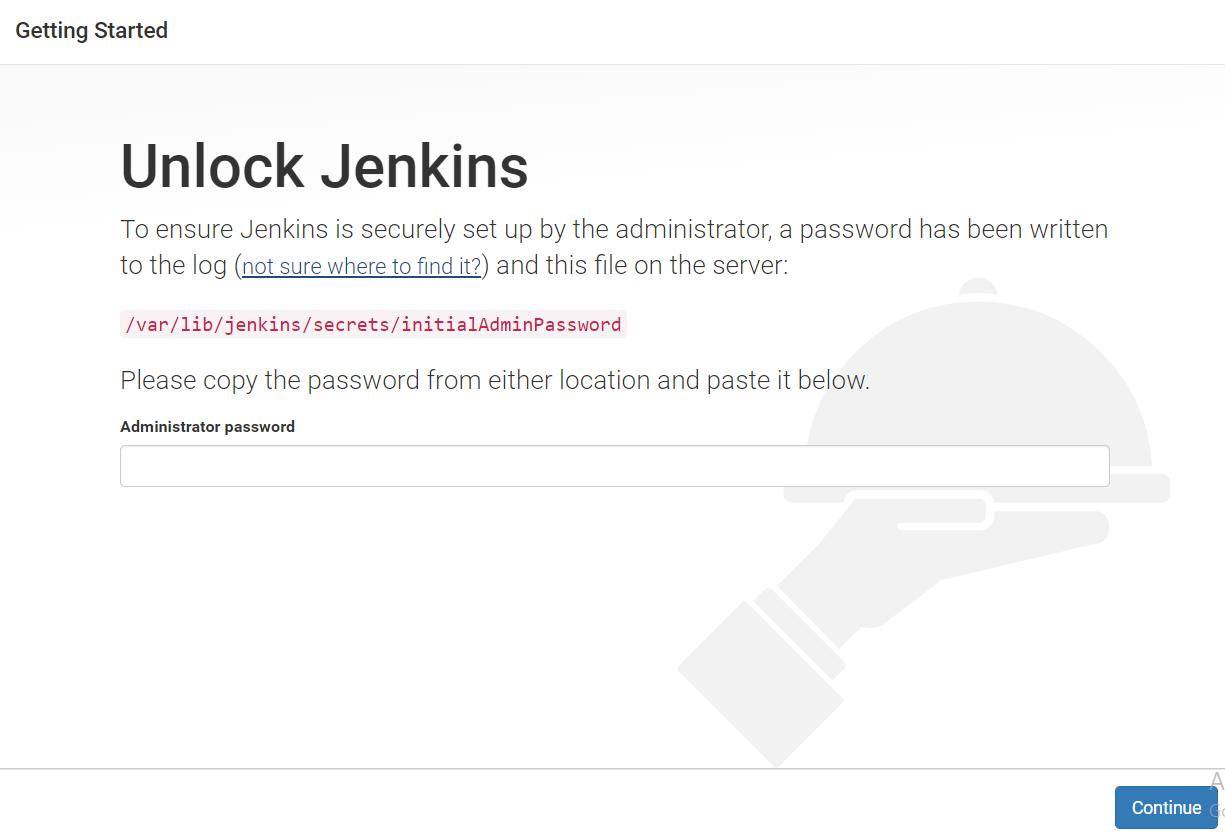
sudo rpm --import https://jenkins-ci.org/redhat/jenkins-ci.org.key

sudo yum install jenkins -y

service jenkins start

Open Jenkins in UI:

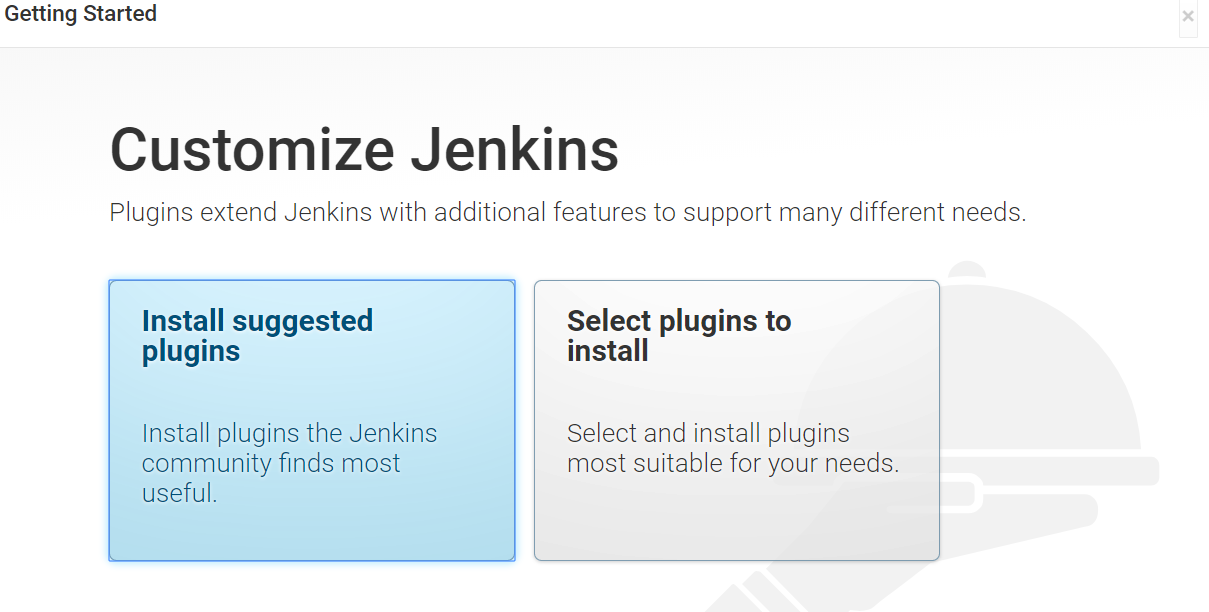
<ip-address>:8080



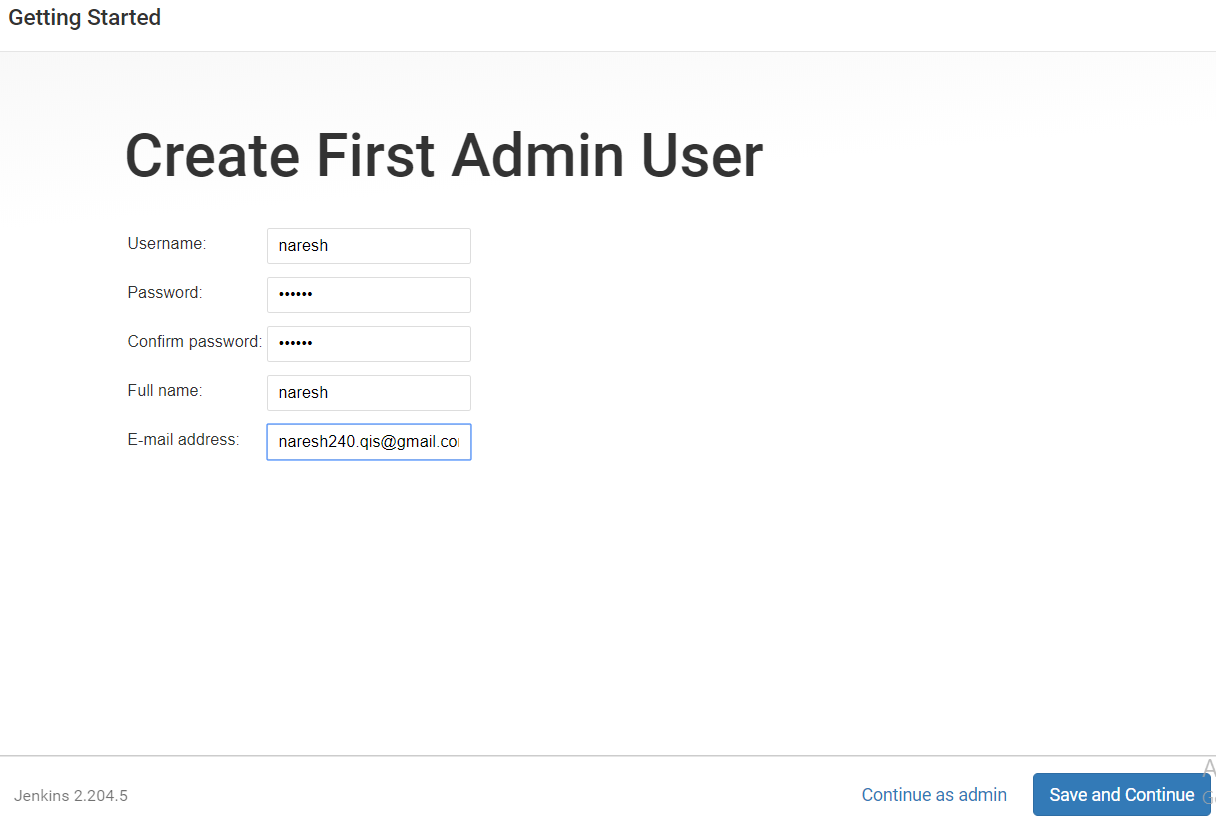
cat /var/lib/jenkins/secrets/initialAdminPassword



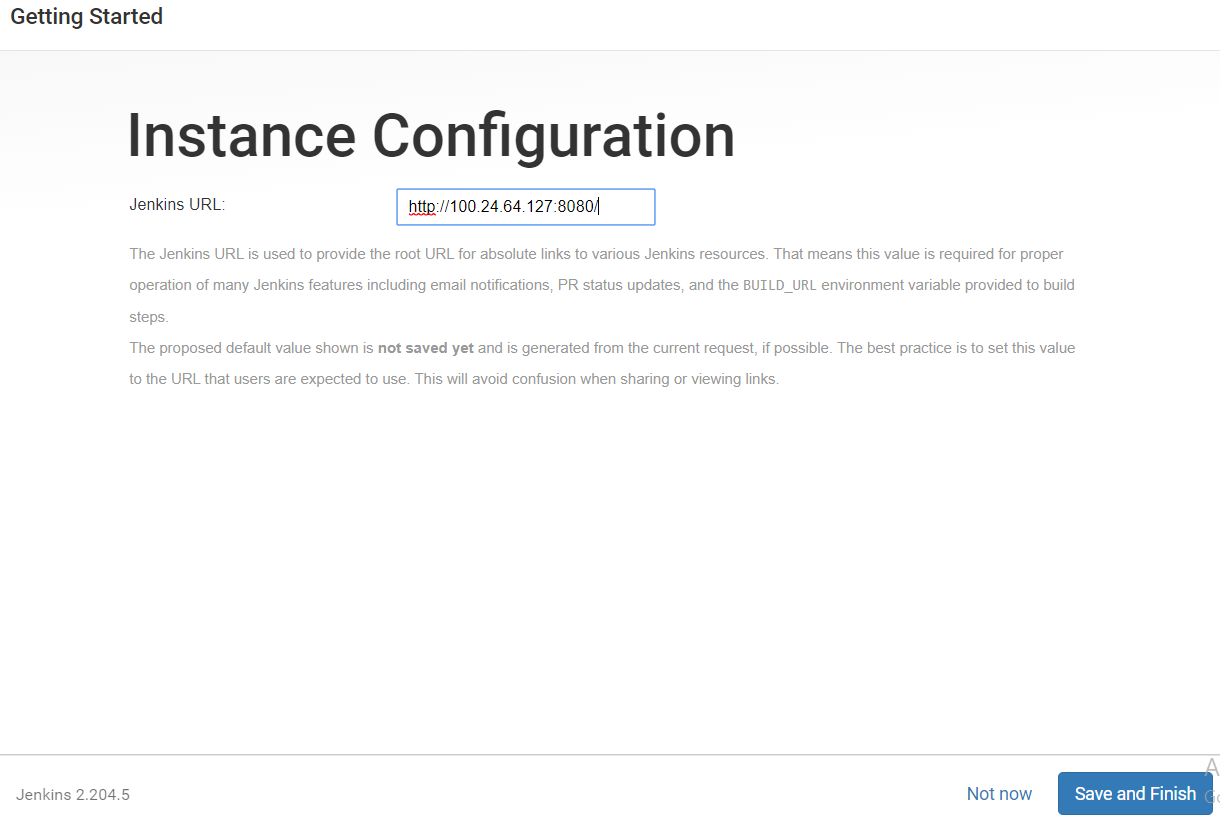
Copy password and paste in UI and then Click on **Continue**



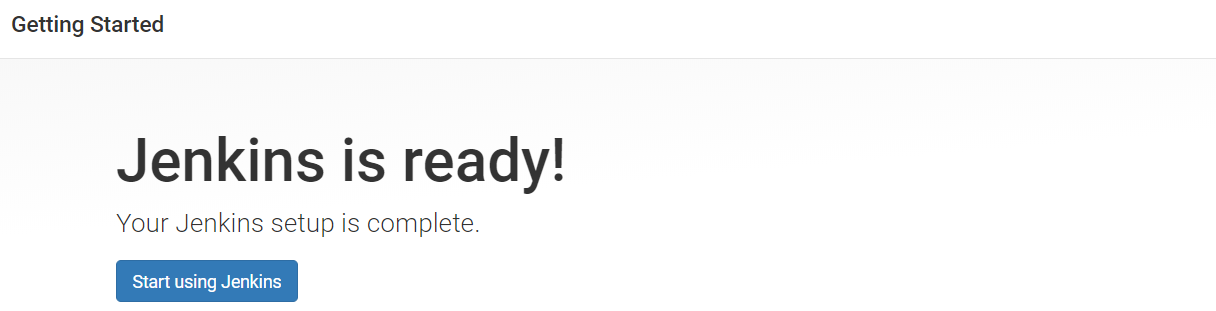
Click on **Install suggested plugins**



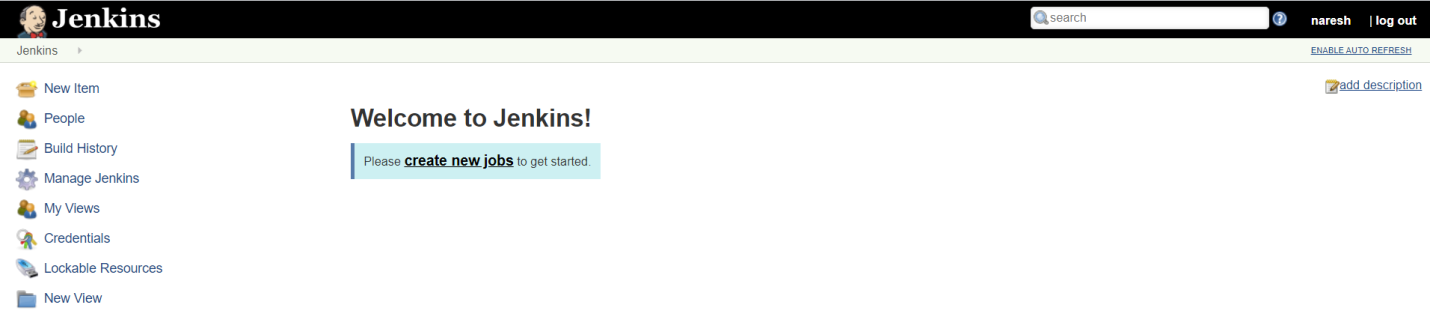
Provide details as above and click on **Save and Continue**



Click on **Save and Finish**



Click on **Start using Jenkins**



Install Terraform:

wget <https://releases.hashicorp.com/terraform/0.12.24/terraform_0.12.24_linux_amd64.zip>

unzip terraform\_0.12.24\_linux\_amd64.zip

mv terraform /usr/bin

terraform –version

Allow SUDO permissions for Jenkins User:

visudo

Add below line inside the file and save it

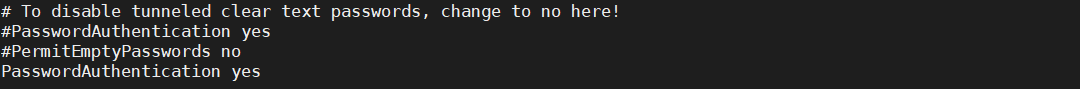
jenkins ALL=(ALL) NOPASSWD: ALL



Make PasswordAuthentication to yes:

vi /etc/ssh/sshd\_config

Change PasswordAuthentication value from **no** to **yes**



Restart **sshd** service:

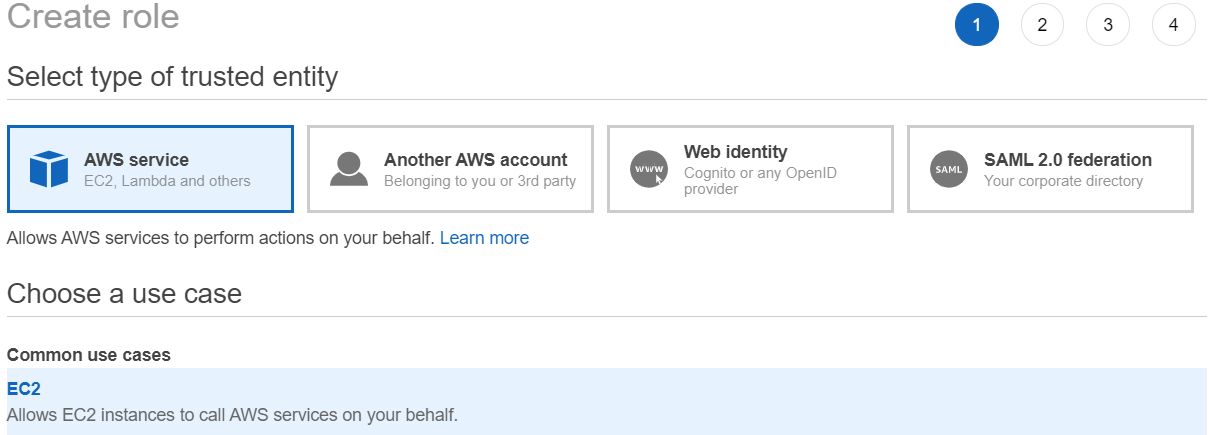
service sshd restart

Create IAM Role to Access AWS Services:

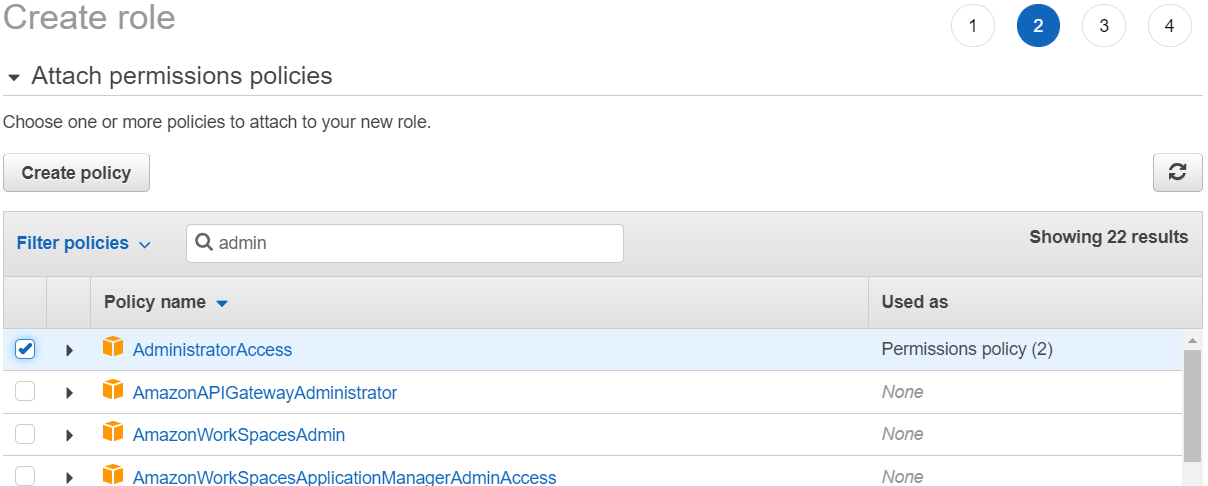
Go to IAM service 🡪 Roles



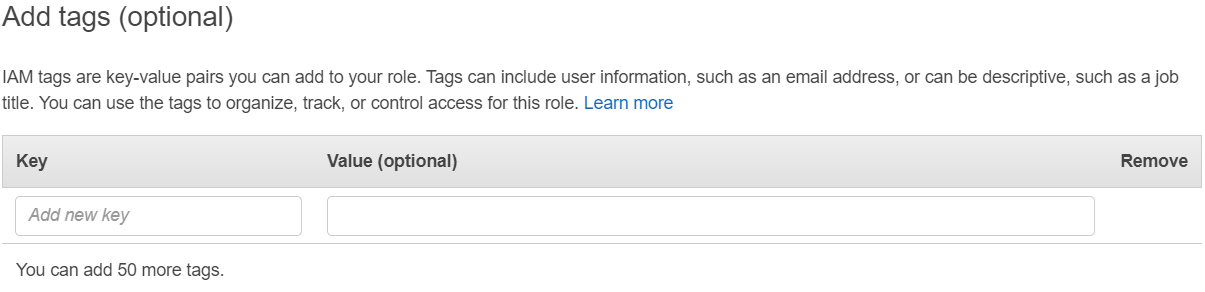
Click on **Create role**



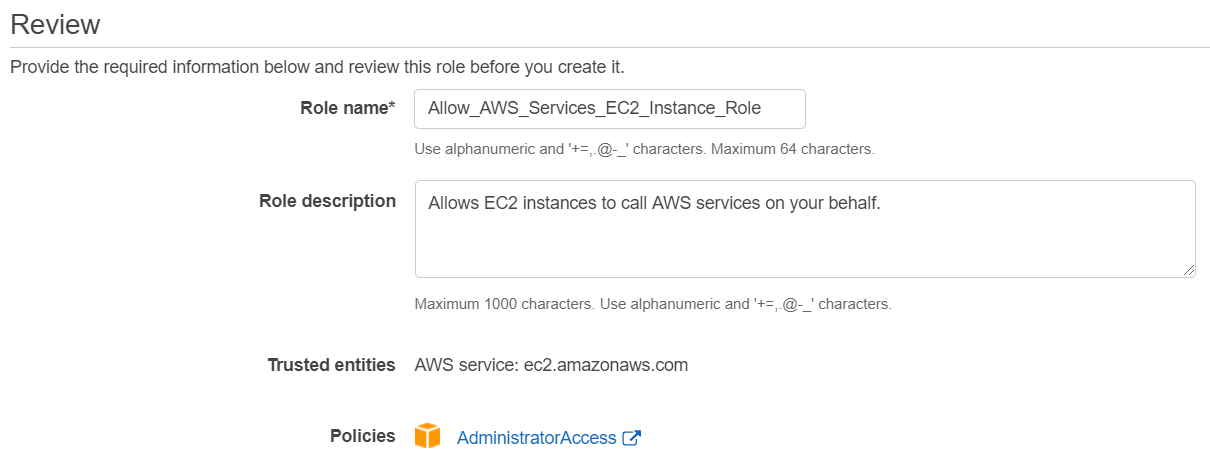
Click on **Next**



Click on **Next**



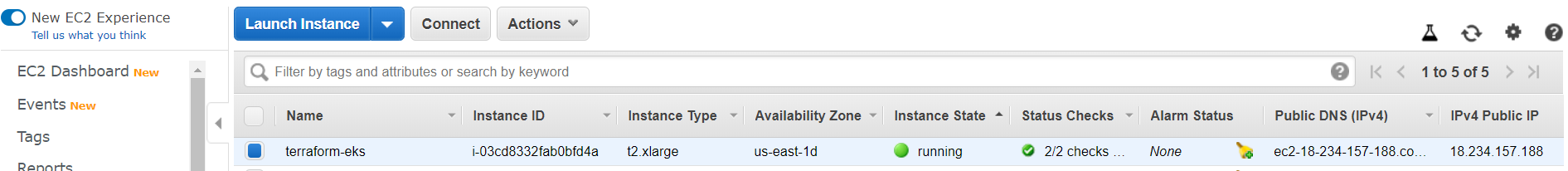
Click on **Next**



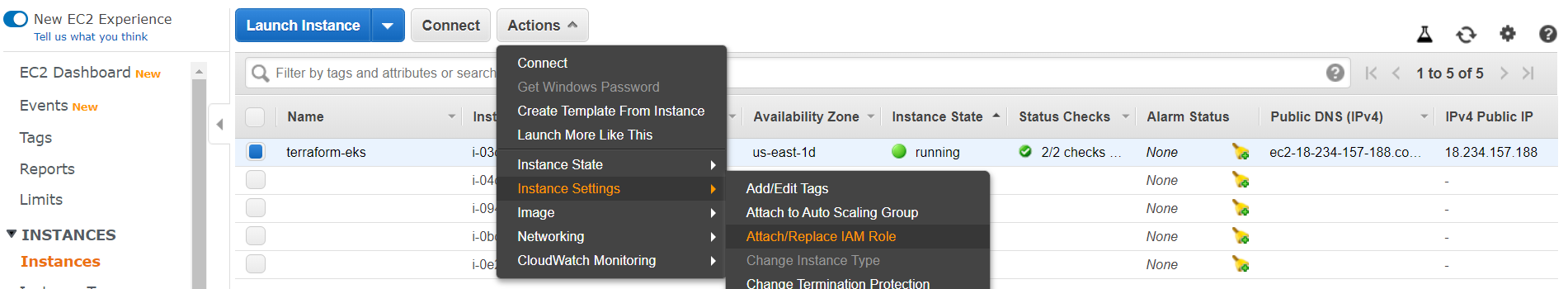
Click on **Create**

Attach Role to EC2 Instance:

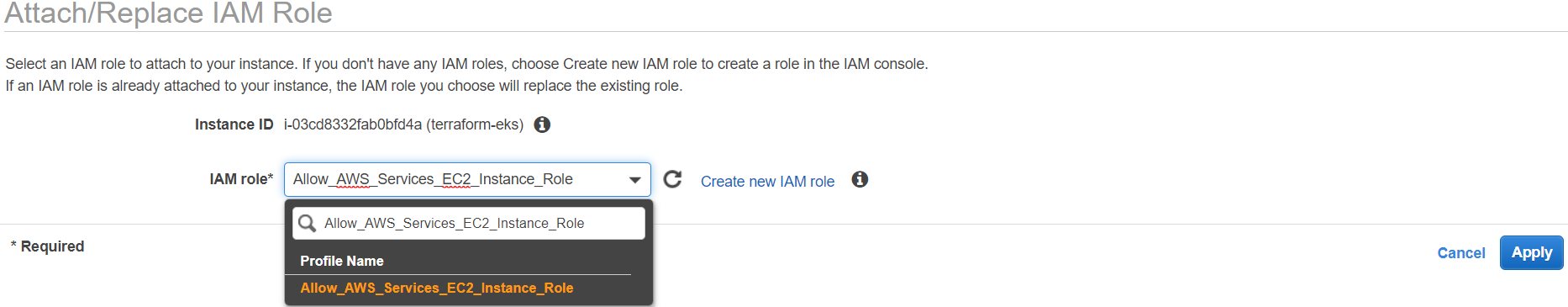
Go to EC2 Service 🡪 Instances



Click on **Actions**



Click on Instance Settings 🡪 Attach/Replace IAM Role



Select Role and Click on **Apply**

Install kubectl:

curl -o kubectl https://amazon-eks.s3-us-west-2.amazonaws.com/1.14.6/2019-08-22/bin/linux/amd64/kubectl

chmod +x ./kubectl

mkdir -p /usr/bin

cp ./kubectl /usr/bin/kubectl

export PATH=/usr/bin:$PATH

echo 'export PATH=/usr/bin:$PATH' >> ~/.bashrc

source /usr/.bashrc

kubectl version --short –client

Install aws-iam-authenticator:

curl -o aws-iam-authenticator https://amazon-eks.s3-us-west-2.amazonaws.com/1.14.6/2019-08-22/bin/linux/amd64/aws-iam-authenticator

chmod +x ./aws-iam-authenticator

cp ./aws-iam-authenticator /usr/bin/aws-iam-authenticator

export PATH=/usr/bin:$PATH

echo 'export PATH=/usr/bin:$PATH' >> ~/.bashrc

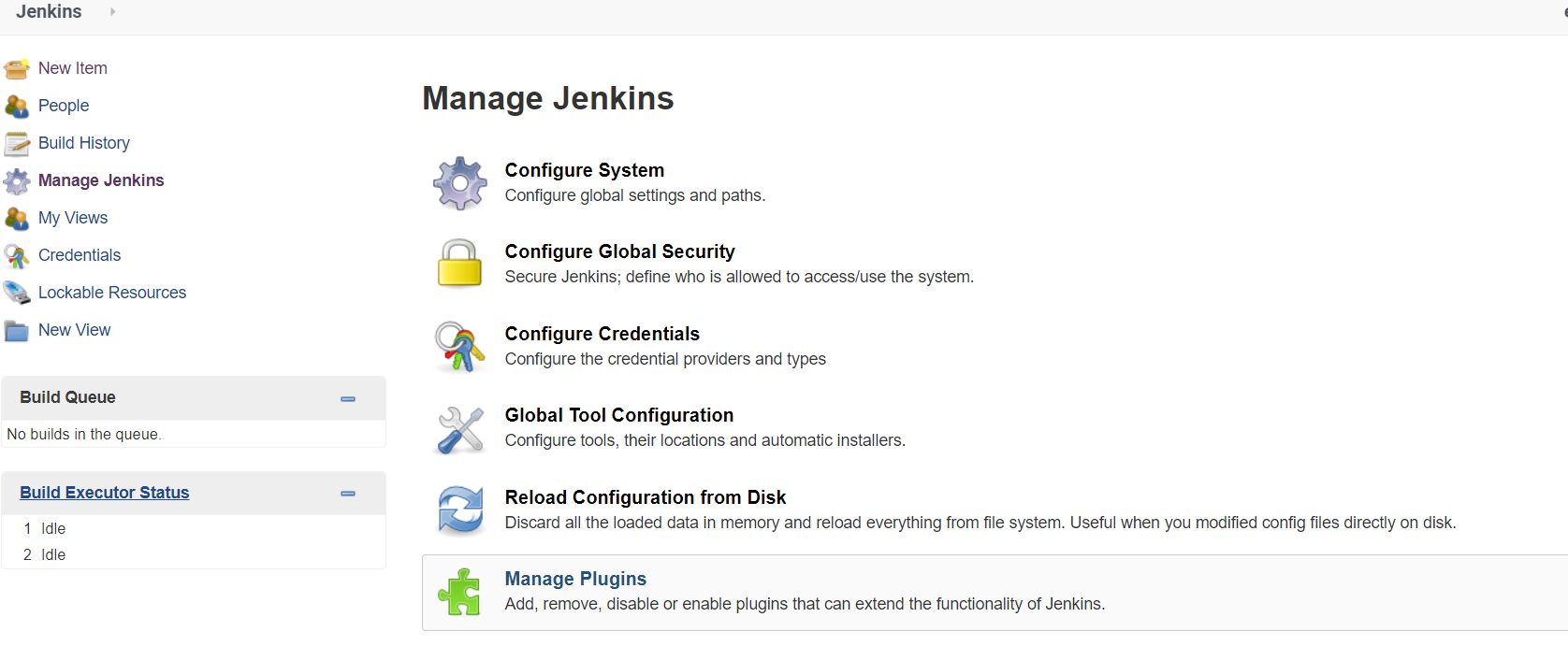
source ~/.bashrc

aws-iam-authenticator --help

Configure Terraform with Jenkins:

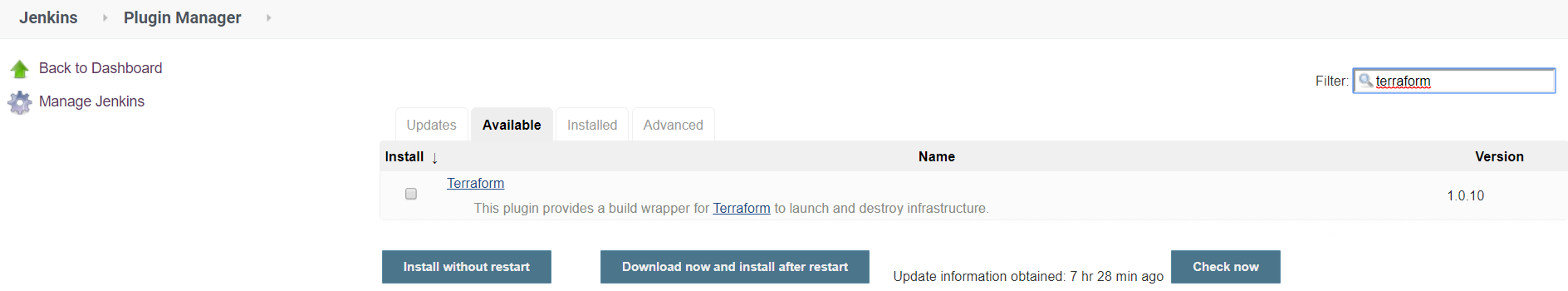
Go to Jenkins Dashboard and add **Terraform** plugin

Click on Manage Jenkins



Click on Manage plugins:

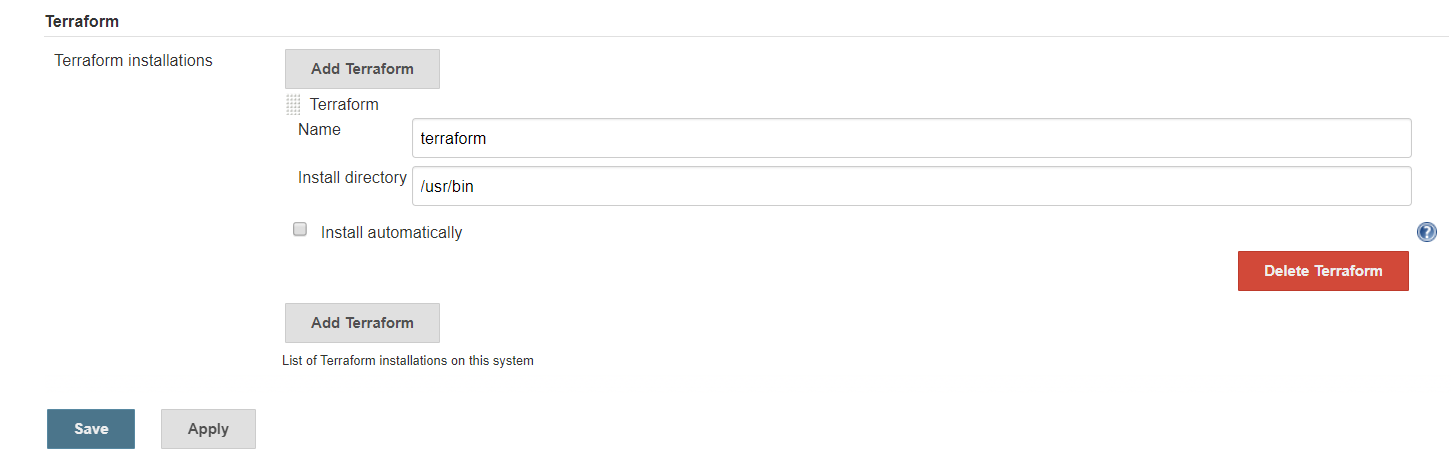
Go to Jenkins Dashboard 🡪 Manage Jenkins 🡪 Manage Plugins



Click on Available and search Terraform

Select Terraform plugins and click on Install without restart

Manage Jenkins 🡪 Global Tool Configuration

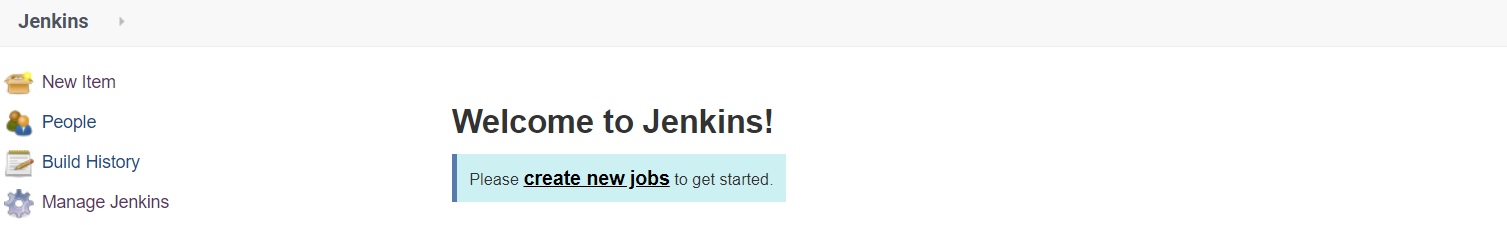


Add Name and Path of Terraform file

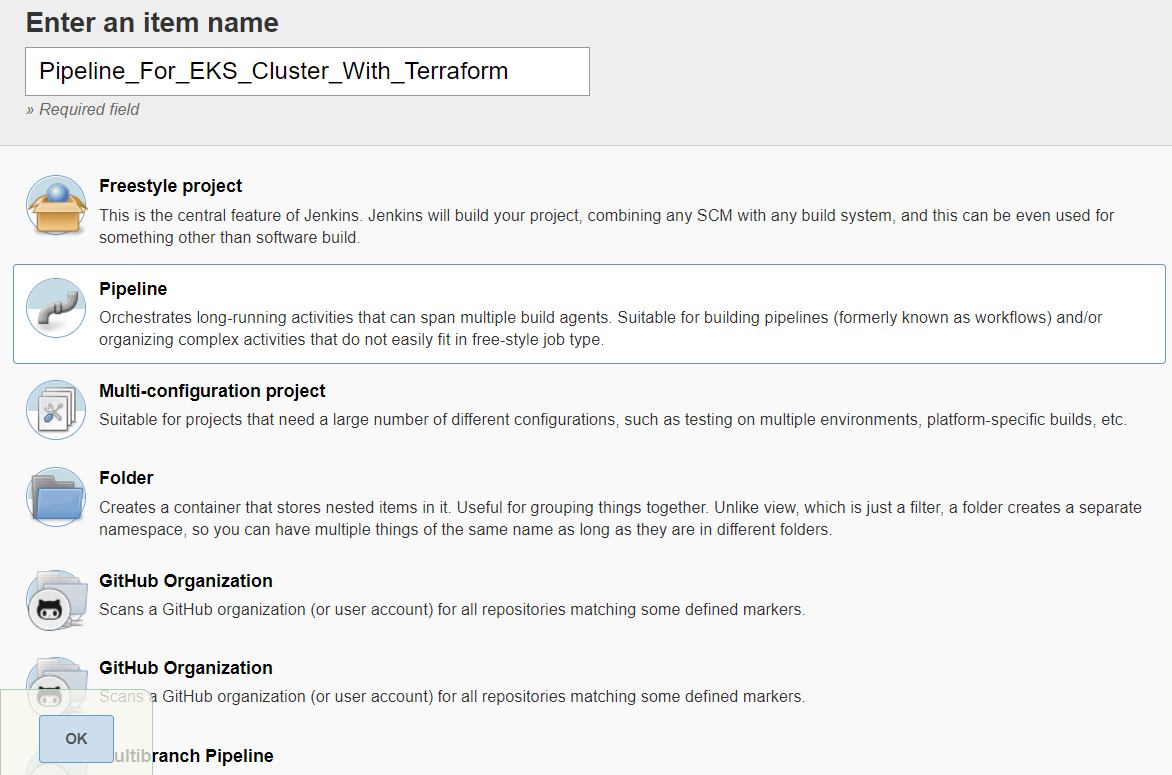
Click on **Save**

Create Jenkins Job (Pipeline) to create EKS cluster with Terraform:

Go to Jenkins Dashboard

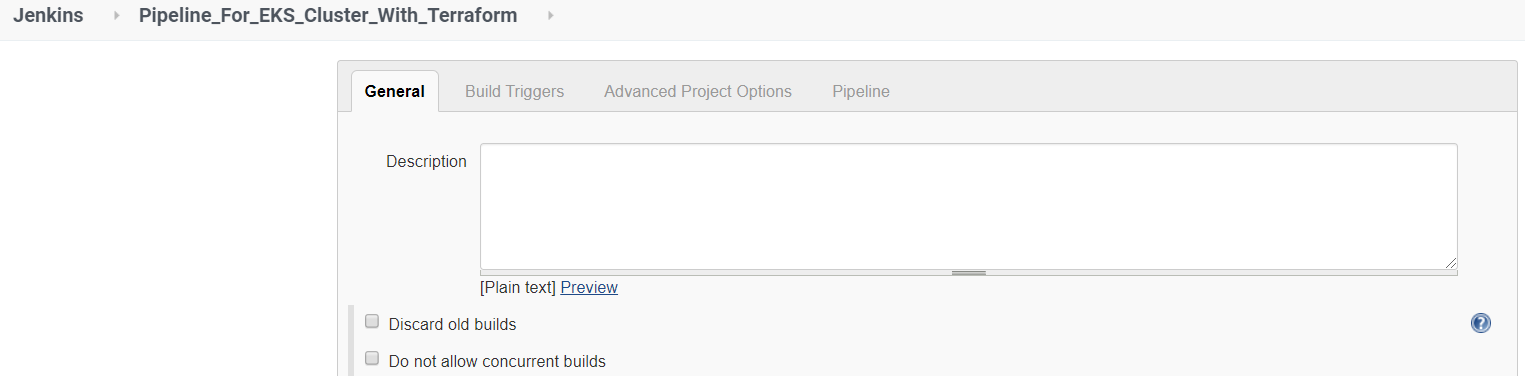


Click on **New Item**



Give a name for the Jenkins job and select pipeline

Click on **OK**



Click on **Pipeline**



Give below pipeline Script and Click on **Save**

pipeline {

parameters {

choice(name: 'action', choices: 'create\ndestroy', description: 'Create/update or destroy the eks cluster.')

string(name: 'aws\_access\_key', defaultValue : 'AKIA3QQQQARMBV3FNX6R', description: "aws\_access\_key")

string(name: 'aws\_secret\_key', defaultValue : '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', description: "aws\_secret\_key")

string(name: 'cluster', defaultValue : 'demo', description: "EKS cluster name;eg demo creates cluster named eks-demo.")

}

agent any

stages {

stage('checkout') {

steps {

git 'https://github.com/Naresh240/eksterraform.git'

}

}

stage('Setup') {

steps {

script {

currentBuild.displayName = "#" + env.BUILD\_NUMBER + " " + params.action + " eks-" + params.cluster

plan = params.cluster + '.plan'

}

}

}

stage('Set Terraform path') {

steps {

script {

def tfHome = tool name: 'terraform'

env.PATH = "${tfHome}:${env.PATH}"

}

sh 'terraform -version'

}

}

stage('TF Plan') {

when {

expression { params.action == 'create' }

}

steps {

script {

// Format cidrs into a list array

sh """

terraform init

terraform workspace new ${params.cluster} || true

terraform workspace select ${params.cluster}

terraform plan \

-var cluster-name=${params.cluster} \

-out ${plan}

echo ${params.cluster}

"""

}

}

}

stage('TF Apply') {

when {

expression { params.action == 'create' }

}

steps {

script {

// input "Create/update Terraform stack eks-${params.cluster} in aws?"

sh """

terraform apply -input=false -auto-approve ${plan}

"""

}

}

}

stage('TF Destroy') {

when {

expression { params.action == 'destroy' }

}

steps {

script {

//input "Destroy Terraform stack eks-${params.cluster-name} in aws?"

sh """

terraform workspace select ${params.cluster}

terraform destroy -auto-approve

"""

}

}

}

stage('Creating Config file') {

when {

expression { params.action == 'create' }

}

steps {

script {

if (fileExists('$HOME/.kube')) {

echo '.kube Directory Exists'

} else {

sh 'mkdir -p $HOME/.kube'

// sh 'sudo chown $(id -u):$(id -g) $HOME/.kube/'

}

sh 'terraform output kubeconfig > $HOME/.kube/config'

sh 'sudo chown $(id -u):$(id -g) $HOME/.kube/config'

sleep 40

}

}

}

stage('Checking nodes of the cluster') {

when {

expression { params.action == 'create' }

}

steps {

script {

sh 'kubectl get nodes'

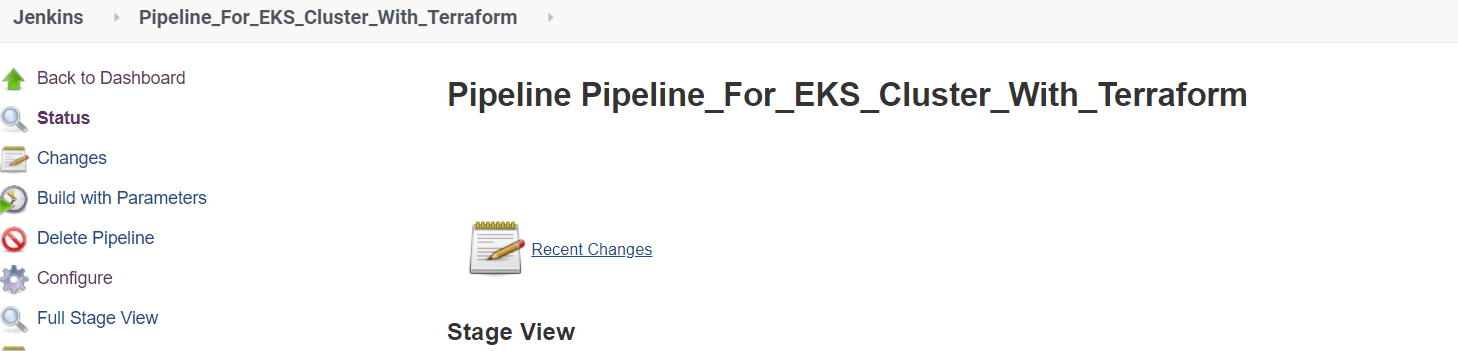
}

}

}

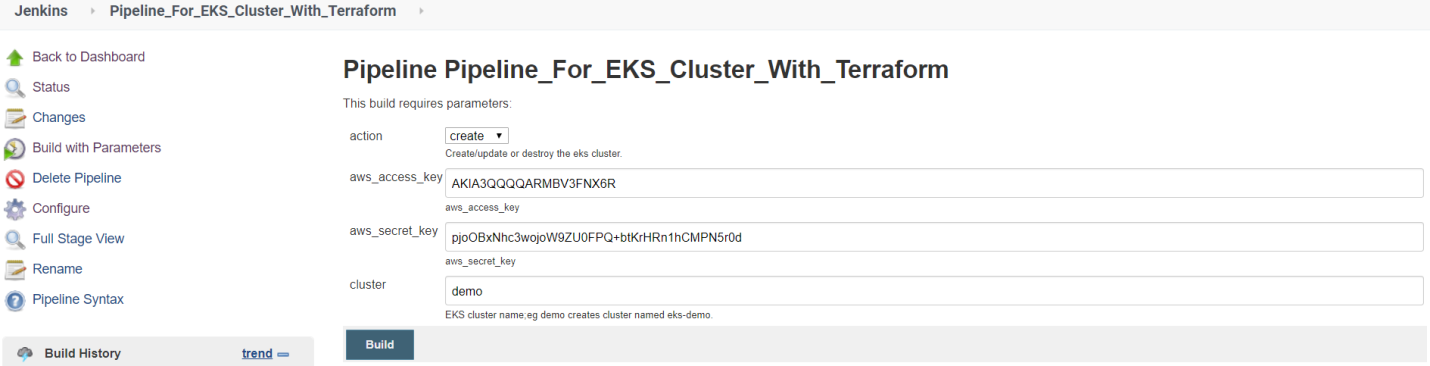
}

}

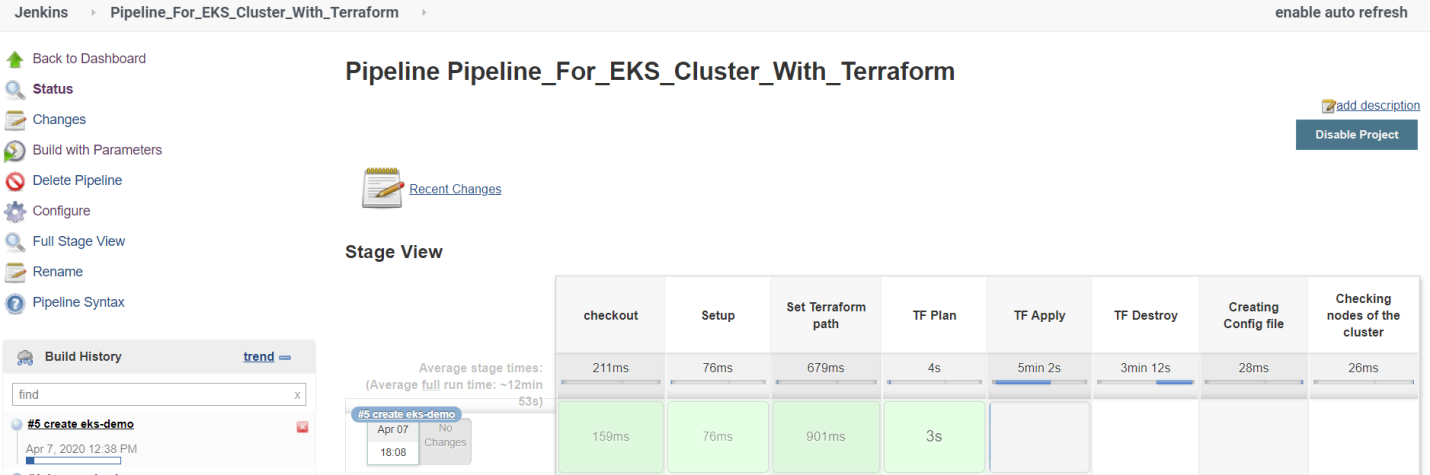


Click on **Build with Parameters**

Action**: Create**

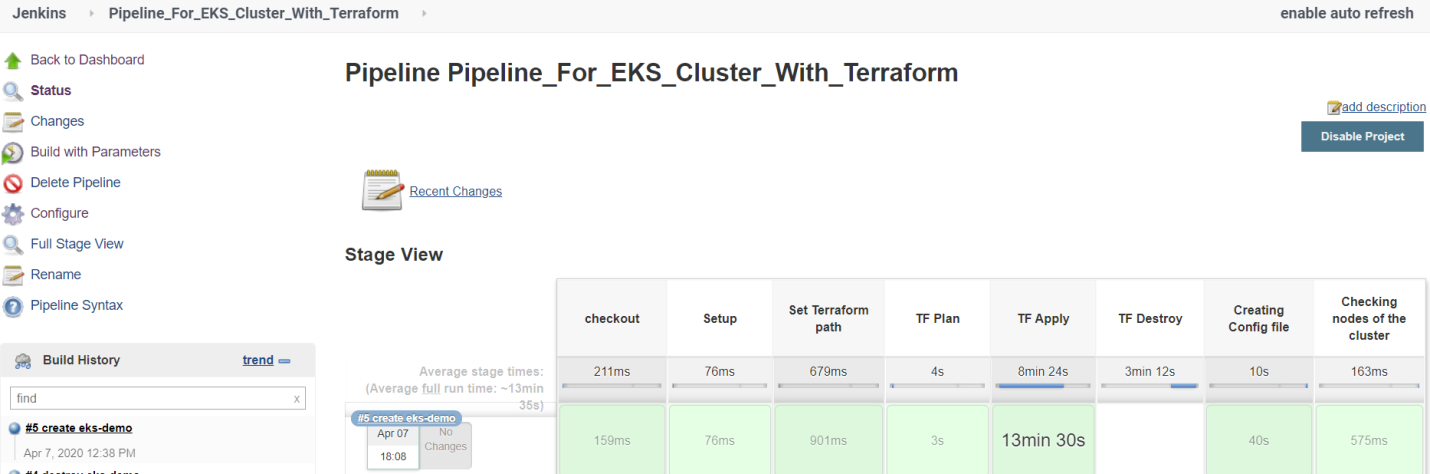


Click on **Build**

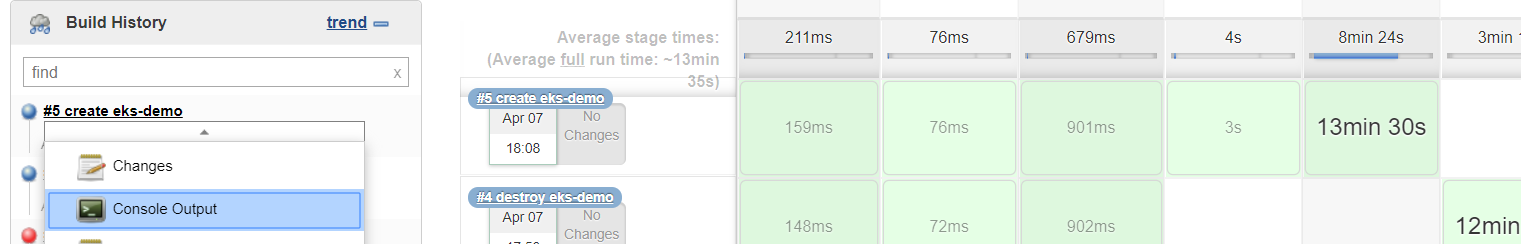


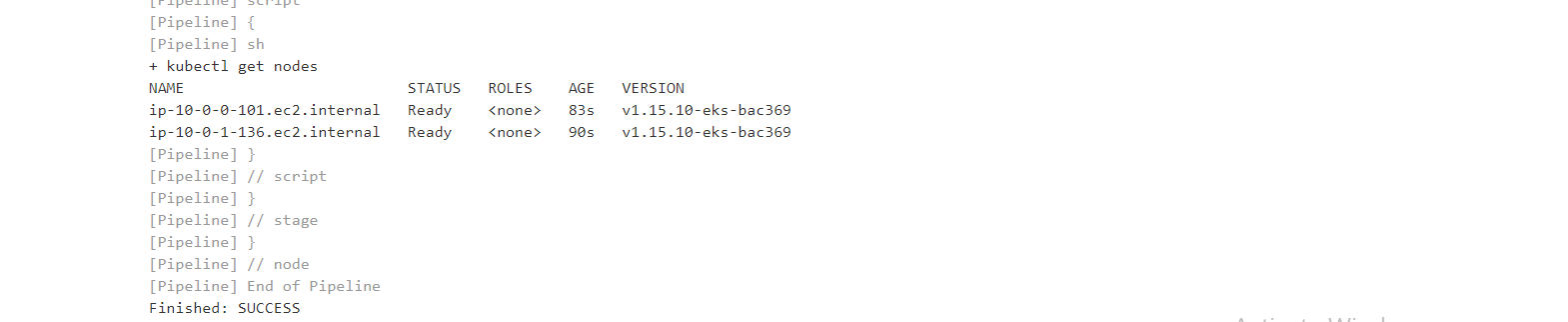
Process is going on

Please wait until it’s complete (It will take 15min to 20min)



Check **Console Output:**

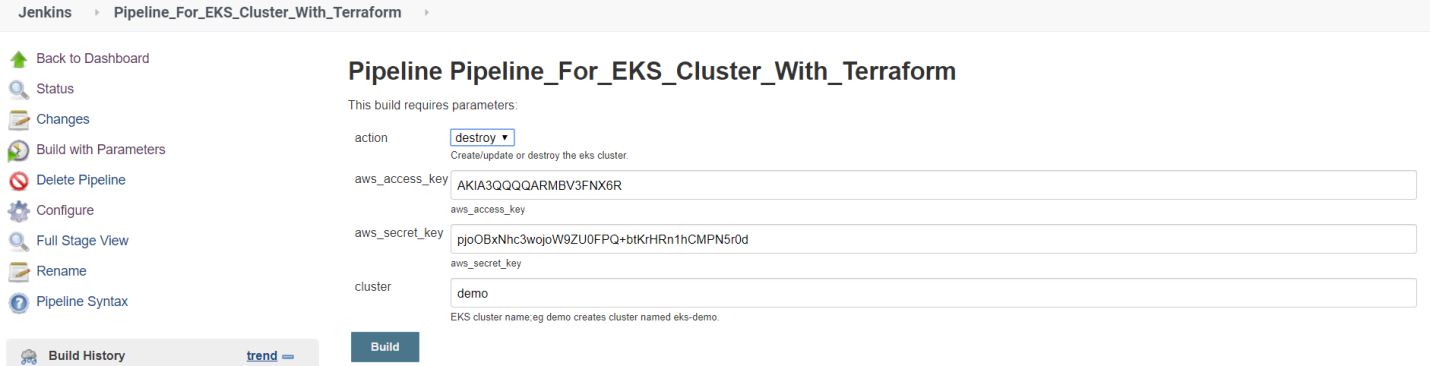
Click on **Console Output**



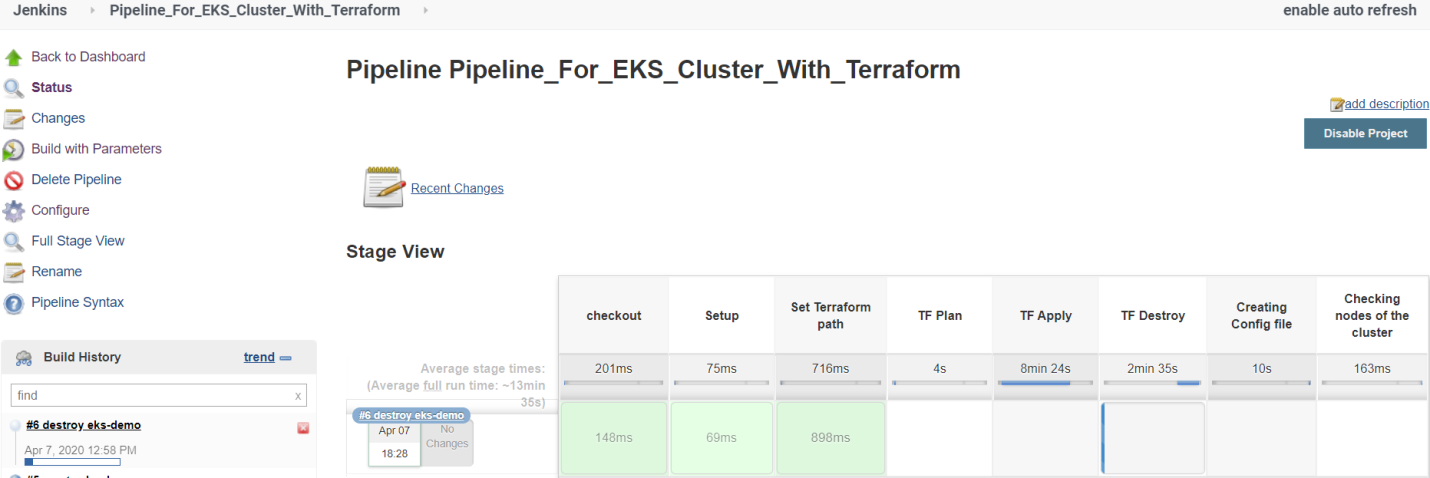
Action**: Destroy**

Go to Jenkins Job 🡪 Click on Build with Parameters

Select action as **Destroy**

****

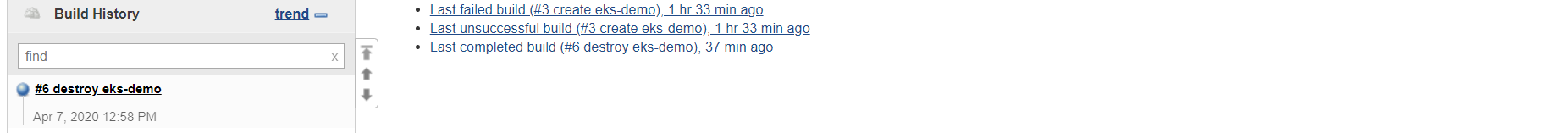
Click on **Build**



Process is going on

Please wait until it’s complete (It will take 15min to 20min)

Check **Console Output:**



Click on **Console Output**

