Jenkins job;

Click This project is parameterized

Action

Choose:
Build
Destroy

Build steps:

EXecute shell:

```
#!/bin/bash
sudo cp -r * /home/ubuntu/terraform/
cd /home/ubuntu/terraform/


# Check the value of the ACTION parameter and run either apply or destroy
if [ "$Action" == "Build" ]; then
# Initialize Terraform (this will set up the Terraform environment, download providers, etc.)
sudo terraform init
# Run the Terraform plan to see what will be created or destroyed
sudo terraform plan -out=tfplan
sudo terraform validate
      echo "Applying Terraform plan to build infrastructure..."

sudo terraform apply -auto-approve tfplan
    sudo aws eks update-kubeconfig --name my-eks-cluster
elif [ "$Action" == "Destroy" ]; then
      echo "Destroying infrastructure..."
    sudo terraform destroy -auto-approve
    echo "Removing jenkins workspace to free up the memory..."
    sudo rm -rf * /var/lib/jenkins/workspace/eks-cluster/
```

```
   sudo shutdown
else
    echo "Invalid action. Please choose either 'build' or 'destroy'."
    exit 1
fi
# Clean up the plan file
rm -f tfplan
```

Visudo

```
jenkins ALL=(ALL:ALL) NOPASSWD: ALL
```

Install terraform in instance

```
wget -O - https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
sudo apt update && sudo apt install terraform
```

Install Jenkins:

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
```

AWS CLI:

```
sudo apt-get install python3-pip
sudo pip install awscli
```

Give IAM Ec2Admin role to this instance.

Start build and create cluster

```
sudo apt install docker.io -y

sudo systemctl   enable docker
sudo systemctl status docker
sudo systemctl start docker
```

Aws cli install:

```
sudo apt install curl unzip
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

Aws configure: give secret access keys

Kubectl:

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sha256"

echo "$(cat kubectl.sha256)   kubectl" | sha256sum --check

chmod +x kubectl

sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

kubectl version --client
```

kubectl version --client --output=yaml. Detailed view

EksCtl:

```
sudo curl --silent --location
"https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$
(uname -s)_amd64.tar.gz" | sudo tar xz -C /usr/local/bin
```