Generated .pem file on aws console

And genearted publickey using git-bash

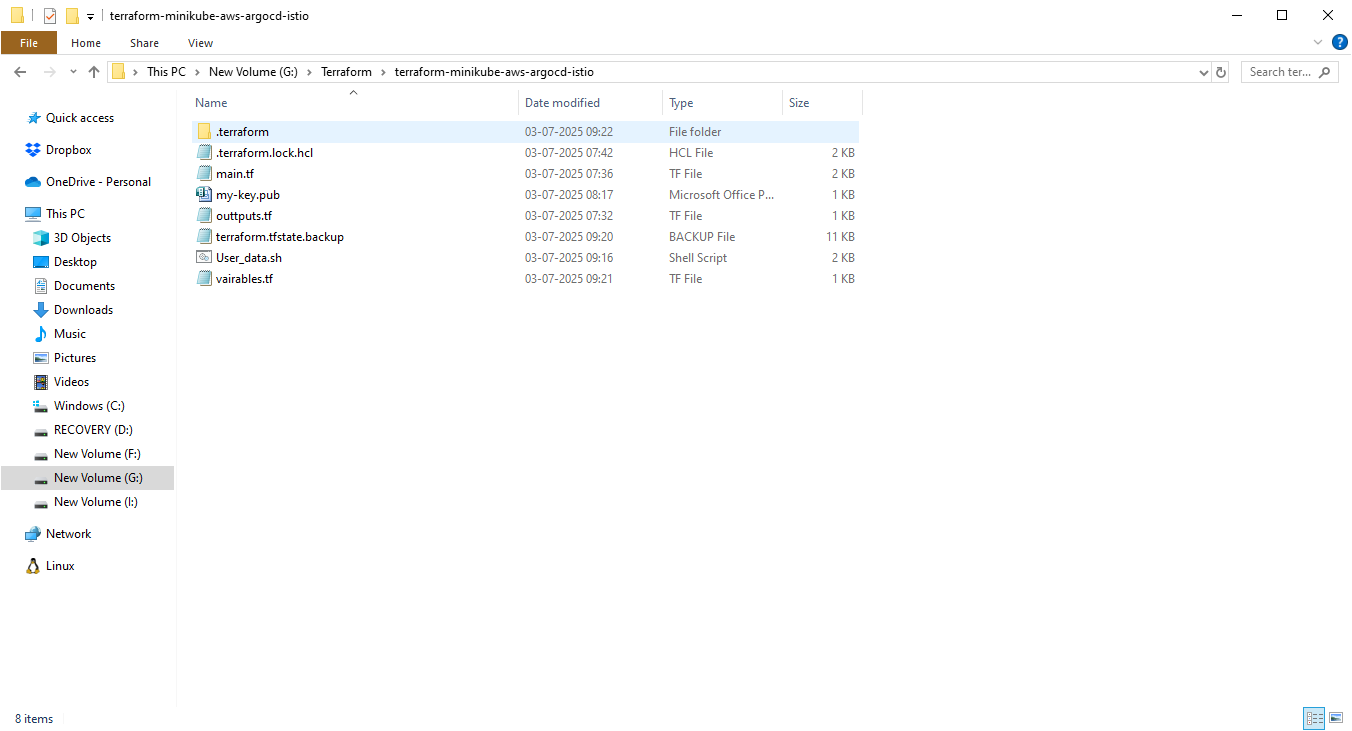
ssh-keygen -y -f G:\\Terraform\\minikube\_key.pem

Terraform folder

asrkr@srkr MINGW64 ~

$ ssh -i G:\\Terraform\\my-aws-key.pem [ubuntu@13.235.81.107](mailto:ubuntu@13.235.81.107)

Terraform Folder structure in local



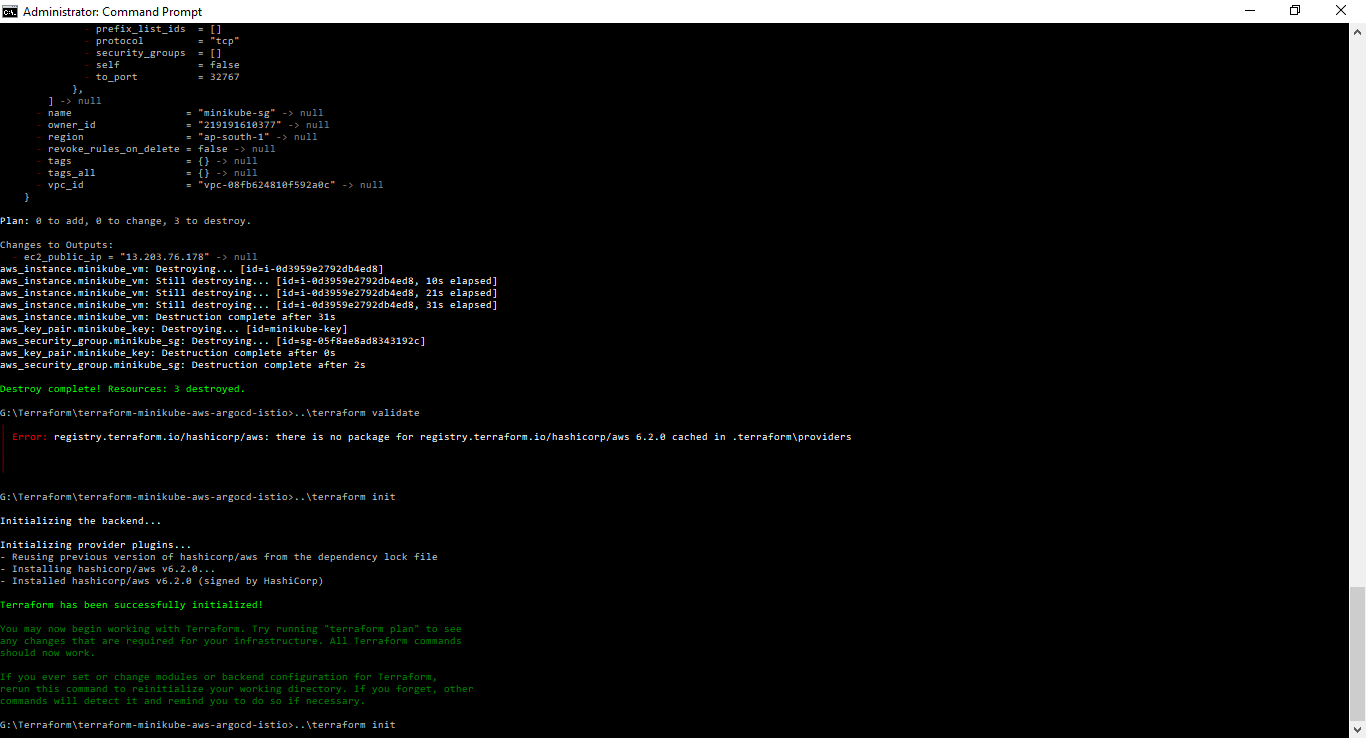
Terraform commands used as below and their ouputs

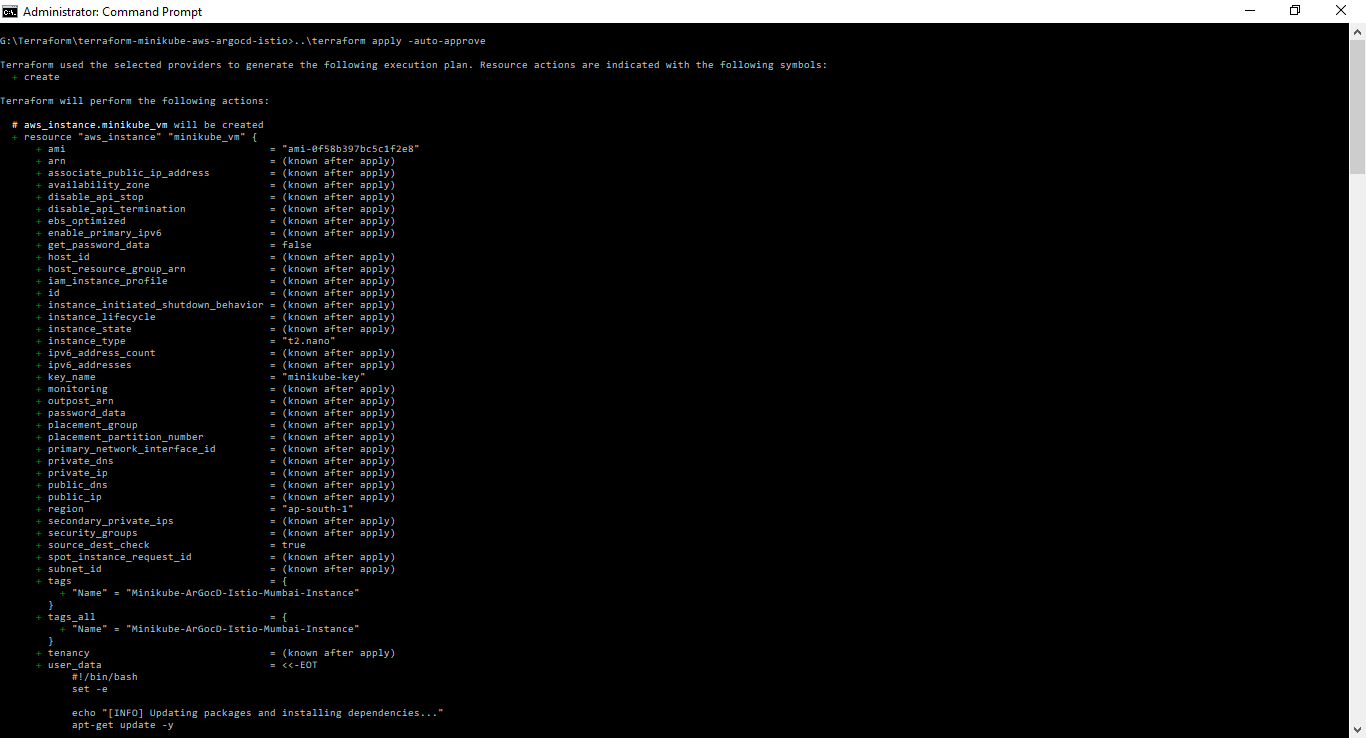
Terraform init

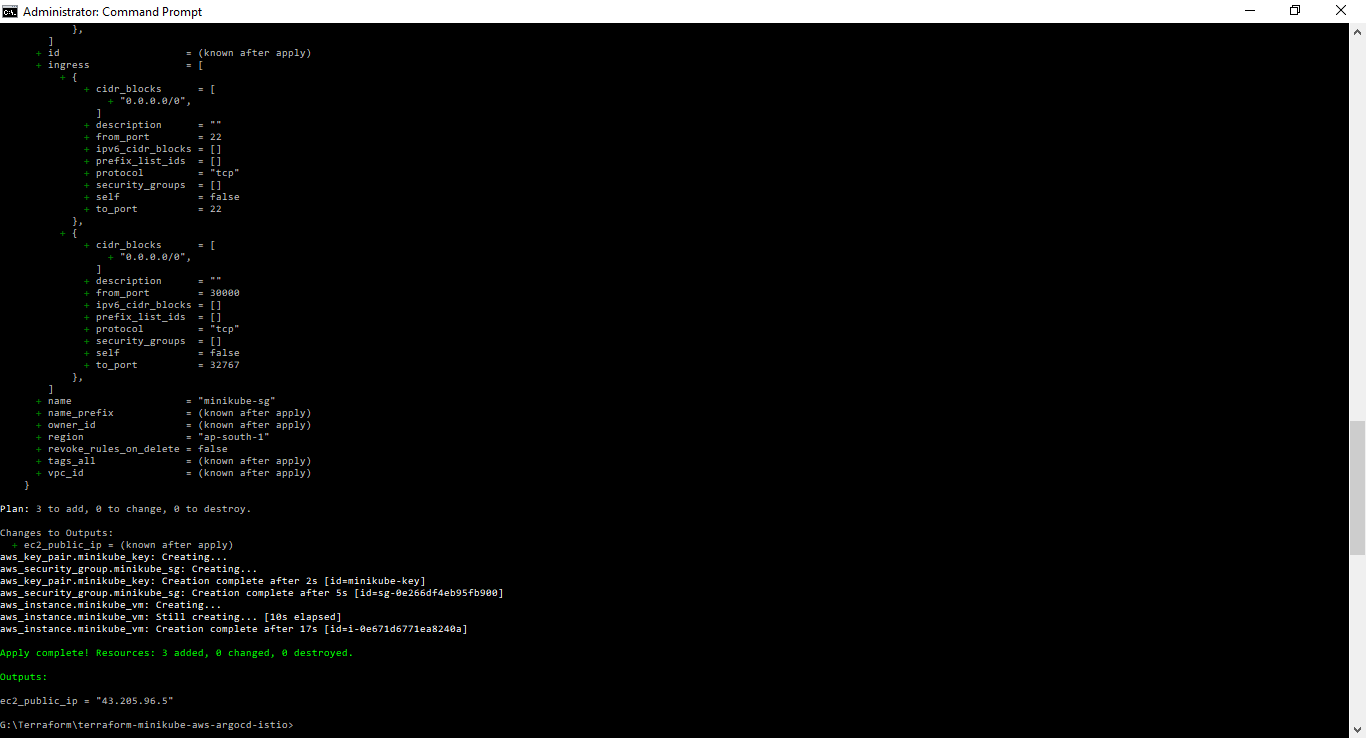
Terraform validate

Terraform Plan

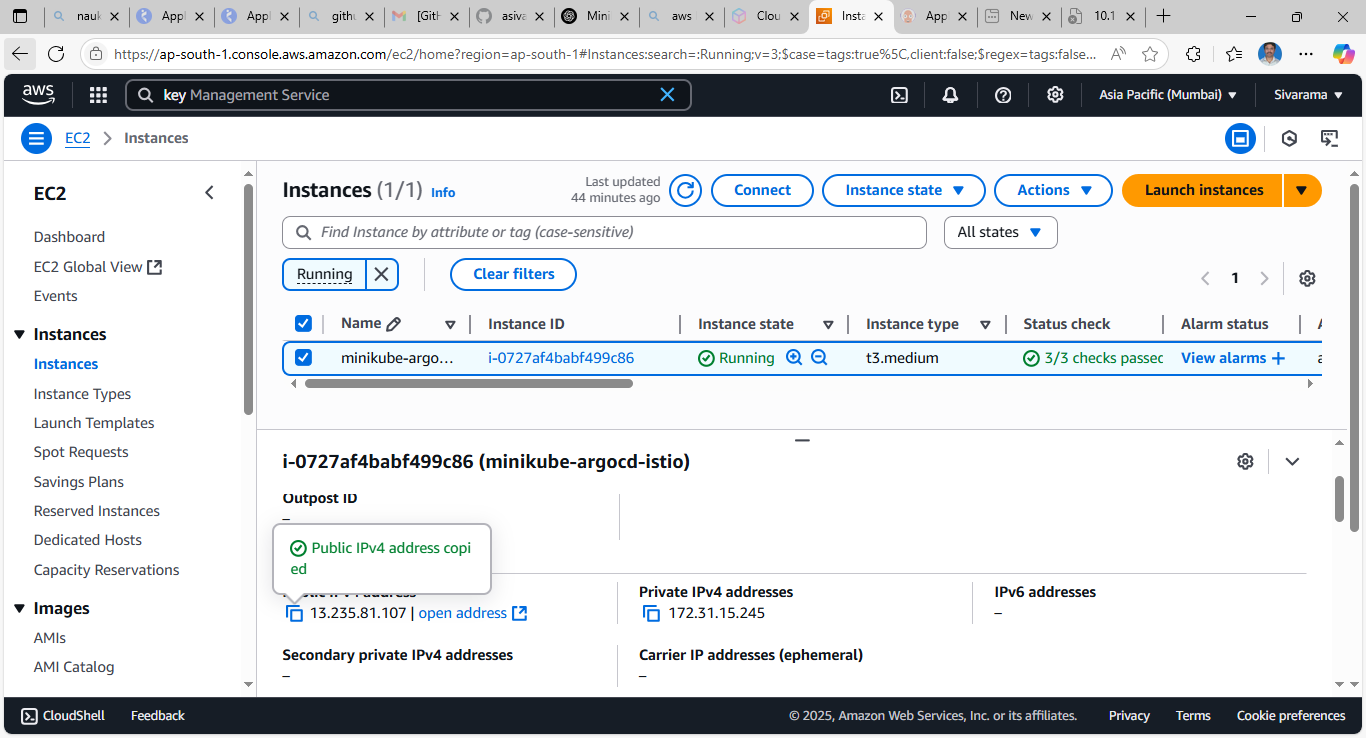
Terrafform apply

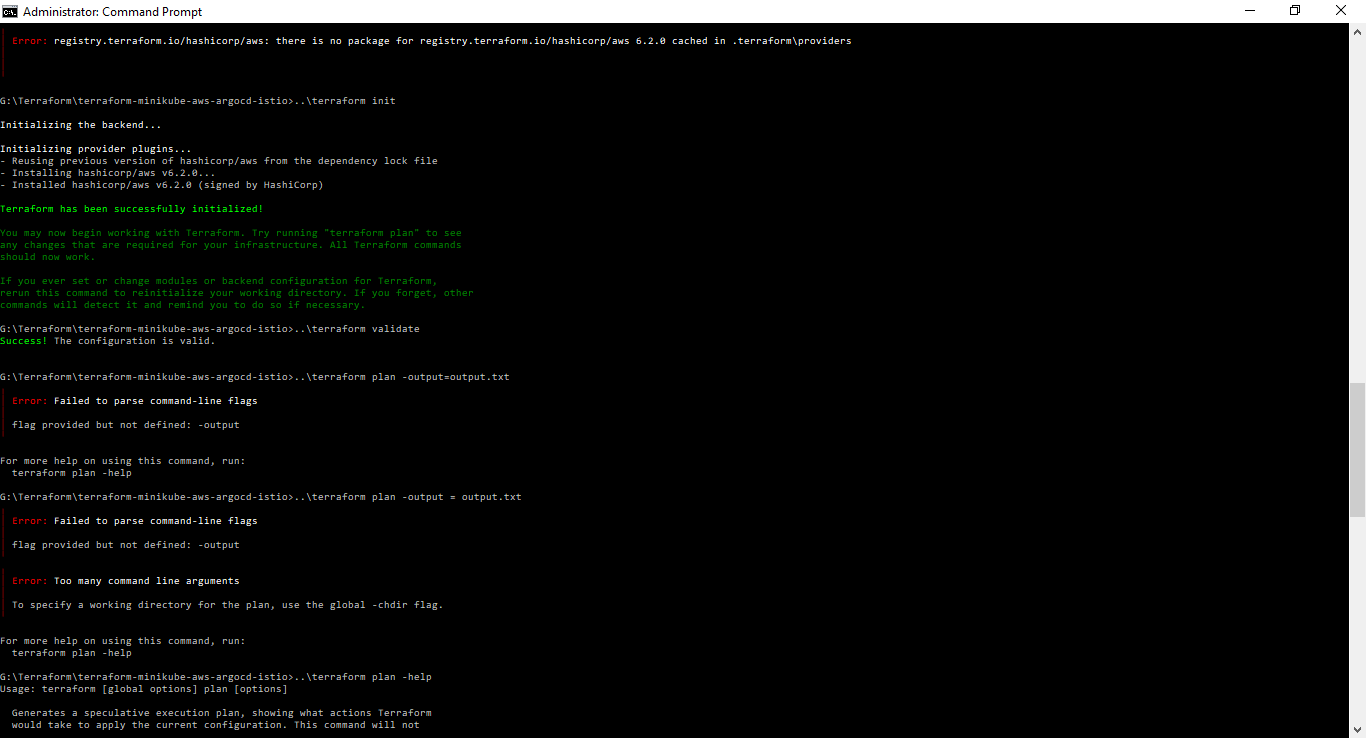


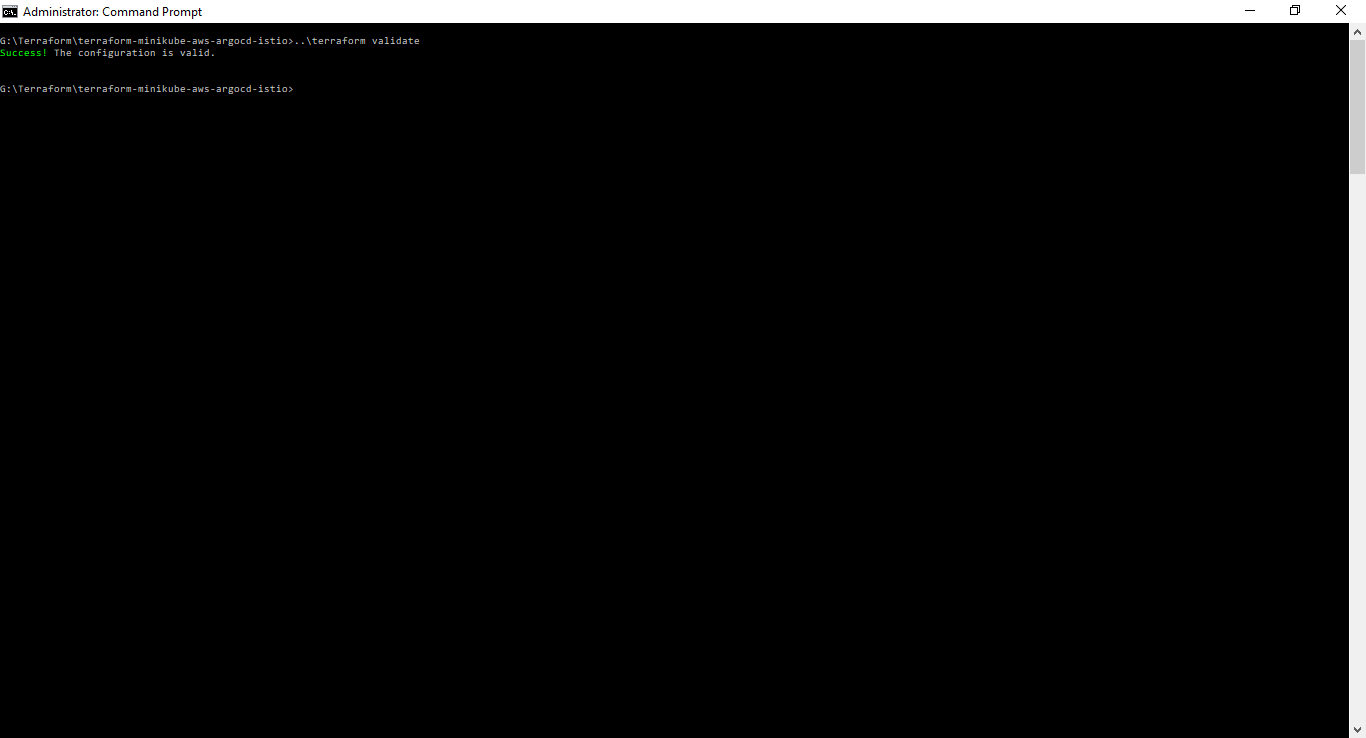


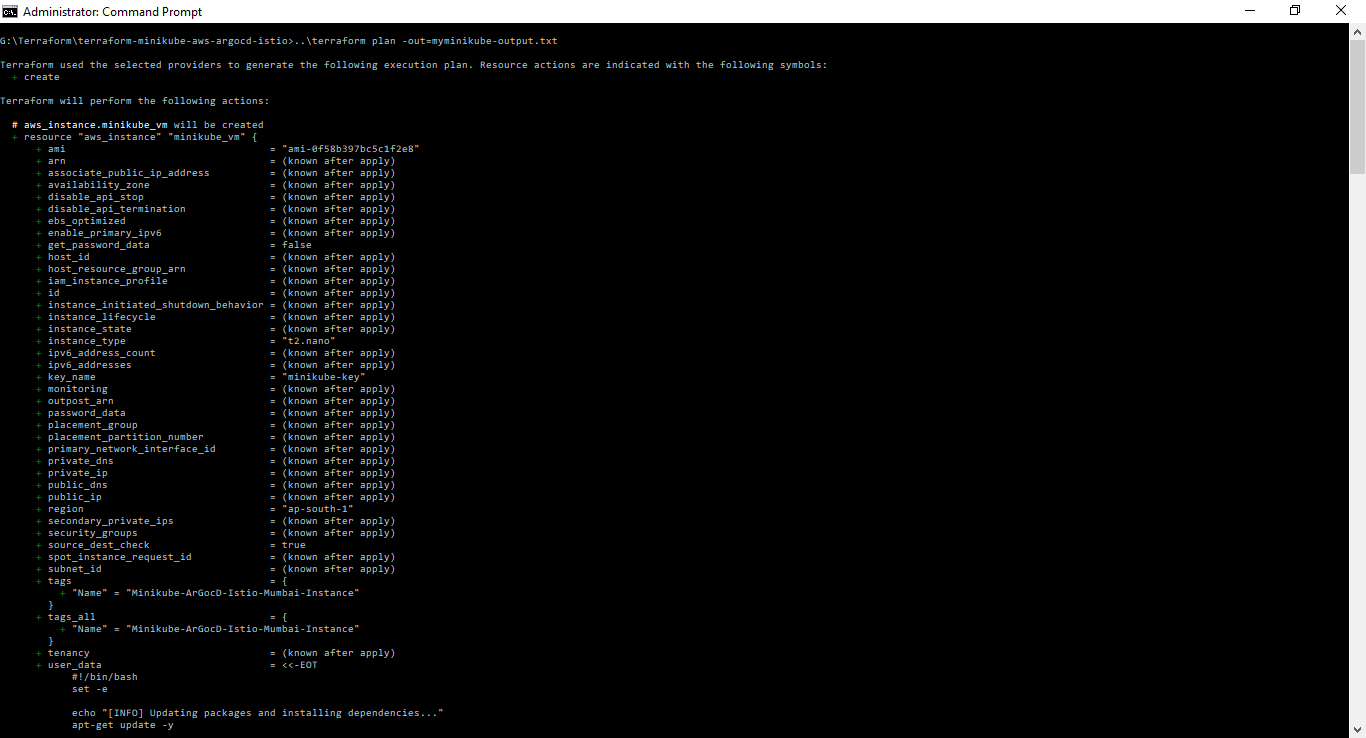


Minikube EC2 on AWS console











**Minikube + ArgoCD + Istio Setup on AWS EC2 Ubuntu**

**1. Access EC2 Instance**

bash

CopyEdit

ssh -i your-key.pem ubuntu@<EC2-public-ip>

**2. Install Docker**

* Update packages and install Docker:

bash

CopyEdit

sudo apt update

sudo apt install -y docker.io

* Start and enable Docker service:

bash

CopyEdit

sudo systemctl start docker

sudo systemctl enable docker

* Add ubuntu user to Docker group:

bash

CopyEdit

sudo usermod -aG docker ubuntu

* Re-login or run newgrp docker to activate group changes.

**3. Install Minikube**

* Download and install Minikube:

bash

CopyEdit

curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

sudo install minikube-linux-amd64 /usr/local/bin/minikube

**4. Install kubectl**

* Download, make executable, and move kubectl binary:

bash

CopyEdit

curl -LO https://dl.k8s.io/release/v1.33.1/bin/linux/amd64/kubectl

chmod +x kubectl

sudo mv kubectl /usr/local/bin/

kubectl version --client

**5. Install Minikube Dependencies for none Driver**

* **Install cri-dockerd:**

bash

CopyEdit

sudo apt install -y git golang-go

git clone https://github.com/Mirantis/cri-dockerd.git

cd cri-dockerd

mkdir bin

go build -o bin/cri-dockerd

sudo cp bin/cri-dockerd /usr/local/bin/

sudo systemctl daemon-reload

sudo systemctl enable cri-docker.service

sudo systemctl start cri-docker.service

* **Install Container Network Interface (CNI) plugins:**

bash

CopyEdit

curl -LO https://github.com/containernetworking/plugins/releases/download/v1.1.1/cni-plugins-linux-amd64-v1.1.1.tgz

sudo mkdir -p /opt/cni/bin

sudo tar -xzvf cni-plugins-linux-amd64-v1.1.1.tgz -C /opt/cni/bin

* **Install socat for port forwarding:**

bash

CopyEdit

sudo apt install -y socat

**6. Start Minikube**

bash

CopyEdit

minikube start --driver=none

* Verify status:

bash

CopyEdit

minikube status

* Check nodes and pods:

bash

CopyEdit

kubectl get nodes

kubectl get pods -A

**7. Deploy ArgoCD**

* Create ArgoCD namespace:

bash

CopyEdit

kubectl create namespace argocd

* Install ArgoCD:

bash

CopyEdit

kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml

* Check ArgoCD pods:

bash

CopyEdit

kubectl get pods -n argocd

* Expose ArgoCD server (with port forwarding):

bash

CopyEdit

kubectl port-forward --address 0.0.0.0 svc/argocd-server -n argocd 8080:443

* Access ArgoCD UI via browser:

cpp

CopyEdit

http://<EC2-public-ip>:8080

* Retrieve ArgoCD initial admin password:

bash

CopyEdit

kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d; echo

**8. Install Istio**

* Download Istio CLI and add to PATH:

bash

CopyEdit

curl -L https://istio.io/downloadIstio | ISTIO\_VERSION=1.22.0 sh -

export PATH="$PATH:$HOME/istio-1.22.0/bin"

* Run pre-installation check:

bash

CopyEdit

istioctl x precheck

* Install Istio demo profile:

bash

CopyEdit

istioctl install --set profile=demo -y

* Verify Istio pods and services:

bash

CopyEdit

kubectl get pods -n istio-system

kubectl get svc -n istio-system

**9. Deploy Sample Application (Bookinfo)**

bash

CopyEdit

kubectl apply -f $HOME/istio-1.22.0/samples/bookinfo/platform/kube/bookinfo.yaml

kubectl apply -f $HOME/istio-1.22.0/samples/bookinfo/networking/bookinfo-gateway.yaml

* Check Bookinfo pods:

bash

CopyEdit

kubectl get pods

**10. Access Sample Application**

* Get Istio Ingress Gateway service details:

bash

CopyEdit

kubectl -n istio-system get svc istio-ingressgateway

* Note the NodePort exposed (e.g., port 30221 on port 80)
* Access app in browser:

php-template

CopyEdit

http://<EC2-public-ip>:<NodePort>/productpage

