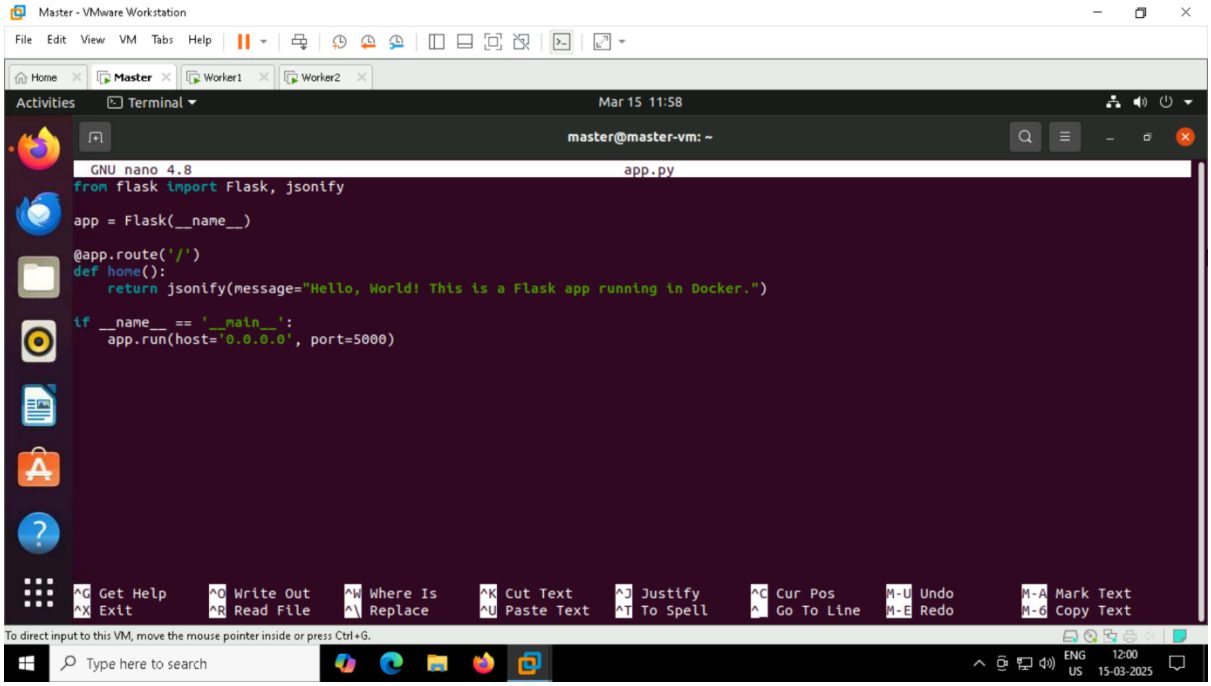
Kubernetes Case Study

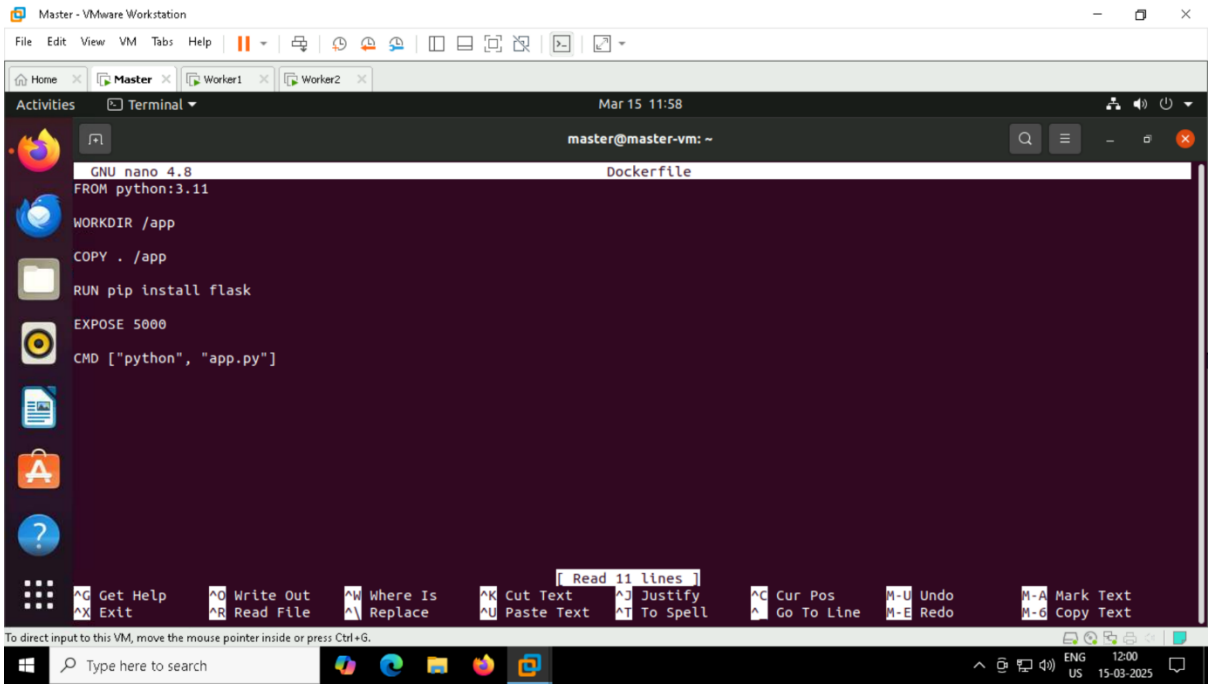
Deploying a Flask Application on Kubernetes with Auto-Scaling

Step 1: Building and Containerizing the Flask Application

- Flask Application (app.py)

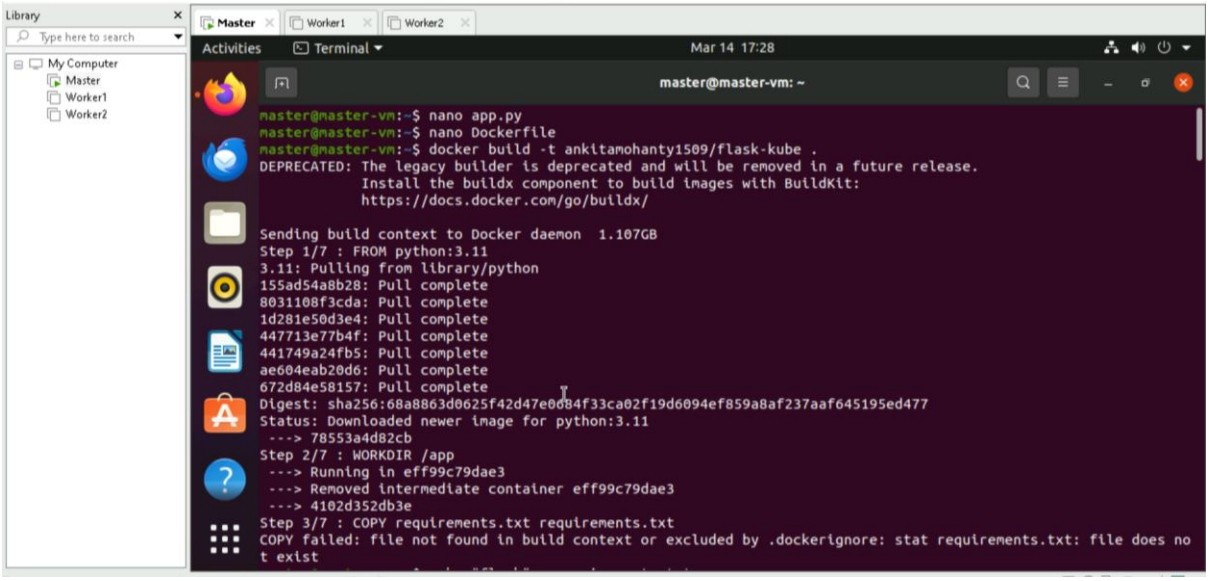


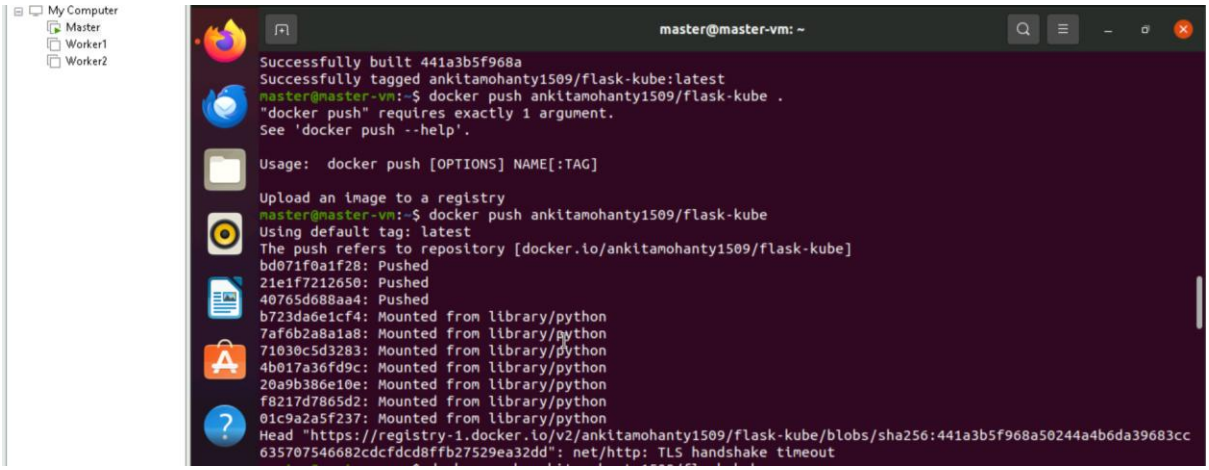
- Create a Dockerfile



Step 2: Build and Push the Image

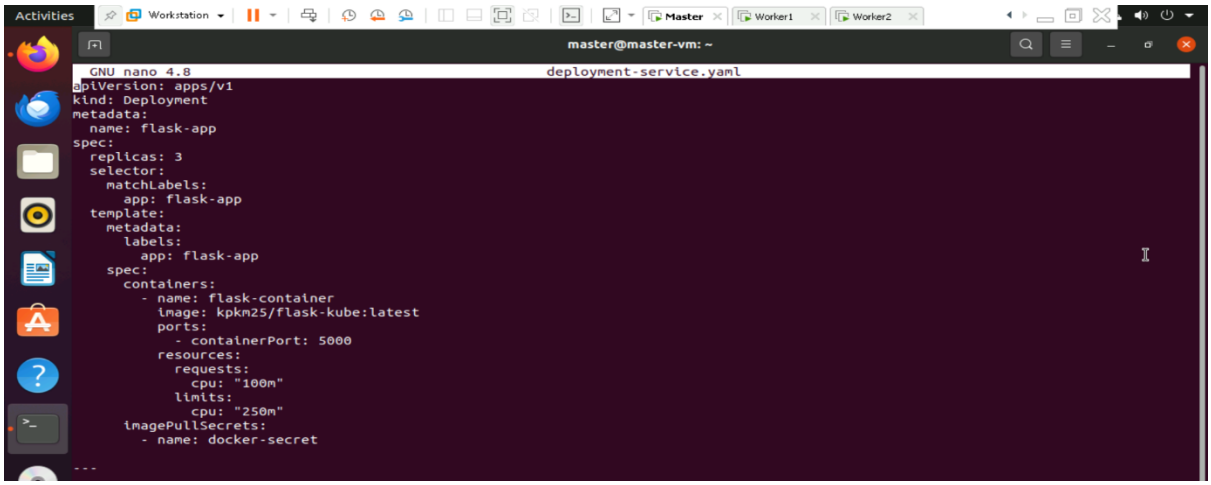
- docker build -t ankitamohanty1509/flask-kube .
- docker push ankitamohanty1509/flask-kube

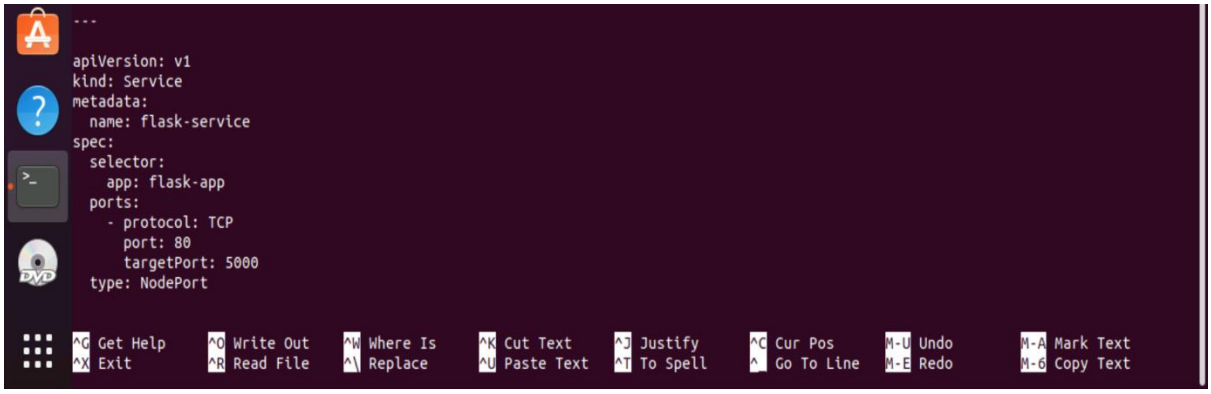




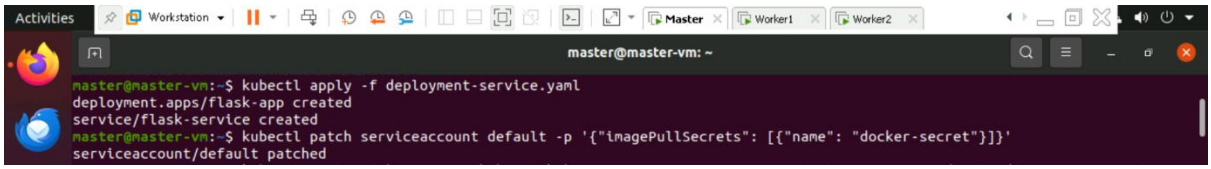
Step 3: Deploying Flask App on Kubernetes

- Create Deployment & Service YAML (deployment-service.yaml)

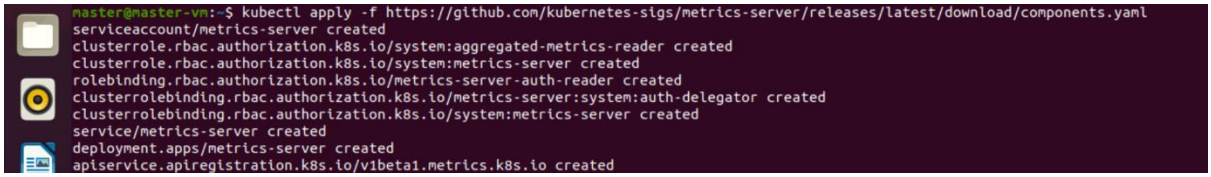




- Apply Deployment
- Patch Default Service Account

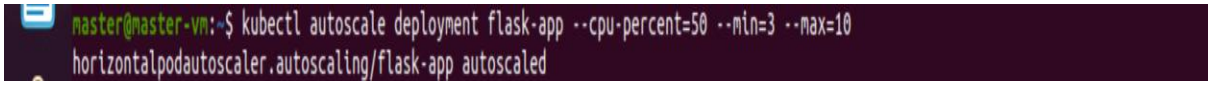


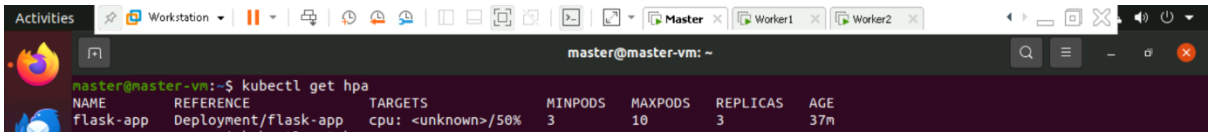
Step 4: Installing and Troubleshooting Metrics Server



Step 5: Enabling HPA (Horizontal Pod Autoscaler)

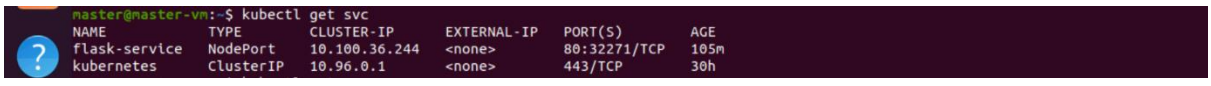
- kubectl autoscale deployment flask-app --cpu-percent=50 --min=3 --max=10
- kubectl get hpa





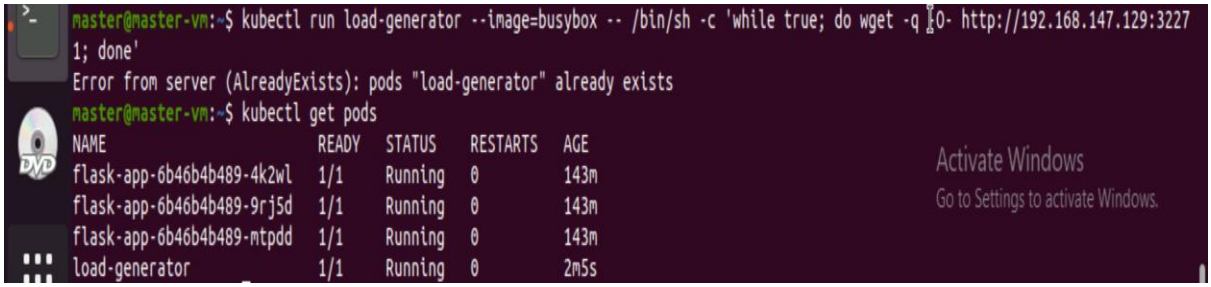
Step 6: Finding NodePort and Testing External Access

- kubectl get svc



Step 7: Simulating Load for HPA

- kubectl run -it --rm load-generator --image=busybox -- /bin/sh
- while true; do wget -q -O- http://192.168.147.129:32271; done
- kubectl get pods



Step 8: View the json output in the browser by entering the IP address