**BlueSpire Assessment**

**1. Deploy a Jenkins container and create a job to demonstrate Continuous**

**Integration.**

**Step 1**: Create a git repository named:BlueSpireAssessment\_Srilakshmich

**Step 2**: Clone the repository to the workspace with below command

git clone https://github.com/Srilakshmicheru/BlueSpireAssessment\_Srilakshmich.git

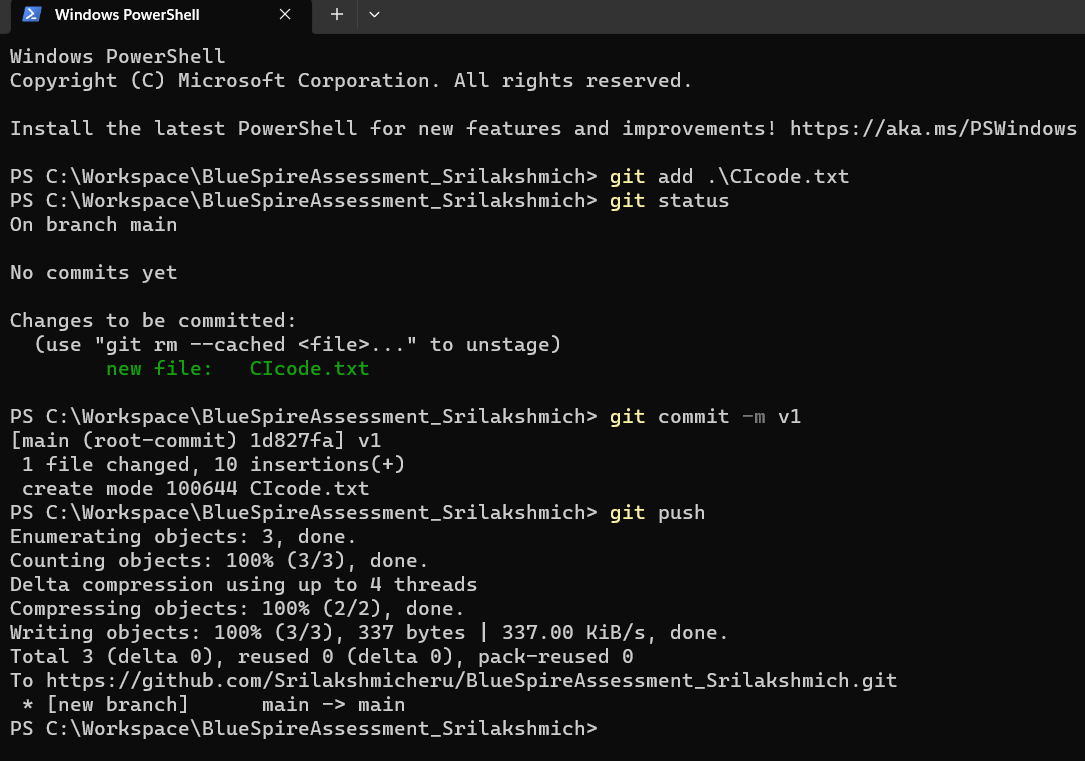
**Step 3**: In workspace, Open the cloned git folder create a document, add code to it.

**Step 4**: Now open the terminal in this folder: Use commands:

**git add CIcode**--used to add the code from working area to staging area

**git commit -m v1** --used to add the code from staging area to local repository

**git push** --used to push the code from local repository to remote repository.



**Step 5**: In Visual studio code, create the Jenkins container through below command:

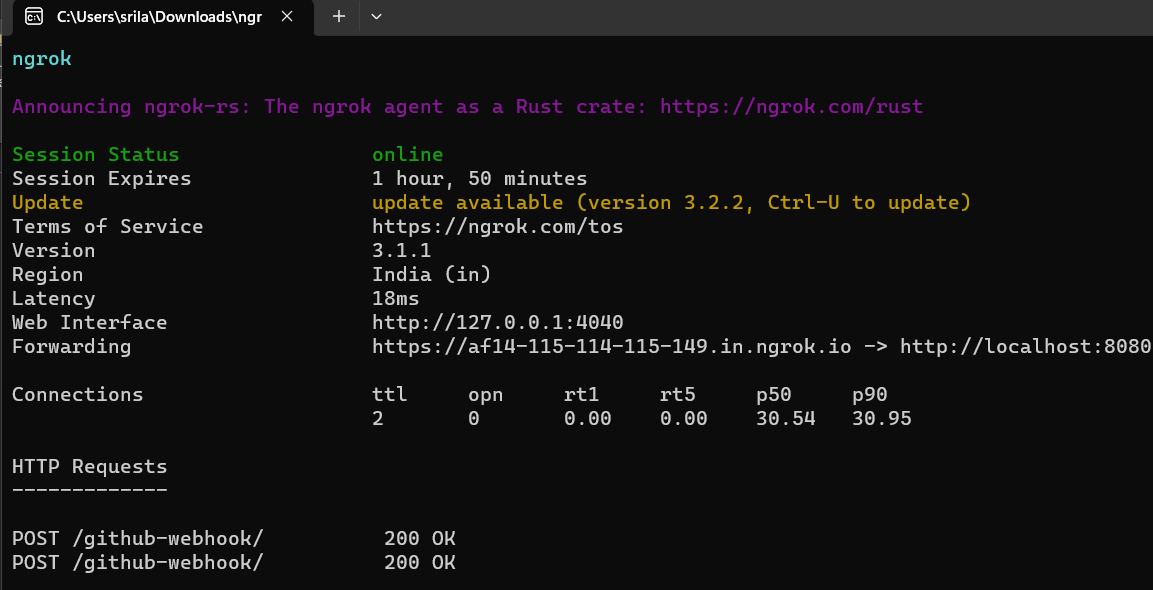
docker run -d --name jenkinstest -v sample/var/jenkins\_home -p 8080:8080 jenkins/jenkins:lts

**Step 6**: For continuous integration, created the webhook in the particular git repository.

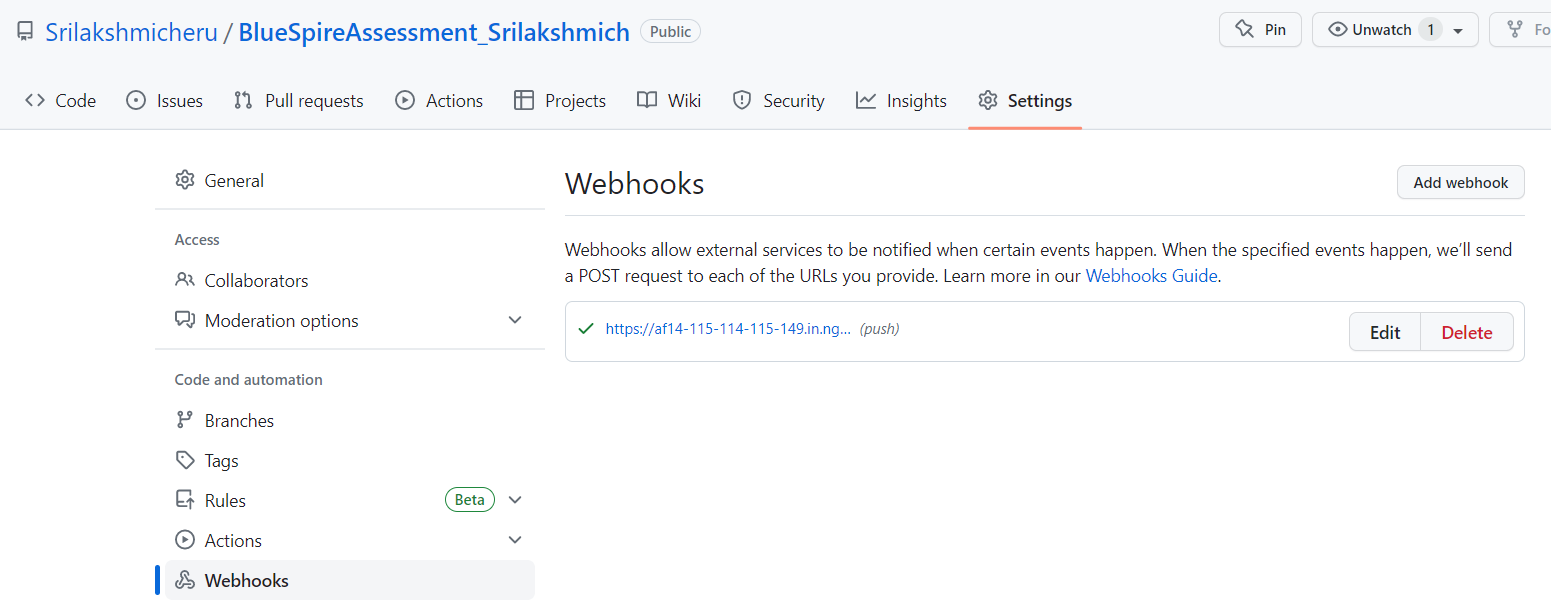
--To create and add a webhook to run the build automatically in jenkins...

--To add webhook first run the ngrok : to attain the jenkins environment url.

--Run the **ngrok http 8080**



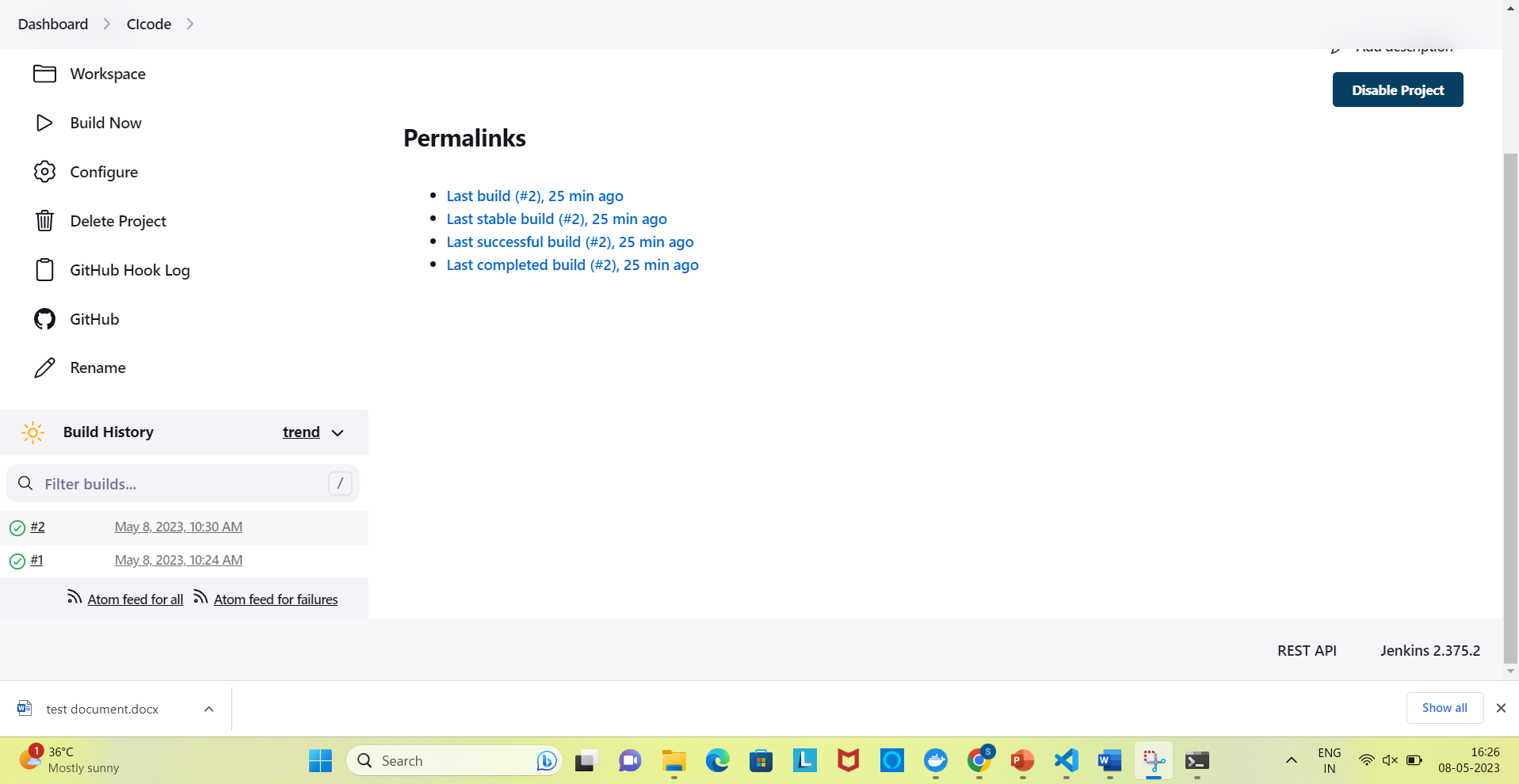
and copy paste the url in git -> Add webhook -> choose app/json -> Add webhook.



**Step 7**: Now Build a freestyle job in Jenkins, build it.

**Step 8**: Edit or update the code in the git repository - Commit the changes.

**Step 9**: Automatically a build will execute in jenkins, whenever a commit pushed in git. (Continuous Integration).



**2. Create a pipeline in Jenkins with the stages of CB(Citizens Bank) end to end**

**implementation of project.**

**Step 1**: Create a git repository named:Assessment\_pipelinesri

**Step 2**: Clone the repository to the workspace with below command

git clone https://github.com/Srilakshmicheru/Assessment\_pipelinesri.git

**Step 3**: In workspace, Open the cloned git folder create a document, add code to it.

**Step 4**: Now open the terminal in this folder: Use commands:

**git add pipeline**--used to add the code from working area to staging area

**git commit -m v1** --used to add the code from staging area to local repository

**git push** --used to push the code from local repository to remote repository.

**//pipeline in Jenkins with the stages of CB(Citizens Bank) end to end**

**implementation of project.**

pipeline {

agent none

stages {

stage('bitbucket') {

steps {

script {

echo "application code will stored here..."

}

}

}

stage('maven') {

steps {

script {

echo "application package will be done..."

}

}

}

stage('build') {

steps {

script {

echo "Building the application..."

}

}

}

stage('test') {

steps {

script {

echo "Testing the application..."

}

}

}

stage('sonarqube') {

steps {

script {

echo "application will be tested for quality..."

}

}

}

stage('convert') {

steps {

script {

echo "convert the file"

}

}

}

stage('nexus') {

steps {

script {

echo "build an nexus image for the code..."

}

}

}

stage('push') {

steps {

script {

echo "convert push"

}

}

}

stage('deploy') {

steps {

script {

echo "Deploying the application..."

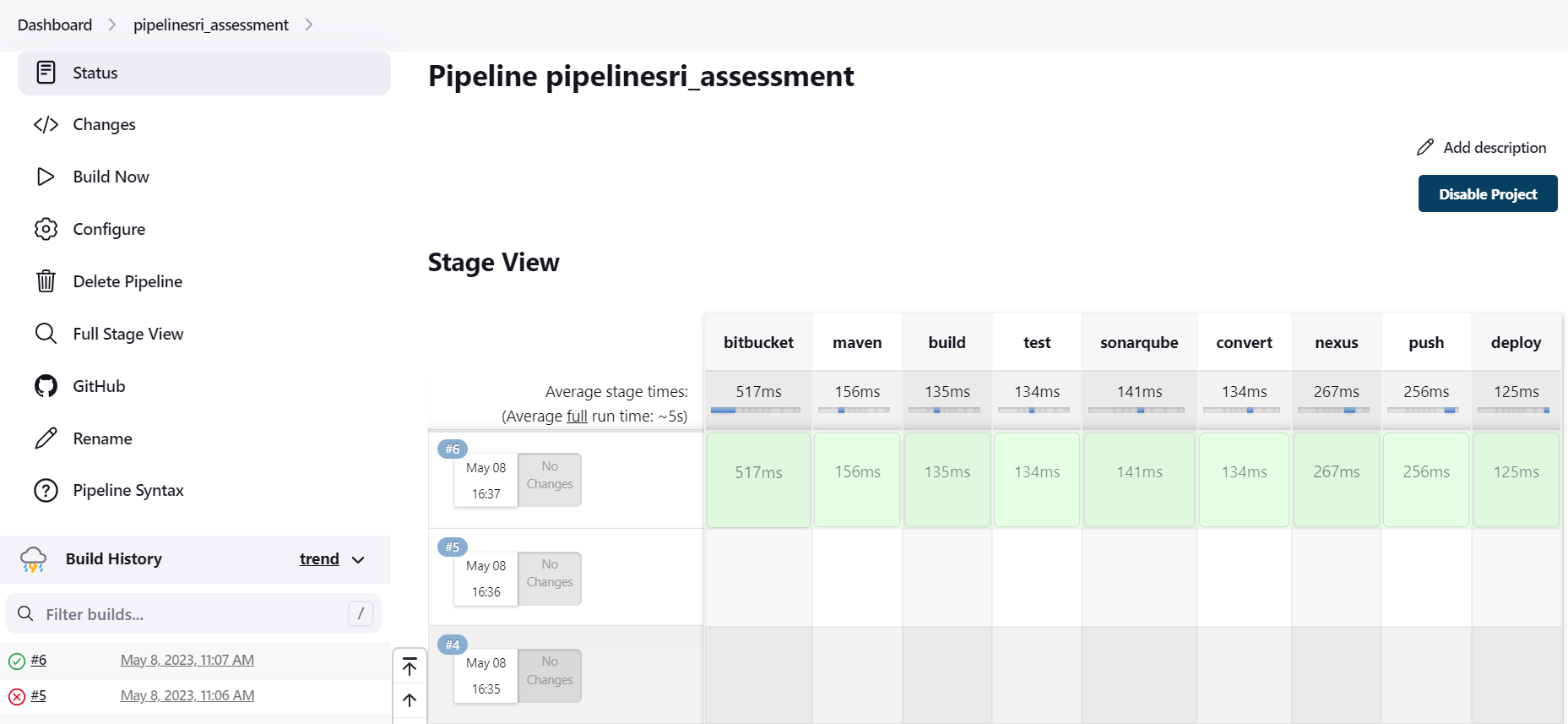
}

}

}

}

}

* Build the pipeline job in Jenkins.
* 

**3. Create a sample application; create the dockerfile ->Docker image🡪Docker container  Deploy in k8’s**

**Step 1**: Create a repository in git

**Step 2**: clone it in workspace

Git clone <https://github.com/Srilakshmicheru/Dockerfile_image.git>

**Step 3**: Create Dockerfile and index.html file in workspace folder

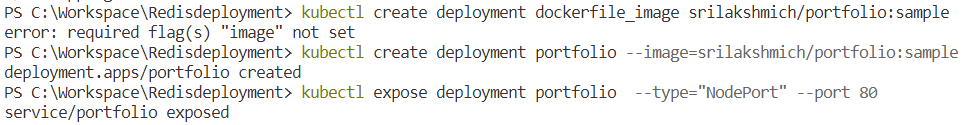
**Step 4**: Use commands

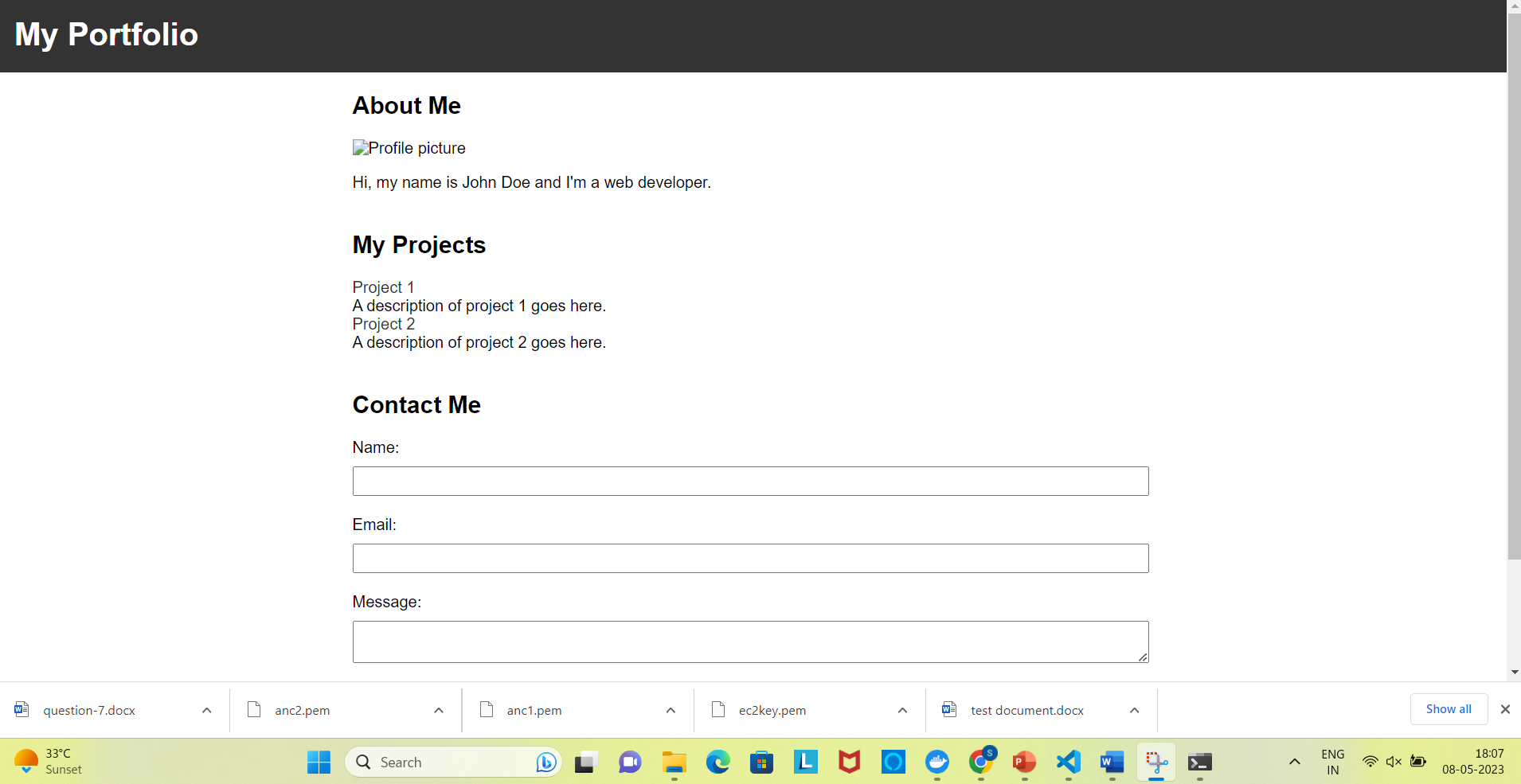
git add .

git commit -m v1

git push

**Step 5**: Give below commands to deploy docker image in k8’s:

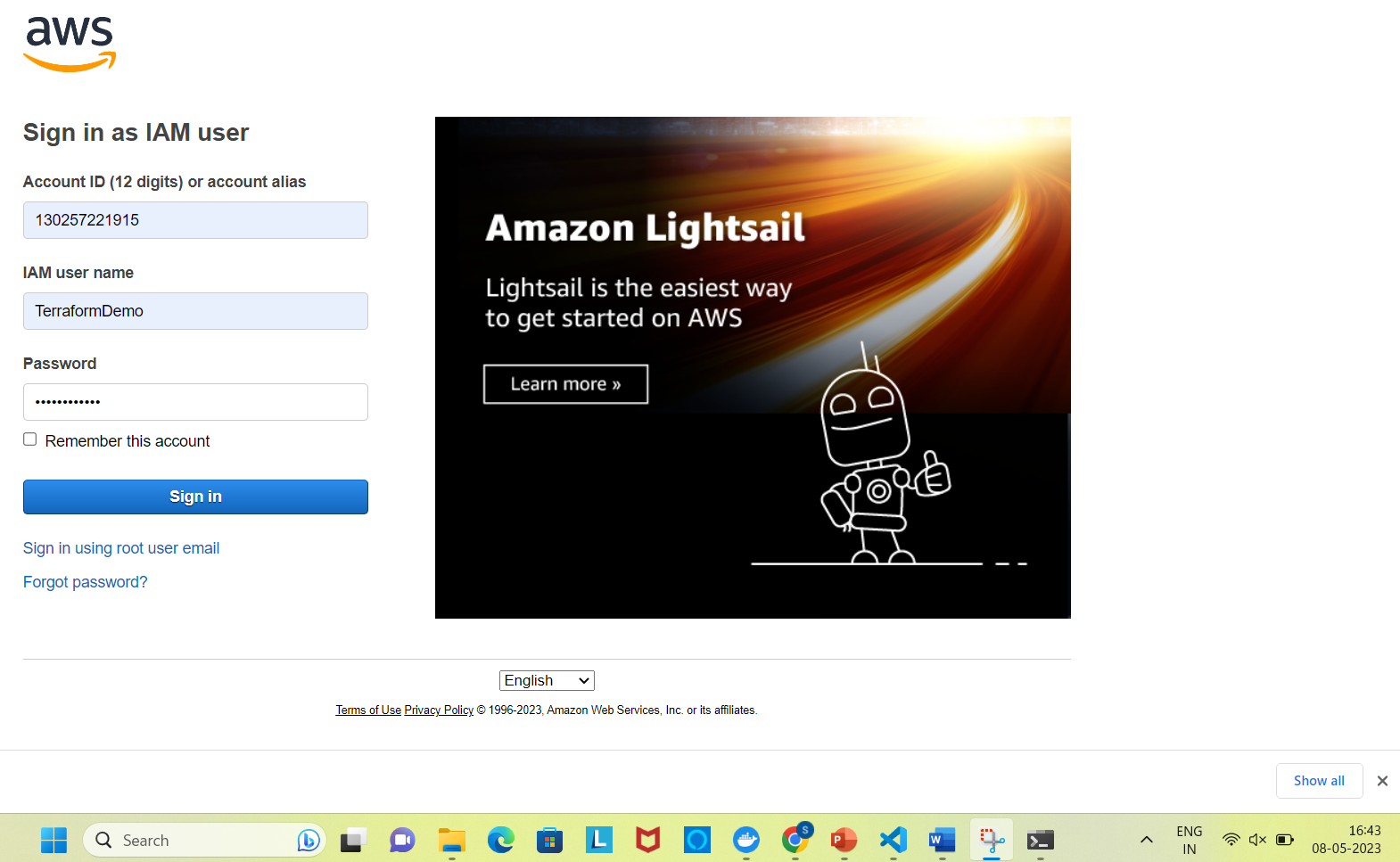




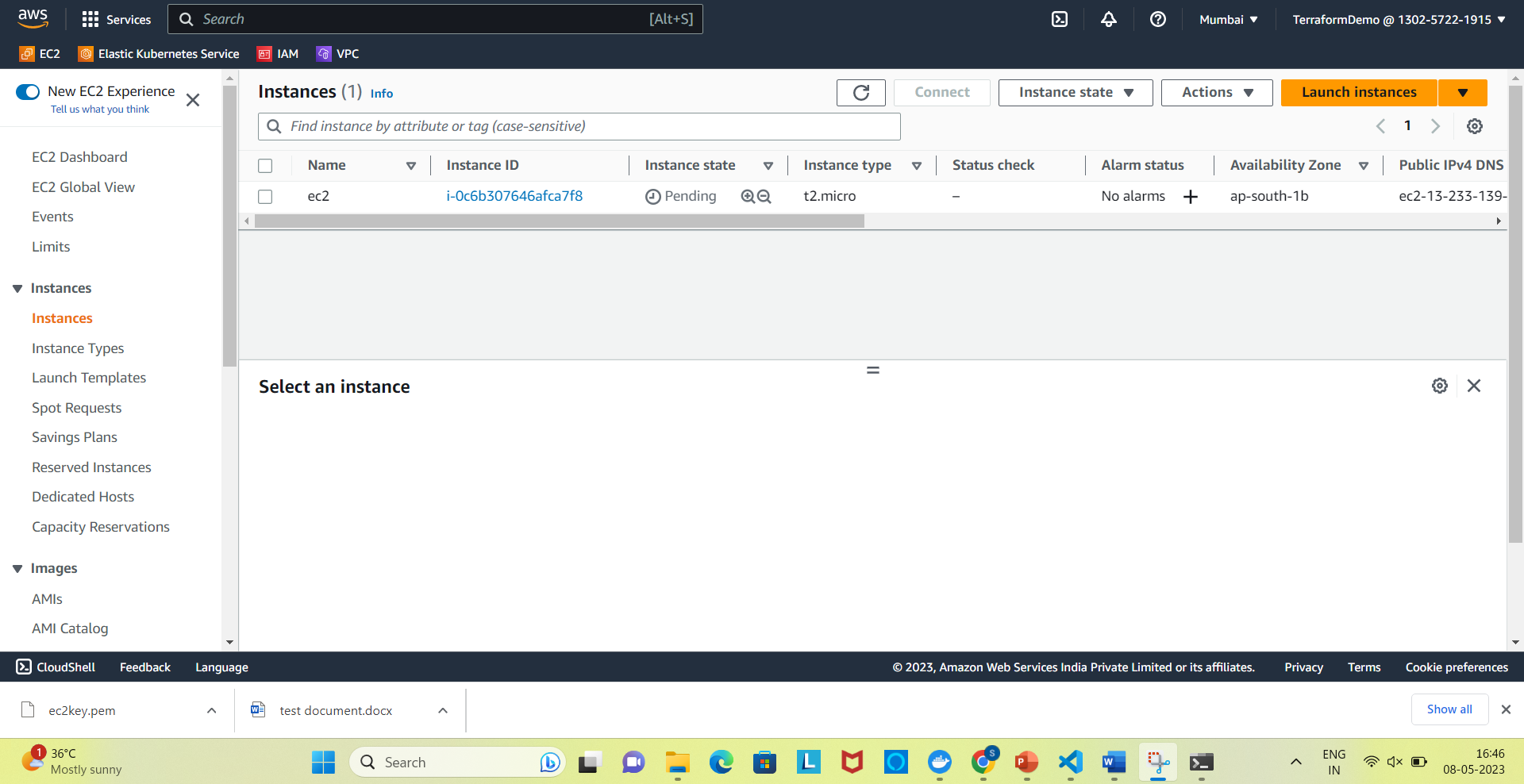
**4. Create an ec2 instance using terraform and destroy the resource.**

**Step 1**: Login to the aws console.

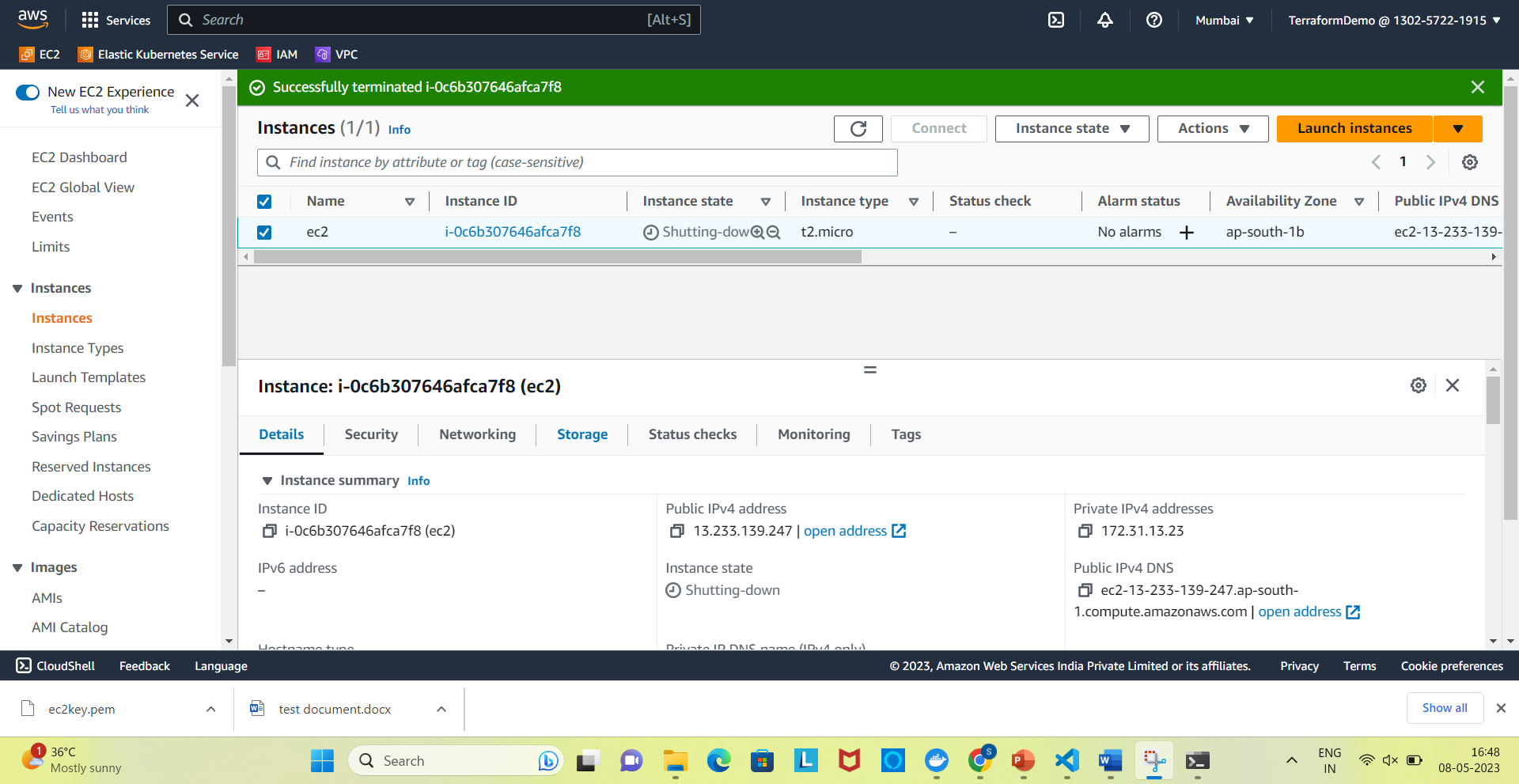
**Step 2**: Create terraform I am user profile.

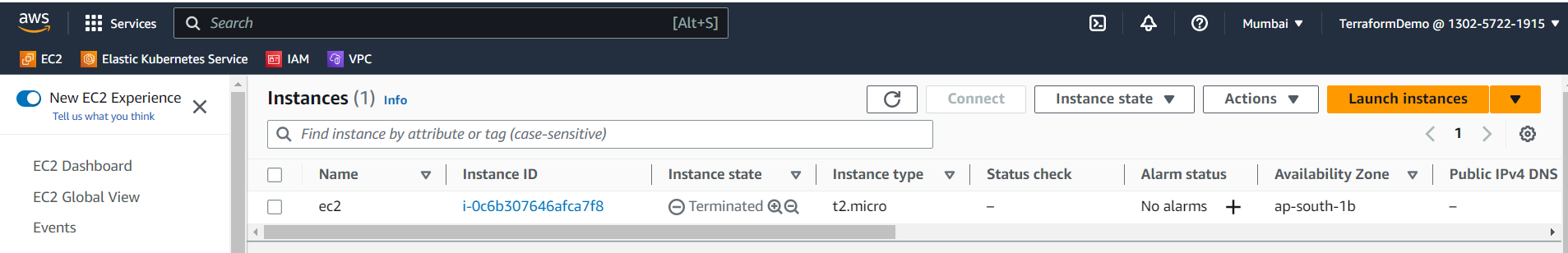


**Step 3**: Create ec2 instance.



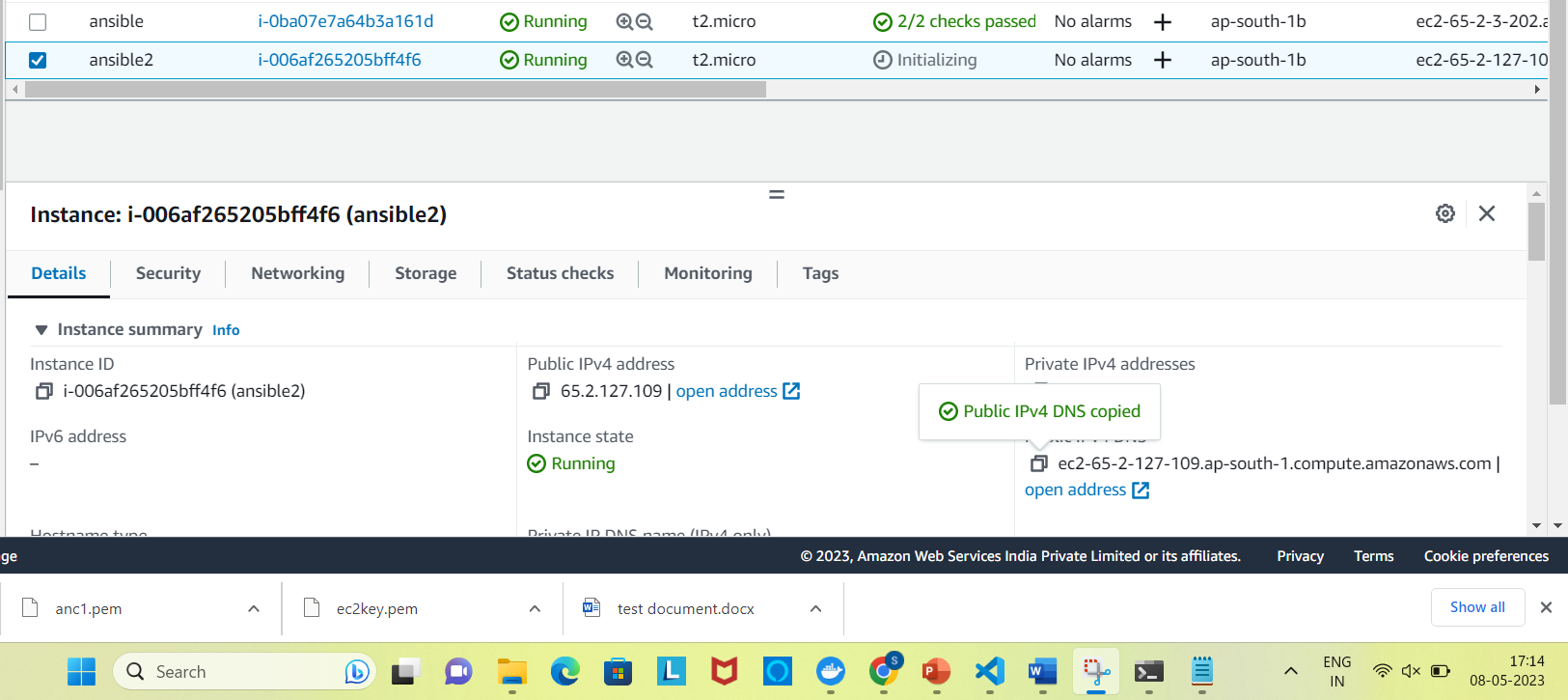
**Step 4**: Destroy the instance.

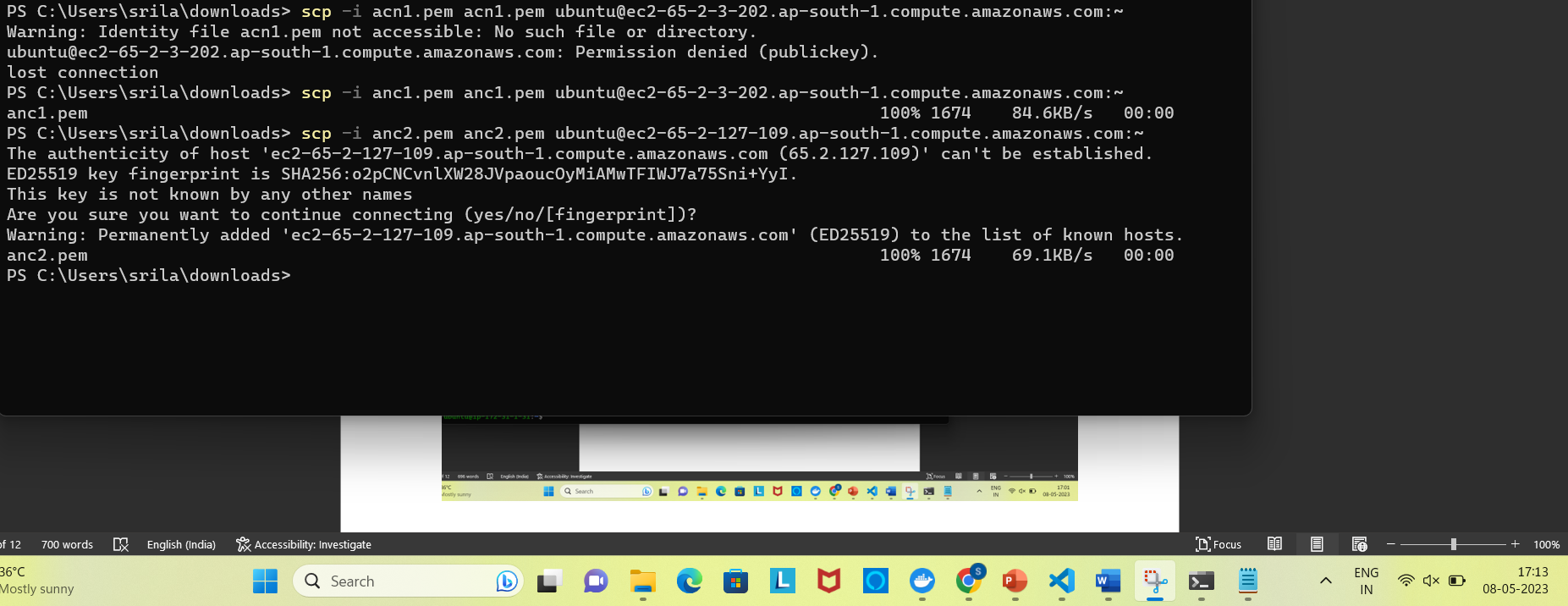




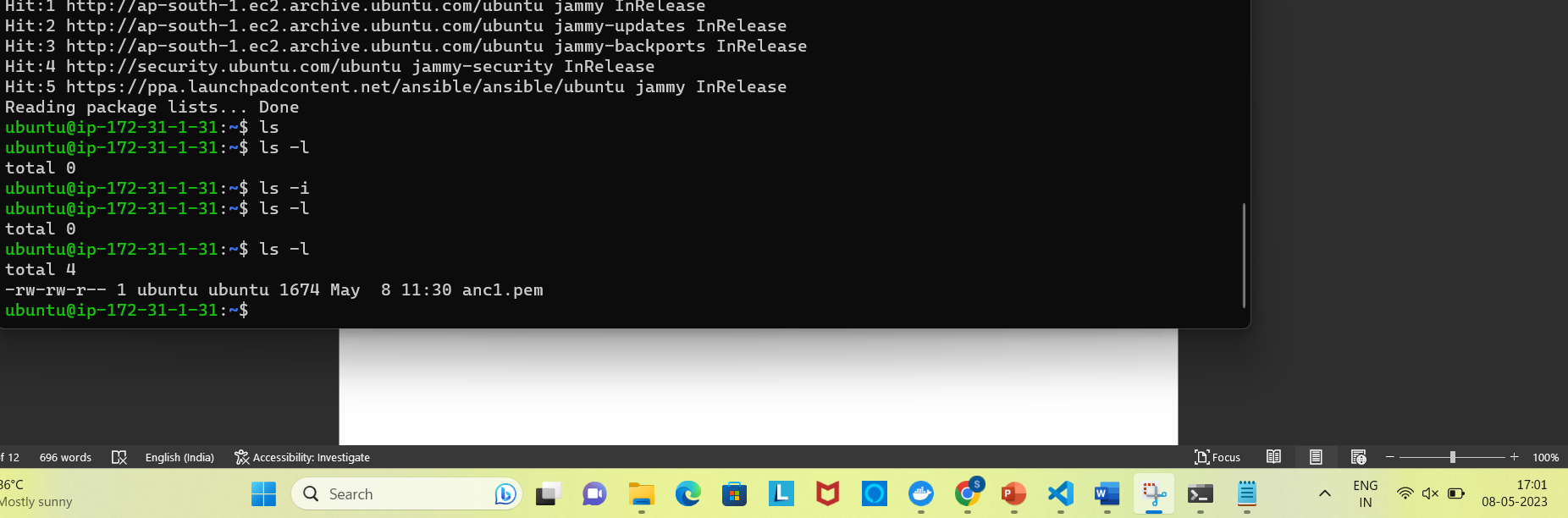
**5. Install apache using Ansible on a ec2 machine.**

**Step 1**: Create 2 ansible instances.

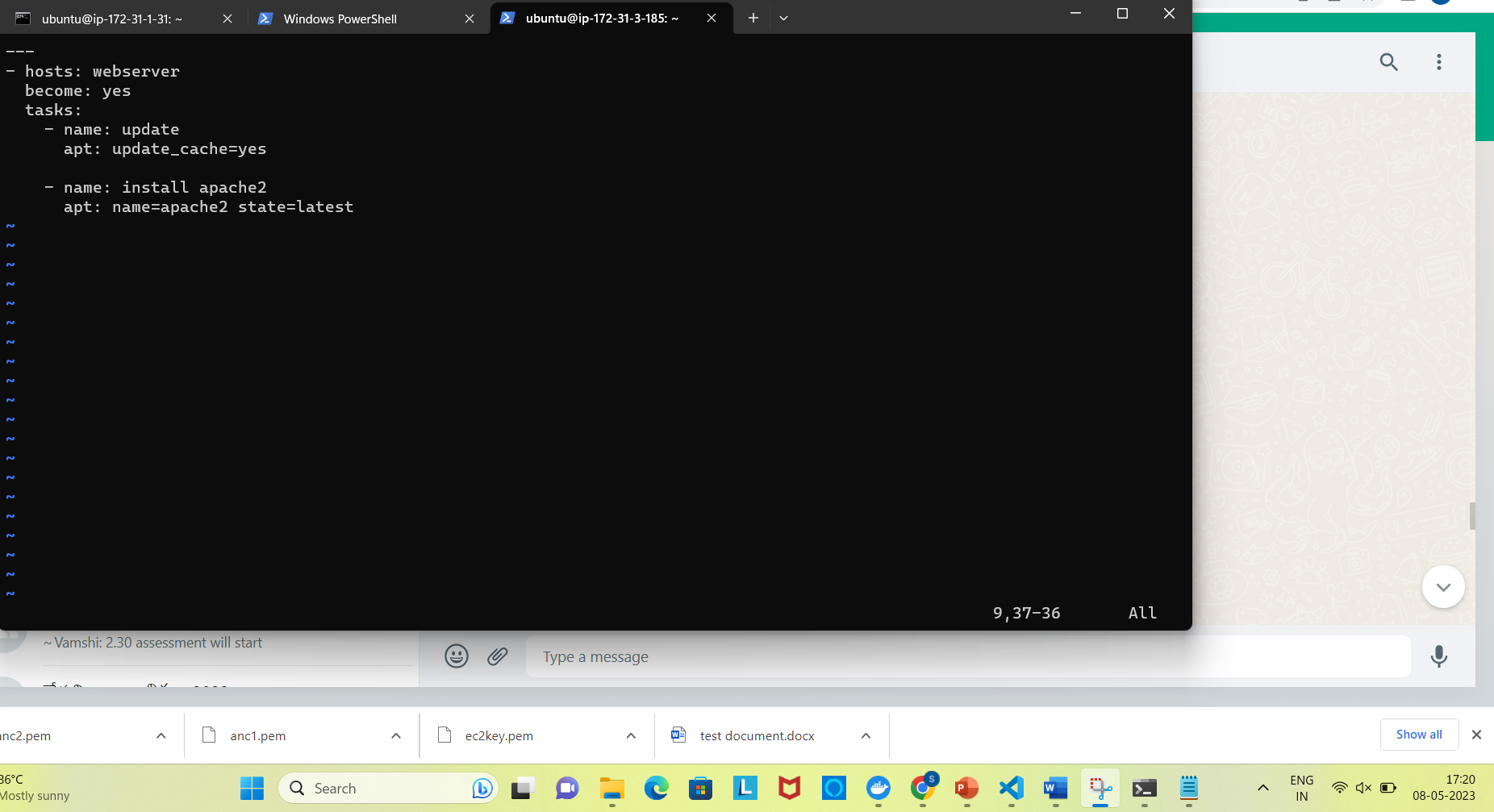


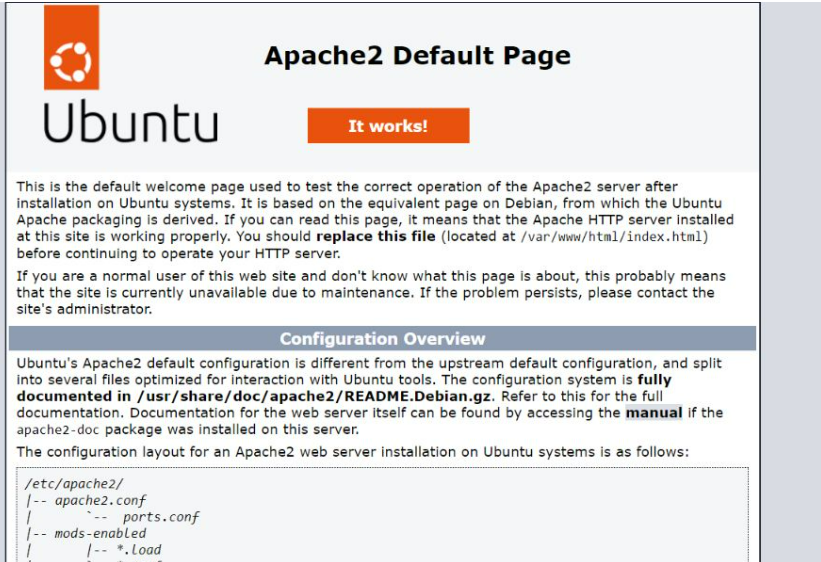
**Step 2: **

**Step 3**: Click on connect to use ssh key. Copy it in command prompt.



Step 4: sudo vi /etc/ansible/hosts



**Step 5**: 

**6. Deploy a e-commerce micro services…Redis Guestbook Deployment**

**Step 1**: create repository Redis\_ecommerce

**Step 2**: git clone https://github.com/Srilakshmicheru/Redis\_ecommerce.git

**Step 3**: create Yaml files in workspace folder

**Step 4**: Use commands

git add .

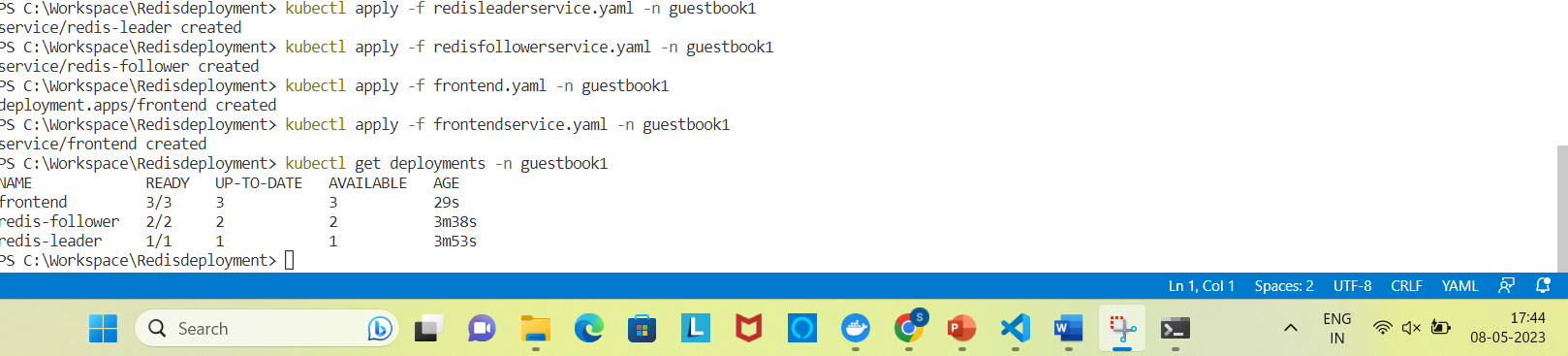
git commit -m v1

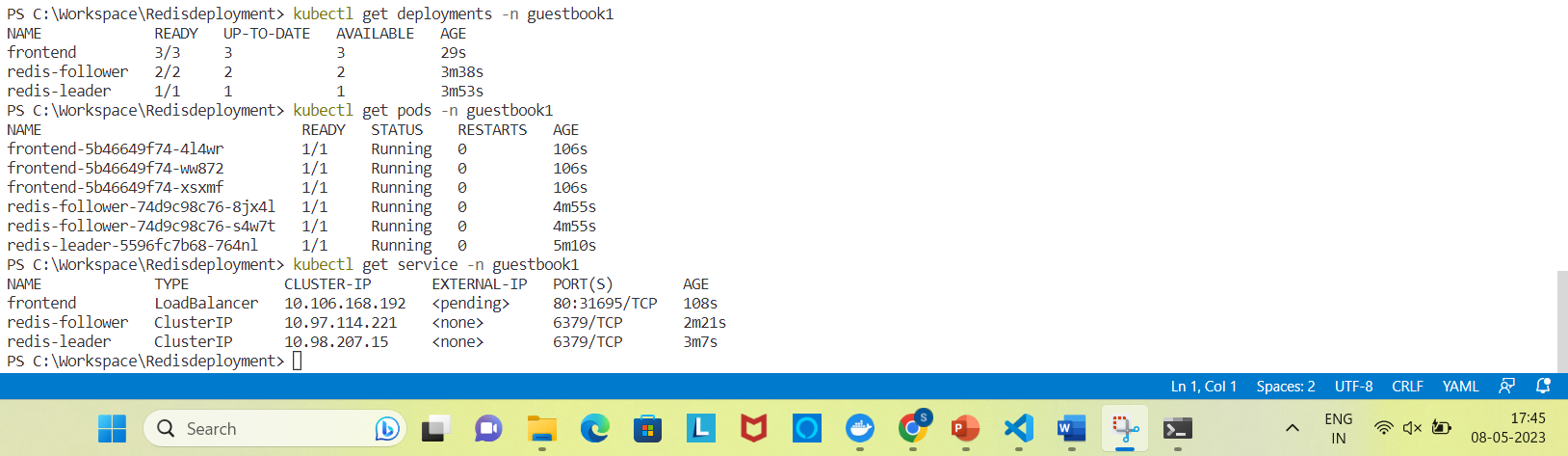
git push

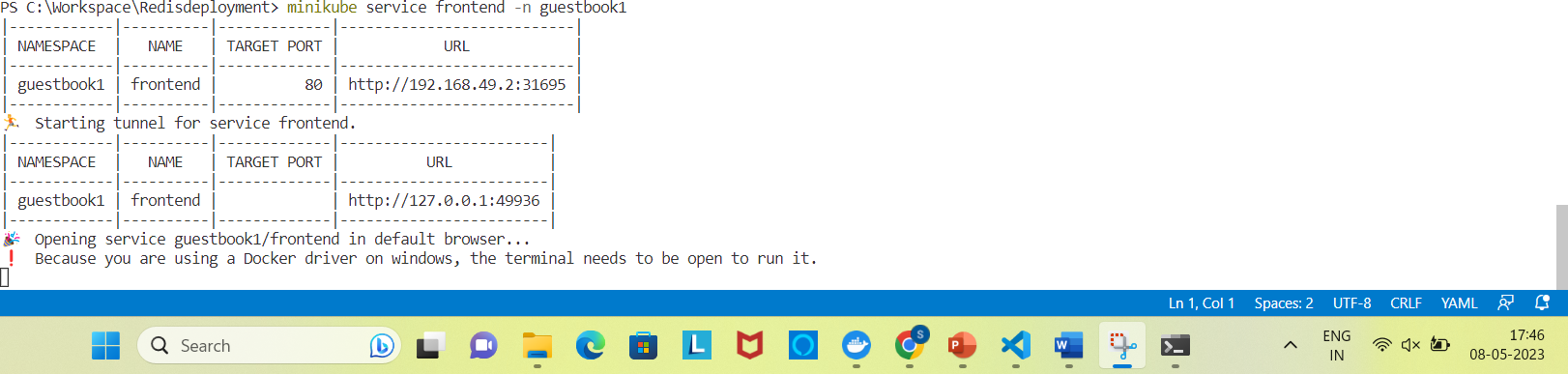
Step 5:minikube start

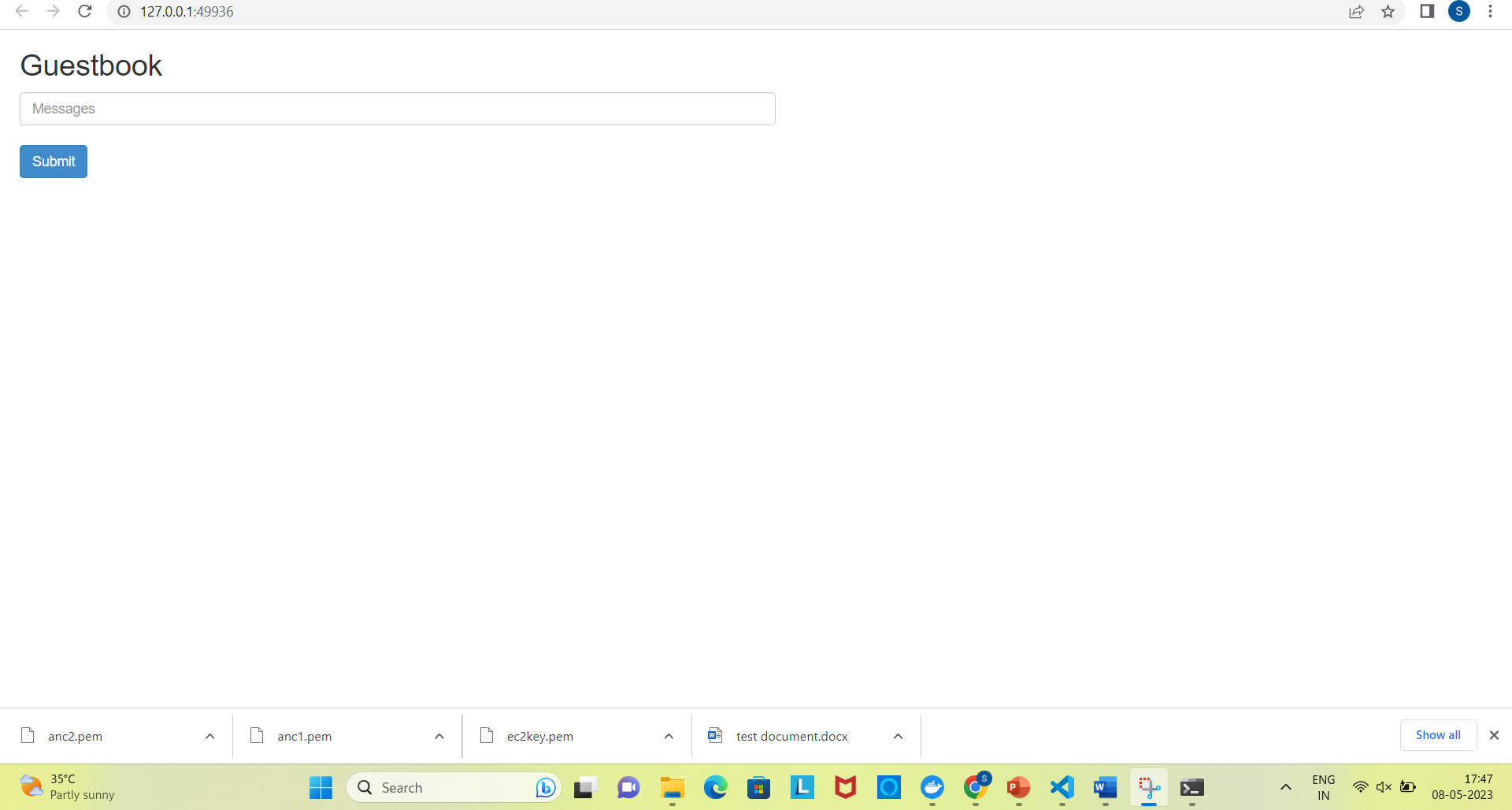
PS C:\Workspace\Redisdeployment> kubectl create ns guestbook1

namespace/guestbook1 created

Step 6: 

Step 7: 

Step 8: 

Step 9: 

7. Citizens bank project explain in details and draw flow diagrams of

applications.

**Citizens Bank project::::::**

* In this project we are using DevOps methodology.
  + We are using tools like
    - Bitbucket (code storage)
    - Jenkins (pipeline)
    - Service now (ticket monitoring tool)
    - Sonarkube (code quality)
    - Maven (build and package)
    - Nexus (images repo)
    - Aws (instances)
    - Openshift(deploying the microservices)
    - Datadog(monitoring and obeservability)
* There are many projects happing like migration and updating
* But, our main project is L2 ops
  + Which involves monitoring the openshift cluster and pods and its health and its rescources
  + And if there is any issue in the cluster we need to set the trashoulds and we will get a ticket
* Steps when the ticket is raised:
  + First accept the ticket
  + Check the logs in datadog, and find where the issue is from
    - Like in network
    - Or pods etc
* And once we come to know the issue
  + We need to inform to the particular team about the issue
* And by this we can achive no down time and provide a reliable site