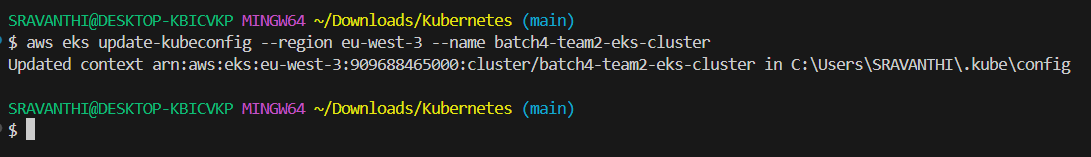


I created an namespace.yaml

Then firstly we need to create a local cluster and set kubeconfig.

Since, we have created an eks clusterfor our team, I just run the command as follows by changing the region and cluster name

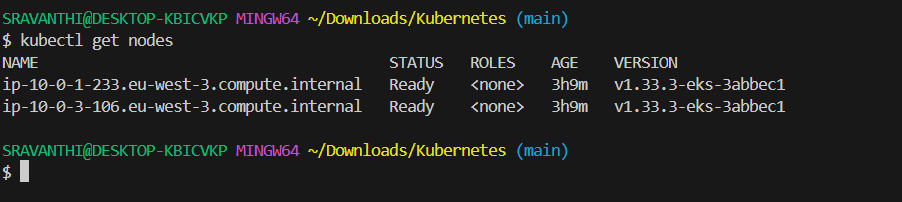
**aws eks update-kubeconfig --region eu-west-3 --name batch4-team2-eks-cluster**

****

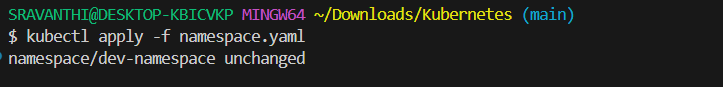
This will gives the kubeconfiguration

Then once kubeconfig is set, run the commands as follows,

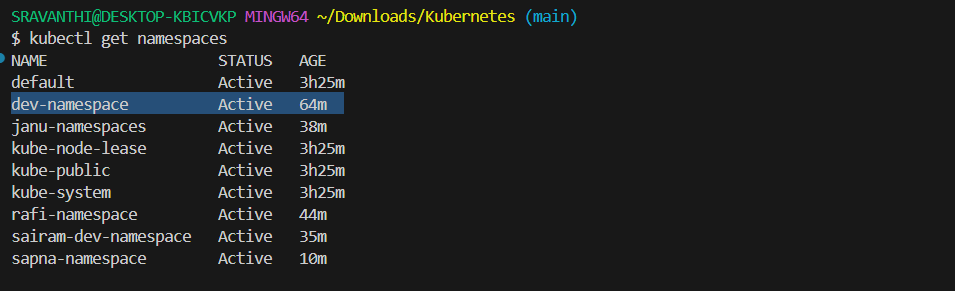
1: For test connection to our cluster run the command “ **kubectl get nodes**”, which shows the status of our worker nodes as below,



Then for applying our namespace run the command as “**kubectl apply -f namespace.yaml**”



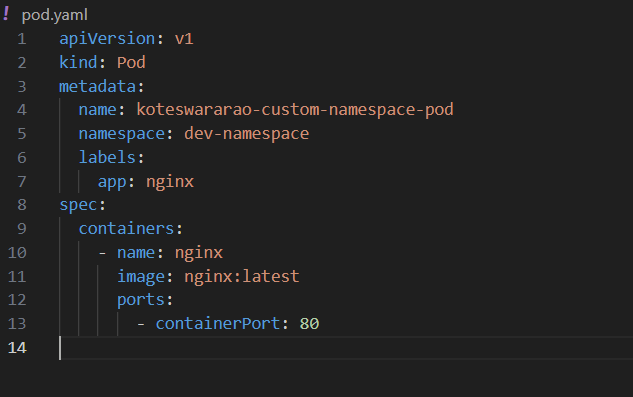
Then for verifying the namespace is created or not run “**kubectl get namespaces**”



Here is my namespace which is created.

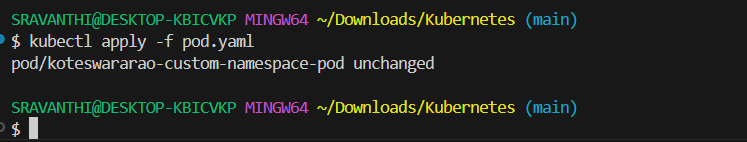
Now we can start deploying workloads inside **dev-namespace**

So for that I created an pod.yaml file

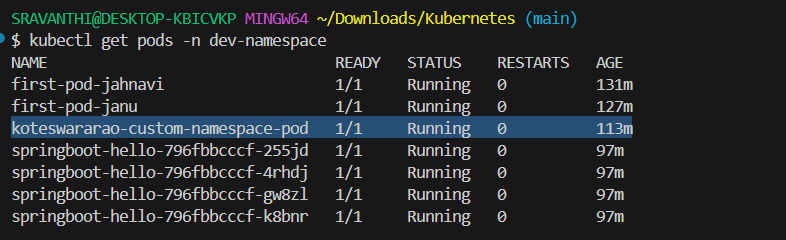


Then run the commands as follows,

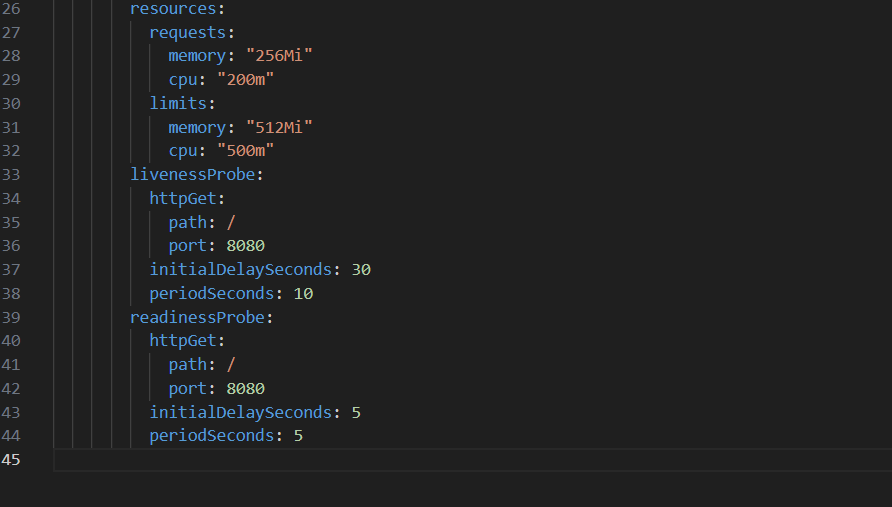
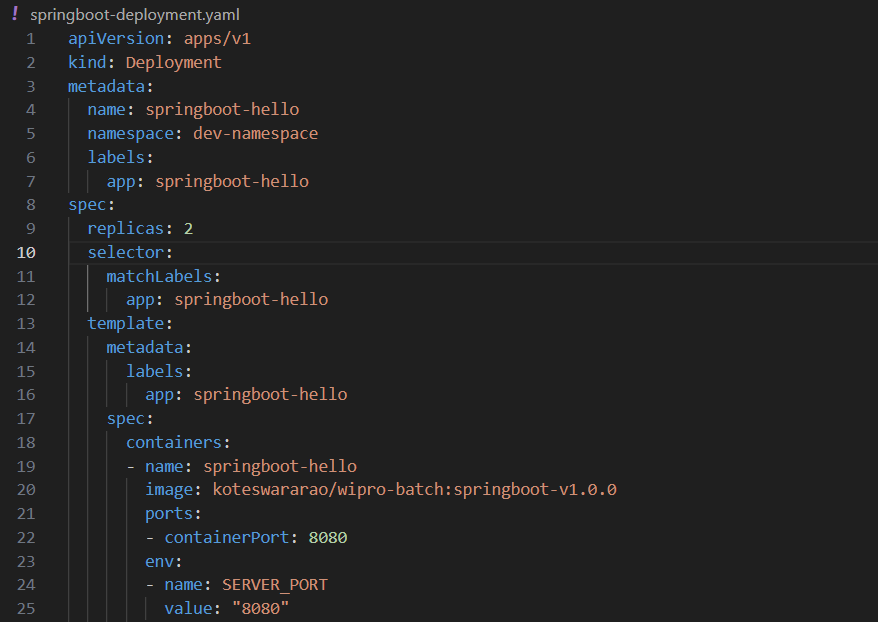
**kubectl apply -f pod.yaml**



**kubectl get pods -n dev-namespace**

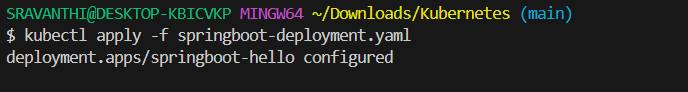
****

Now create an **springboot-deployment.yaml** that runs **2 replicas** of our nginx pod in the **dev-namespace**.

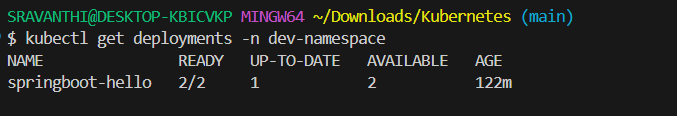


Then run the commands,

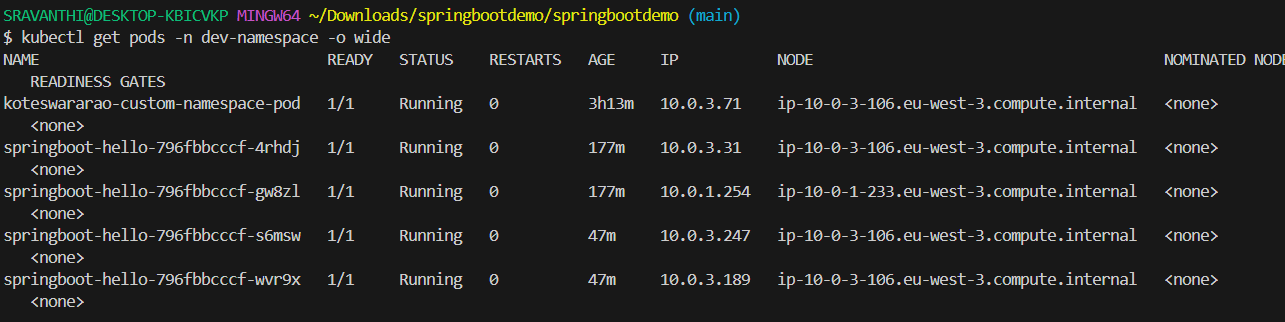
**kubectl apply -f springboot-deployment.yaml**

****

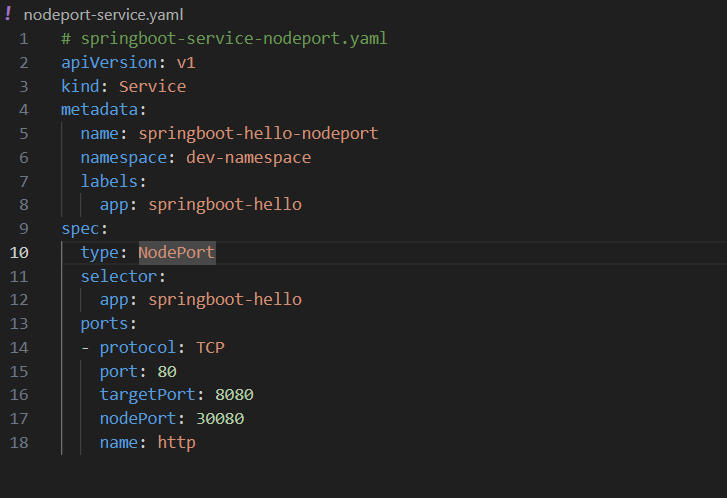
**kubectl get deployments -n dev-namespace**

****

**kubectl get pods -n dev-namespace -o wide**

****

Then created an nodeport-service.yaml

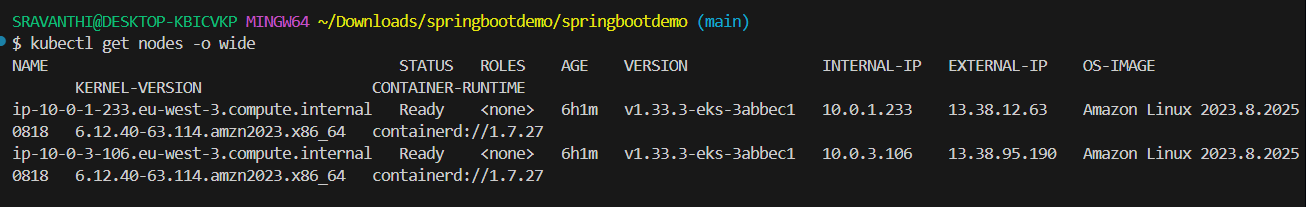
****

Then run the commands,

**Kubectl get svc -n dev-namespace**

****

**kubectl get nodes -o wide**

****

Then check in browser with [**https://13.38.95.190/**](https://13.38.95.190/)

