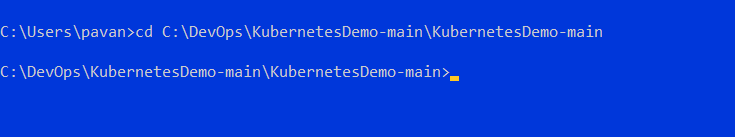
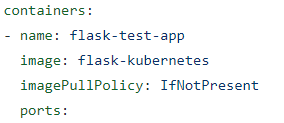
Kubernets Demonstartion

1. Download the code from github <https://github.com/sandeepdoodigani/KubernetesDemo>
2. Open command prompt and navigate to the code file. Here we are creating the docker image

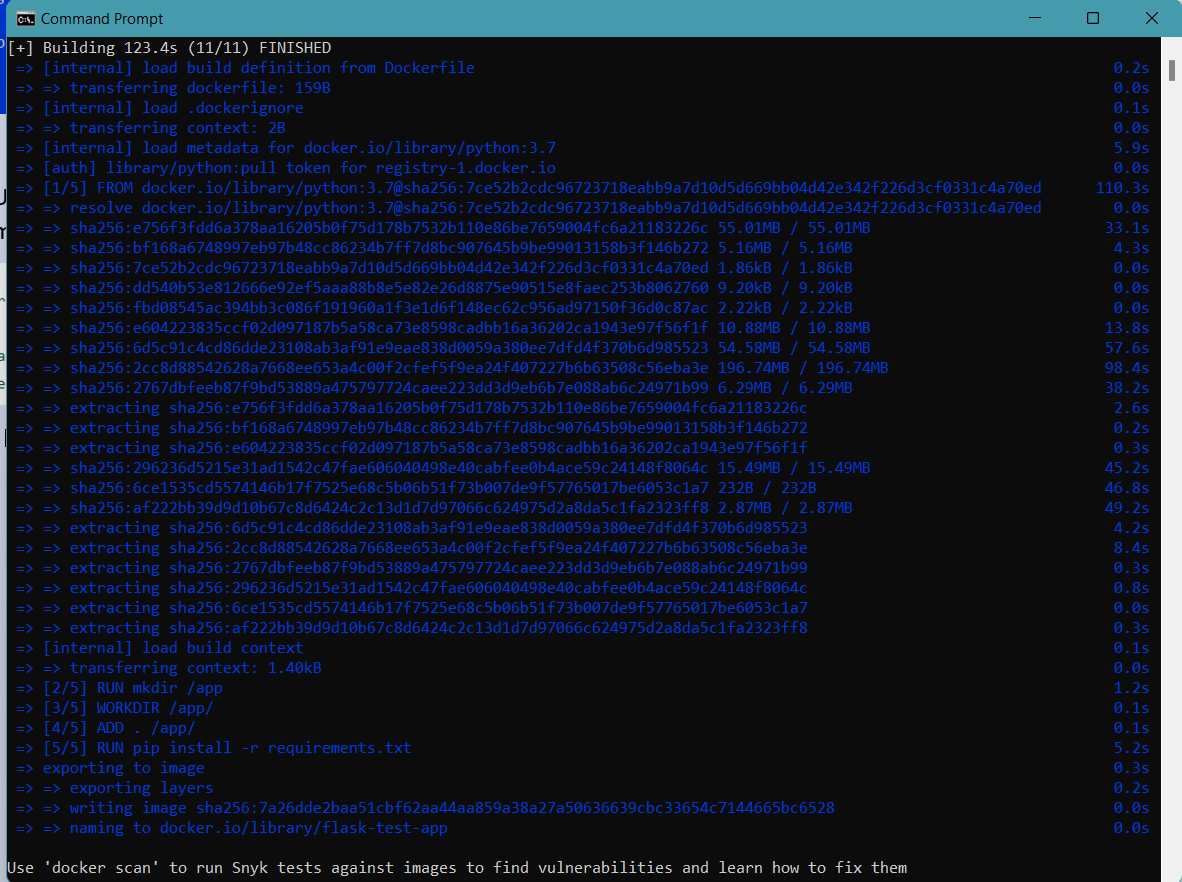
cd C:\DevOps\KubernetesDemo-main\KubernetesDemo-main



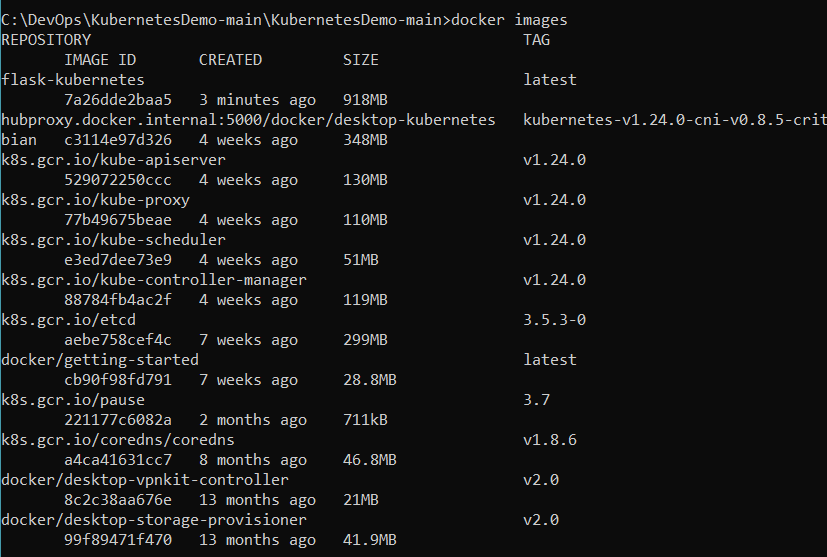
1. Use the following commands.Remember the name which we had mentioned in kubernetes should be the same as the name of the image



docker build -t flask-kubernetes .

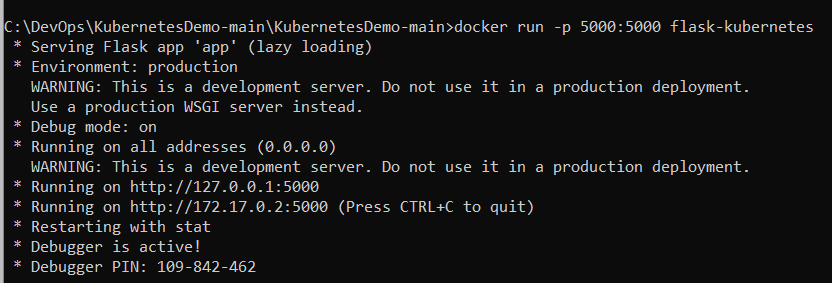


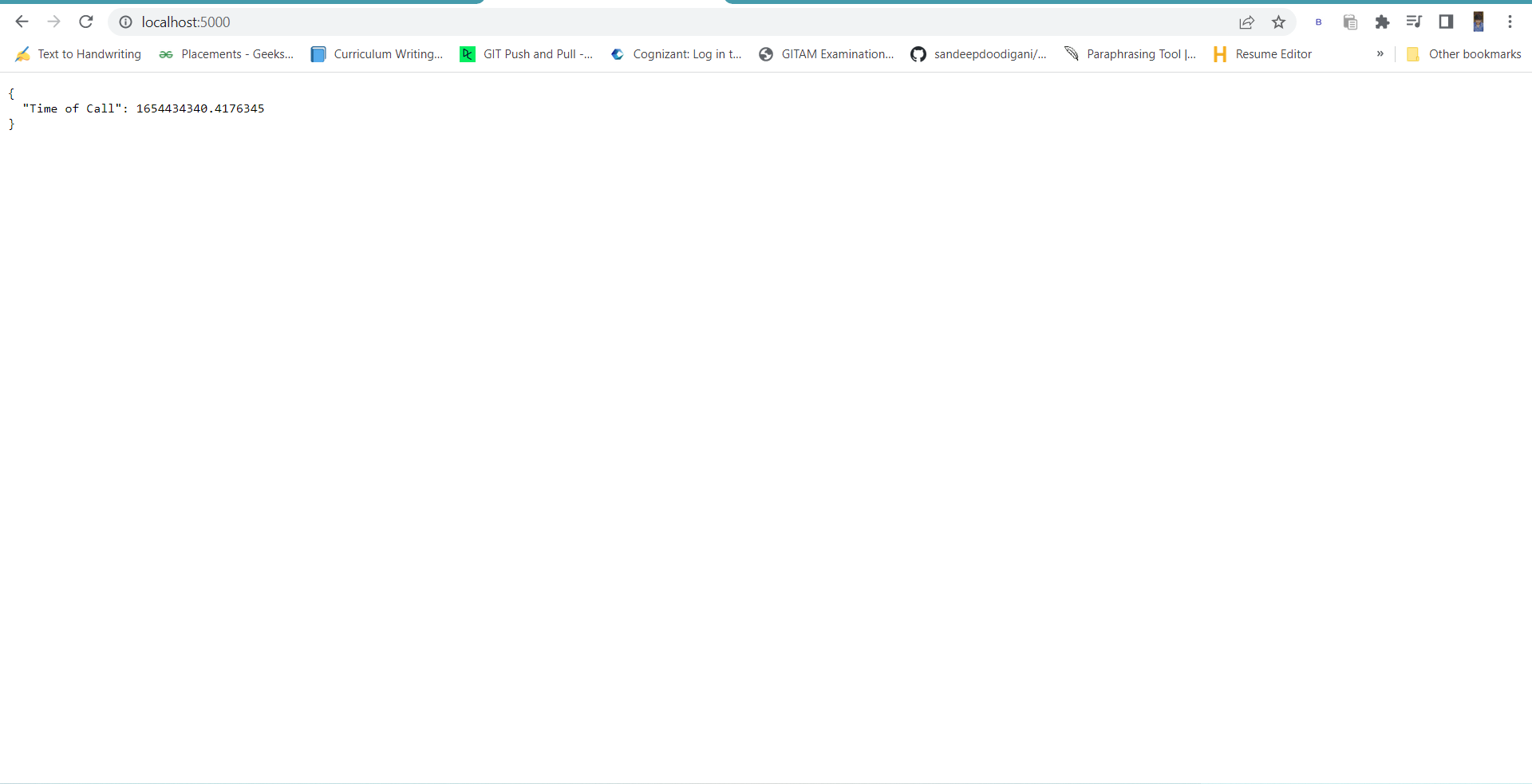
1. Now check the image is present in the repository



1. Now we are running the application on the local host

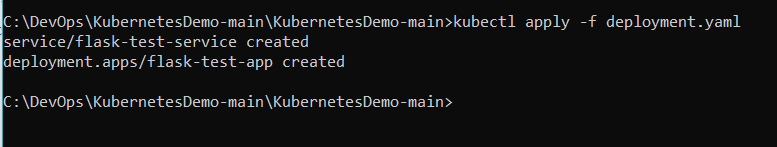
docker run -p 5000:5000 flask-kubernetes



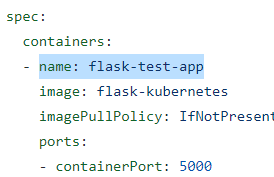


1. Now we are going to deploy the application in kubernetes cluster

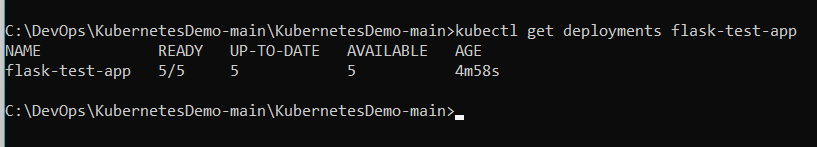
kubectl apply -f deployment.yaml



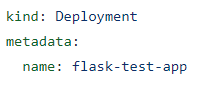
7. In the YAML file, the deployment name is name: flask-test-app



-kubectl get deployments flask-test-app

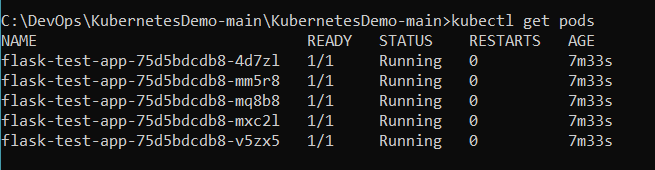


flask-test-app= since the name of the file is so mentioned



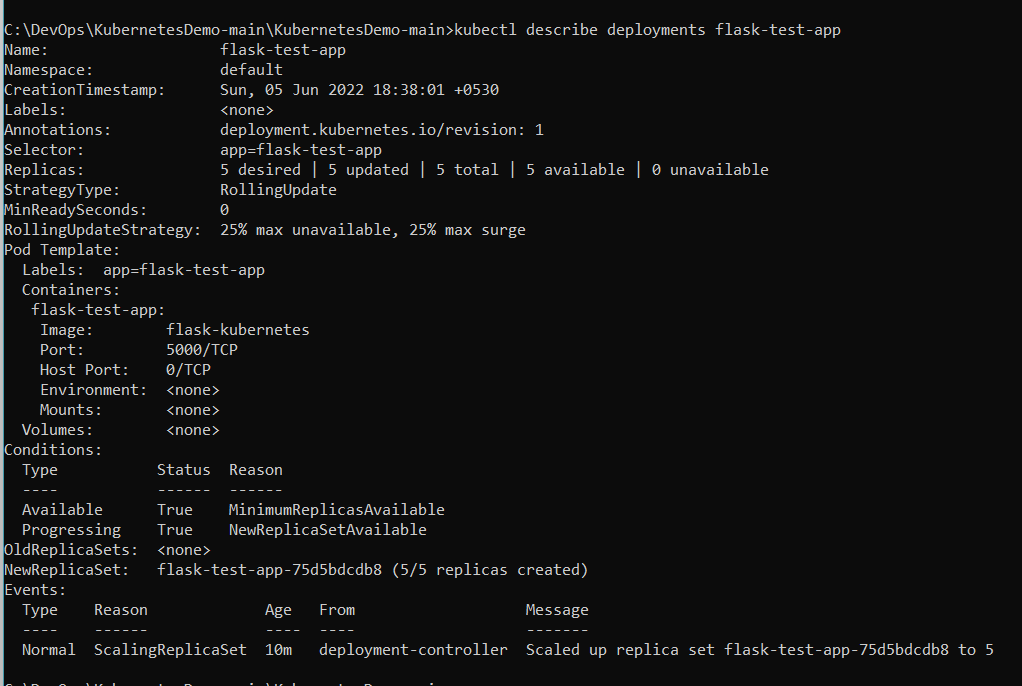
1. We can also get all the pods using the ollowing command

kubectl get pods



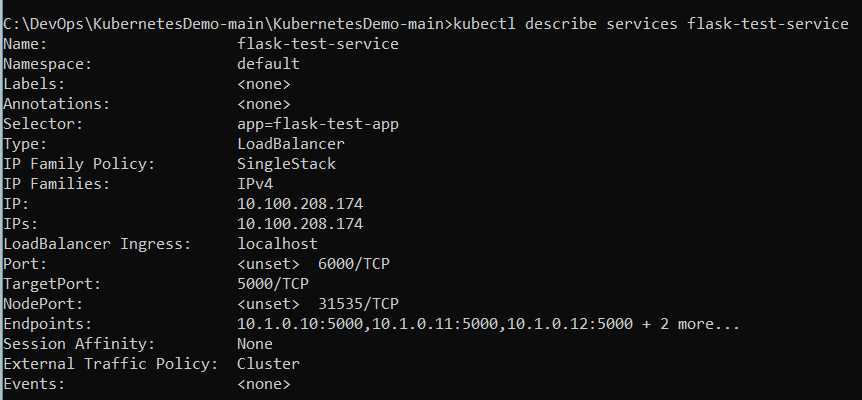
1. Inorder to get the configuration of the deployement we use the following commands

kubectl describe deployments flask-test-app



1. Now we are deploying for the service

kubectl describe services flask-test-service

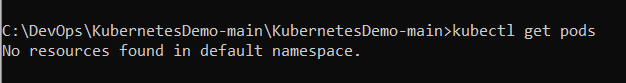


Now we need to run the port

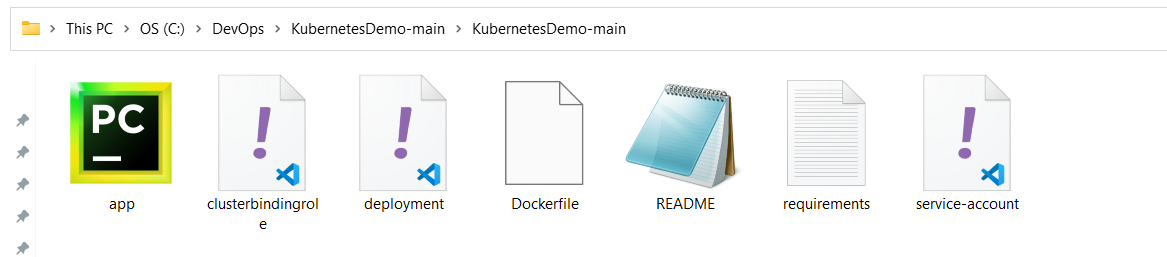
localhost:31535

kubectl delete deployment flask-test-app

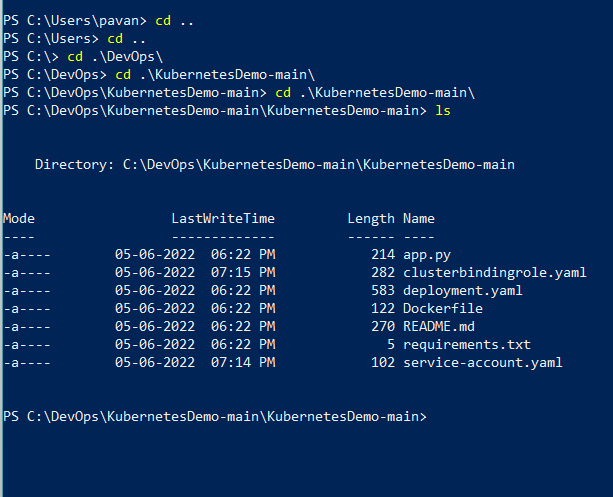




Now set up the required files in the folder

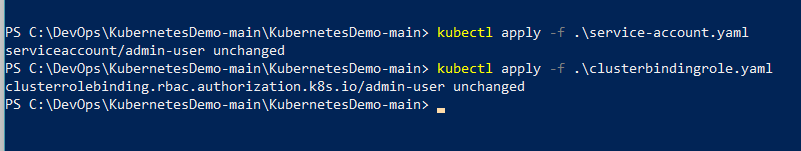


C:\DevOps\KubernetesDemo-main\KubernetesDemo-main>



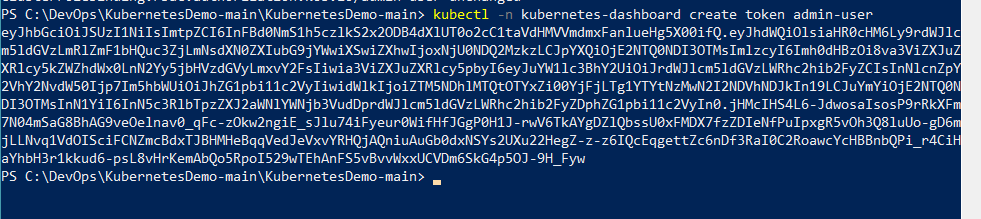
kubectl apply -f .\service-account.yaml

kubectl apply -f .\clusterbindingrole.yaml



kubectl proxy

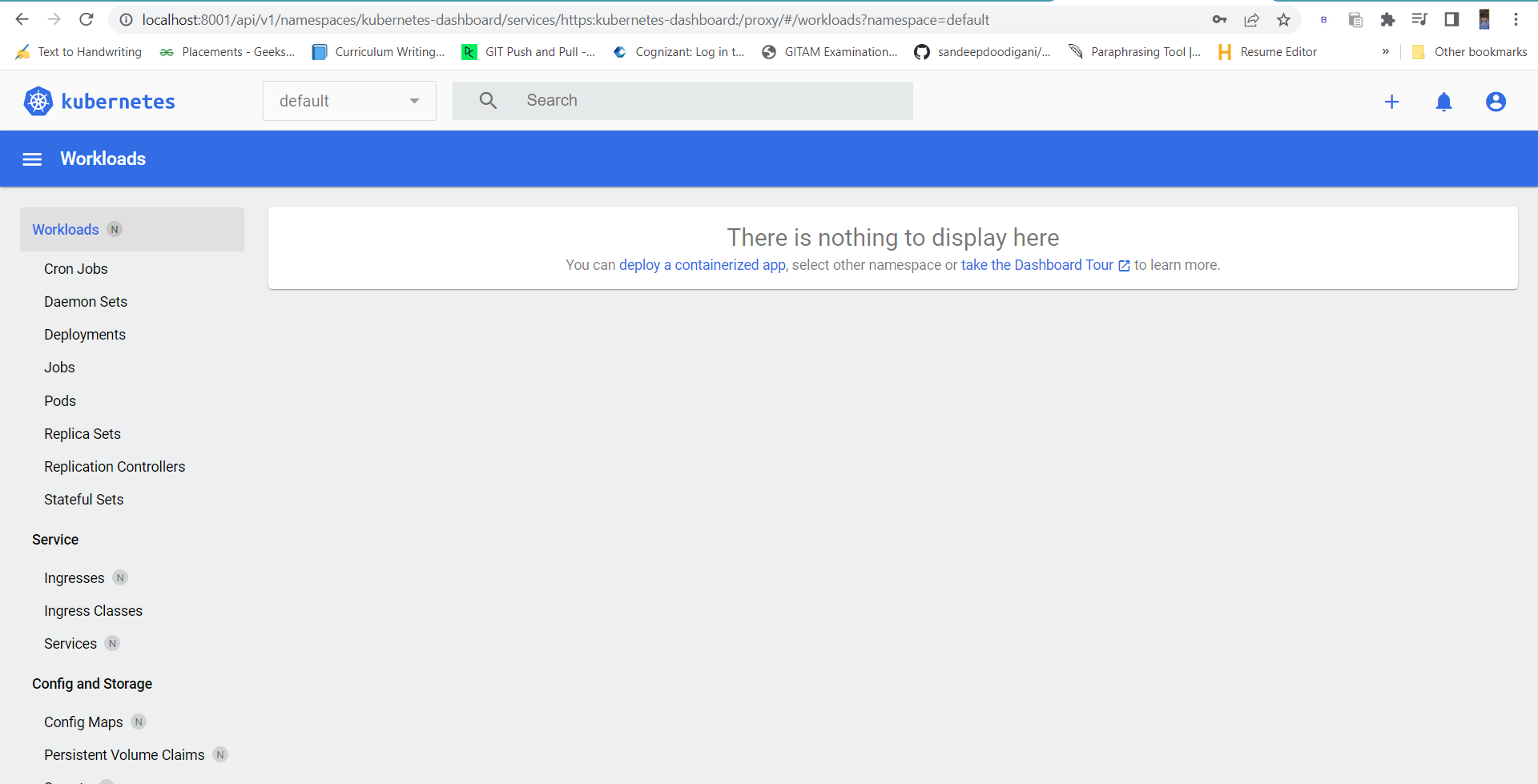
kubectl -n kubernetes-dashboard create token admin-user

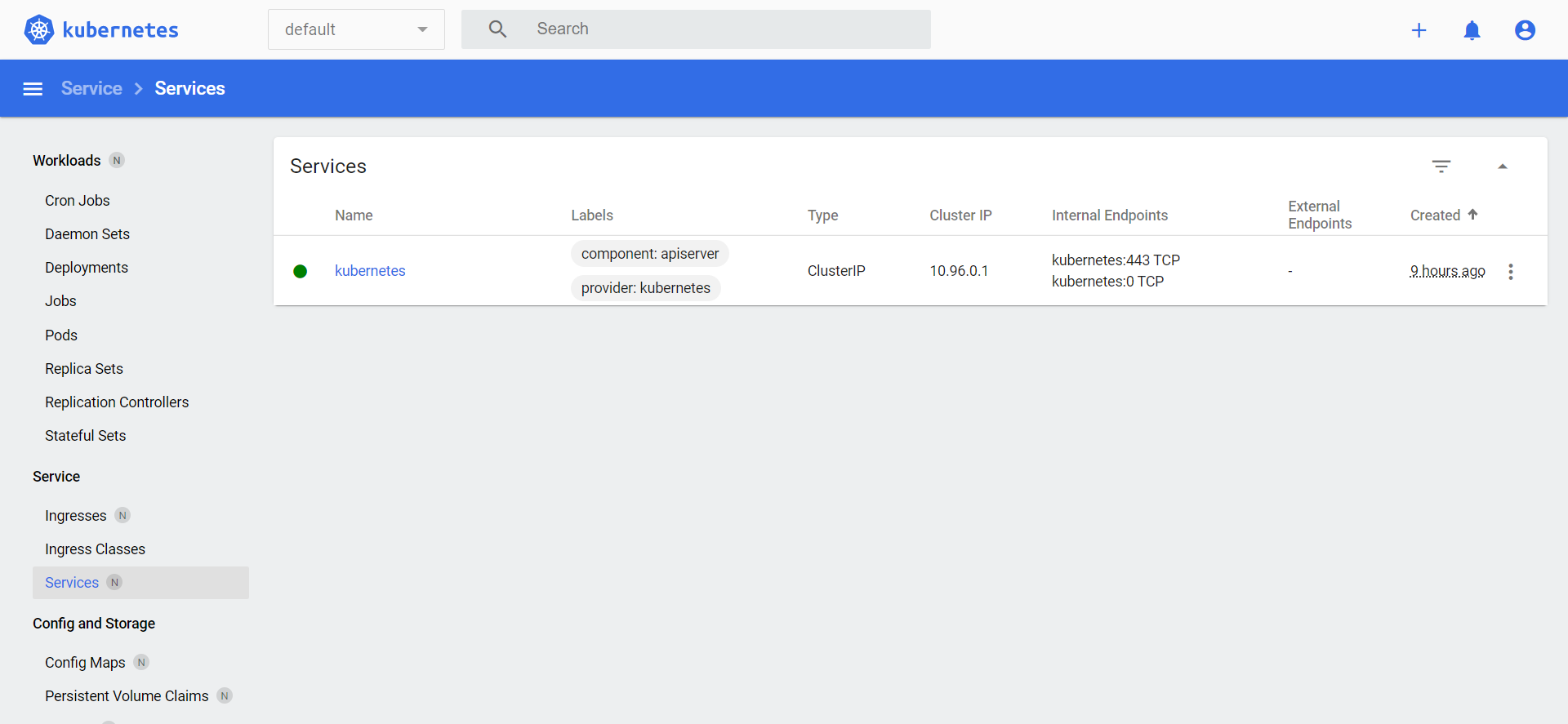


eyJhbGciOiJSUzI1NiIsImtpZCI6InFBd0NmS1h5czlkS2x2ODB4dXlUT0o2cC1taVdHMVVmdmxFanlueHg5X00ifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiXSwiZXhwIjoxNjU0NTE5NTM3LCJpYXQiOjE2NTQ1MTU5MzcsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJrdWJlcm5ldGVzLWRhc2hib2FyZCIsInNlcnZpY2VhY2NvdW50Ijp7Im5hbWUiOiJhZG1pbi11c2VyIiwidWlkIjoiZTM5NDhlMTQtOTYxZi00YjFjLTg1YTYtNzMwN2I2NDVhNDJkIn19LCJuYmYiOjE2NTQ1MTU5MzcsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDprdWJlcm5ldGVzLWRhc2hib2FyZDphZG1pbi11c2VyIn0.jyvK0dAf4BLeHE667-8xXenIRFIyIA\_8VoE56FQhwiO4L5xM39qd95zoRauZ9f4fTF3RDF6h2Nvigeyb6WJWJNFcb6CNEdzSUGBFCO3-zb\_U51P916oh2nqDI-Bwj47pUIohdTjgdteRQ8IK6BPDpoOkEpuFKuyGYxpihvFoBO8GsimKGbnkkhDEkDpkYV8HhFjPZTDl2Q94Dn99FbtucUY9t6ZtH5uN\_AT9kZogUmk-Q-rKh4-yo9iOhKZRiILMrfFkxrlDL-hdyCCqjJh2Rg0CJA8\_Lujp\_ugL0M4E5gsSYekJv8lCyCWwPqbqQ6qkwM6EJ2\_8ZRczRdrCCnXtfw

Enter the password in the token

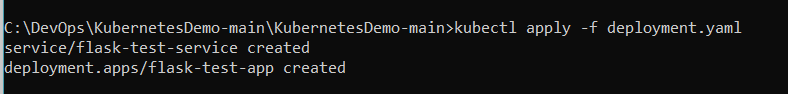
<http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/workloads?namespace=default>





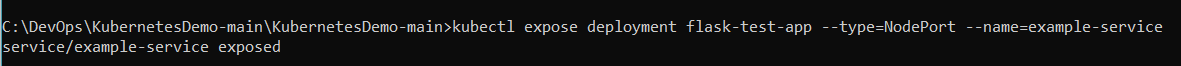
Open command promot and follow the following commands

kubectl apply -f deployment.yaml

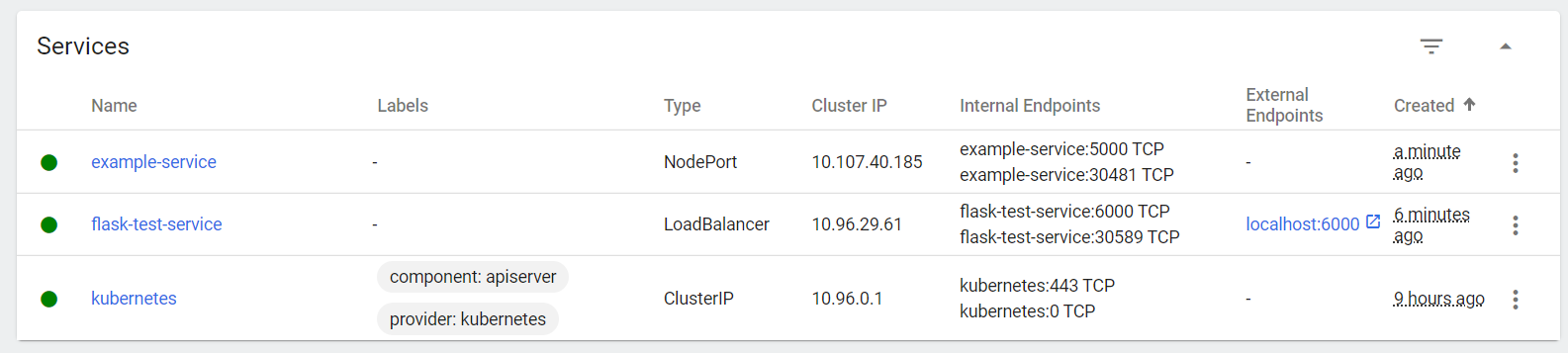


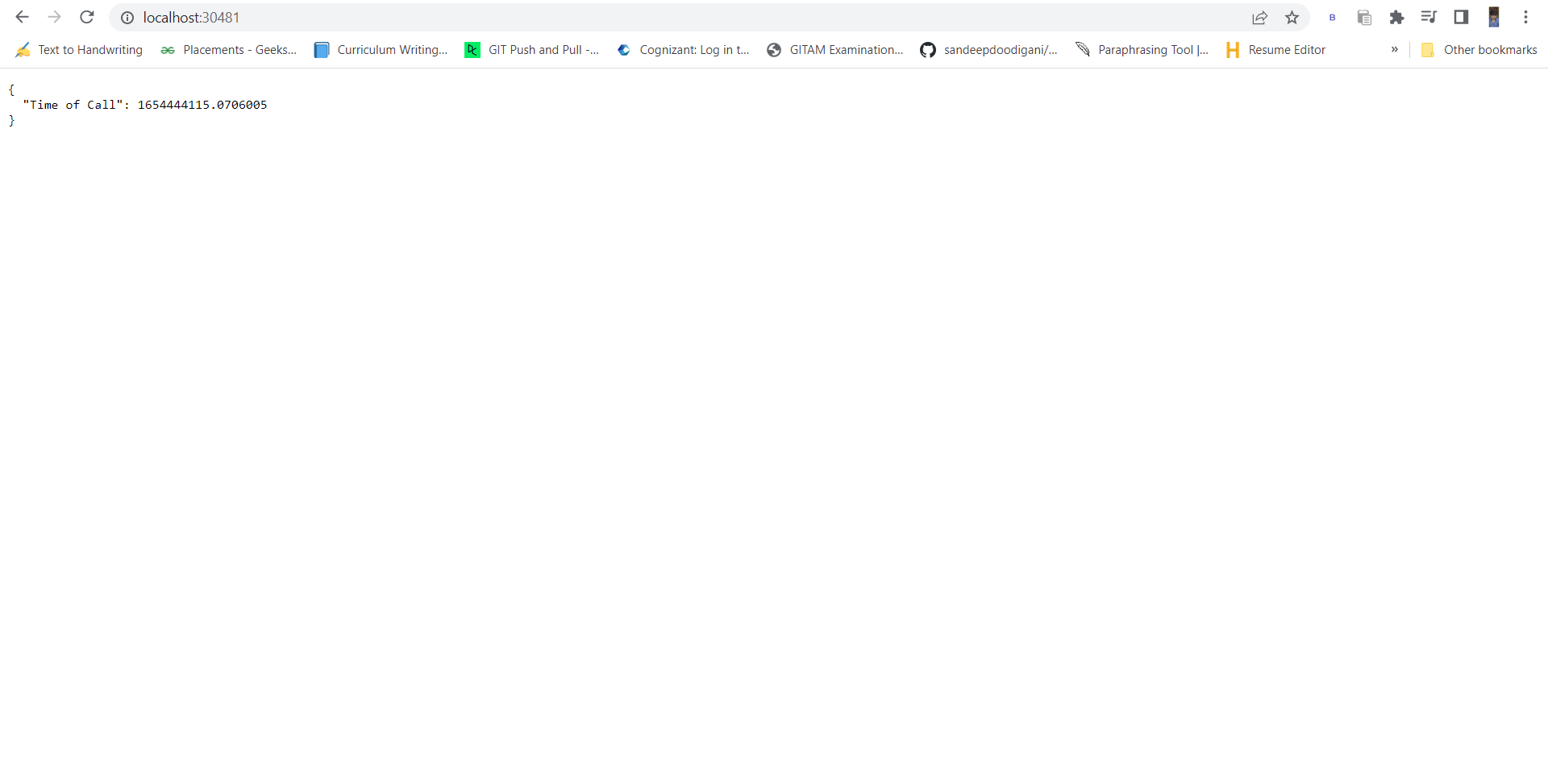
Now we can explore the replica sets, deployement and services

-kubectl expose deployment flask-test-app --type=NodePort --name=example-service

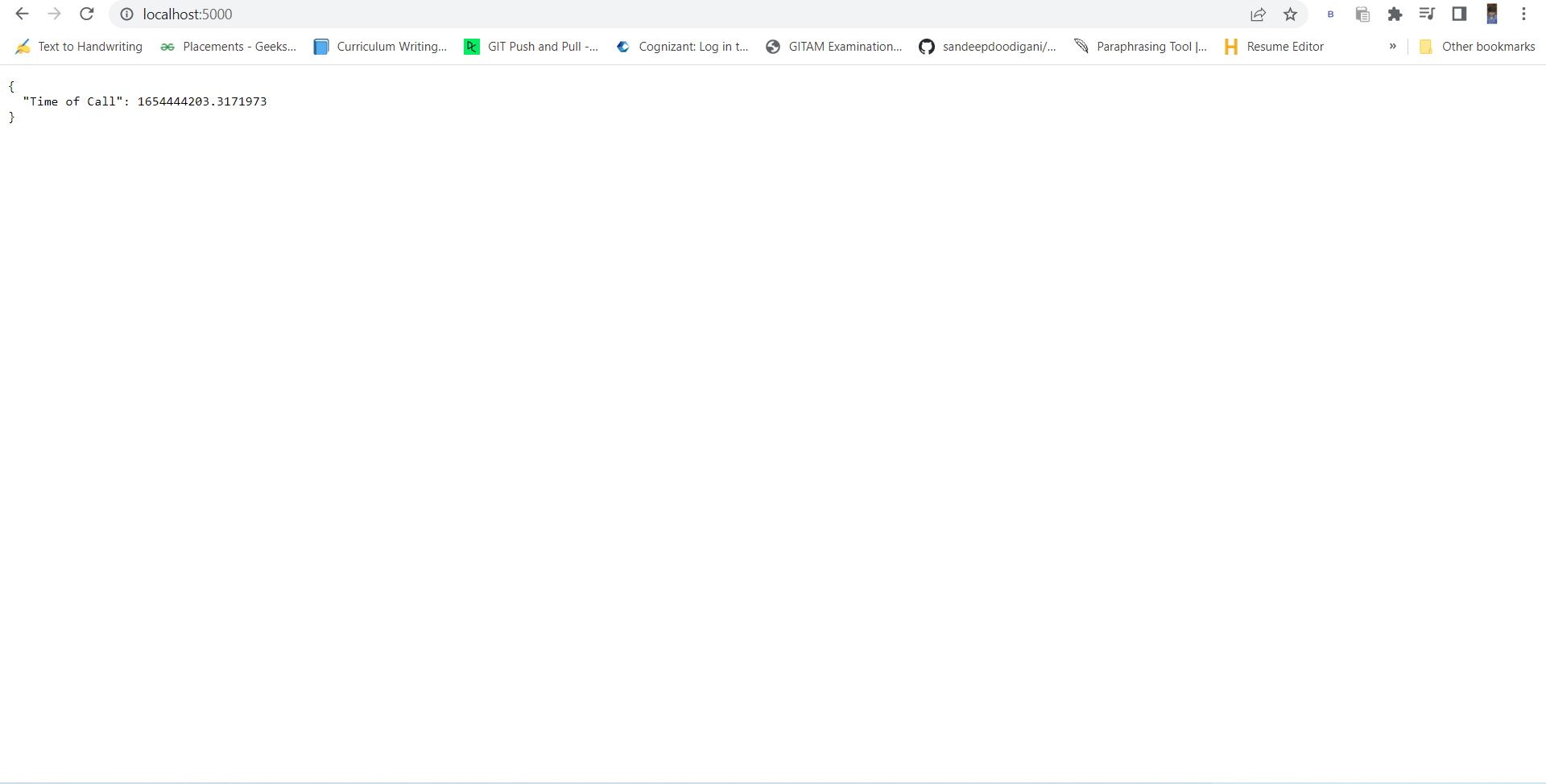


<http://localhost:30481/>

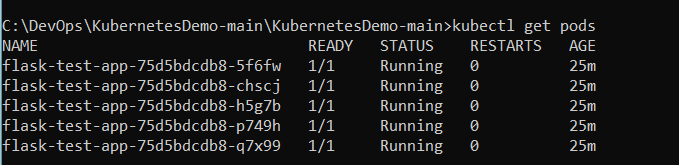




<http://localhost:5000/>



kubectl get pods



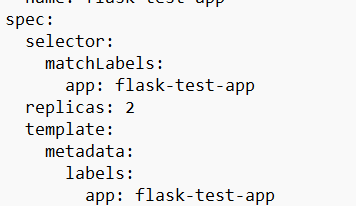
docker images

C:\DevOps\KubernetesDemo-main\KubernetesDemo-main>docker rmi -f 7a26dde2baa5

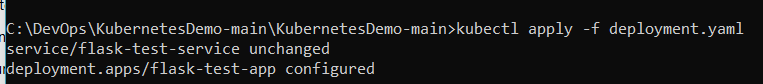
Error response from daemon: conflict: unable to delete 7a26dde2baa5 (cannot be forced) - image is being used by running container 00014c2c9daa

1:09:05

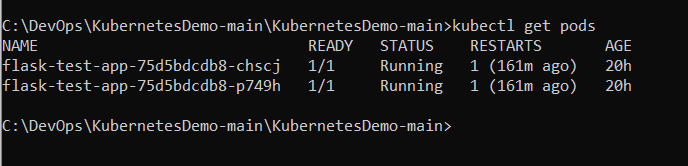
Now change the replica set value from 5 to 2



kubectl apply -f deployment.yaml



We get 2 pods instead of 5 pods



\*\* Can also access kubernetes using minikube website