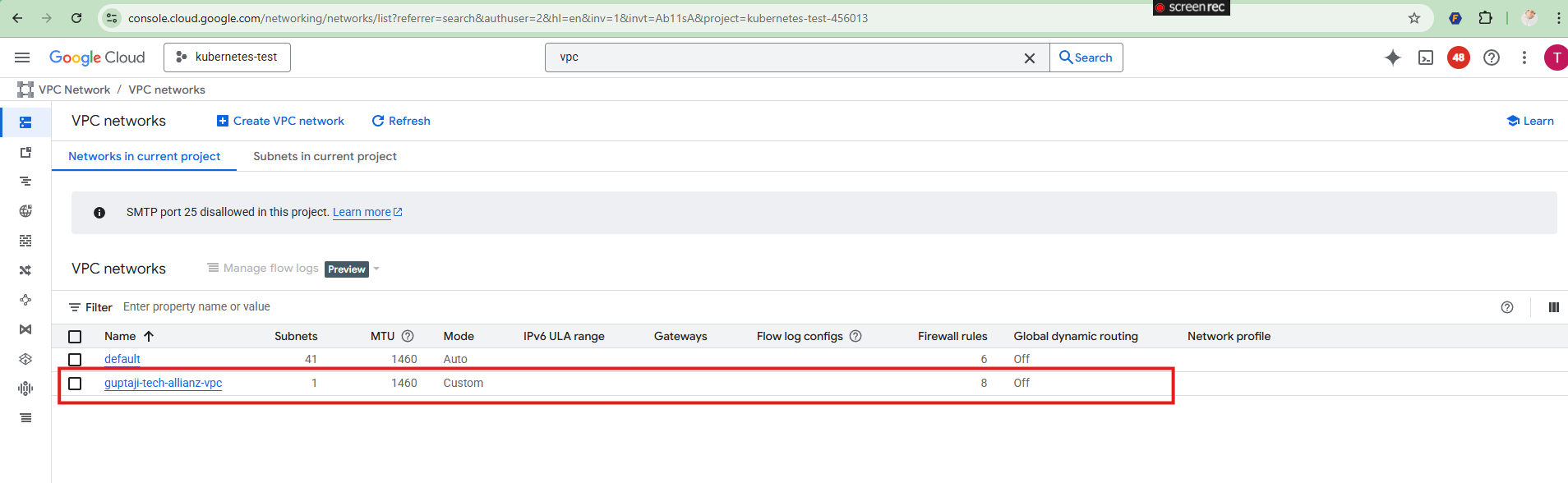
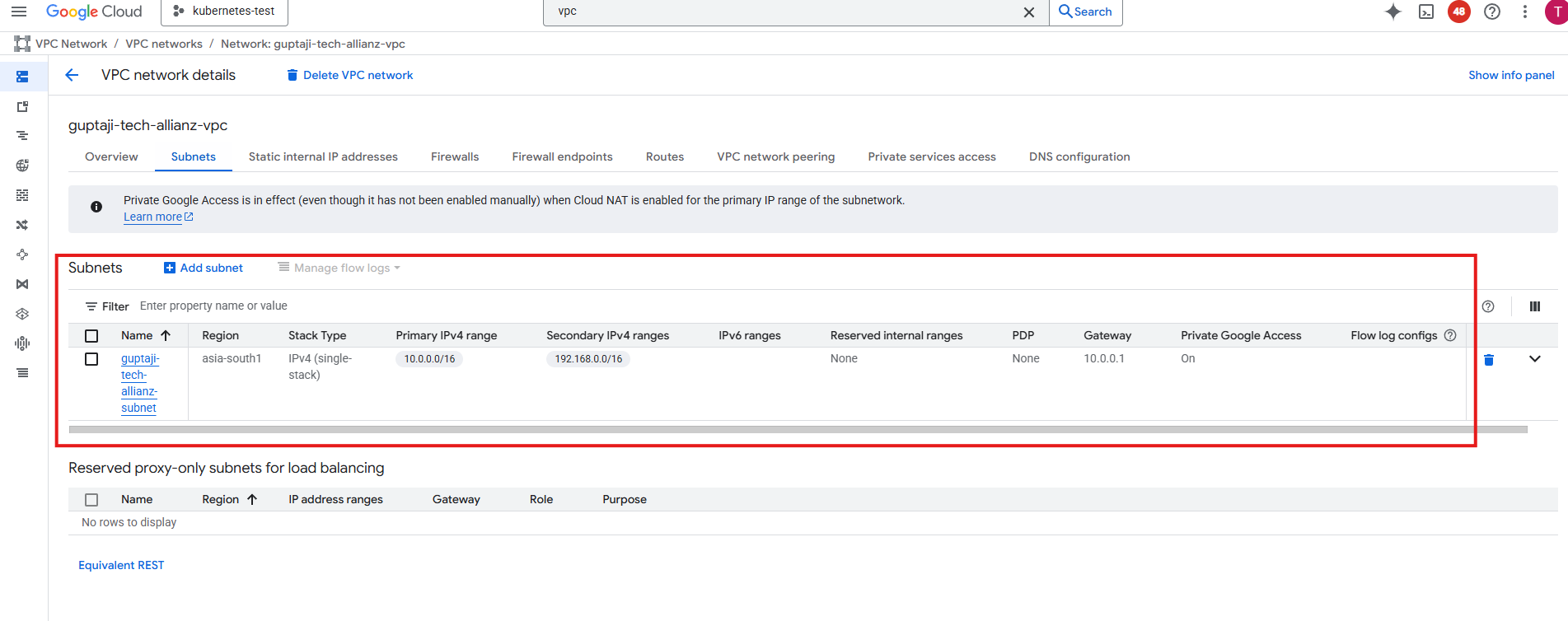
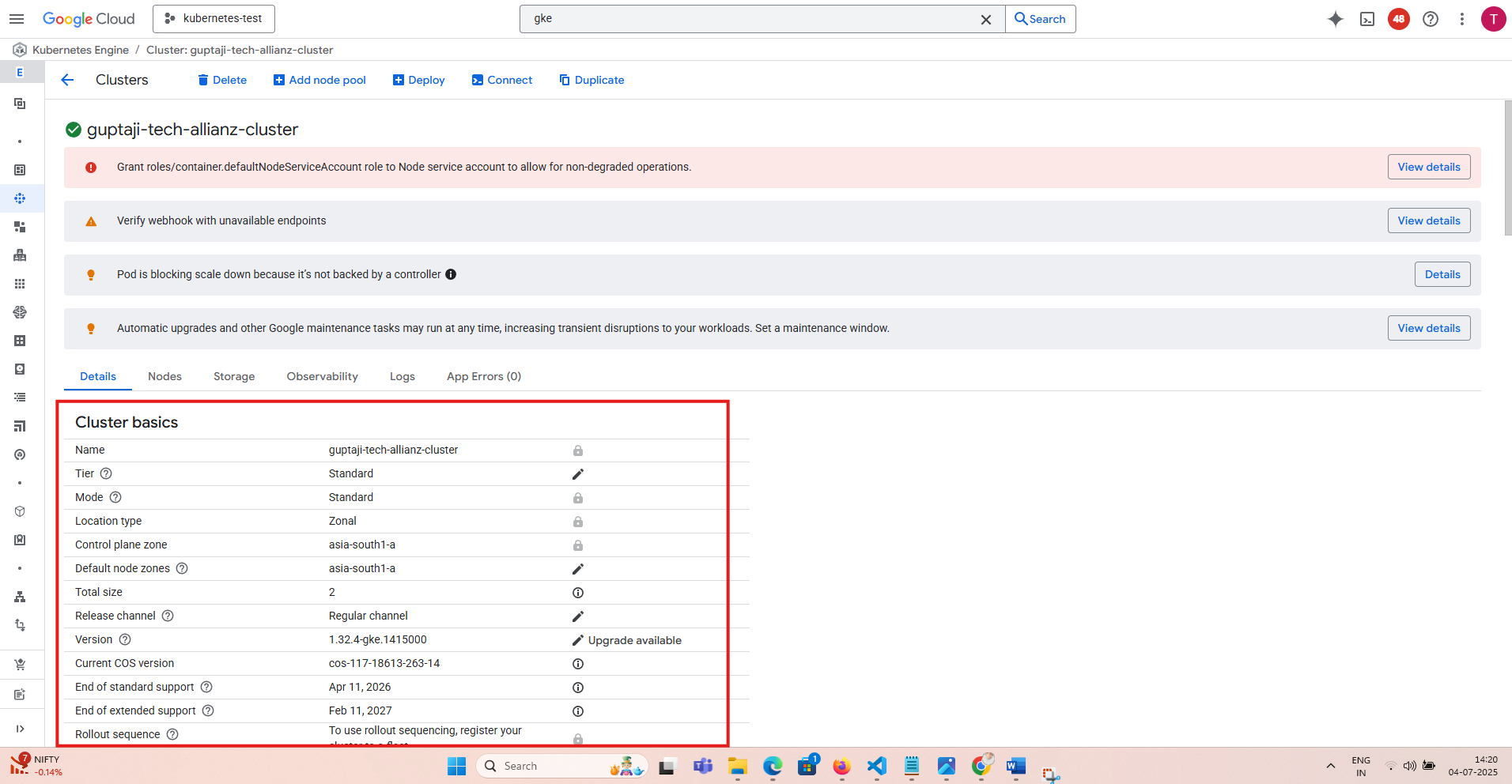
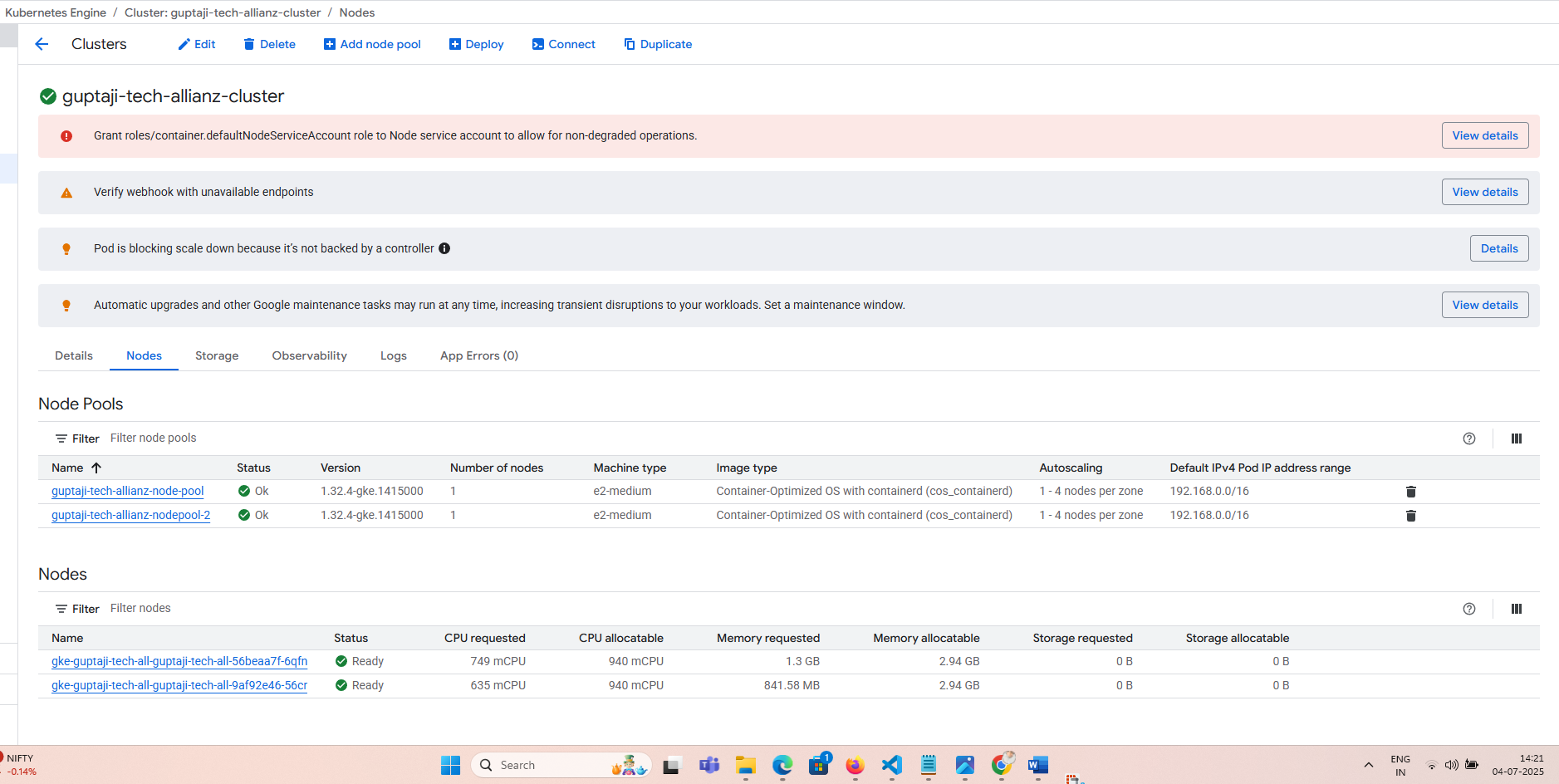
**Implementation Steps –**

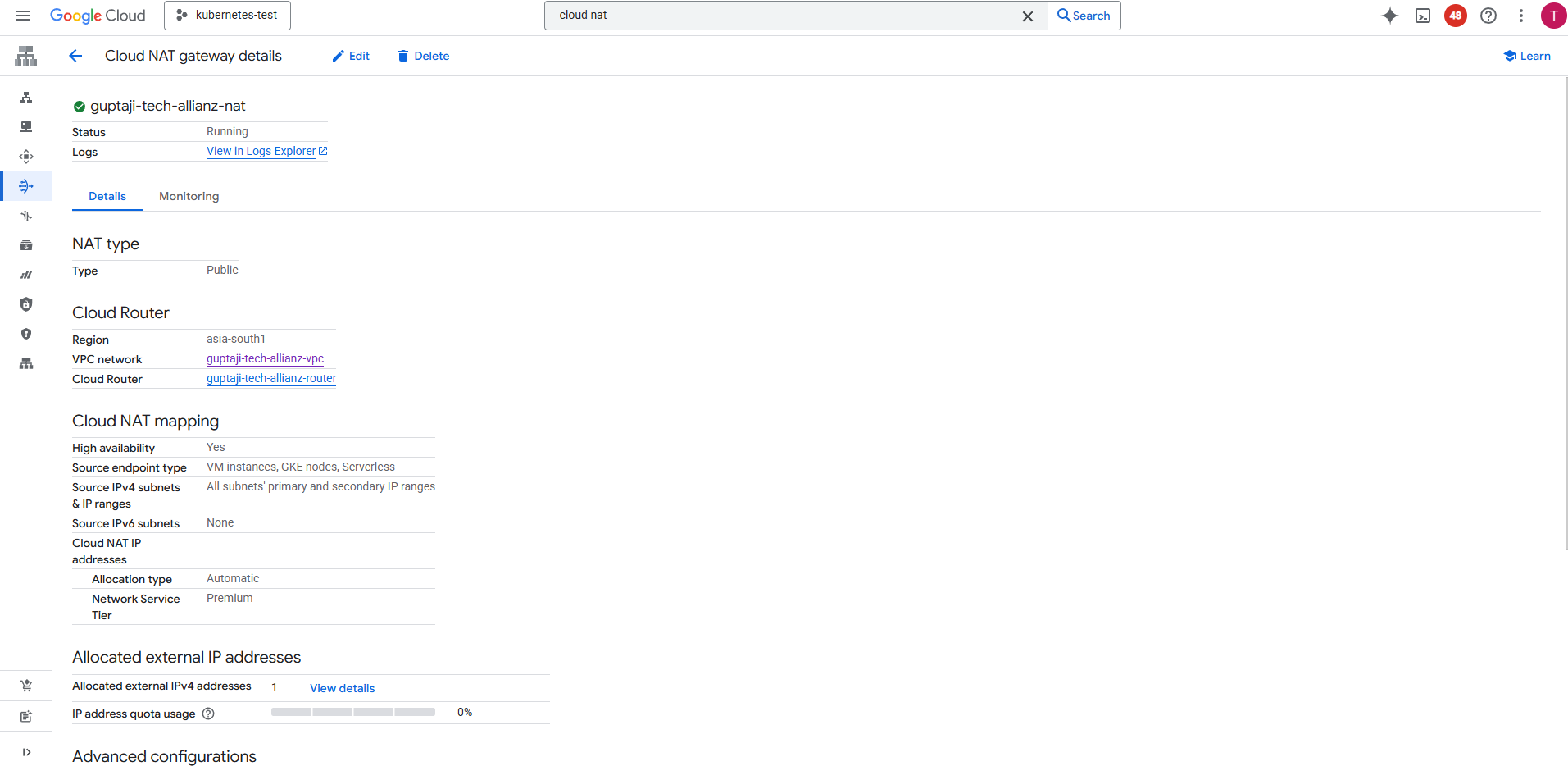
1. Prepared terraform modules for VPC, Subnet, GKE and Cloud NAT.
2. Built the infrastructure using terraform. Below are the screenshots of components in GCP.







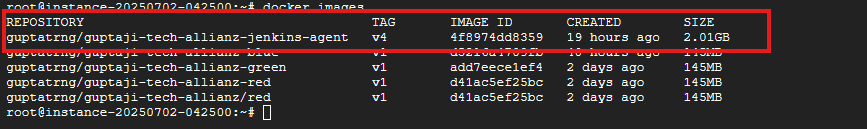




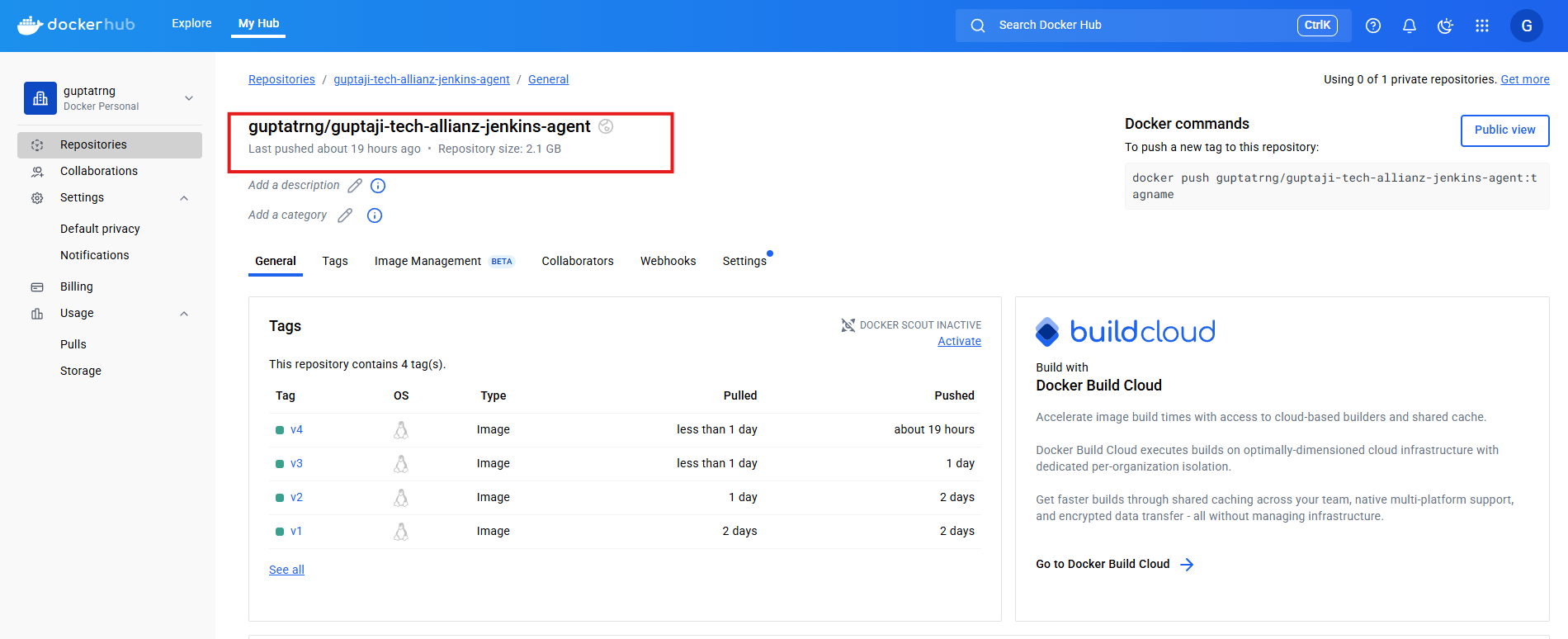
1. Prepared docker file for customer Jenkins slave/agent which will be deployed as a pod in Kubernetes cluster. The docker file is present in below path in repo –

<https://github.com/guptatrng/guptaji-tech-allianz/blob/main/jenkins/dockerfile>

1. Build the docker image using docker build command and give it a appropriate tag.



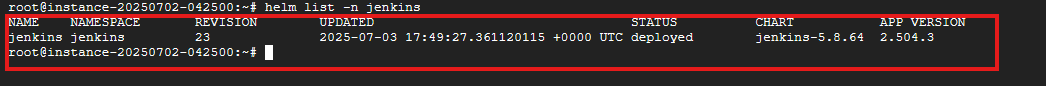
1. Push the image to docker hub repository.

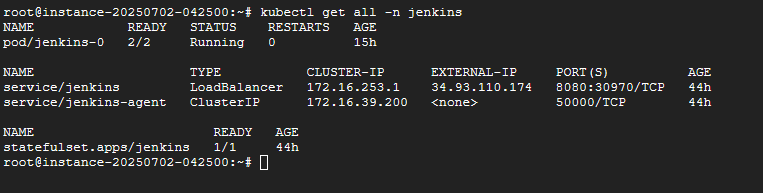


1. Use helm chart - <https://github.com/jenkinsci/helm-charts> to install Jenkins in Kubernetes. Update values.yaml file to add additional configurations in Jenkins deployment. The values.yaml file is present in repository –

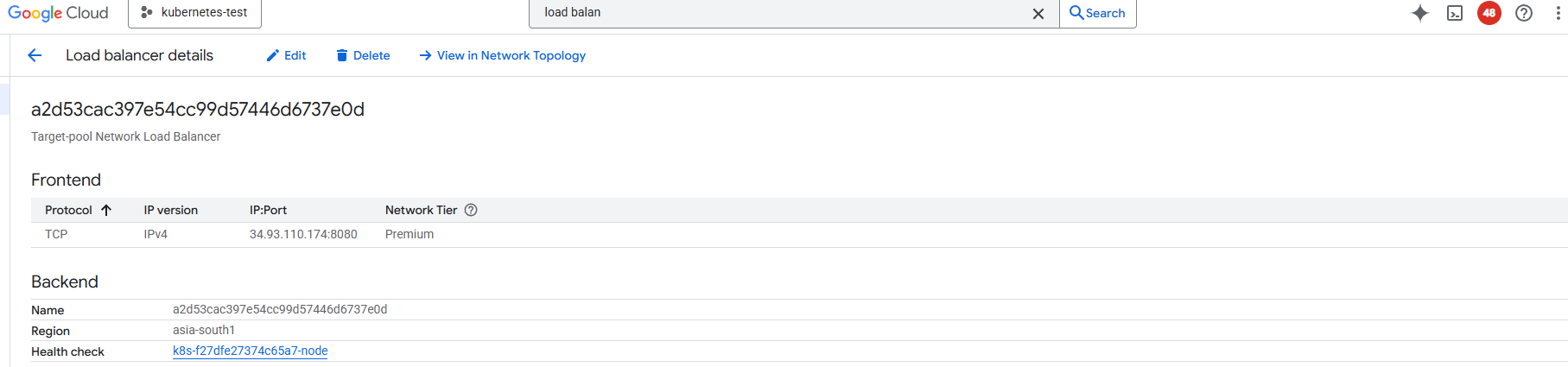
<https://github.com/guptatrng/guptaji-tech-allianz/blob/main/jenkins/values.yaml>

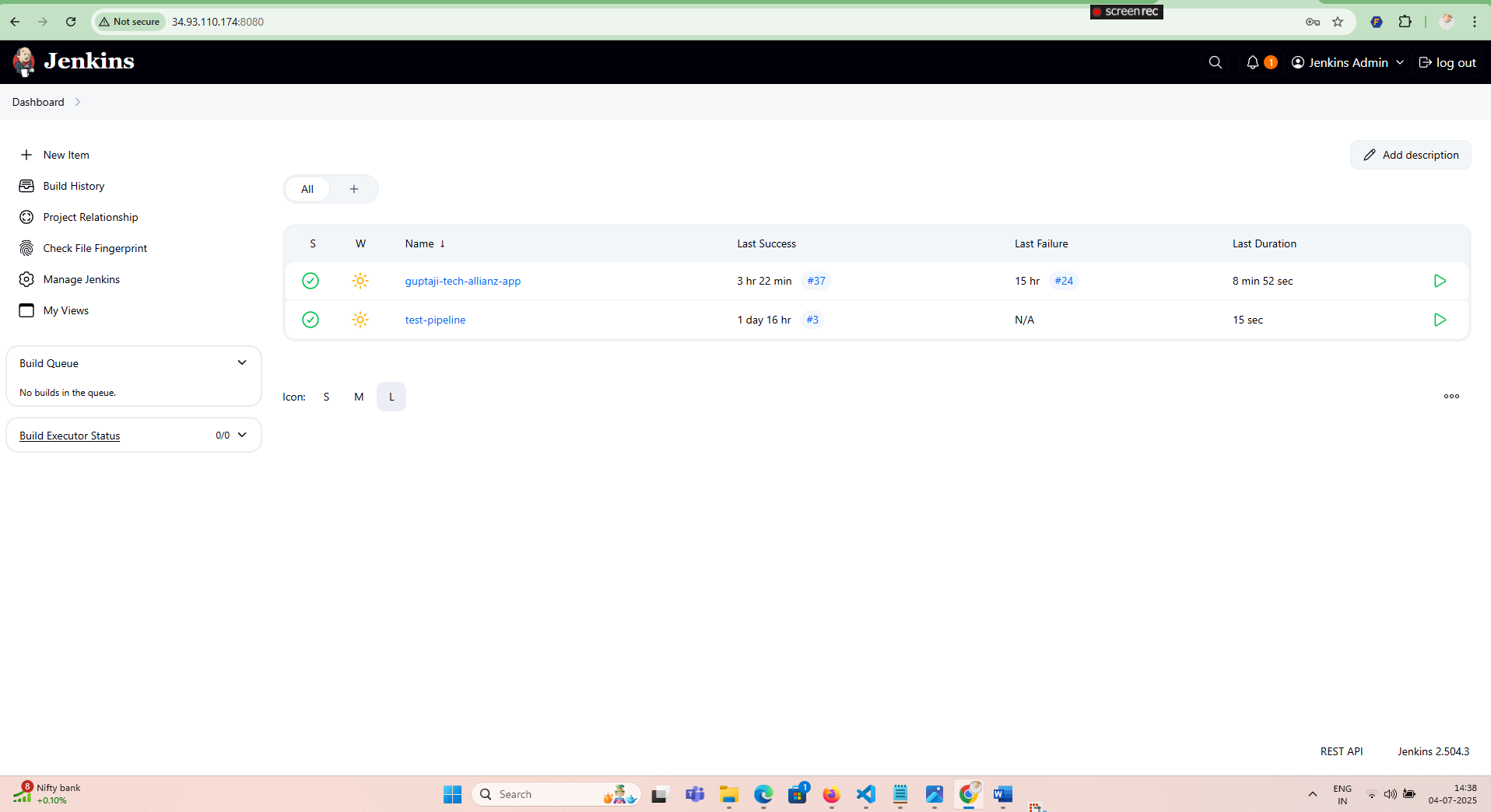
1. Apply the helm chart using helm install command





1. The Jenkins is exposed through load balancer type Kubernetes service.

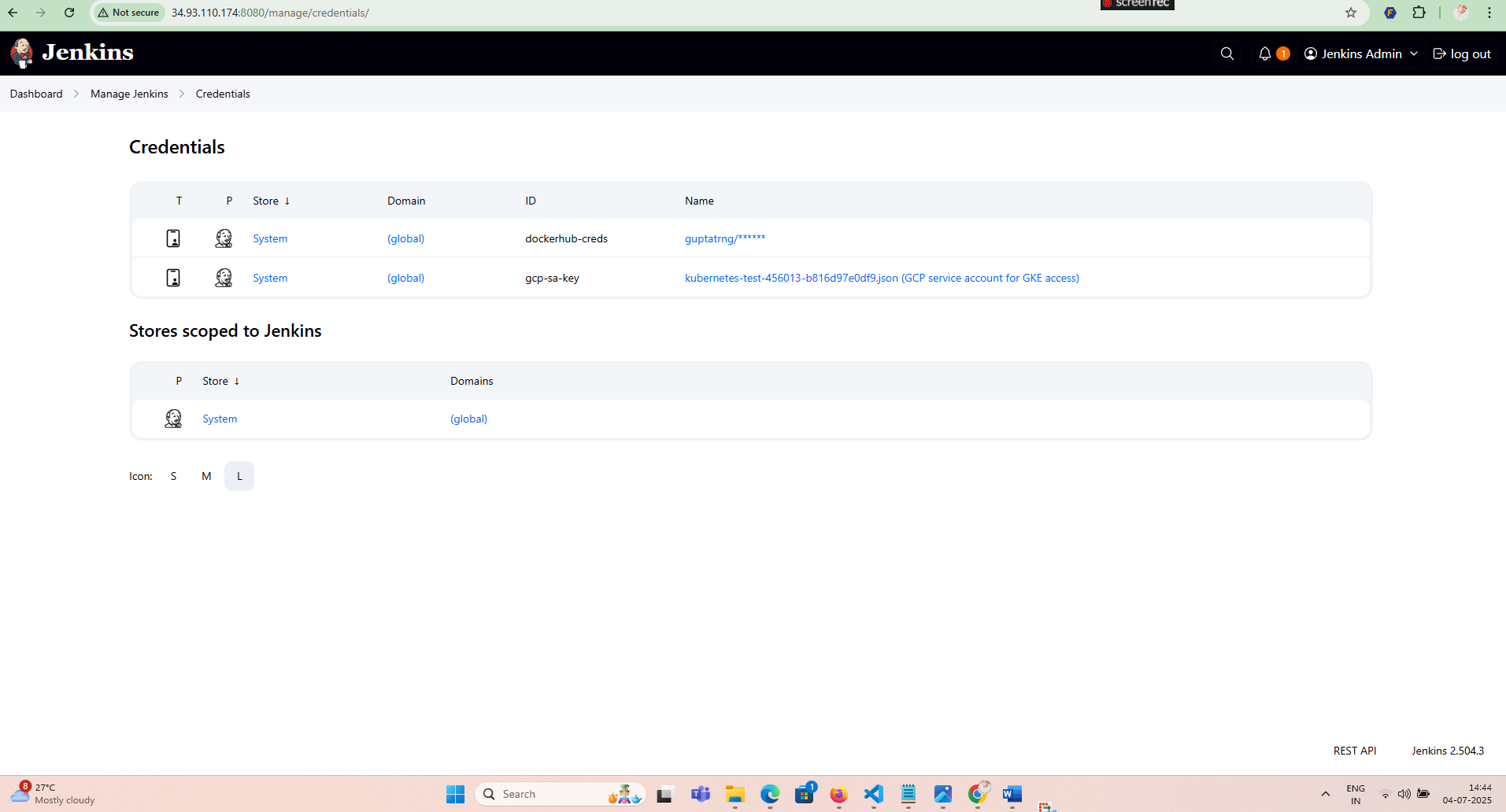




1. Prepare the python application along with the docker files and Kubernetes manifest files. The files are present in below repo, it contains 2 sub folders corresponding to blue and green.

<https://github.com/guptatrng/guptaji-tech-allianz/tree/main/python-app>

1. Add the docker hub credentials and GCP service account keys (for gke authentication) in Jenkins credentials.



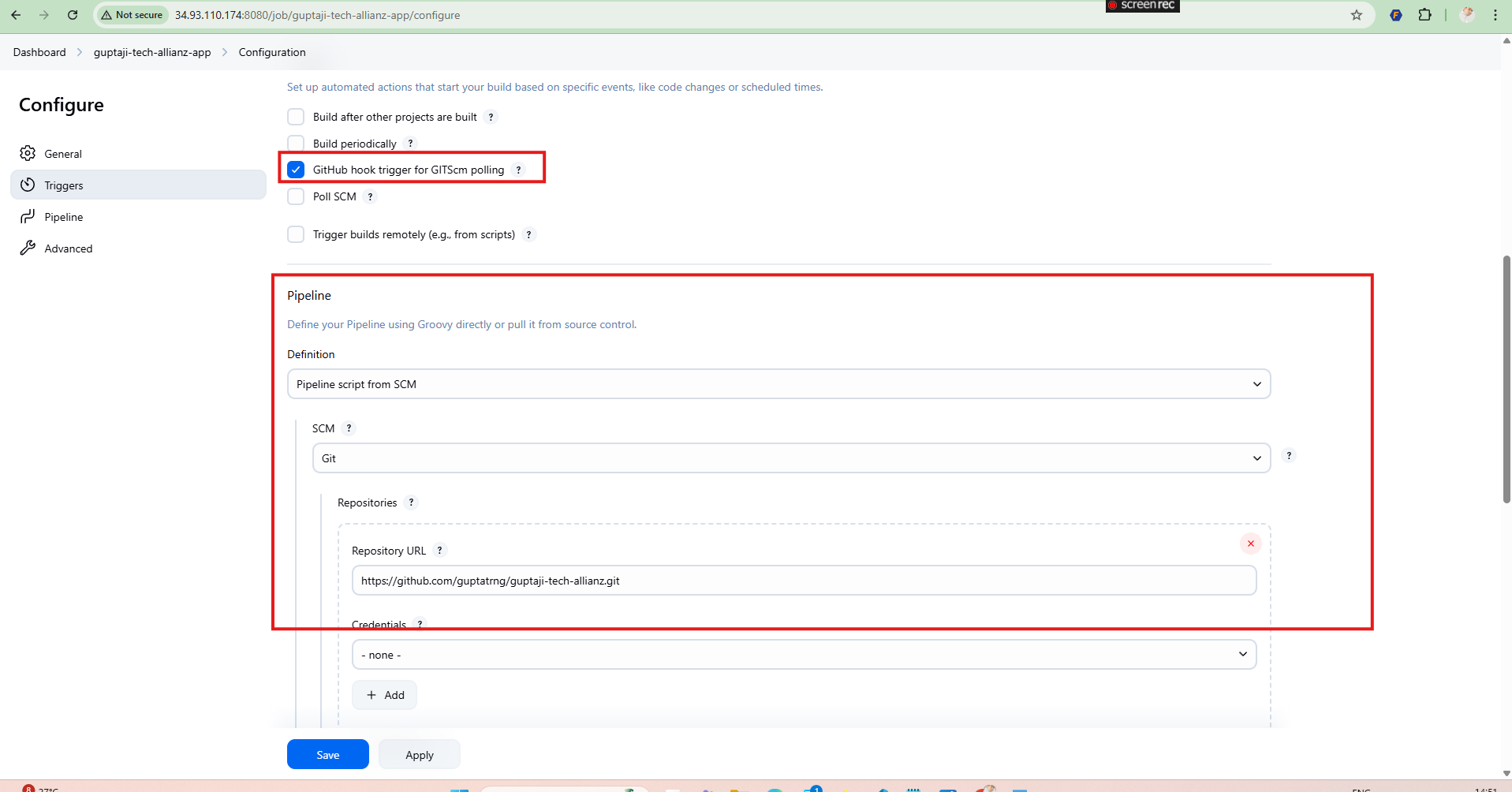
1. Prepare the deployment control file which will specify the application we want to deploy – Blue or Green. This file will be read by Jenkins agent for building image and deployment. The file is present in below repo –

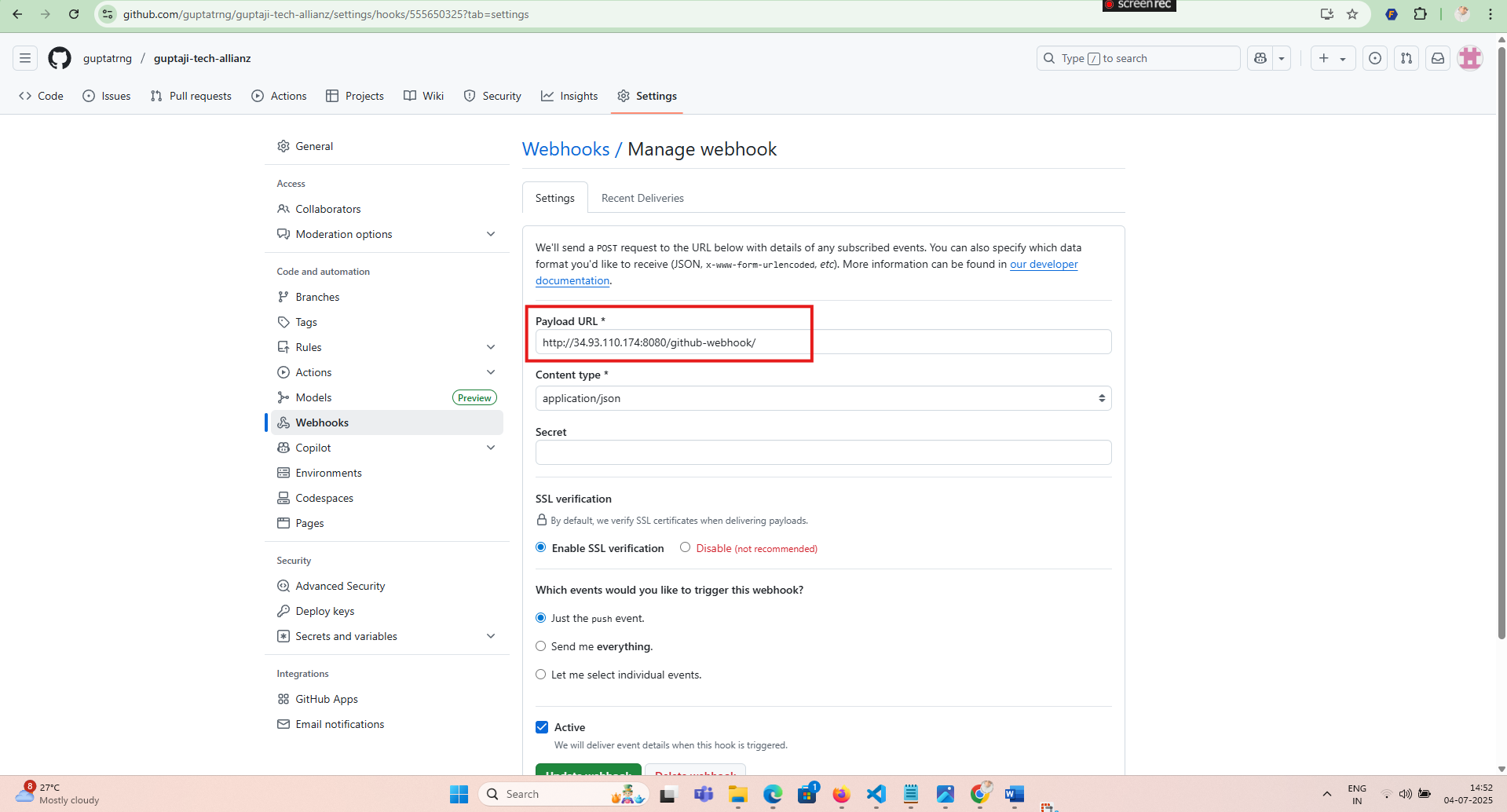
https://github.com/guptatrng/guptaji-tech-allianz/blob/main/python-app/blue-green-control.txt

1. Prepare the Jenkins file having build stages for python application. The Jenkins file present in below repo –

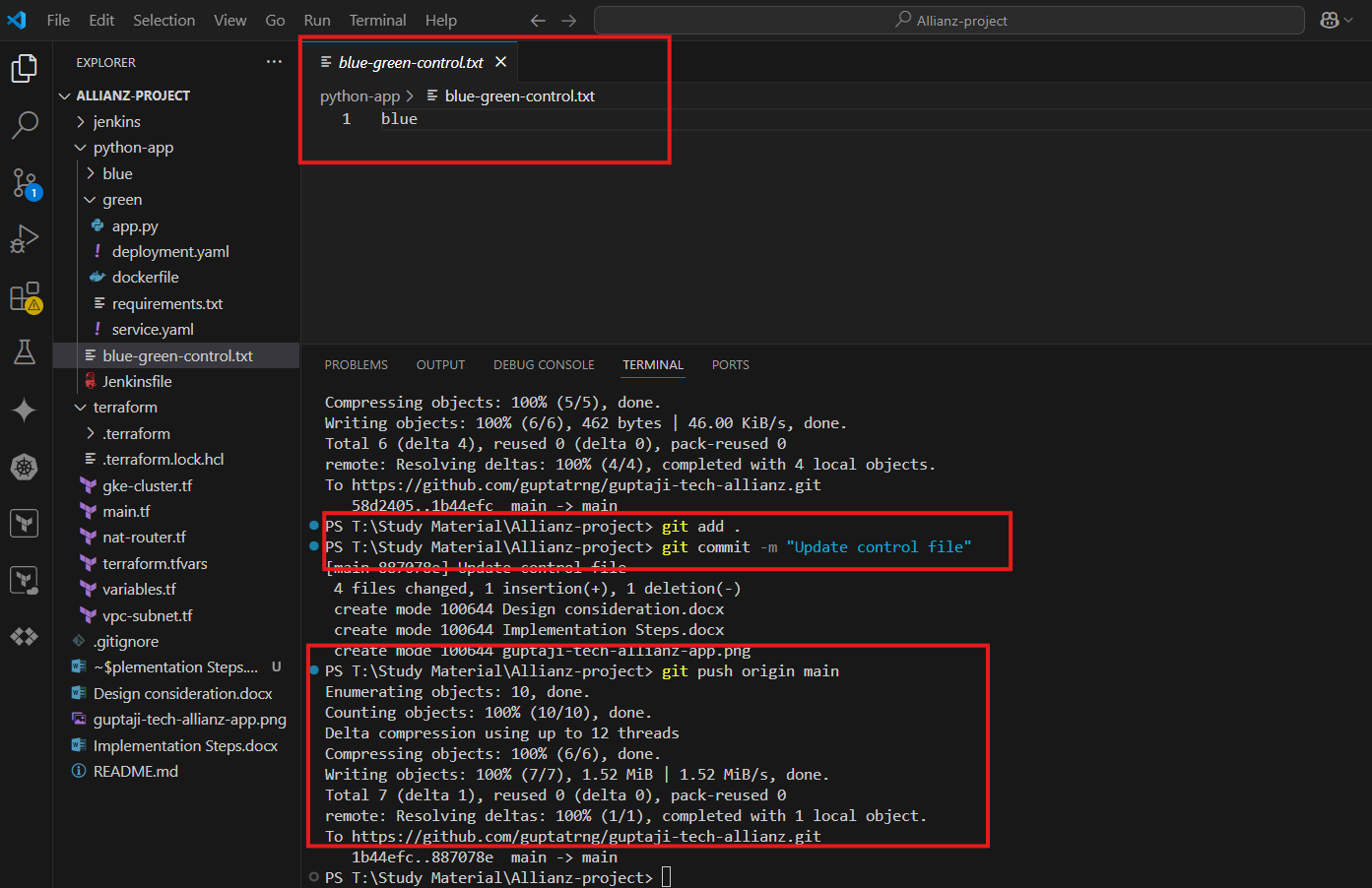
<https://github.com/guptatrng/guptaji-tech-allianz/blob/main/python-app/JenkinsFile>

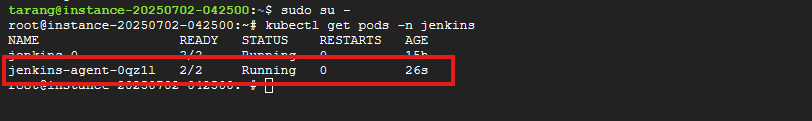
1. Create the pipeline in Jenkins and setup the webhook in GitHub to automatically trigger the pipeline as soon as code is pushed/committed to GitHub repo.

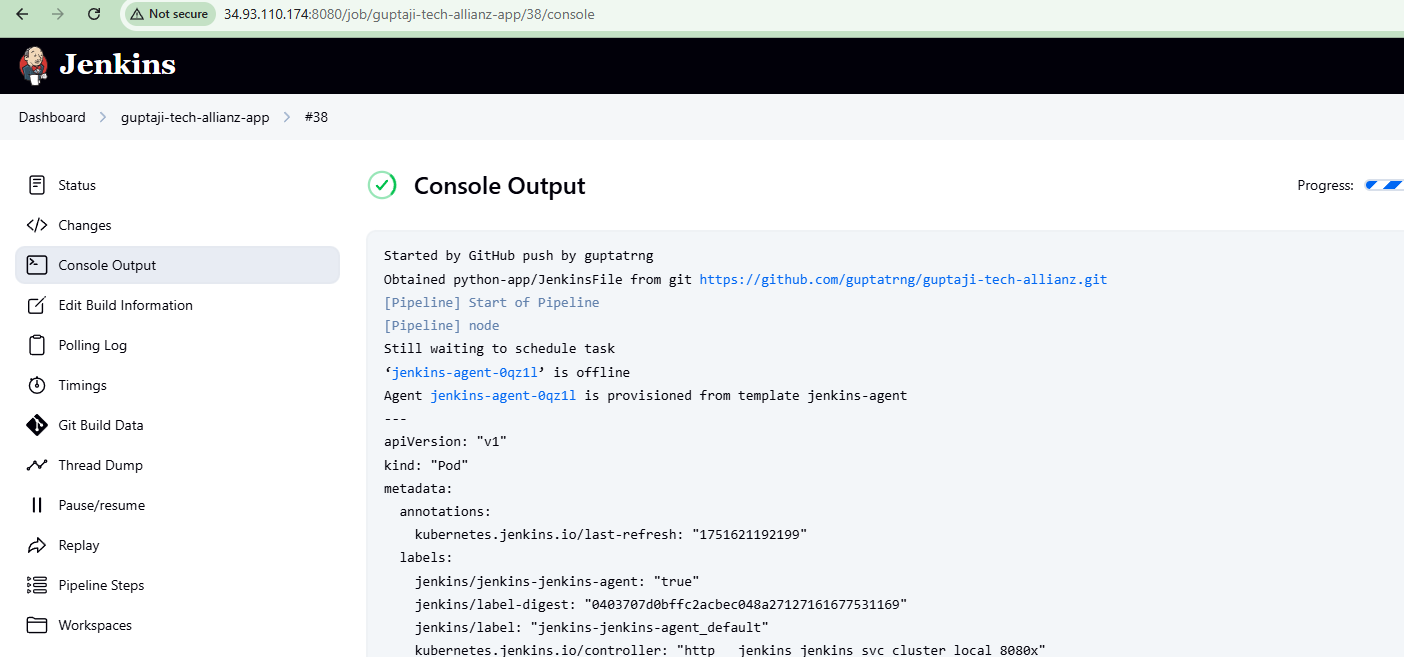




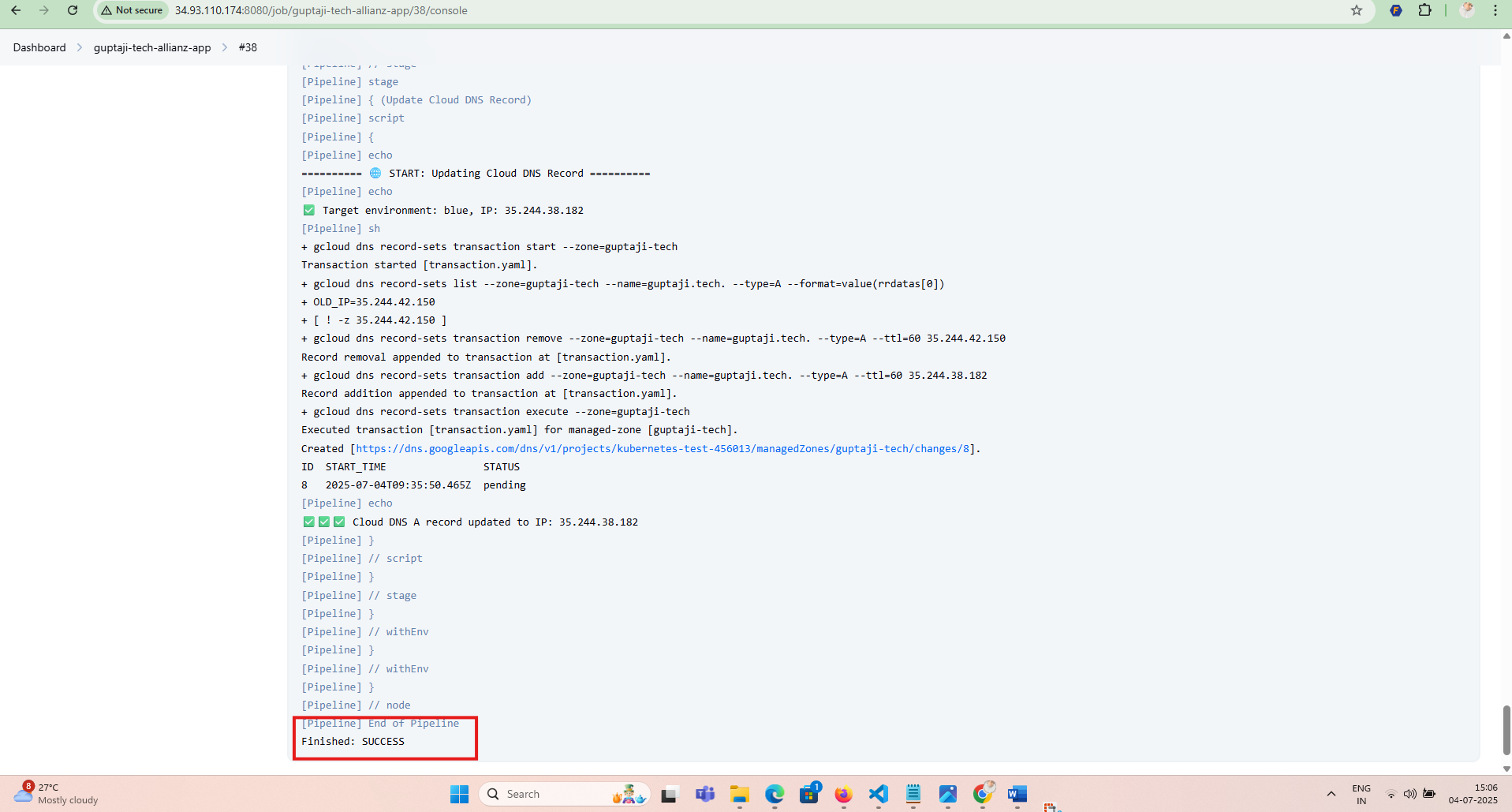
1. Update the deployment control file – blue-green-control.txt to value “blue” and commit the code in GitHub repo. The deployment pipeline will be triggered automatically.

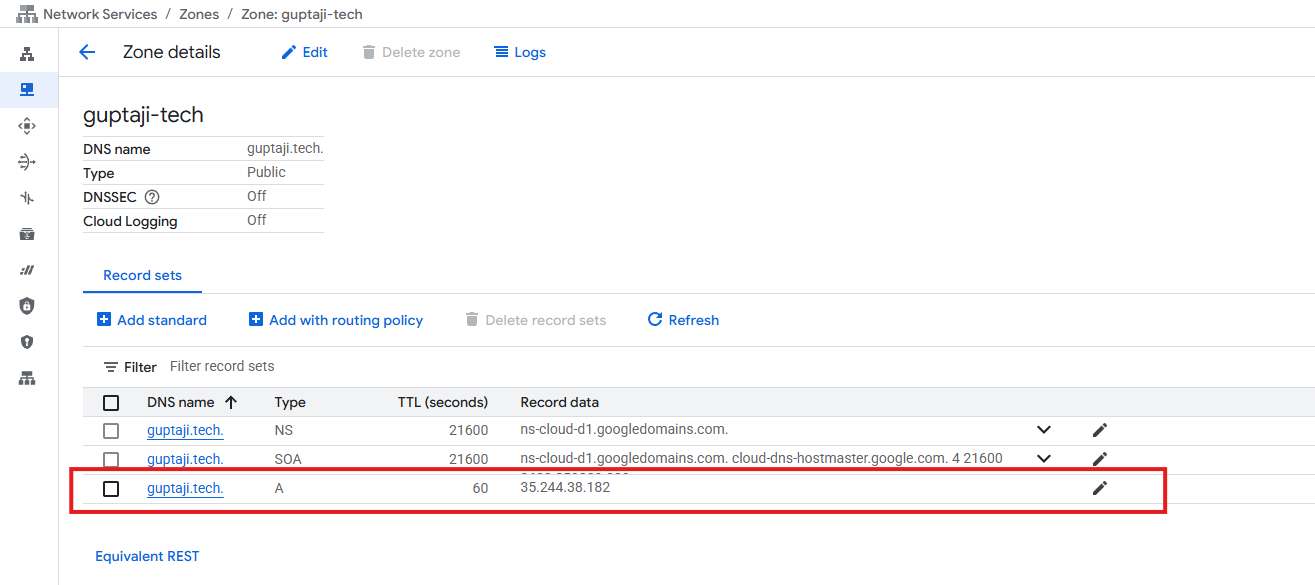


Jenkins controller will spin up new Jenkins-agent pod for the pipeline  
  


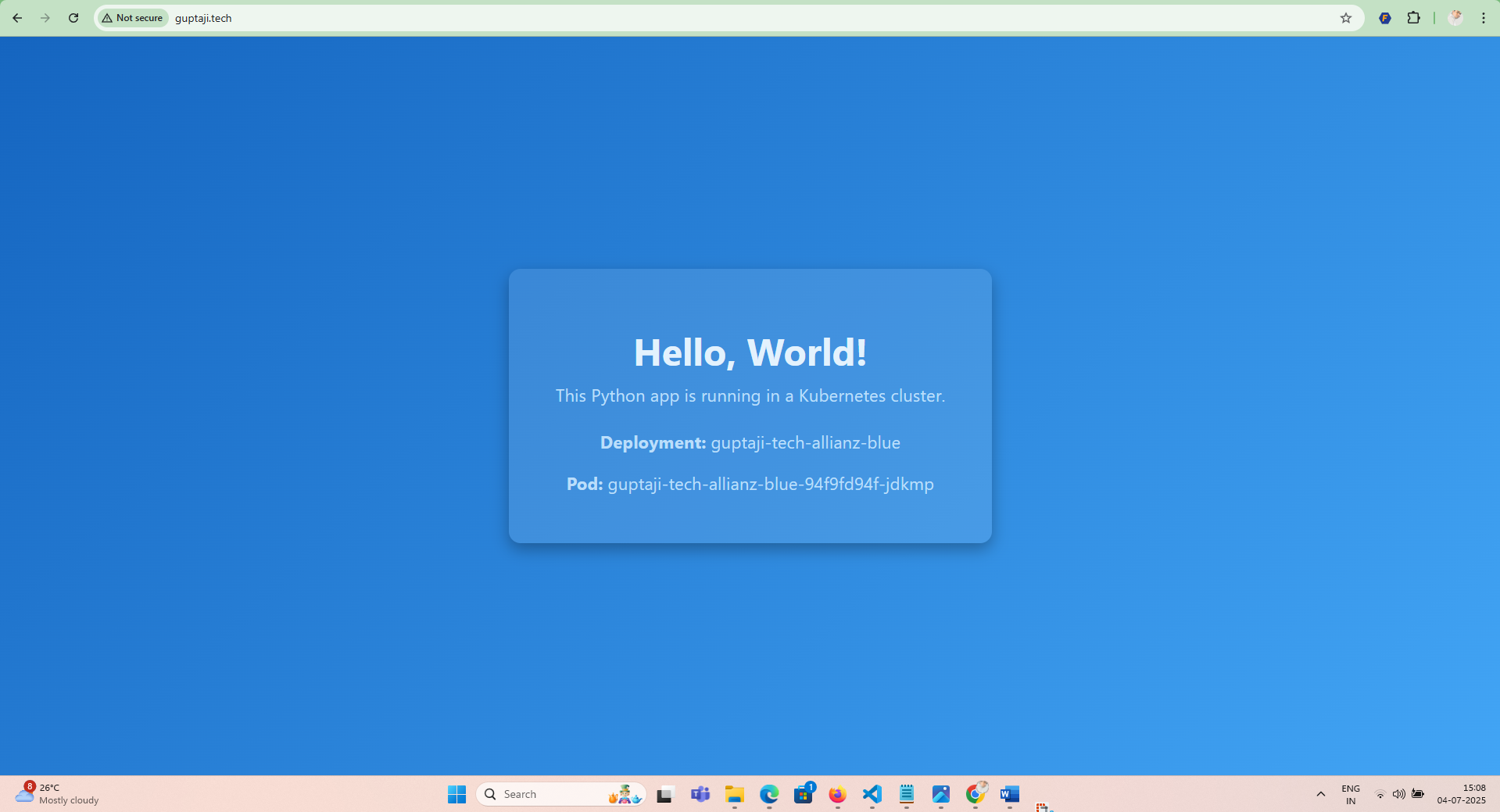
Jenkins pipeline will be triggered automatically  
  


1. Once the pipeline is completed. Check the Cloud DNS A record if it points to IP address of blue load balancer service.





1. Open the application URL => <http://guptaji.tech> to verify if the blue application is deployed successfully.



1. Repeat the deployment steps for green application and verify if it is deployed successfully in <http://guptaji.tech>