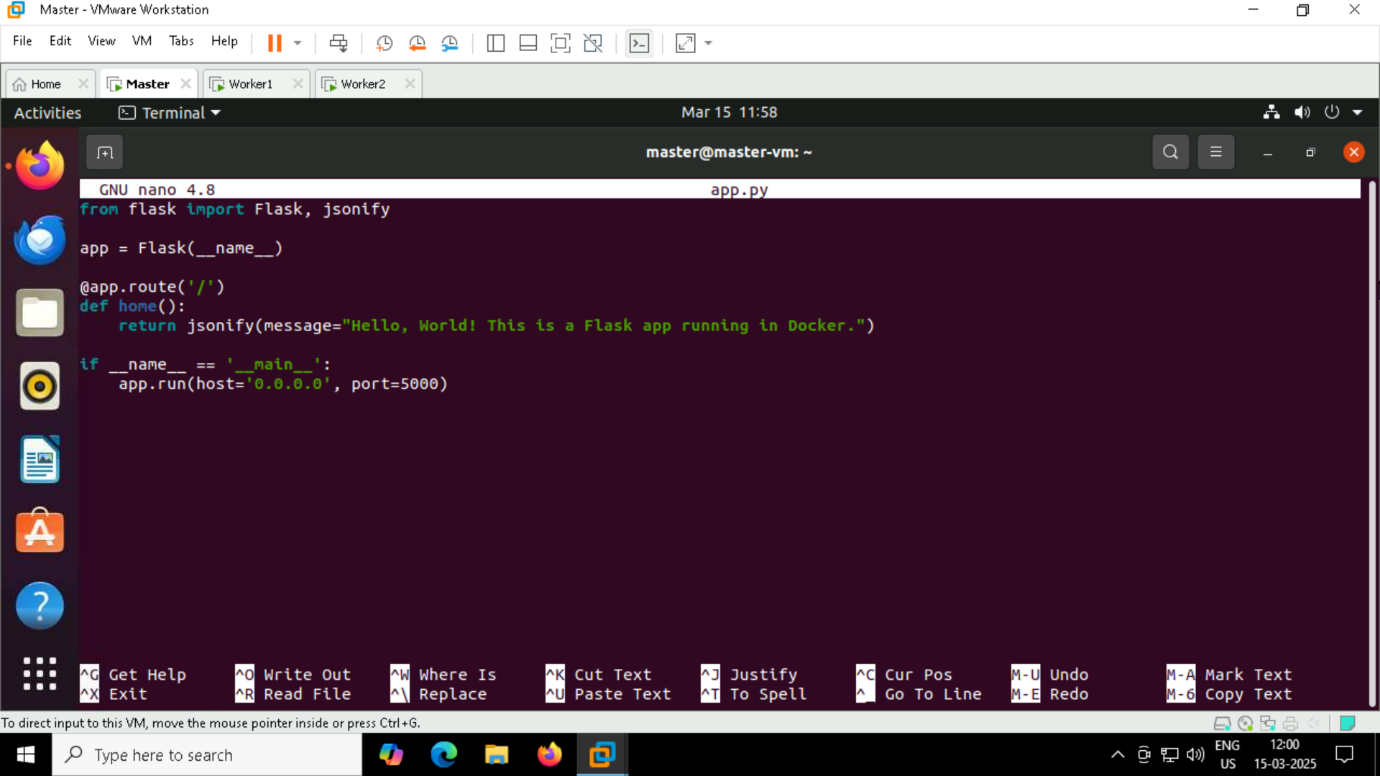
**Kubernetes Project -1**

**Deploying a Flask Application on Kubernetes with Auto-Scaling**

Step 1: Building and Containerizing the Flask Application

* Flask Application (app.py)



* Create a Dockerfile

A computer screen shot of a computer

AI-generated content may be incorrect.

Step 2: Build and Push the Image

* docker build -t pradhisha/flask-kube .
* docker push pradhisha/flask-kube

**A screenshot of a computer program

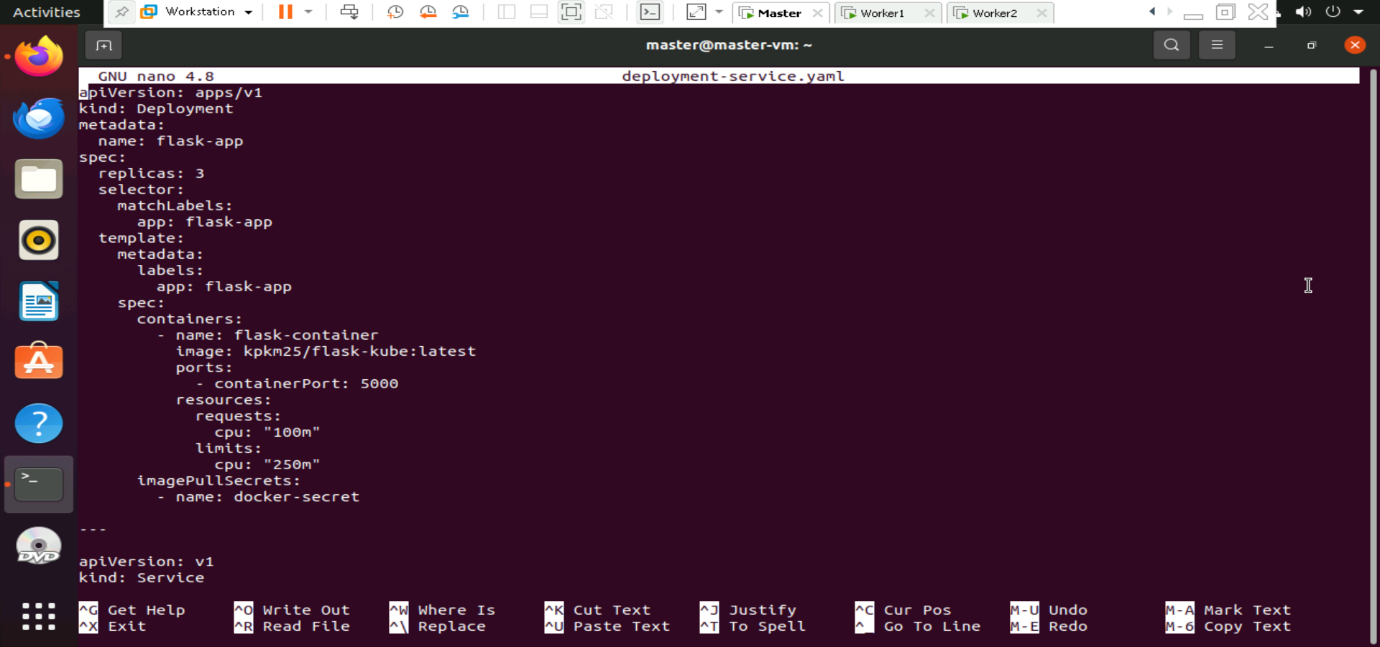
AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

Step 3: Deploying Flask App on Kubernetes

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



A screenshot of a computer

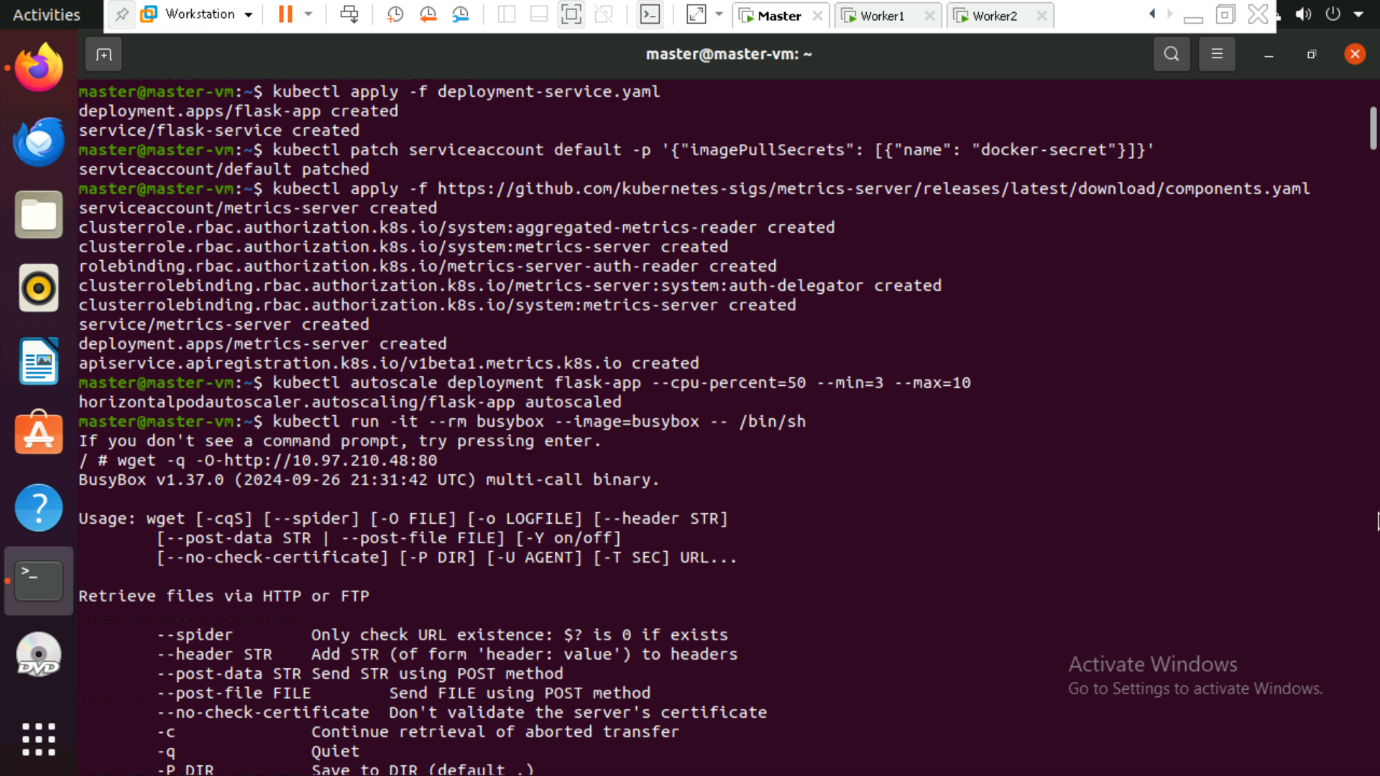
AI-generated content may be incorrect.

* Apply Deployment
* Patch Default Service Account

A screenshot of a computer program

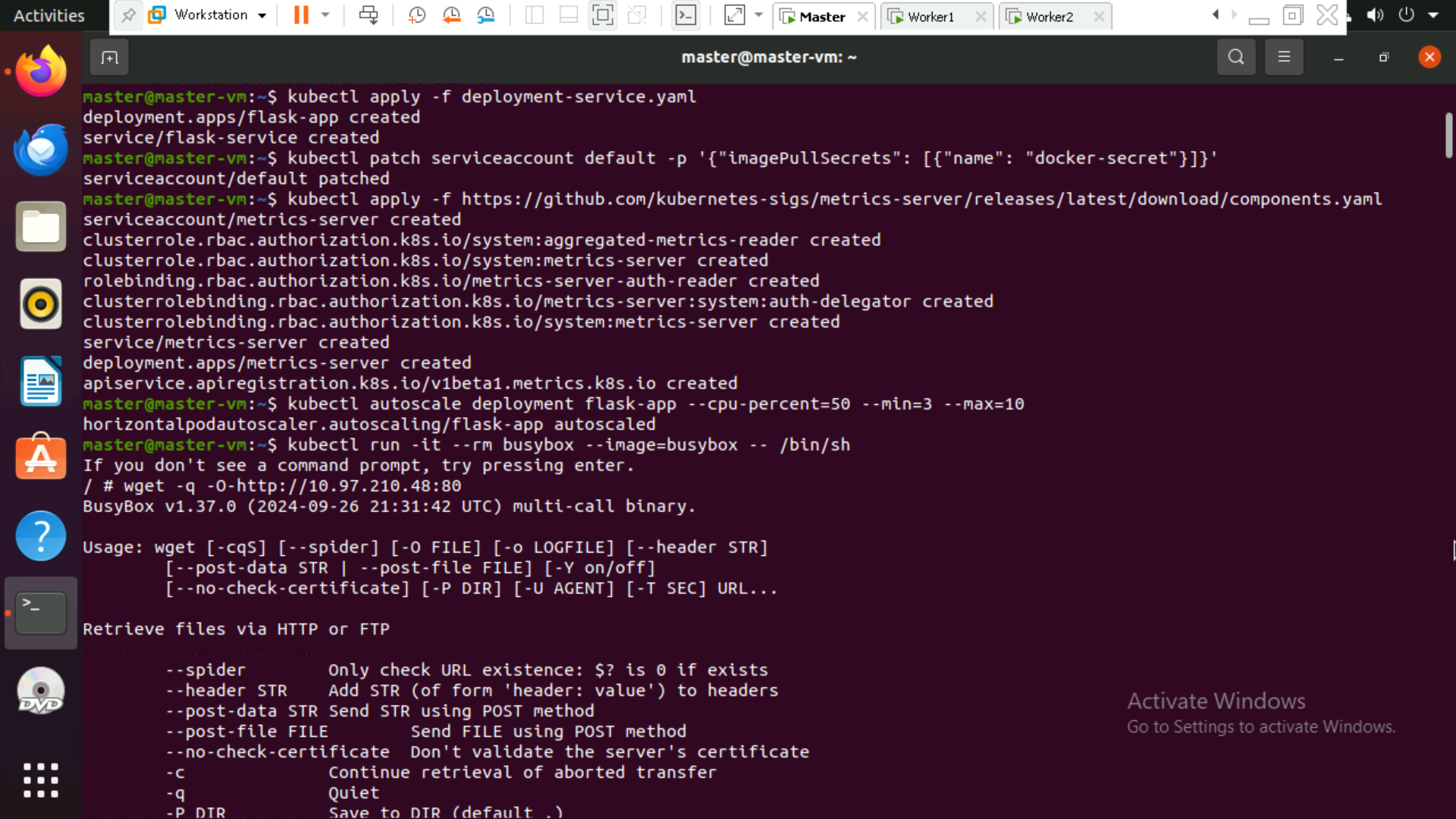
AI-generated content may be incorrect.

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



**A screenshot of a computer program

AI-generated content may be incorrect.**

Step 6: Finding NodePort and Testing External Access

* kubectl get svc

**A screenshot of a computer screen

AI-generated content may be incorrect.**

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

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

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

