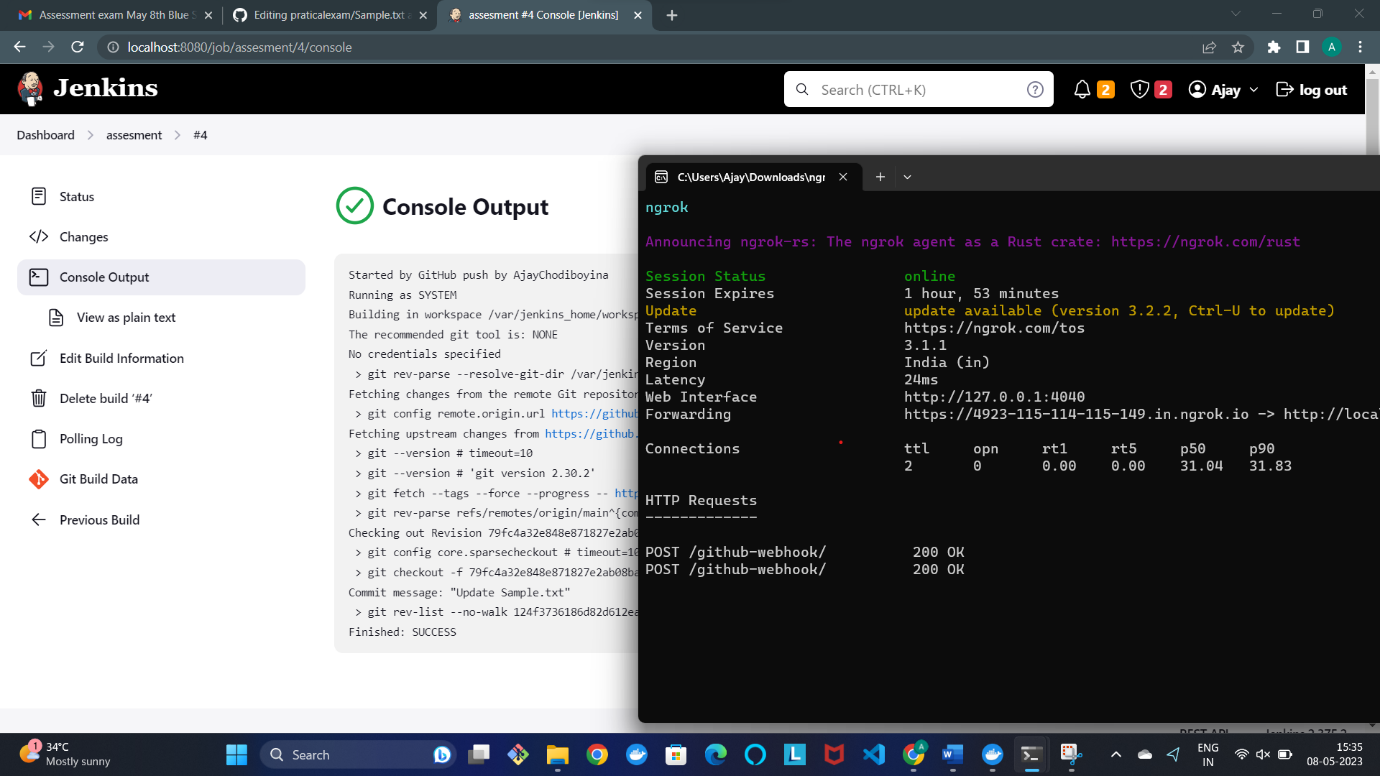
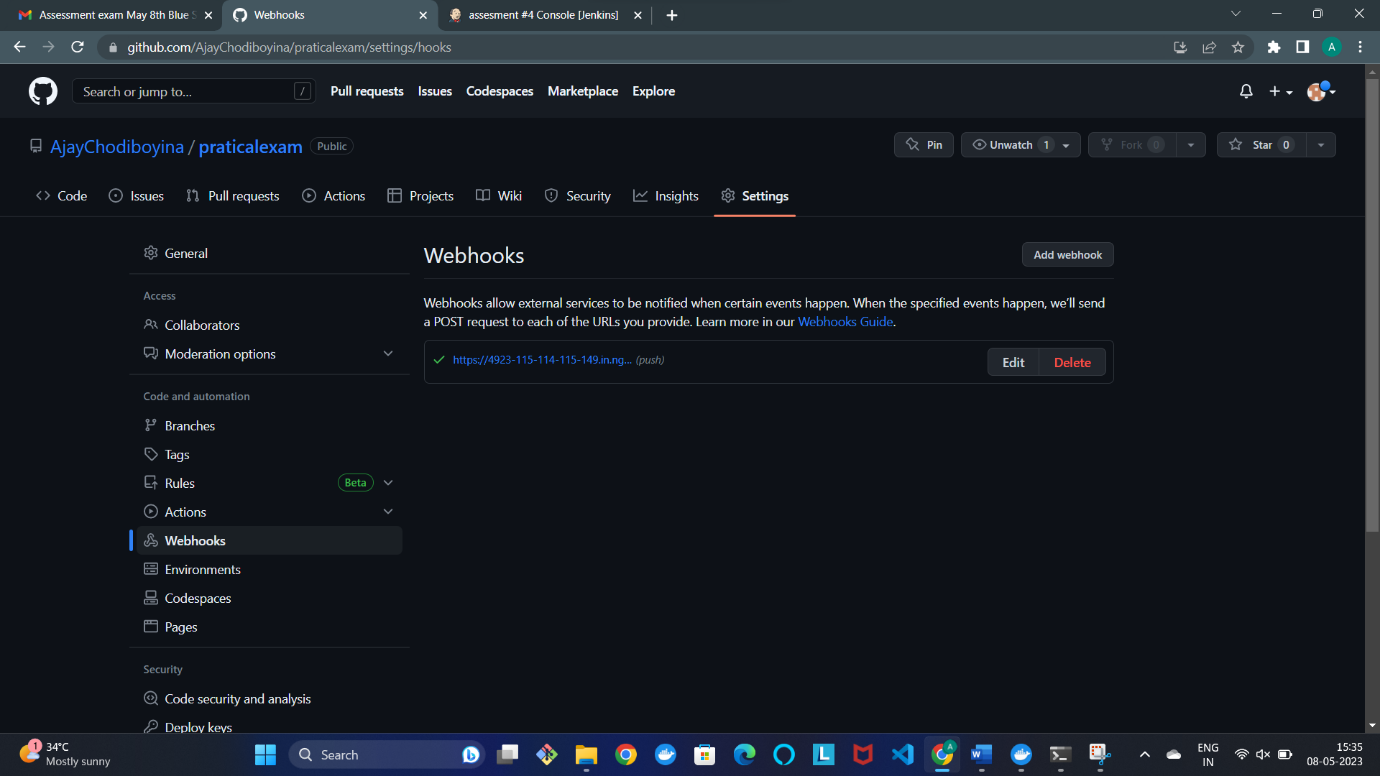
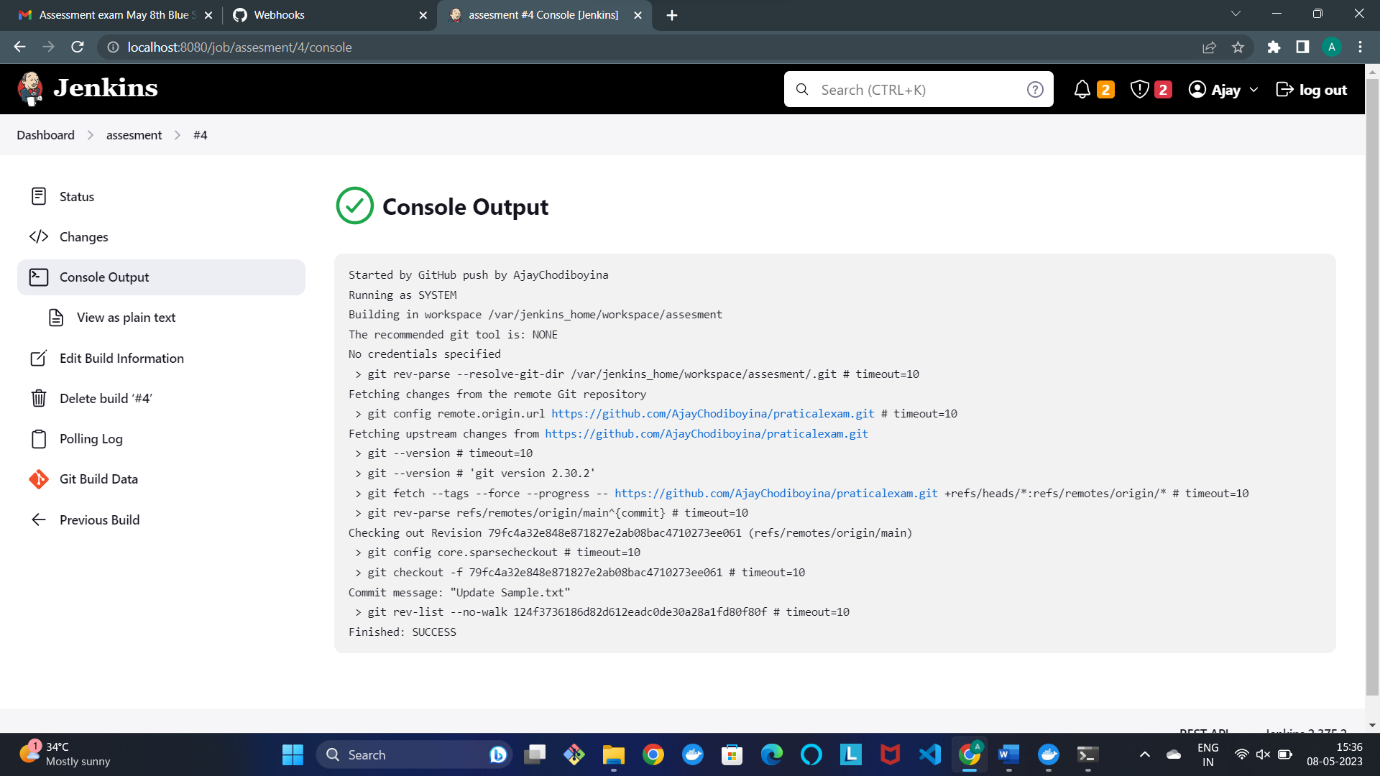
QUESTION-1:::::

Steps :

* Create a new git repo and clone it into your local system
* Create a file and add,push,commit into the git repo
* Open ngrok and give the command
  + Ngrok http 8080
  + Copy the ip address
* Open settings click on webhook in repo settings
* Add the url copied from ngrok and the webhook will be created
* Now, we need to run a jenkins container
* Open localhost:8080
* Create a new item
  + Freestyle job
  + As the webhook is implemented the build will happen automatically







**::::::::::Question 2::::::::::**

* Create a new item in jenkins
  + Select a pipline
  + Add Pipeline script

pipeline {

agent none

stages {

stage('bitbucket') {

steps {

script {

echo "Will store the code developed in bitbucket"

}

}

}

stage('jenkins') {

steps {

script {

echo "will build and package the code in jenkins using maven by implementing webhook "

}

}

}

stage('sonarkube') {

steps {

script {

echo "will check the code quality in sonarkube "

}

}

}

stage('test') {

steps {

script {

echo "will test the code"

}

}

}

stage('staging') {

steps {

script {

echo "code will be in staging area"

}

}

}

stage('push to nexus') {

steps {

script {

echo "push the image into nexus repo"

}

}

}

stage('deploy in openshift') {

steps {

script {

echo "deploy the images from nexus repo to openshift based on aws"

}

}

}

stage('datadog') {

steps {

script {

echo "implemet in datadog"

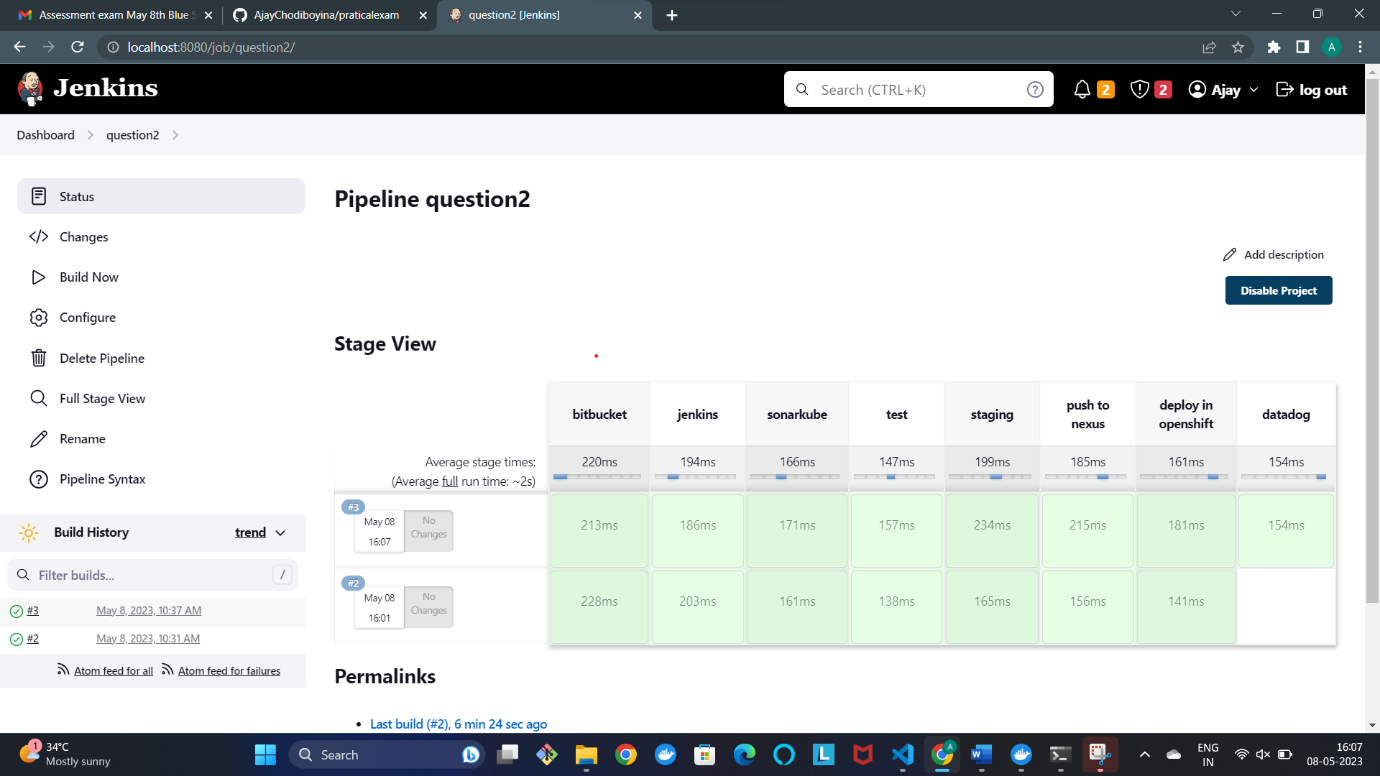
}

}

}

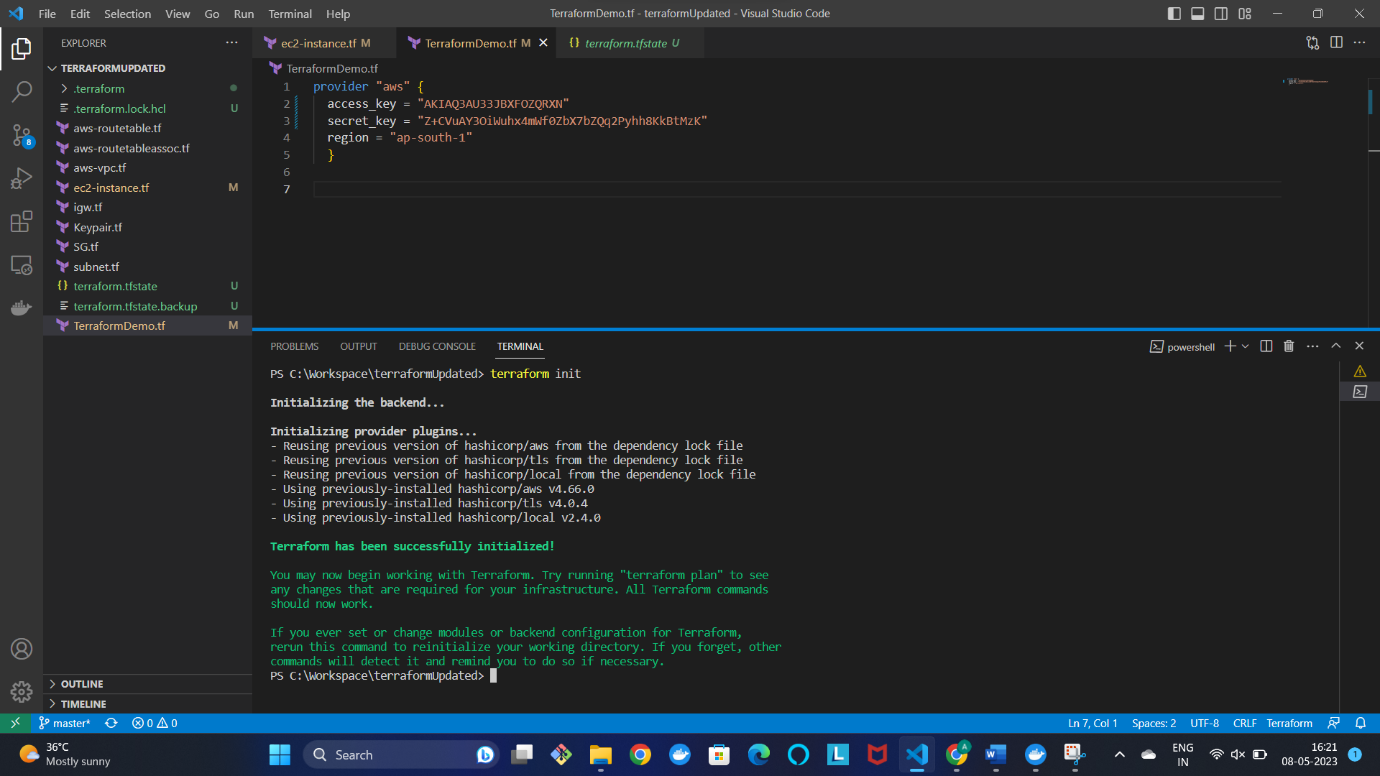
}

}

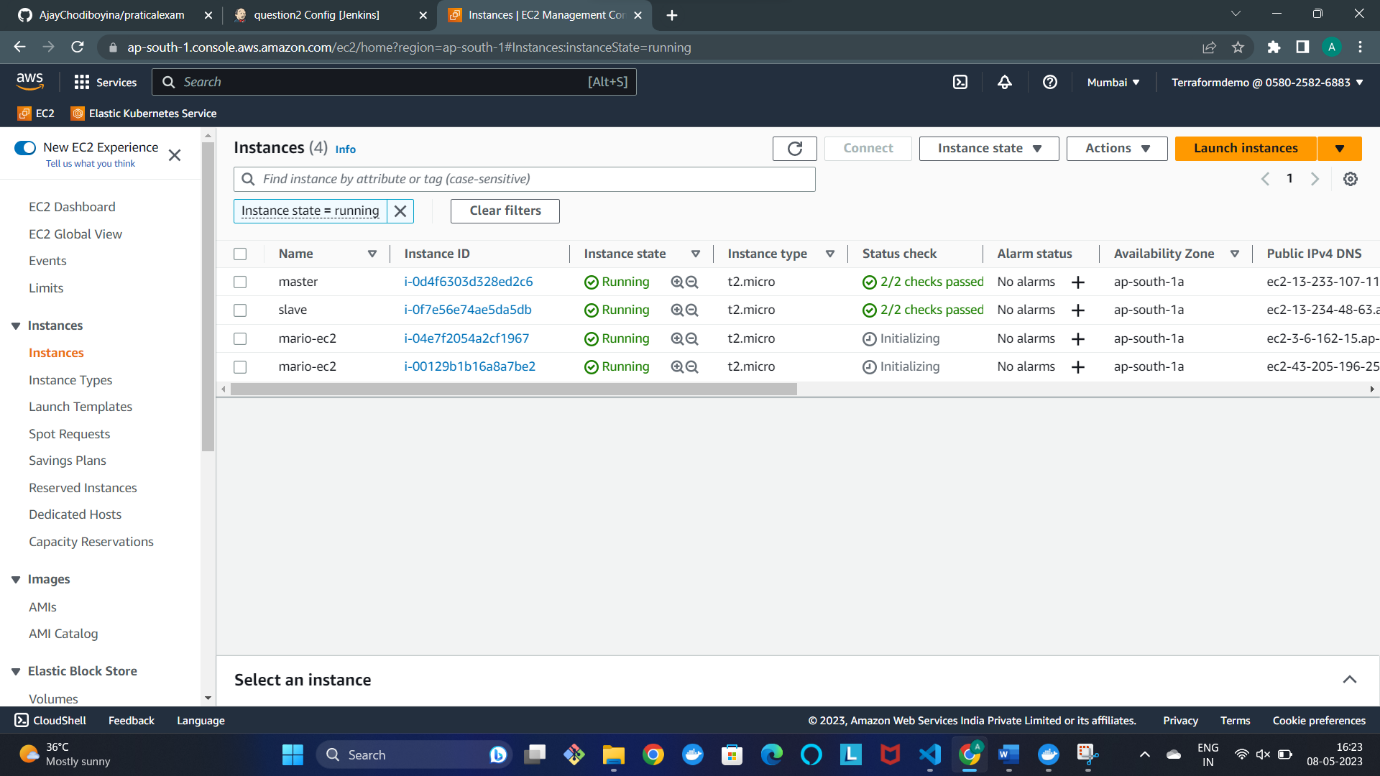


**::::::::Question 4::::::::**

* First we need to open the folder where our terraform file is located and open the terminal over there
* We need to change the access\_key, secret\_key and place your own keys
* There we need to give the commands
  + Terraform init
  + Terraform plan
  + Terraform apply

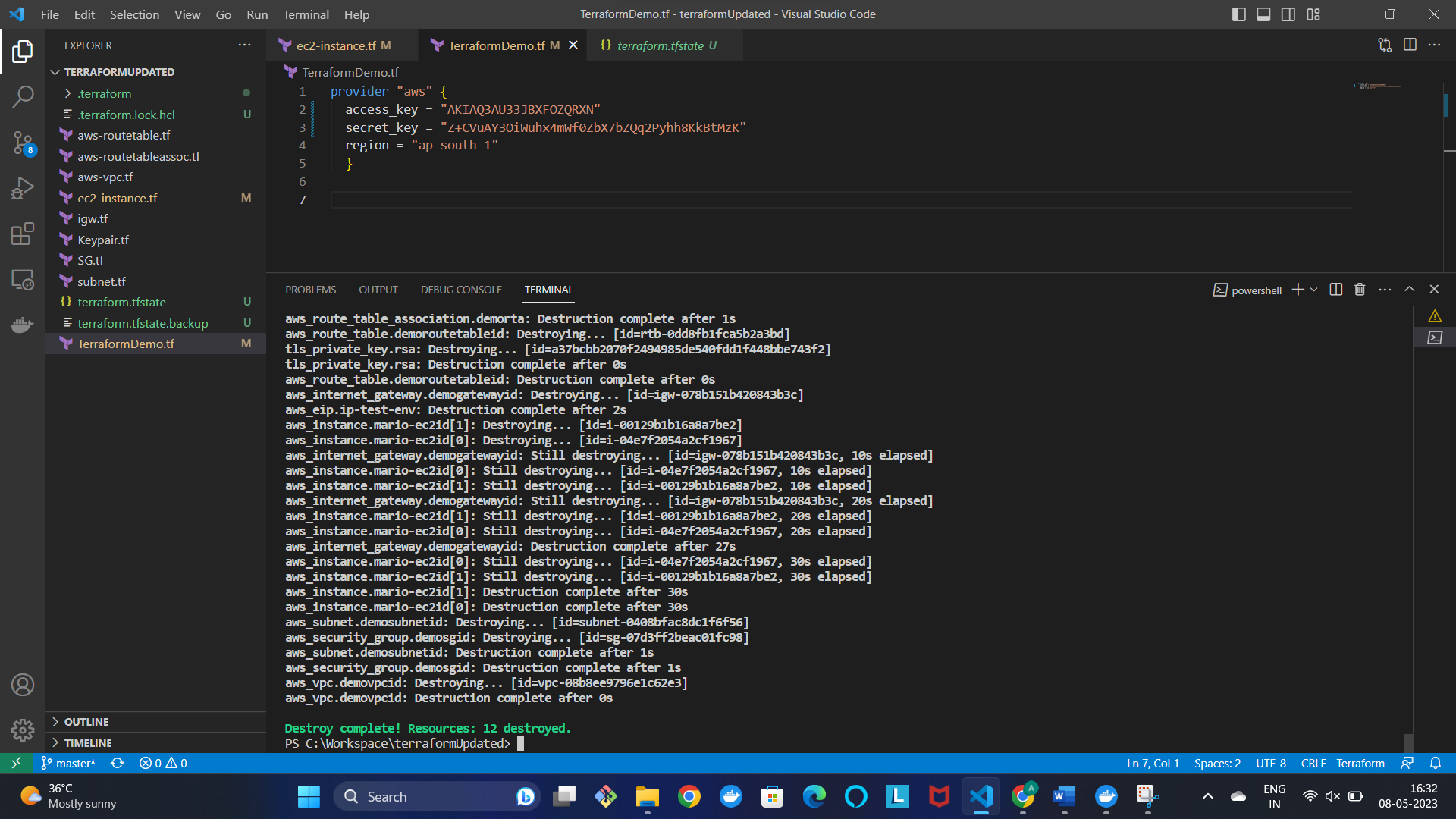
****

**Instances are created in our aws account using the keys provided**

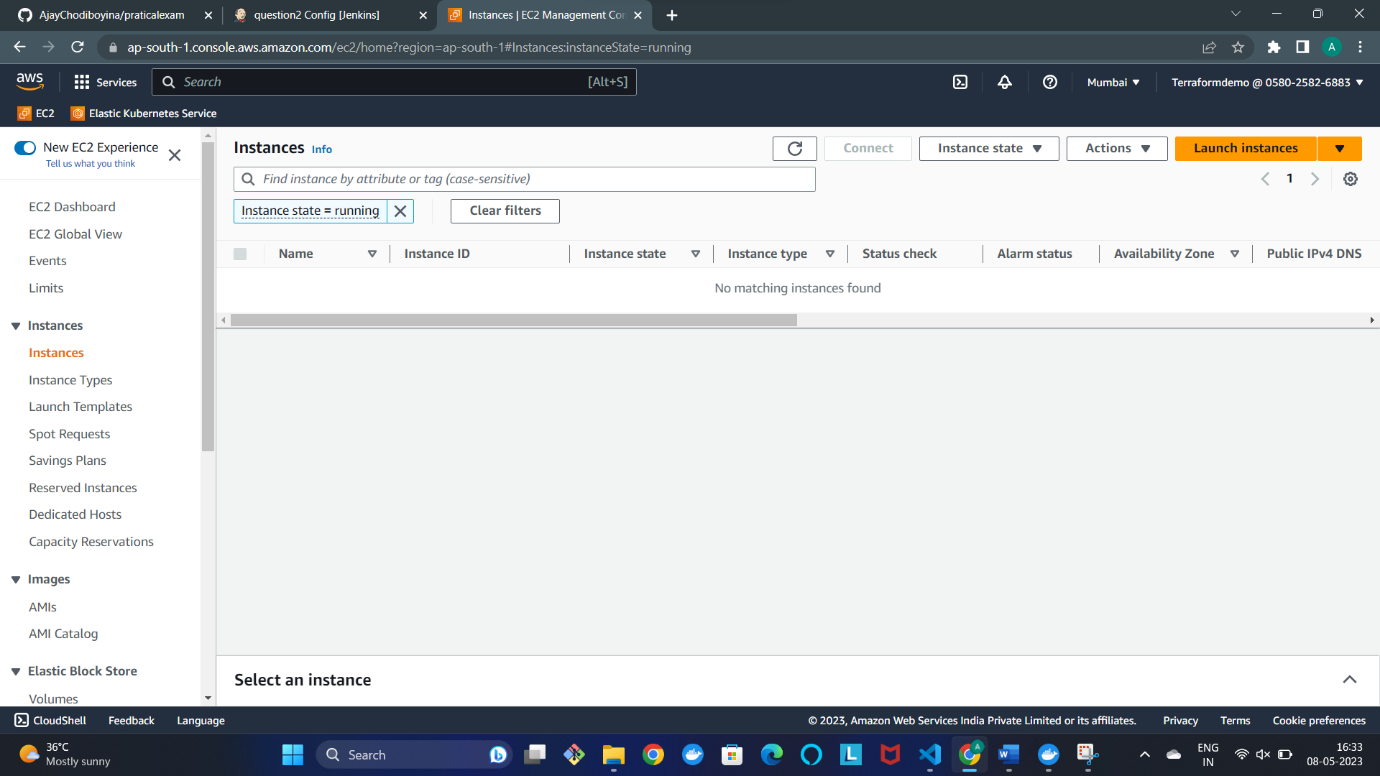
****

**Destroy:::::**

* Use command
  + Terraform destroy



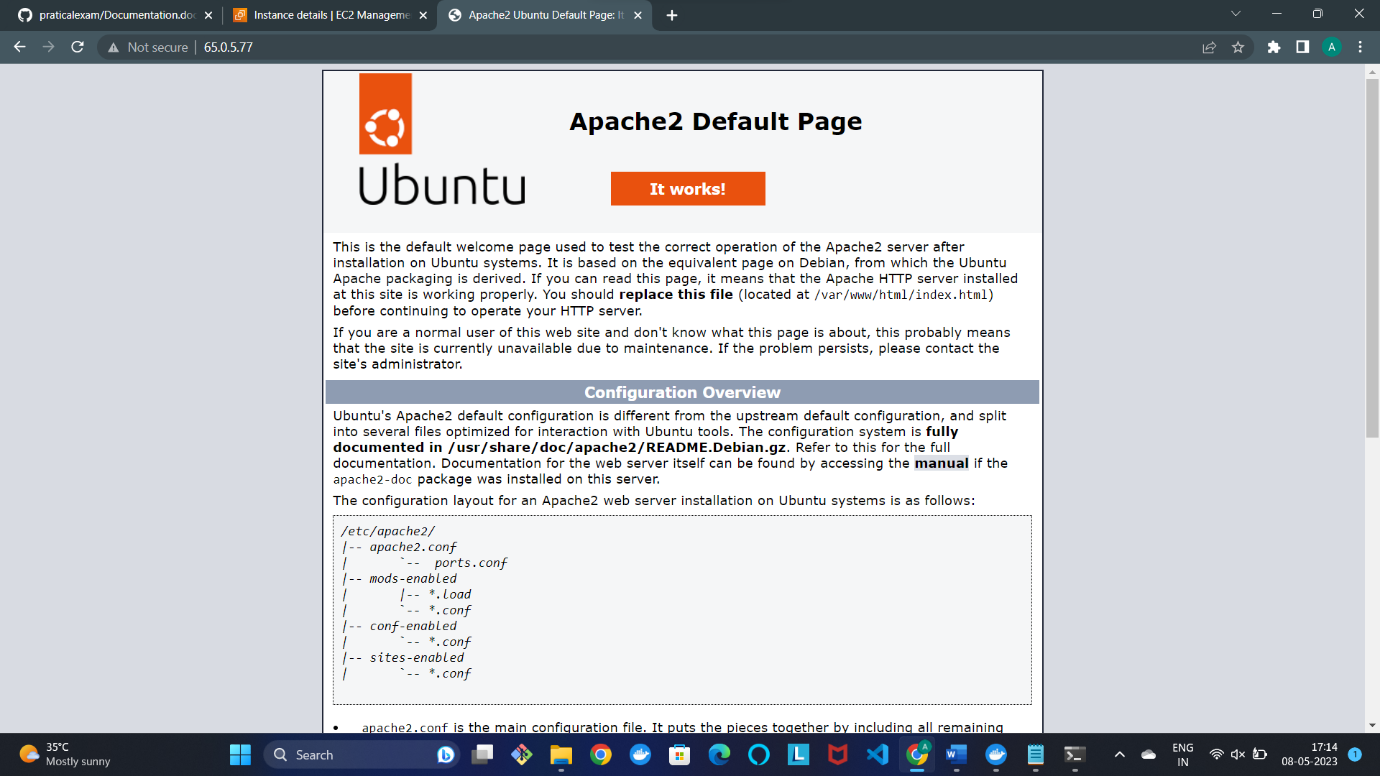
Destroyed in aws

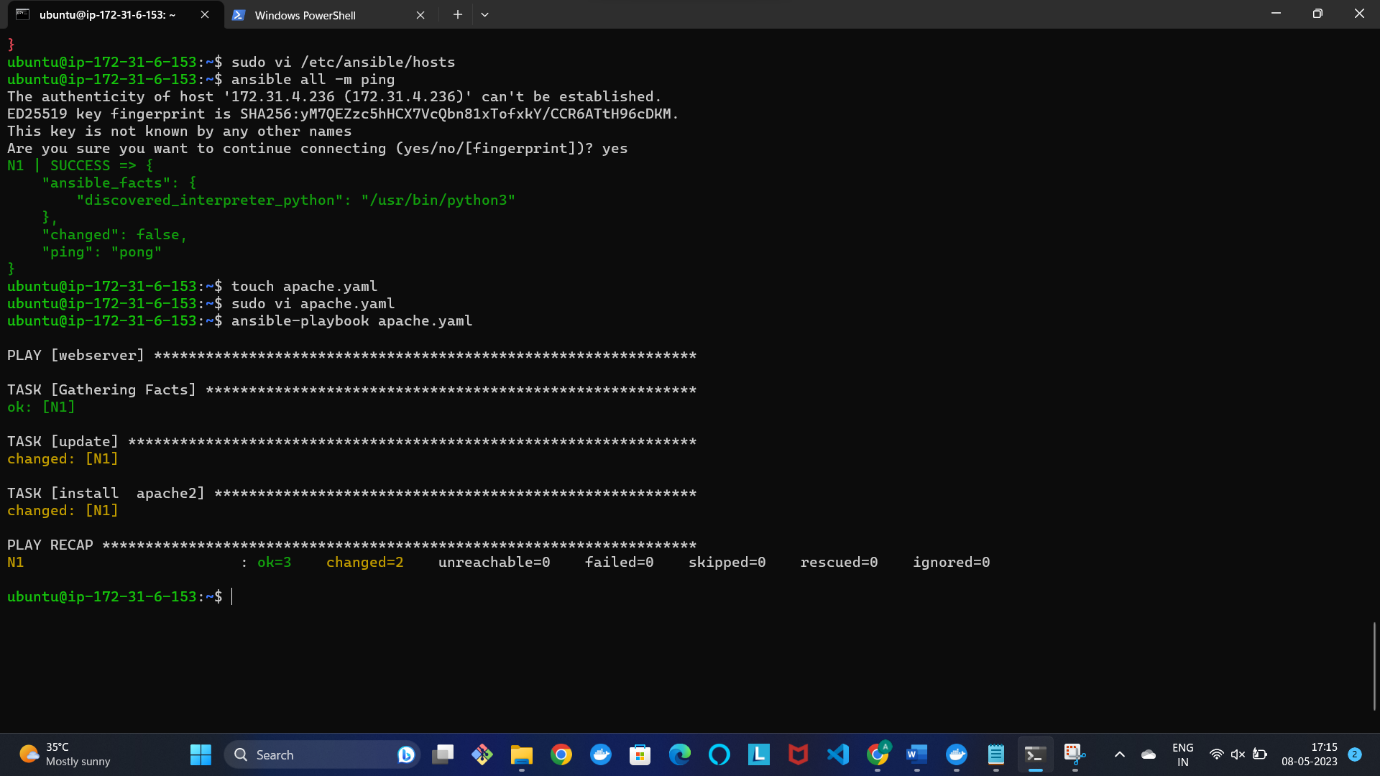


**:::::::;Question\_5:::::::**

* First we need to create two instances in aws using UI
  + Name them as master and slave
  + Click on master and press on connect we will get a ssh key
  + Open the terminal where the .pem file is downloaded and paste the commad

ssh -i "testexam.pem" [ubuntu@ec2-15-206-210-216.ap-south-1.compute.amazonaws.com](mailto:ubuntu@ec2-15-206-210-216.ap-south-1.compute.amazonaws.com)

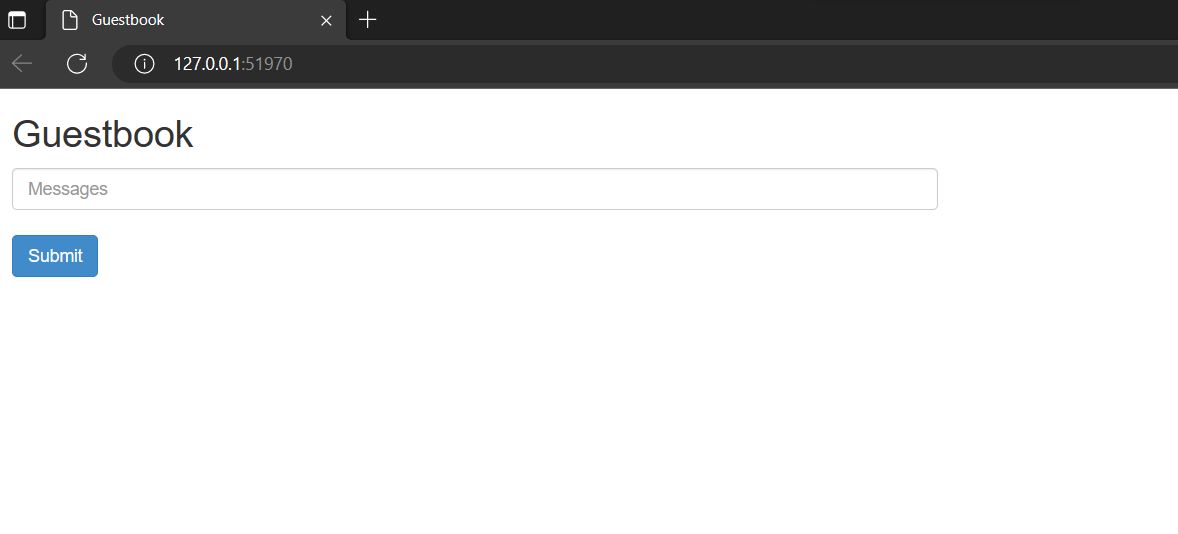
* Now we are connected
  + Now we need to follow these commands to
  + cat /etc/issue
  + sudo apt-add-repository ppa:ansible/ansible
  + sudo apt update
  + sudo apt install ansible
  + scp -i ct.pem ct.pem ubuntu@ec2-13-235-246-27.ap-south-1.compute.amazonaws.com:~
  + pwd
  + ls -l
  + chmod 400 ct.pem
  + cat /etc/ansible/hosts
  + sudo vi /etc/ansible/hosts
  + [webserver]
  + N1 ansible\_host=172.31.41.208 ansible\_user=ubuntu ansible\_private\_key\_file=/home/ubuntu/testexam.pem
  + [webserver]
  + N1 ansible\_host=172.31.43.247 ansible\_user=ubuntu ansible\_private\_key\_file=/home/ubuntu/testexam.pem
  + ansible all -m ping
  + touch apache.yaml
  + sudo vi apache.yaml
  + ansible-playbook apache.yaml
* This will help us install ansible in master node and manage the slave node and install apache in slave node

****

**Question-6::::**

* First we need open the folder where the guestbook yaml files are located
  + Open terminal in that folder
* Now give the commad
  + Minikube start
  + Next we need to apply the yaml files as show below i
* kubectl create ns guestbook
* kubectl apply -n guestbook frontend.yaml
* kubectl apply -n guestbook frontendservice.yaml
* kubectl apply -n guestbook -f redis-follower.yaml
* kubectl get pods –n guestbook
* kubectl apply –n guestbook –f redis-followerservice.yaml
* kubectl apply -n guestbook -f redisleaderdeployment.yaml
* kubectl apply -n guestbook -f redisleaderservice.yaml
* kubectl get pods –n guestbook
* kubectl get service -n guestbook
* minikube service -n guestbook

Now we can see the guestbook home page

****

**Question3::::::**

* First we need to create a index.html and dockerfile in one folder
* No weneed to give commads
* **::::::::docker build and push:::::**
* docker build -t ajaychodiboyina/demo1:sampleproject . (DOT)
* docker push ajaychodiboyina/demo1:sampleproject
* **:::::::Deployming Created image in Kubernetes:::::::::**
* kubectl create deployment praticedemo(Name) --image=ajaychodiboyina/demo1:sampleproject
* kubectl expose deployment praticedemo(Name) --type="NodePort" --port 80
* minikube service mycalcapp1
* Kubectl get deployments
* kubectl get pods
* kubectl get deployments
* kubectl scale --replicas=2 deployment first-deployment
* **:::::::: installing datadog in minikube ::::::::::**
* helm repo add datadog https://helm.datadoghq.com
* helm repo update
* helm install dataproject --set datadog.site='us5.datadoghq.com' --set datadog.apiKey='ca7126172da7a869f43b496baf59d0ad' datadog/datadog
* kubectl get deployments