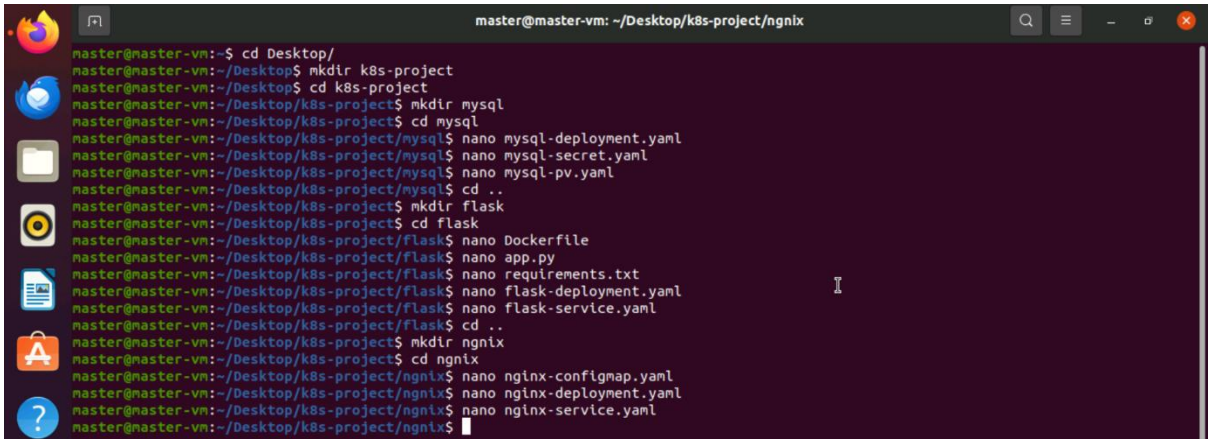
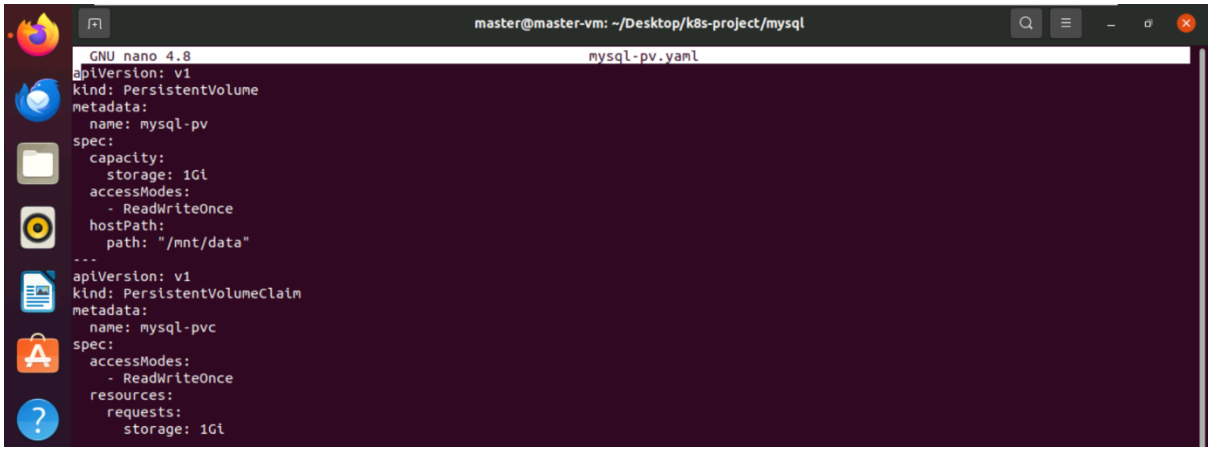
# DEPLOY A MULTI-TIER WEB APPLICATION ON KUBERNETES

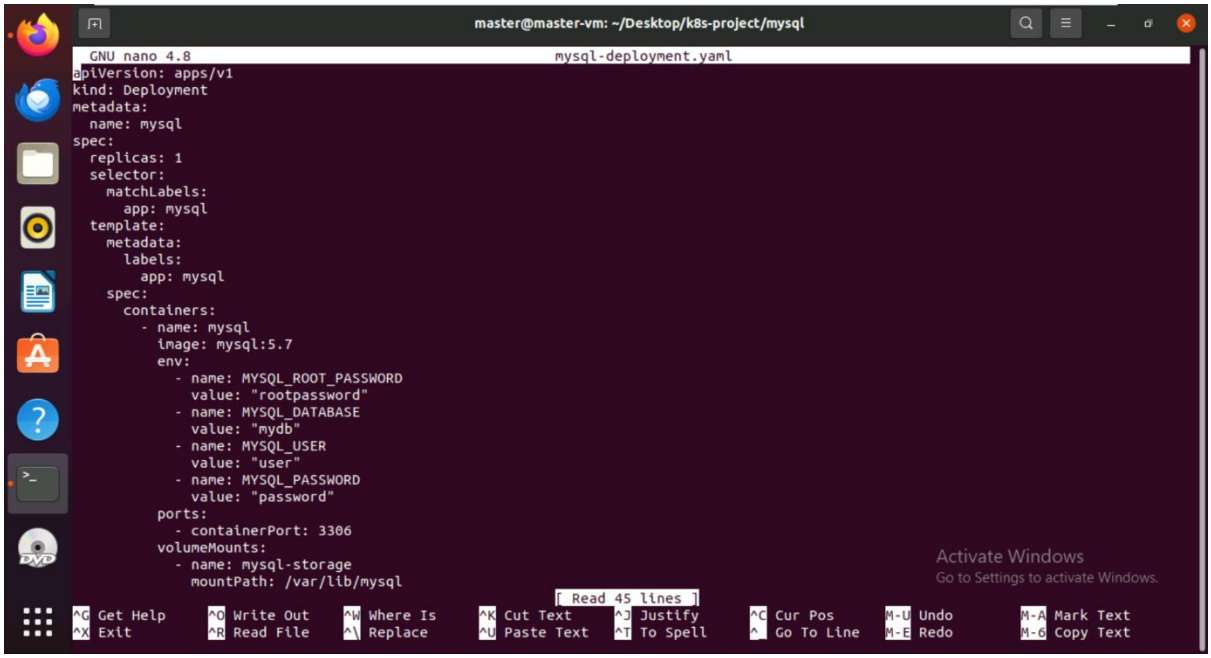
The File Structure



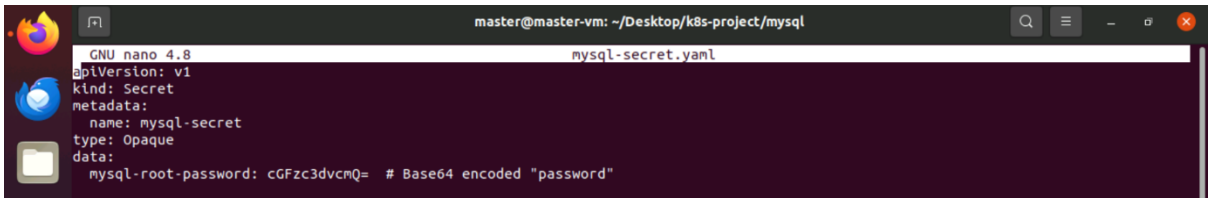

**Persistent Volume (`mysql-pv.yaml`) file**



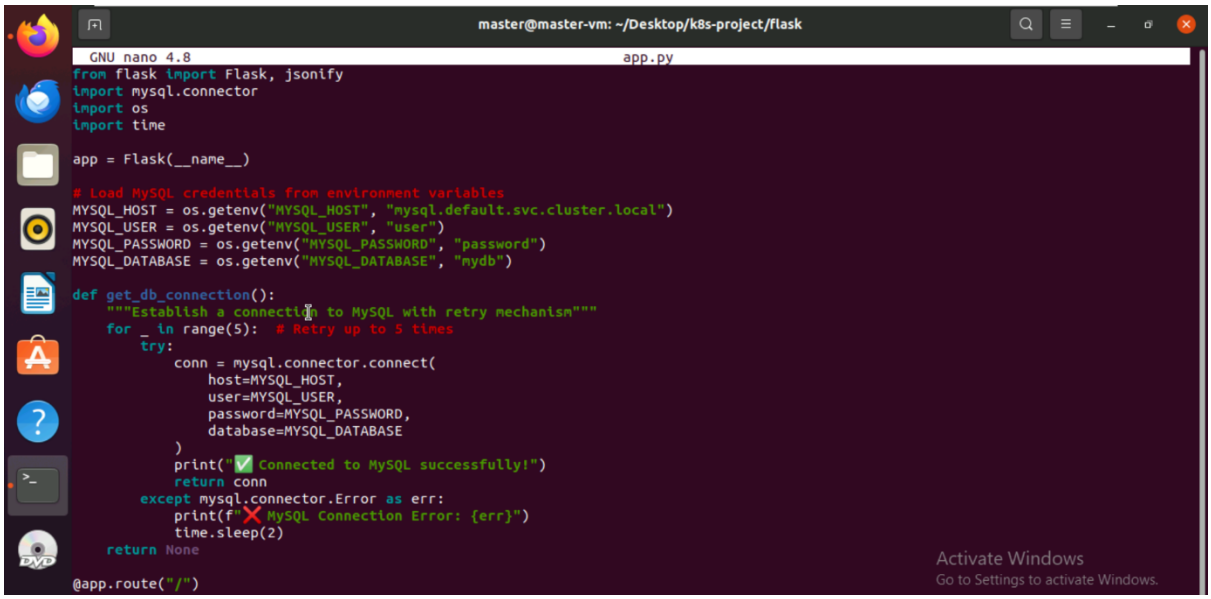

**StatefulSet (`mysql-deployment.yaml`) file**



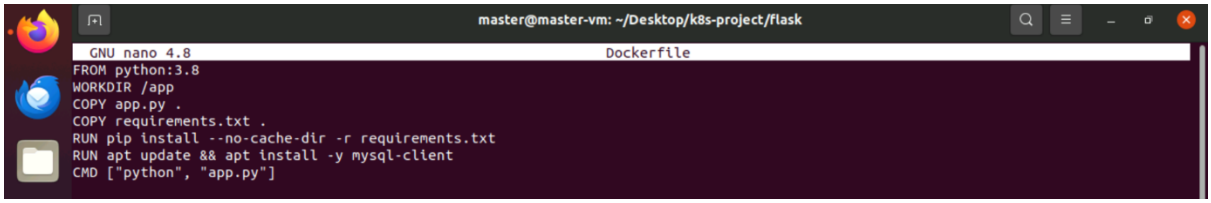
## Secret (`mysql-secret.yaml`) file



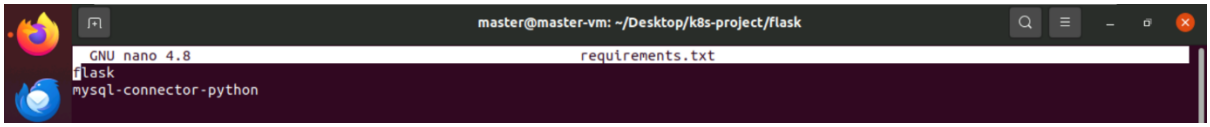

## Flask app.py file



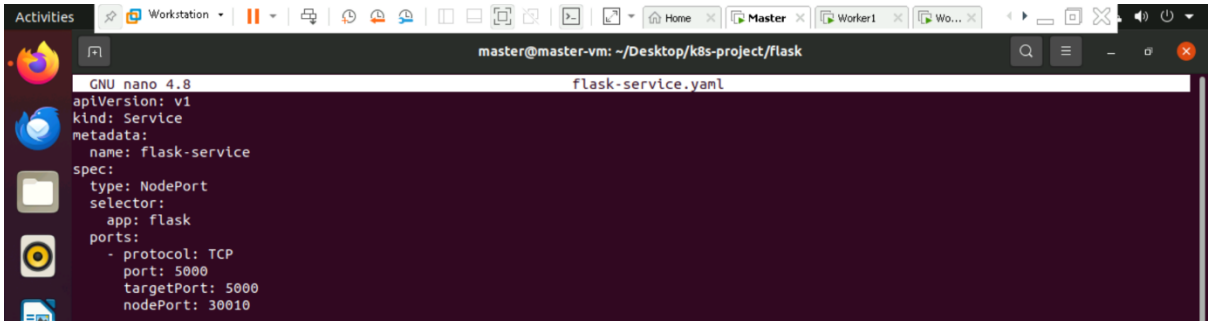

## Flask- Dcokerfile



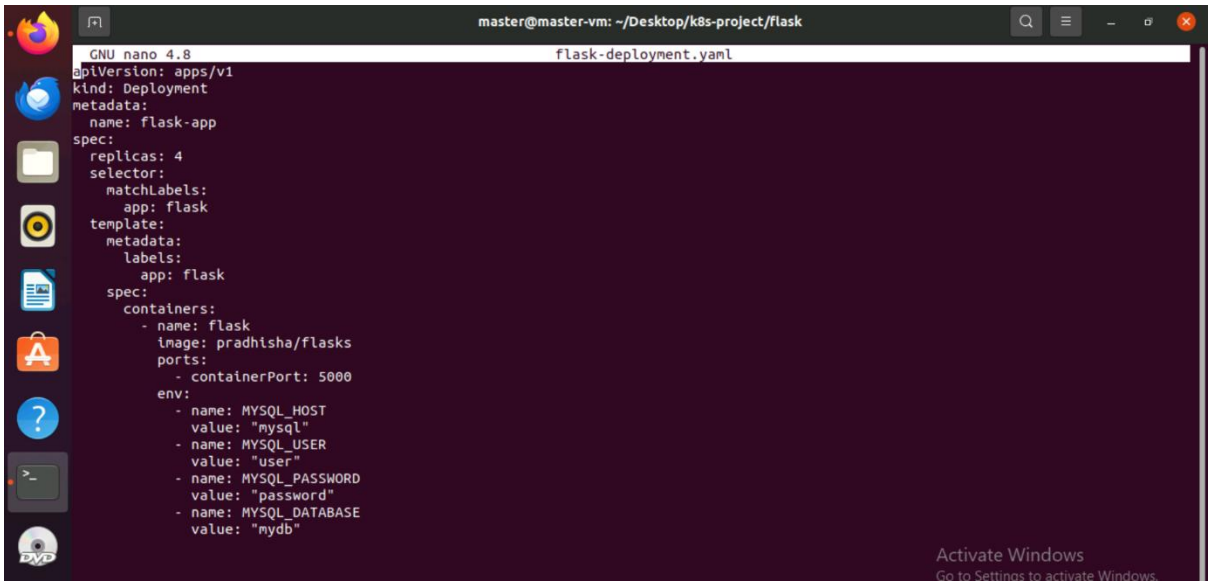

## Flask- Requirements.txt file



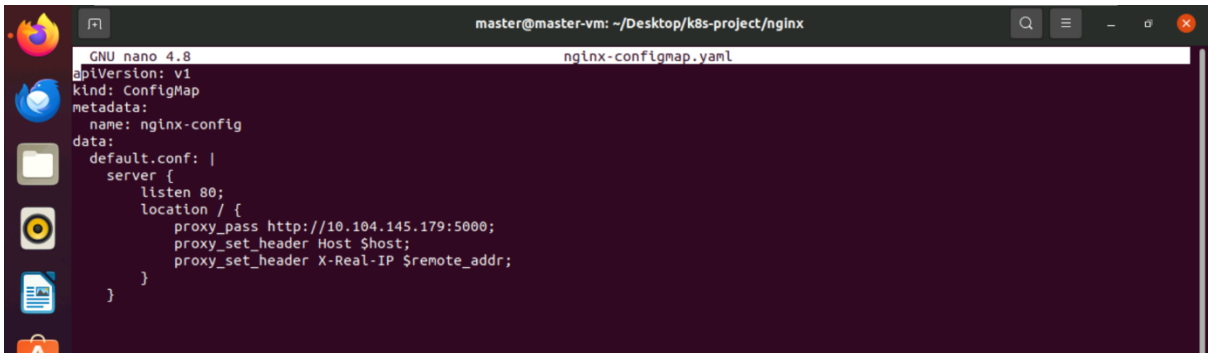

## Service (`flask-service.yaml` )



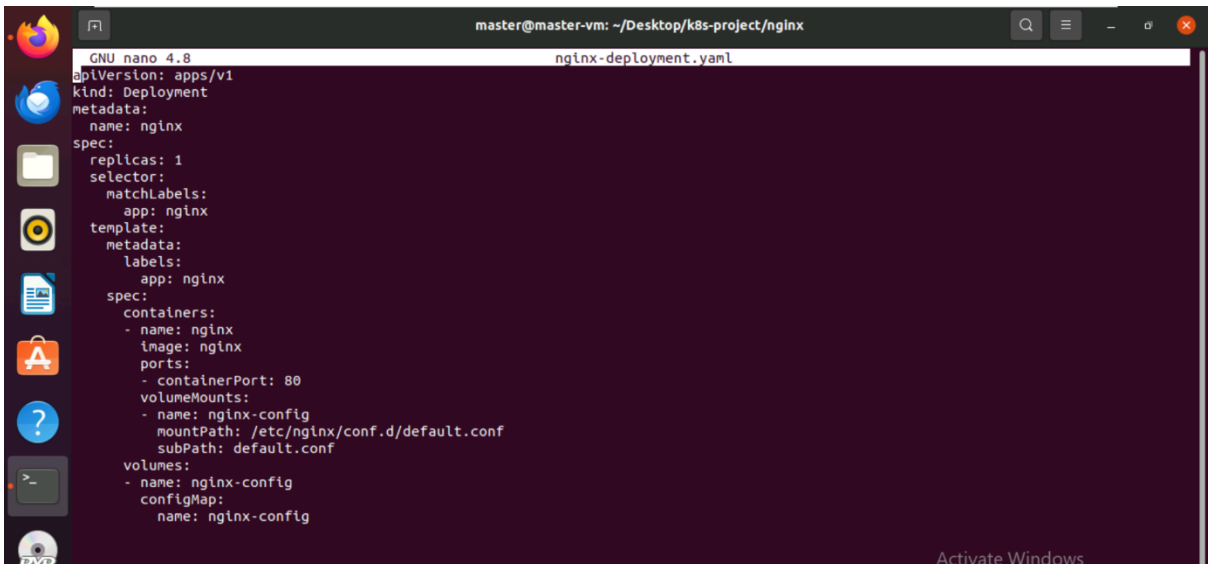
**Deployment (`flask-deployment.yaml`)**



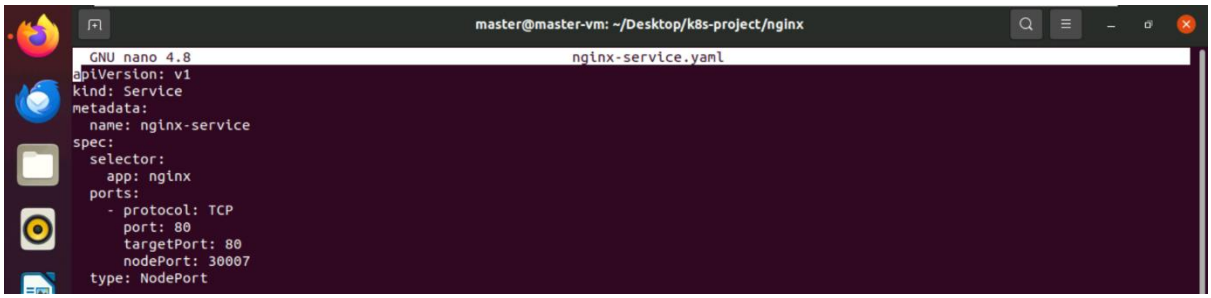

**ConfigMap (`nginx-configmap.yaml`) file**




**Deployment (`nginx-deployment.yaml`) file**



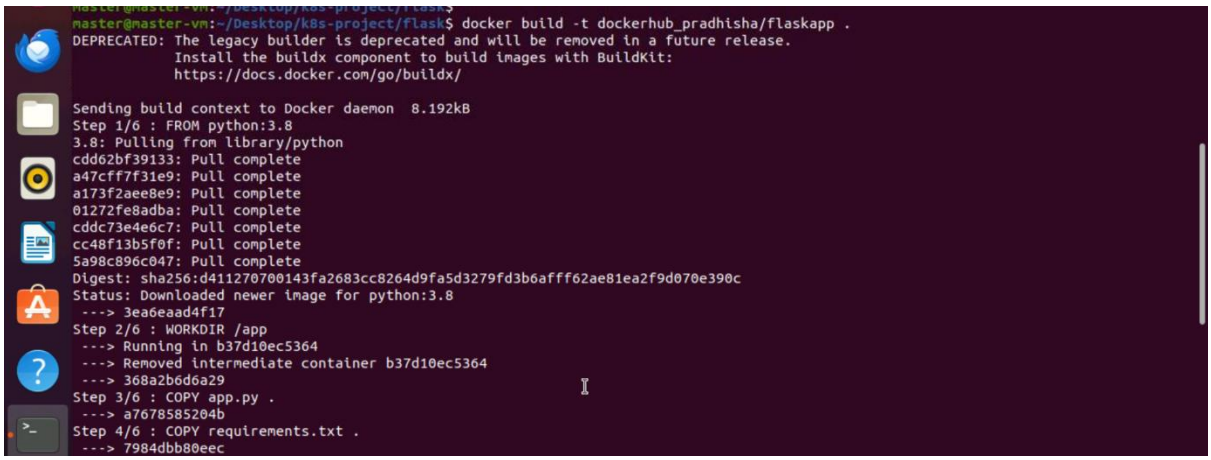
**Service (`nginx-service.yaml`) file**



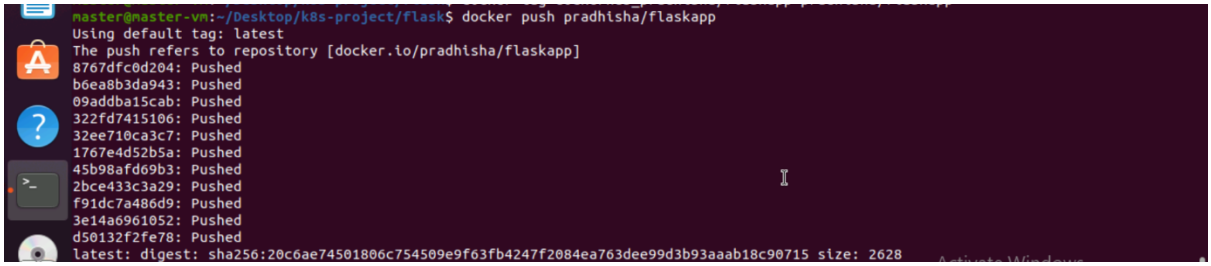

## 1. Build and Push Docker Image

1. Navigate to the Flask application directory:
    - cd flask


2. Build the Docker image:
    - docker build -t dockerhub_username/flaskapp .

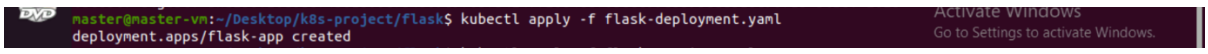



1. Push the image to Docker Hub:
    - docker push dockerhub_username/flaskapp

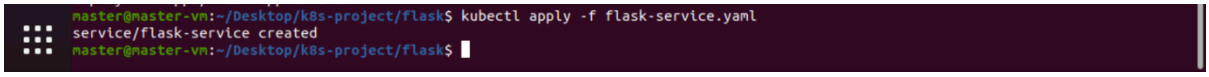



## 2. Deploy the Flask Application

4. Apply the Flask Deployment manifest:
    - kubectl apply -f flask-deployment.yaml



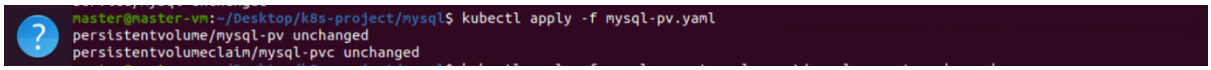
5. Apply the Flask Service manifest:
    - kubectl apply -f flask-service.yaml



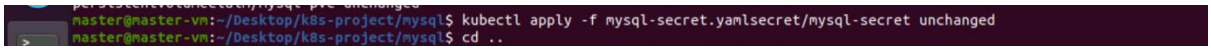

**3. Deploy the MySQL Database**

6. Navigate to the MySQL directory:
    - cd .. && cd mysql
7. Apply the MySQL Deployment manifest:
    - kubectl apply -f mysql-deployment.yaml


8. Apply the MySQL Persistent Volume manifest:
    - kubectl apply -f mysql-pv.yaml

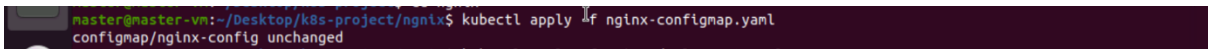



9. Apply the MySQL Secret manifest:
    - kubectl apply -f mysql-secret.yaml

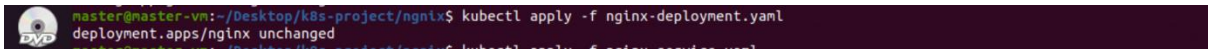



**4. Deploy the Nginx Service**

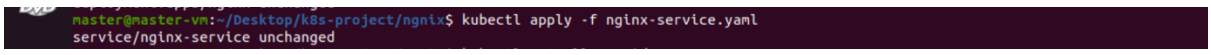
10. Navigate to the Nginx directory:
    - cd .. && cd nginx
11. Apply the Nginx ConfigMap manifest:
    - kubectl apply -f nginx-configmap.yaml




12. Apply the Nginx Deployment manifest:
    - kubectl apply -f nginx-deployment.yaml



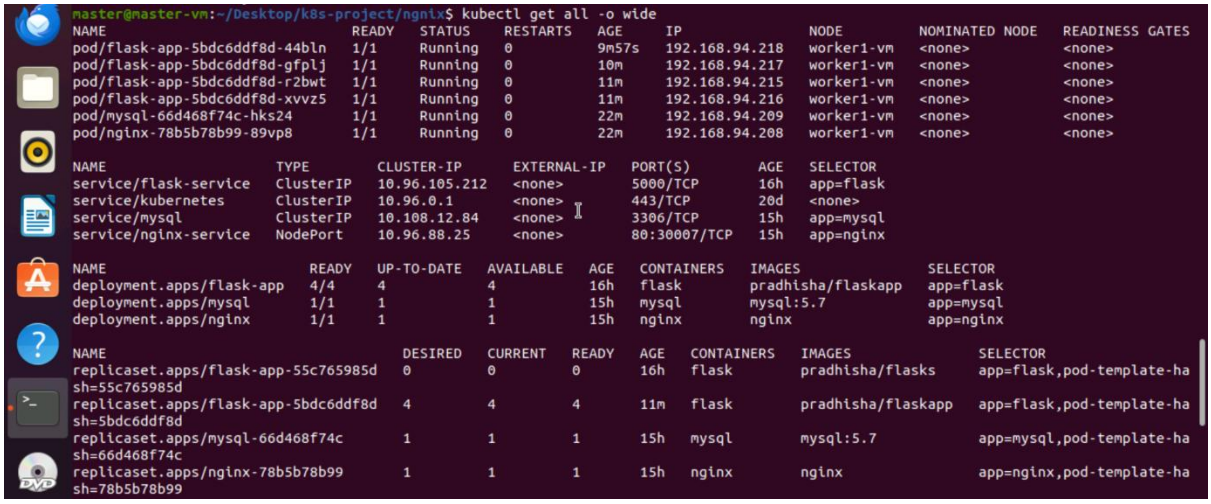

13. Apply the Nginx Service manifest:
    - kubectl apply -f nginx-service.yaml



## 5. Verify the Deployment

14. Check the status of all Kubernetes resources:

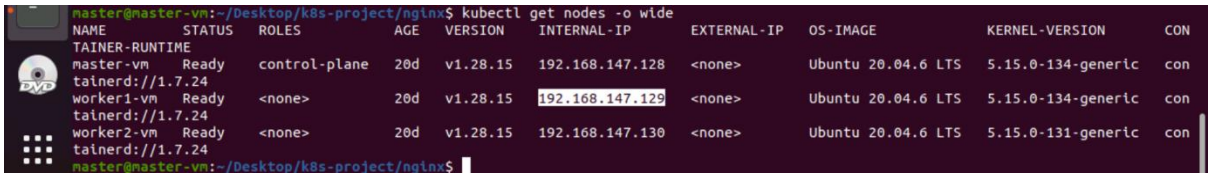
    - kubectl get all -o wide




## 6. Access the Application

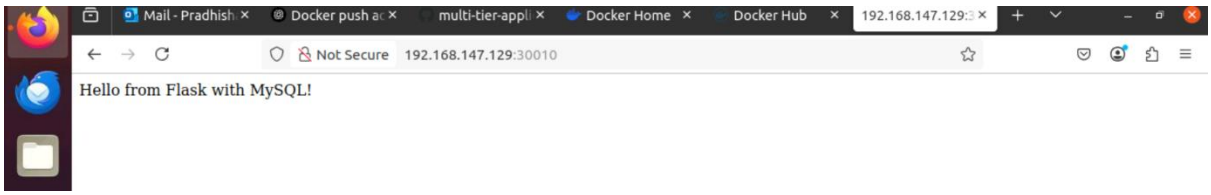
15. Retrieve the worker node IP:

    - kubectl get nodes -o wide




16. Access the application in the browser:

http://192.168.147.129:30010/

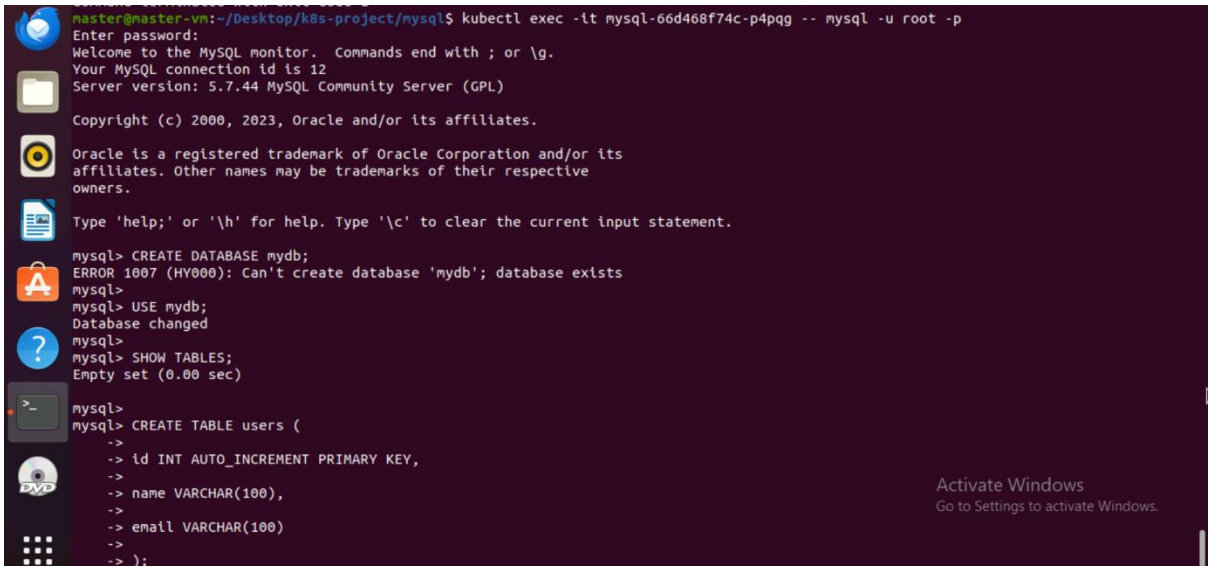


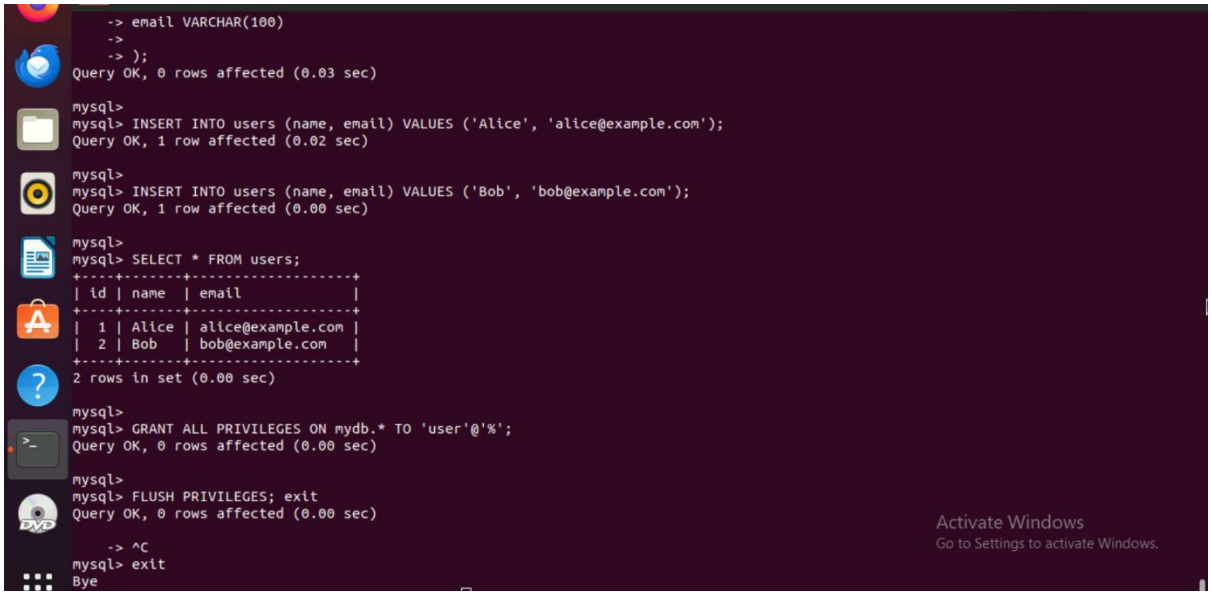

## 7. Configure MySQL Database

17. Access the MySQL pod:

kubectl exec -it mysql-0 -- mysql -u root -p

18. Create and configure the database:




19. Access the users API endpoint:

http://192.168.147.129:30010/users