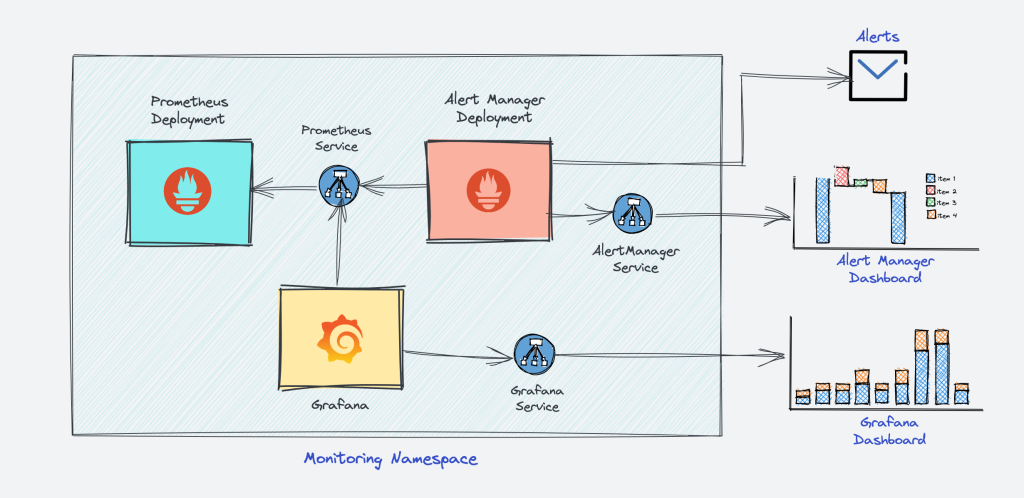
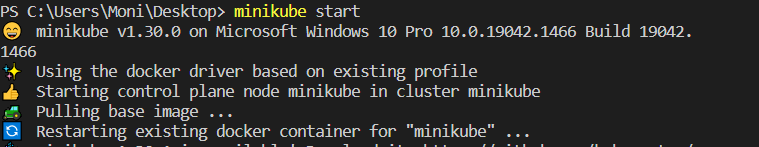
Prometheus Monitoring Setup on Kubernetes



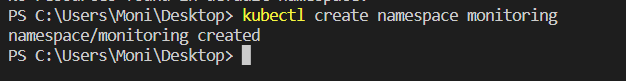
Connect to your Kubernetes cluster



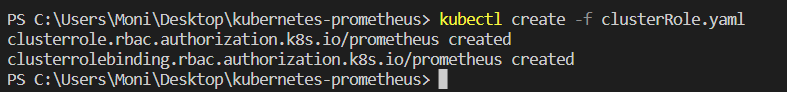
Get all the configuration files from the provided github repo with git clone

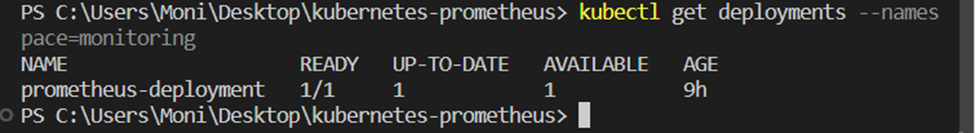
git clone <https://github.com/techiescamp/kubernetes-prometheus>

Create a Namespace monitoring

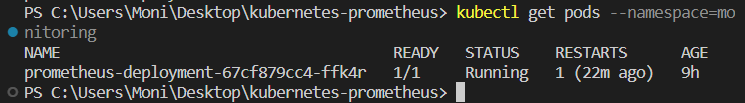


And create the role



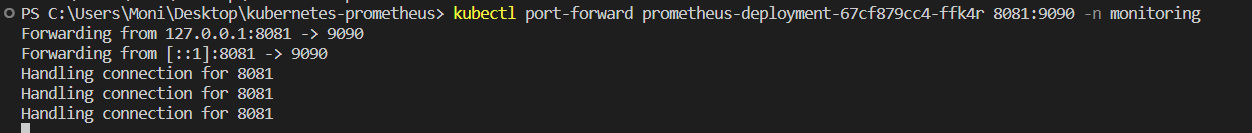
Create the config map in Kubernetes, Create a deployment on monitoring namespace and check the deployment.  


Connecting To Prometheus Dashboard  
  
Using Kubectl port forwarding - Get pods name



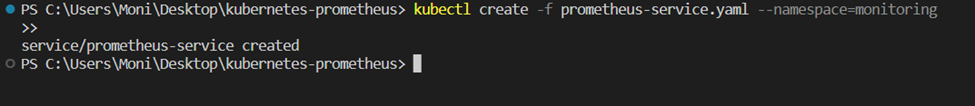
We get the name of the pod and enter the following command

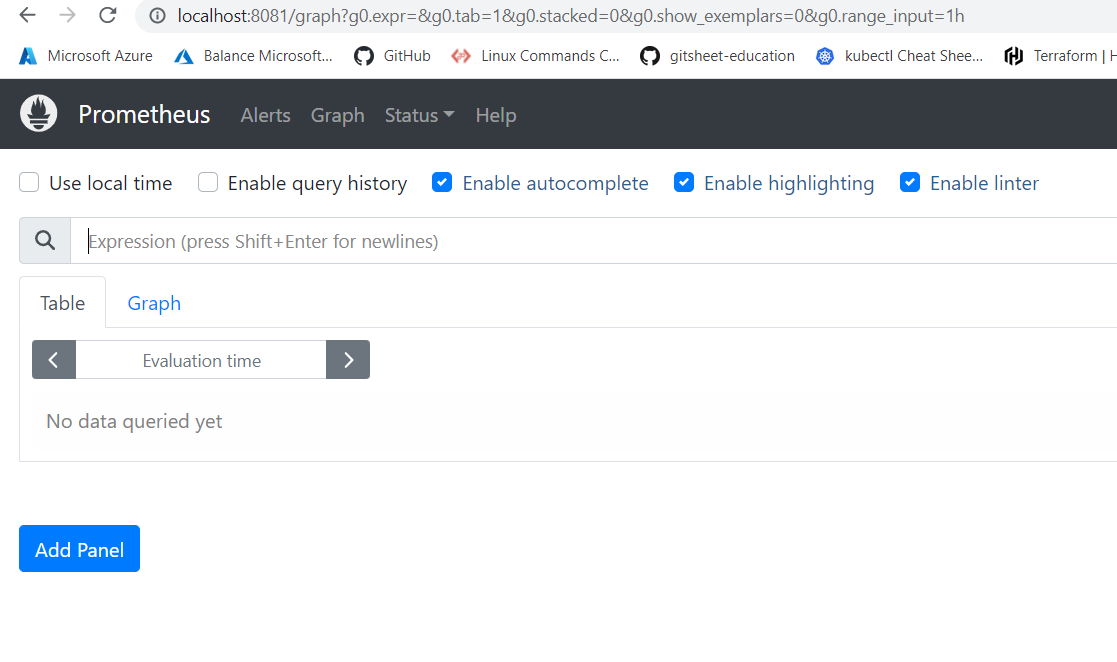
kubectl port-forward prometheus-deployment-67cf879cc4-ffk4r 8081:9090 -n monitoring

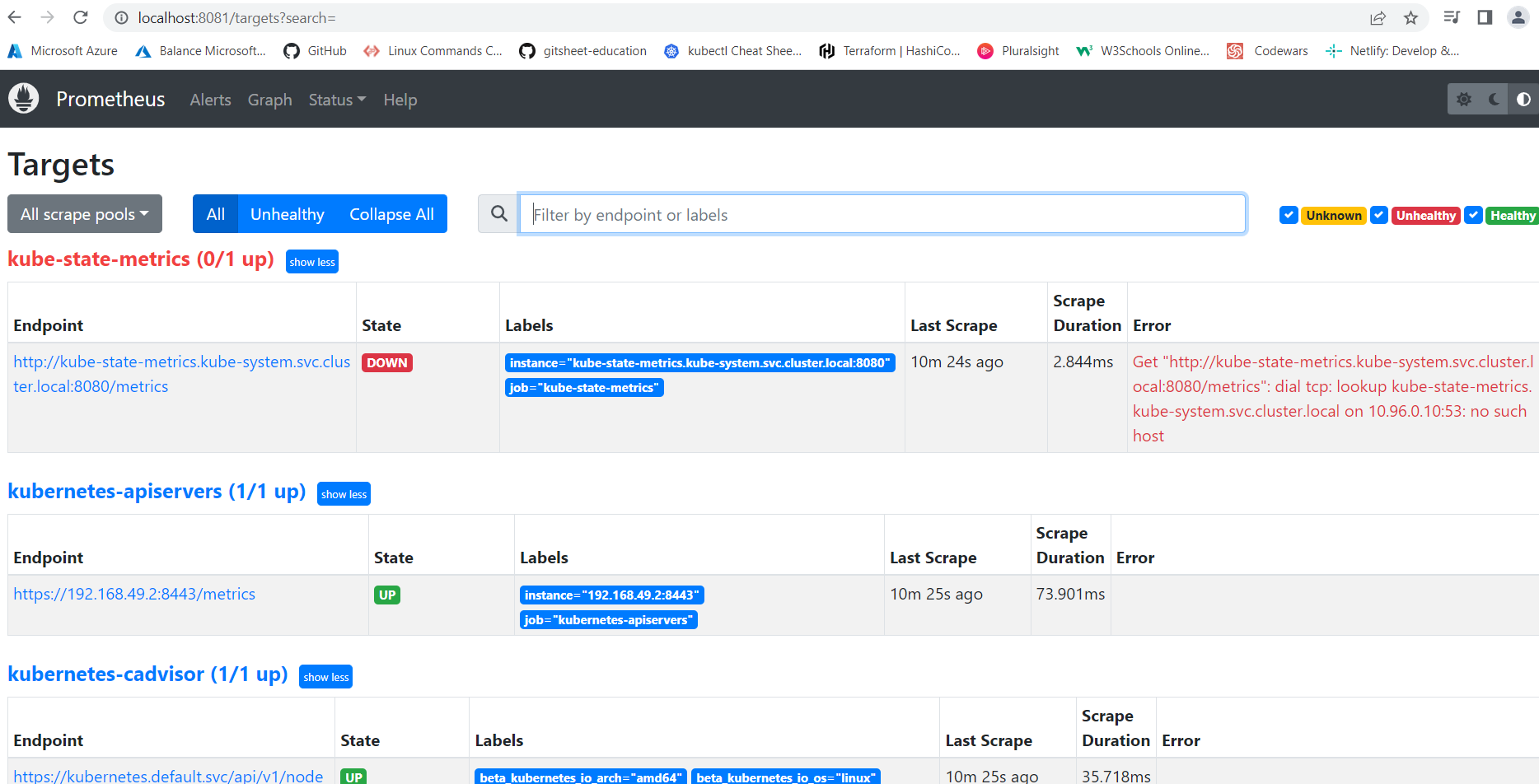


change the port from 8080 to 8081 because it was already in use  
and we can access Prometheus on localhost 8081

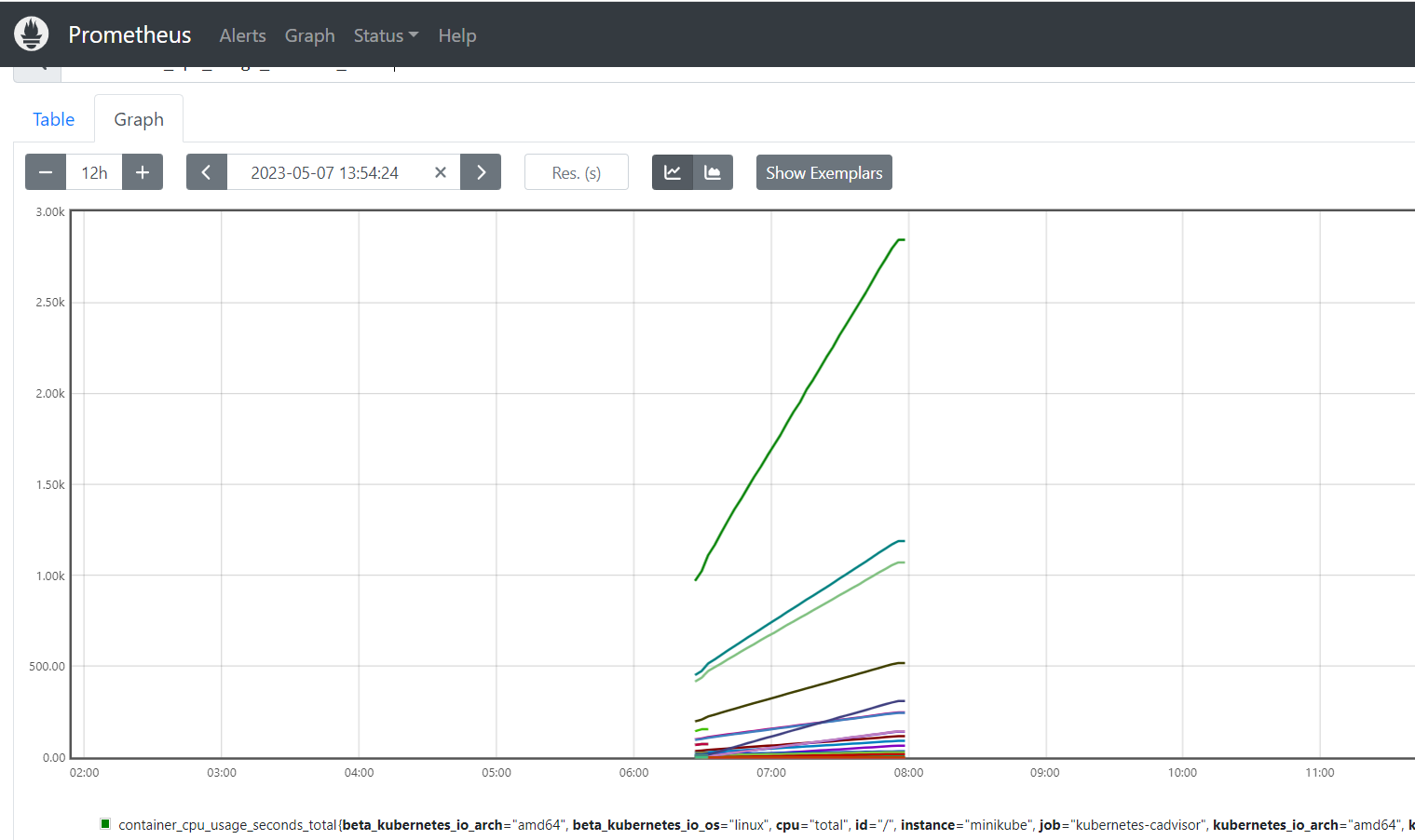
Create the service with the command:



  
If you browse to targets you can see the endpoints connected to Prometheus



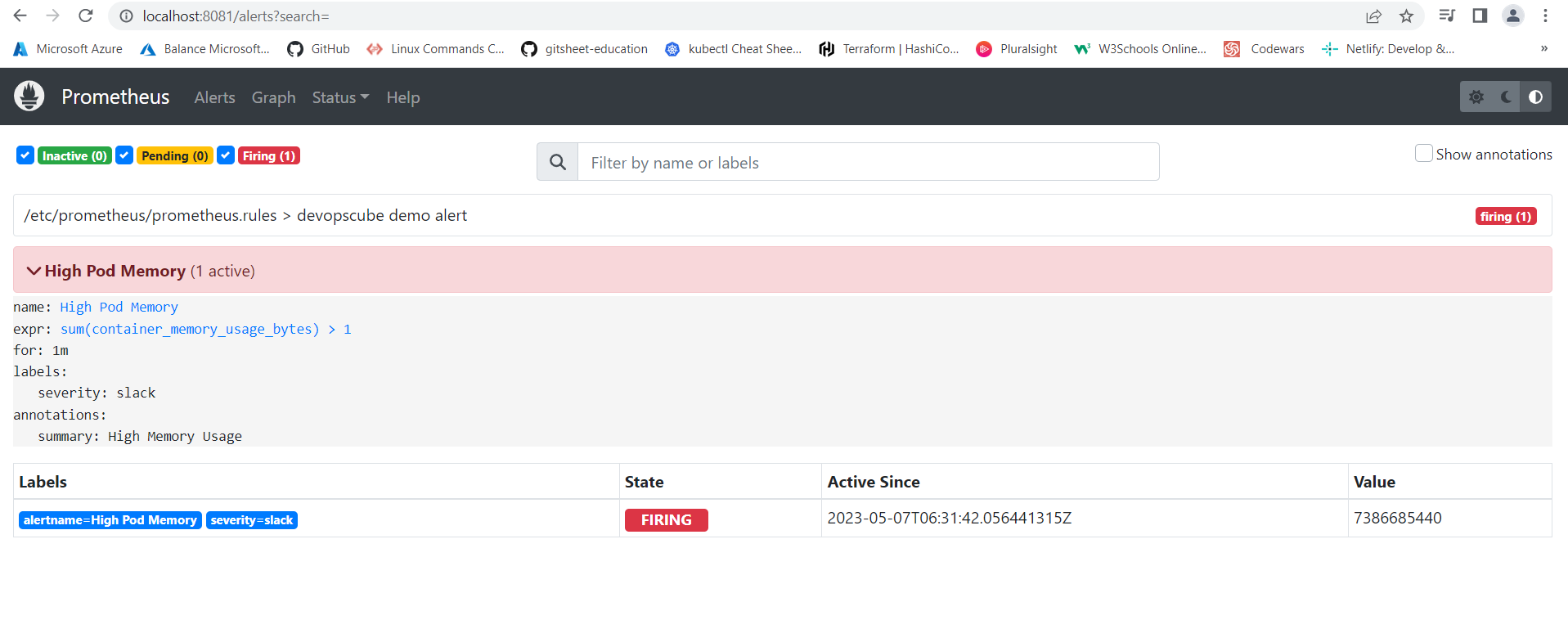
The kube-state-metrics down and it is expected



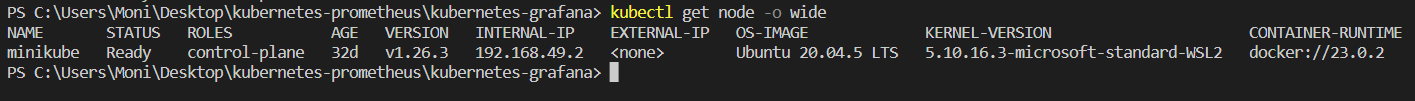
This is a graph for container usage

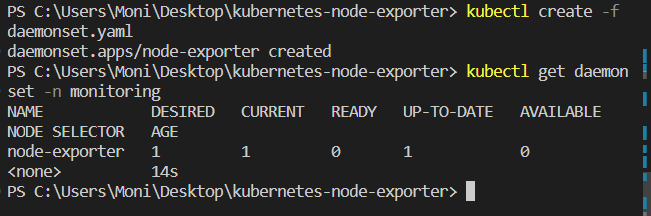
Exposing the Prometheus deployment as a service with NodePort or a Load Balancer.

Setting Up Alertmanager



Setting Up Grafana  
Get the yaml files from github repo

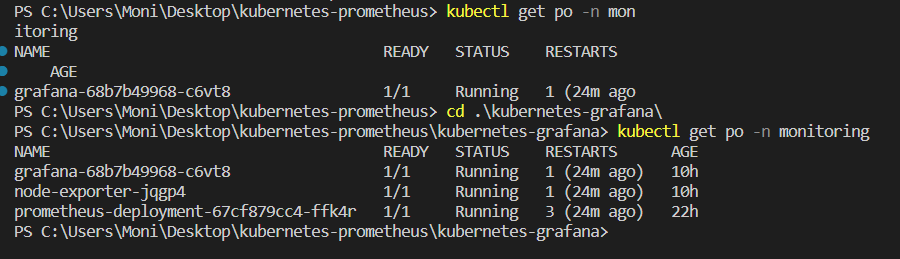
Check the node info to get the IP address  


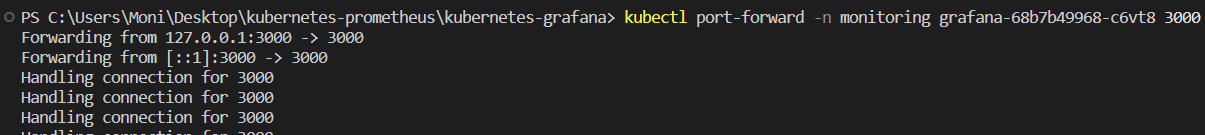
Create the yaml file daemonset

Create a file names service.yaml

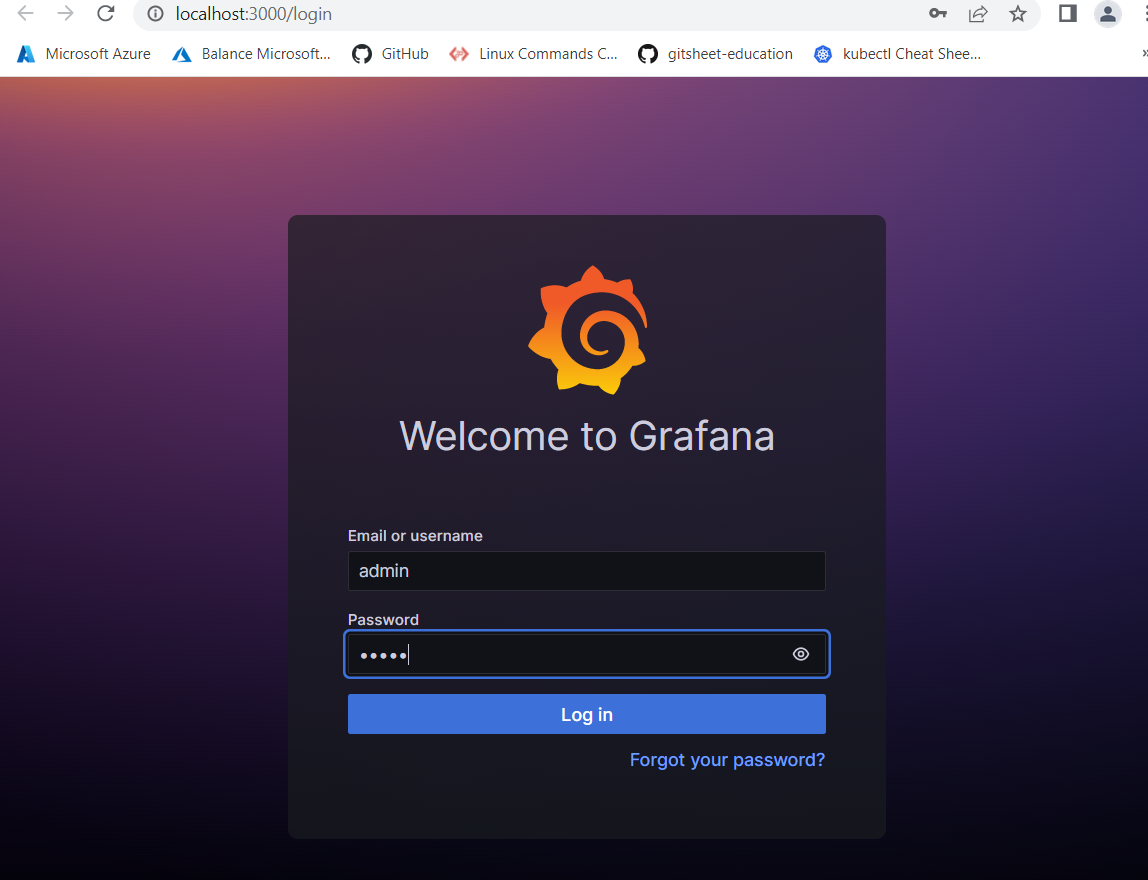
Create the service. 

Get the pods info

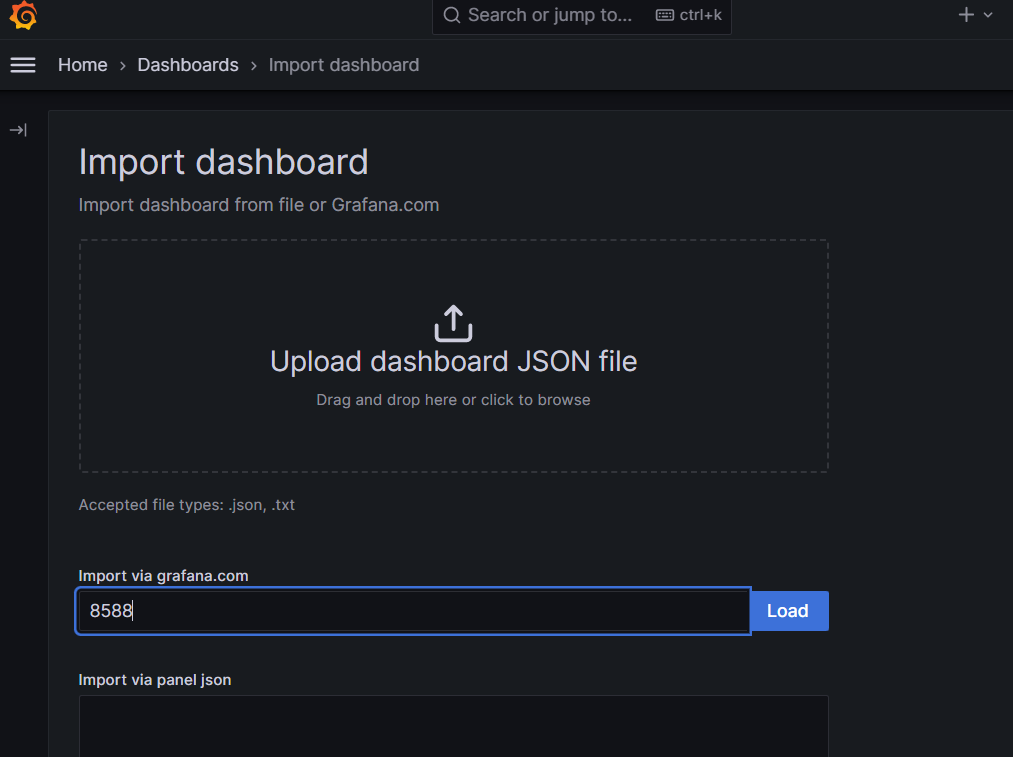


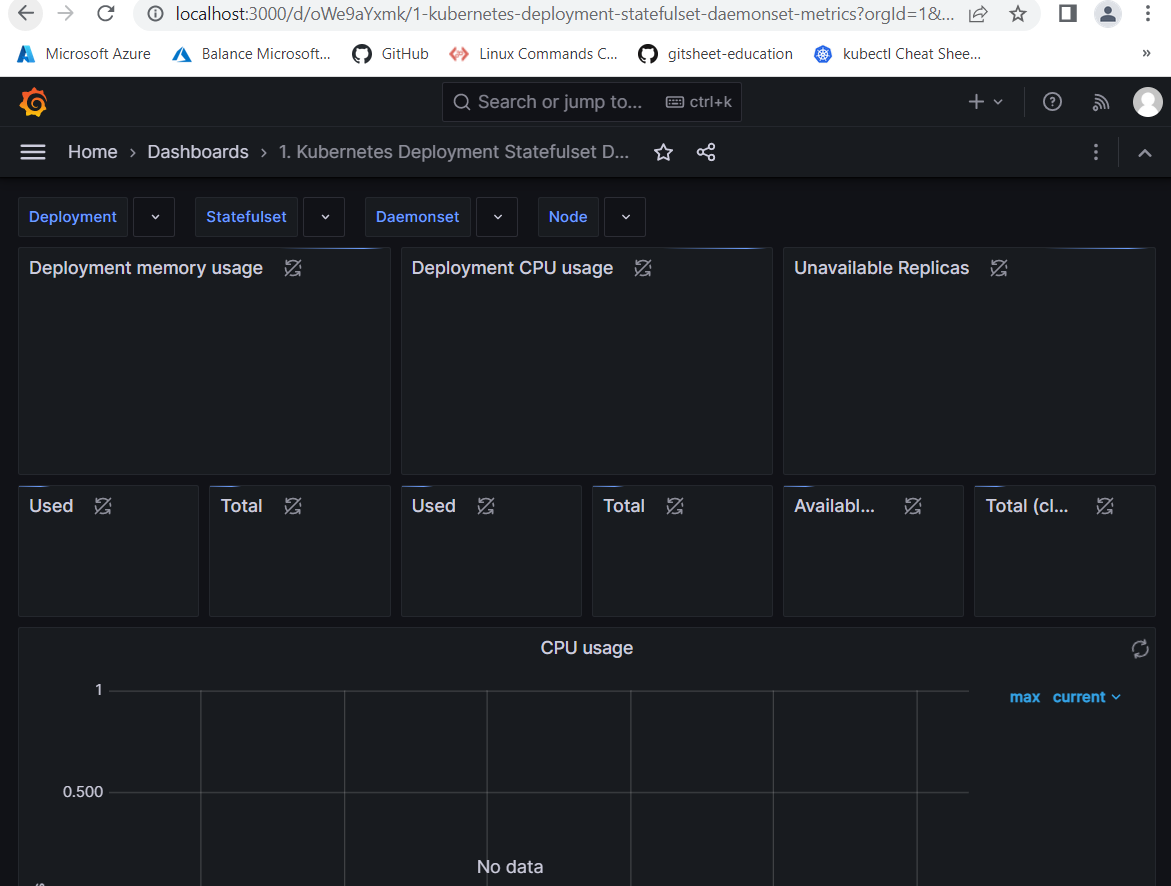
Use port forwarding

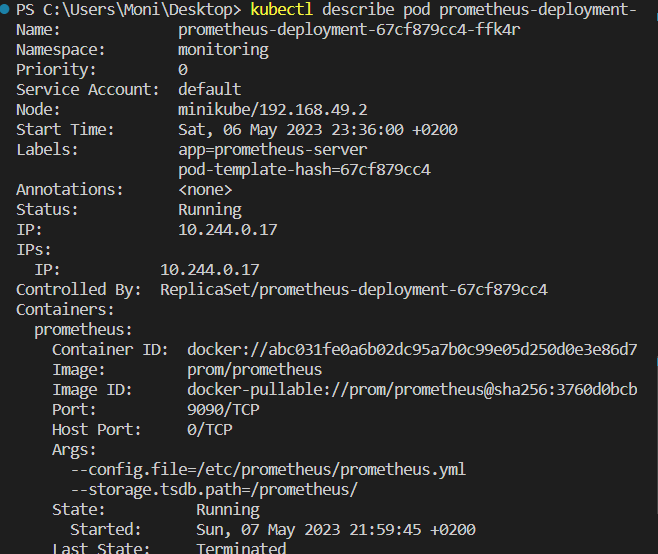
Lon in Grafana



Download the JSON file and import







Update Grafana to pull data from prometeus

