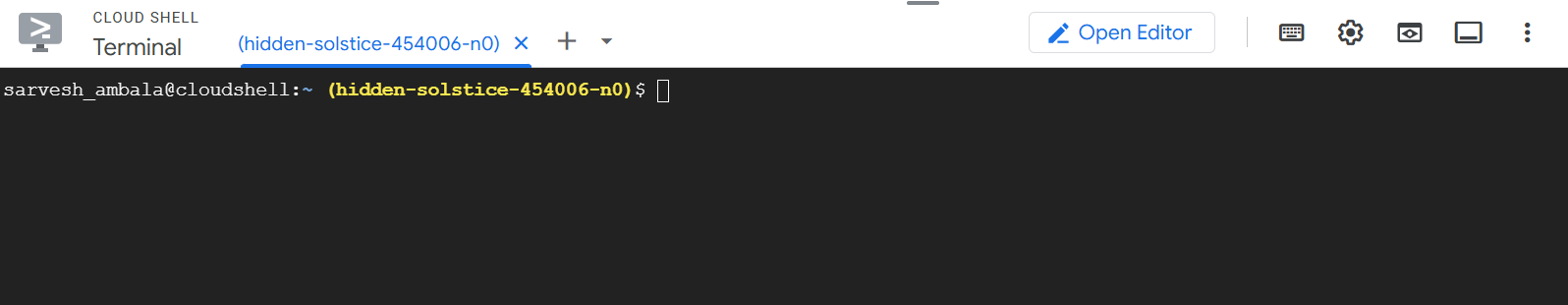
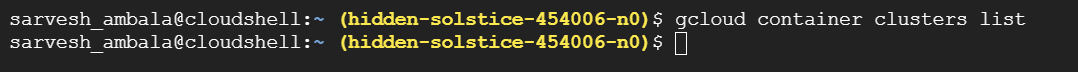
* NOW CREATE THE KUBERNETES CLUSTER

Open the cloud shell



To see the cluster list run the below command

$ gcloud container clusters list ( no clusters are there)



You create the cluster with below command

~~$ gcloud container clusters create my-cluster --zone us-central1-a~~

$ gcloud container clusters create my-cluster \

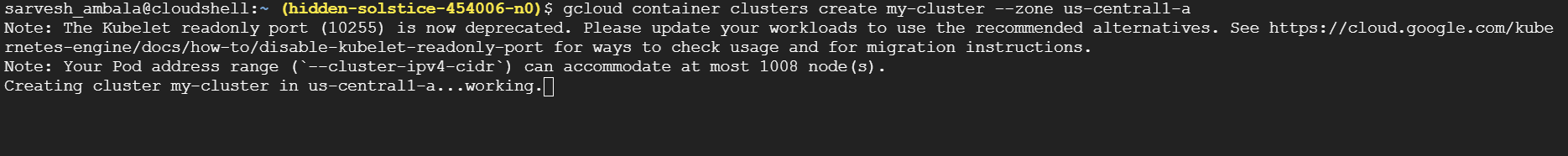
--zone us-central1-a \

--num-nodes=3 \

--disk-size=50 \

--disk-type=pd-ssd

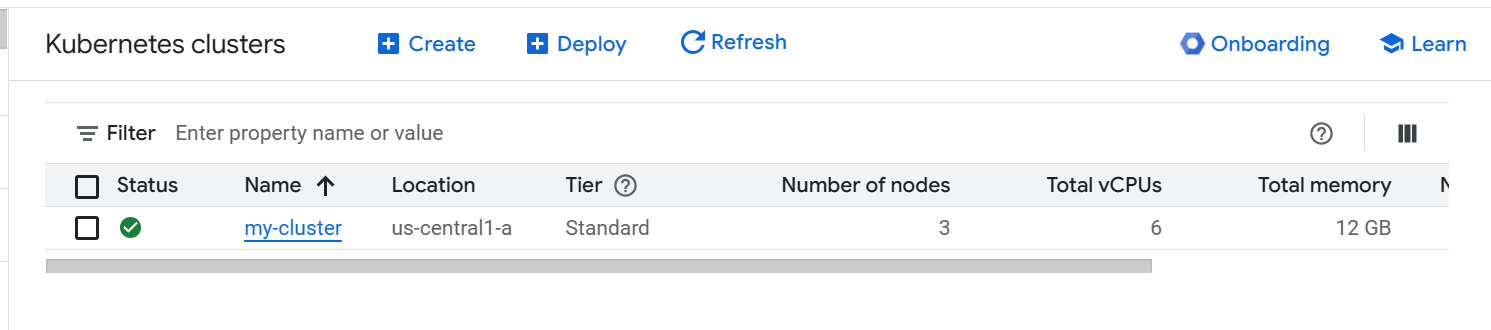
Cluster creation is taking 5 to 10 mints time



Once the cluster is created u can see the below message automatically

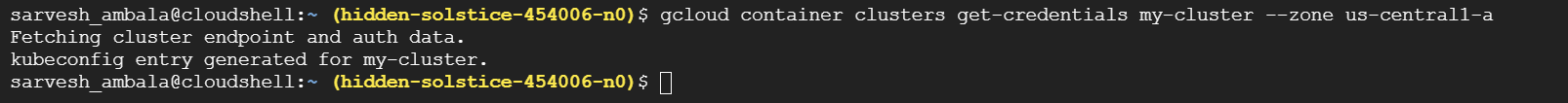


Now u go and check kubernetes engine--->cluster , you can see the my-cluster is running



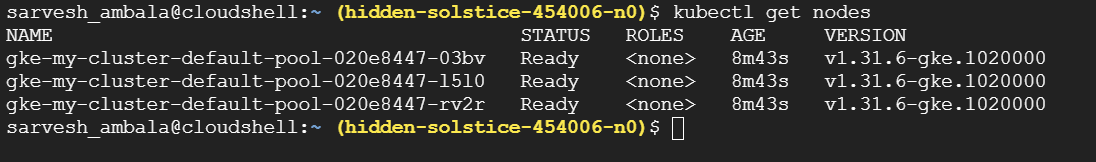
Run the below command

$ gcloud container clusters get-credentials my-cluster --zone us-central1-a



To see the list of nodes

$ kubectl get nodes



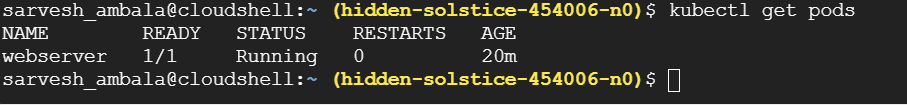
Create the pods

$ kubectl run --image tomcat webserver



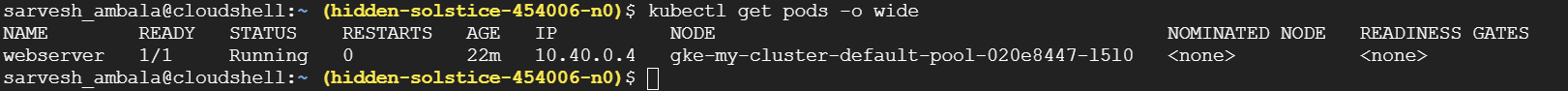
To see the pods list

$ kubectl get pods



To get the list of pods along with ip address and which node the pod is running

$ kubectl get pods -o wide



Actually u can create the pod using definition file

$ create pd-df1.yaml

$ vim pd-df1.yaml

apiVersion: v1

kind: Pod

metadata:

name: jenkins-pod

spec:

containers:

- name: myjenkins

image: jenkins/jenkins

ports:

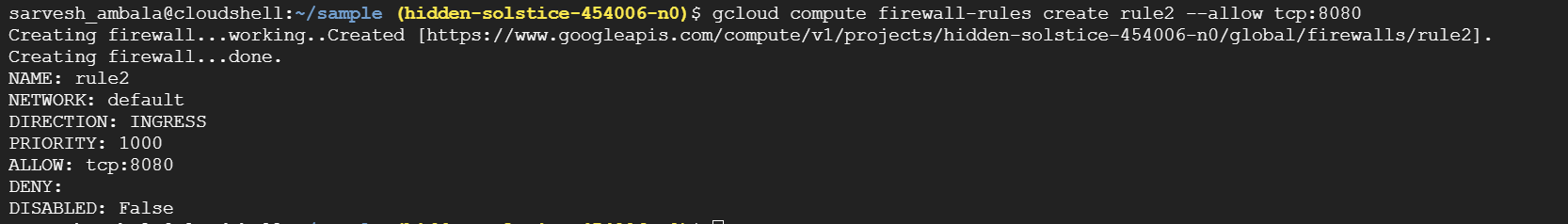
- containerPort: 8080

hostPort: 8080

for accessing the application u need to open the port

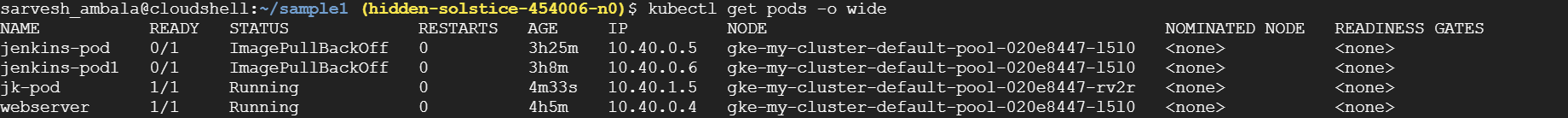
How to open the port

$ gcloud compute firewall-rules create rule2 --allow tcp:8080

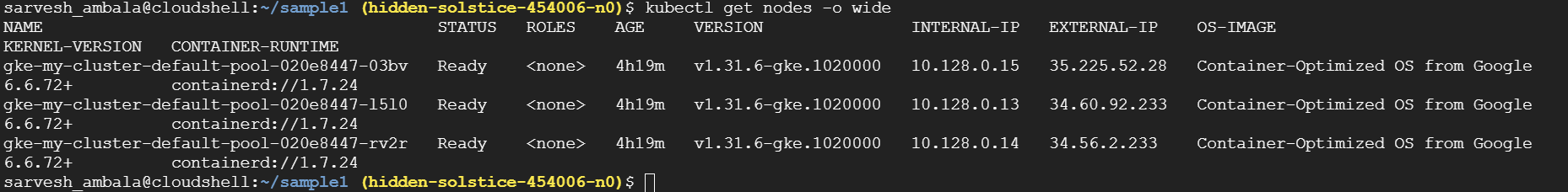


$ kubectl create -f pd-df1.yaml

$ kubectl get pods -o wide



$ kubectl get nodes -o wide

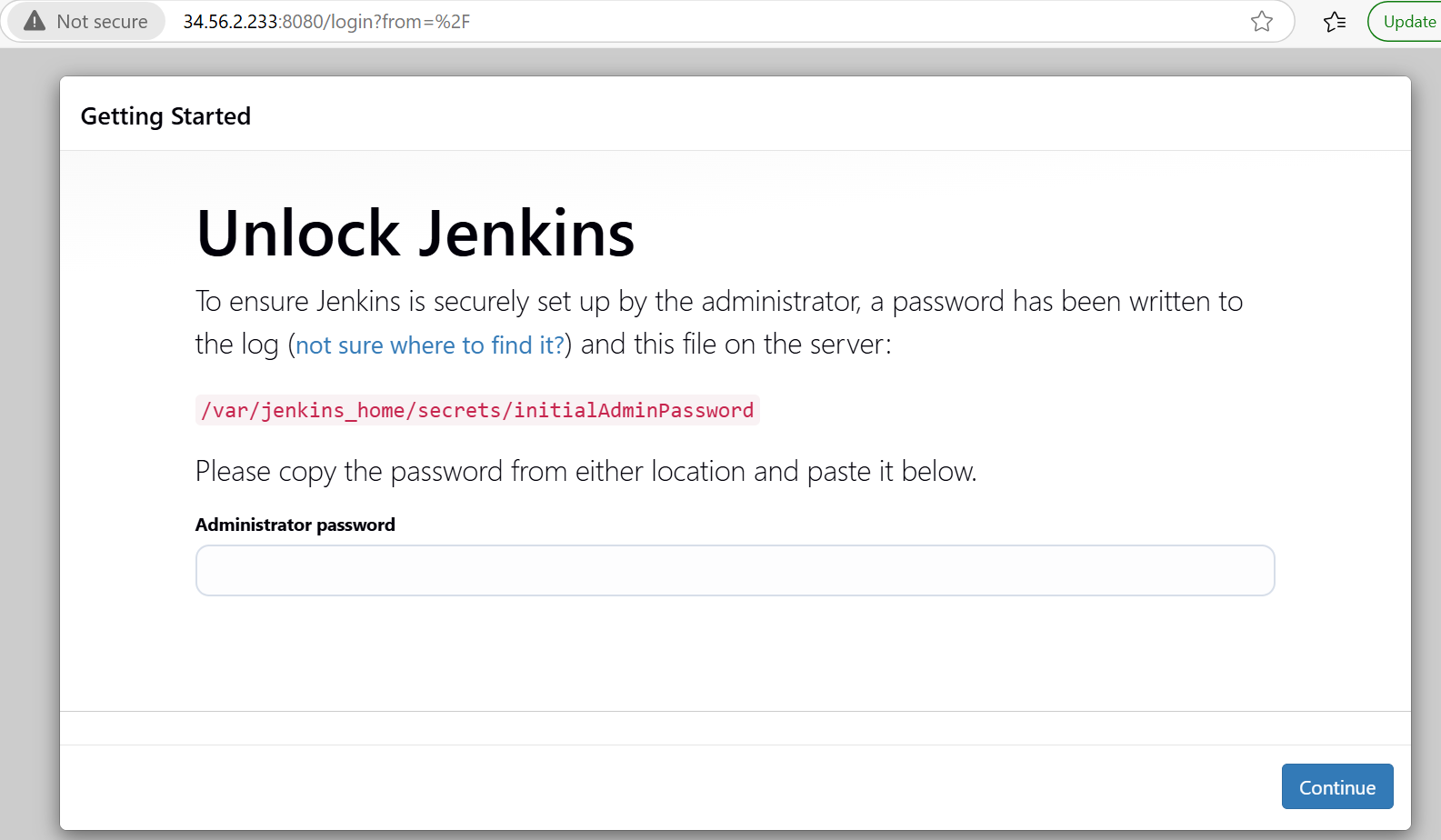


How can we access the pod

Take the external ip add the port no 8080

Open the browser paste ipaddress:8080

Now u can able to see the jenkins



To delete the cluster use the following command:

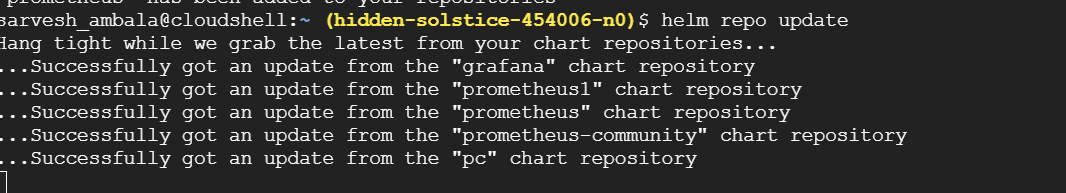
$ gcloud container clusters delete my-cluster --zone us-central1-a

**PROMETHEUS SETUP:**

$ helm repo add prometheus <https://prometheus-community.github.io/helm-charts>

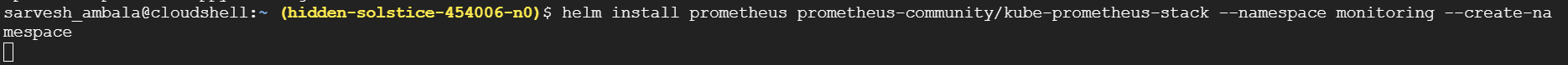


$ helm repo update



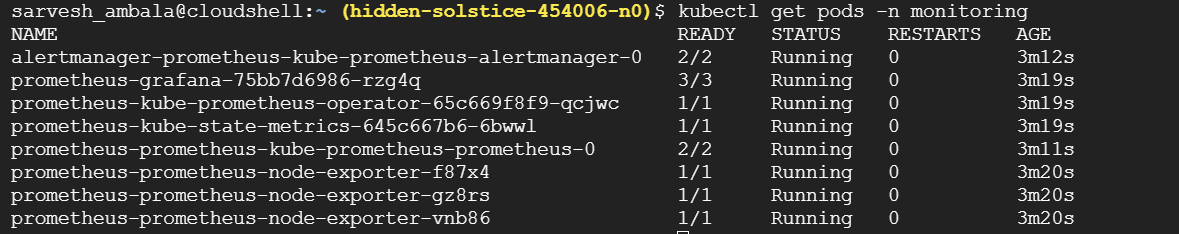
$ helm install prometheus prometheus-community/kube-prometheus-stack --namespace monitoring --create-namespace

This will install prometheus,alermanager and grafana

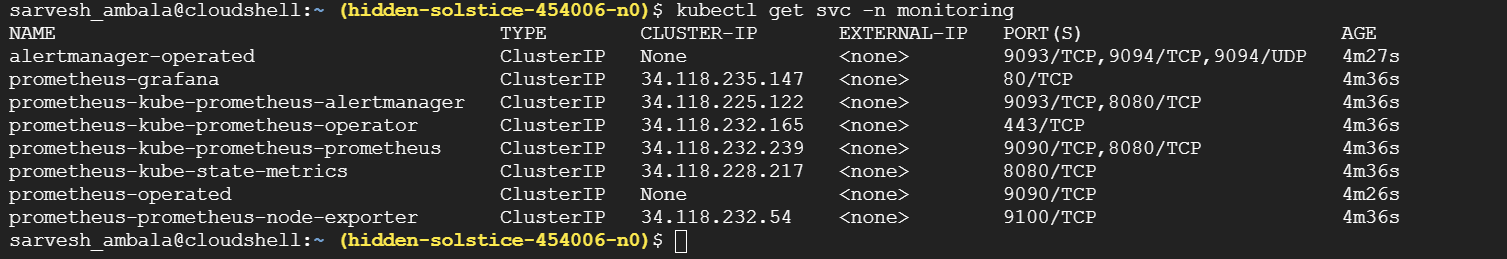


Check the prometheus pods and services

$ kubectl get pods -n monitoring

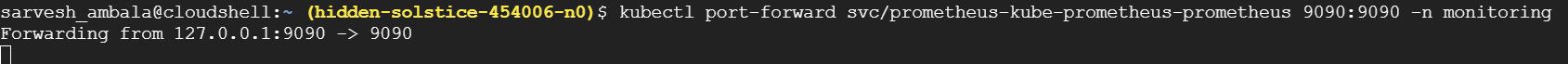


$ kubectl get svc -n monitoring



Access prometheus and port forwarding

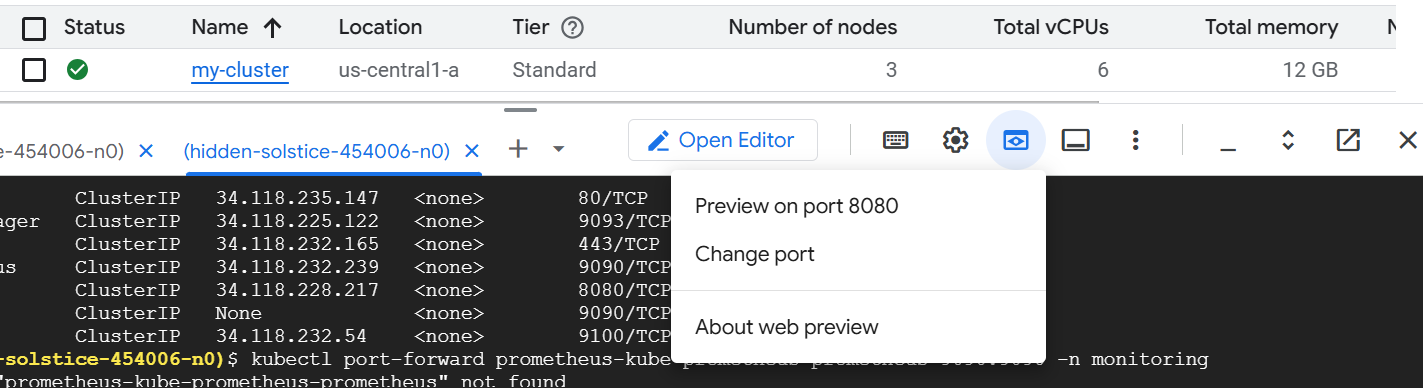
$ kubectl port-forward svc/prometheus-kube-prometheus-prometheus 9090:9090 -n monitoring



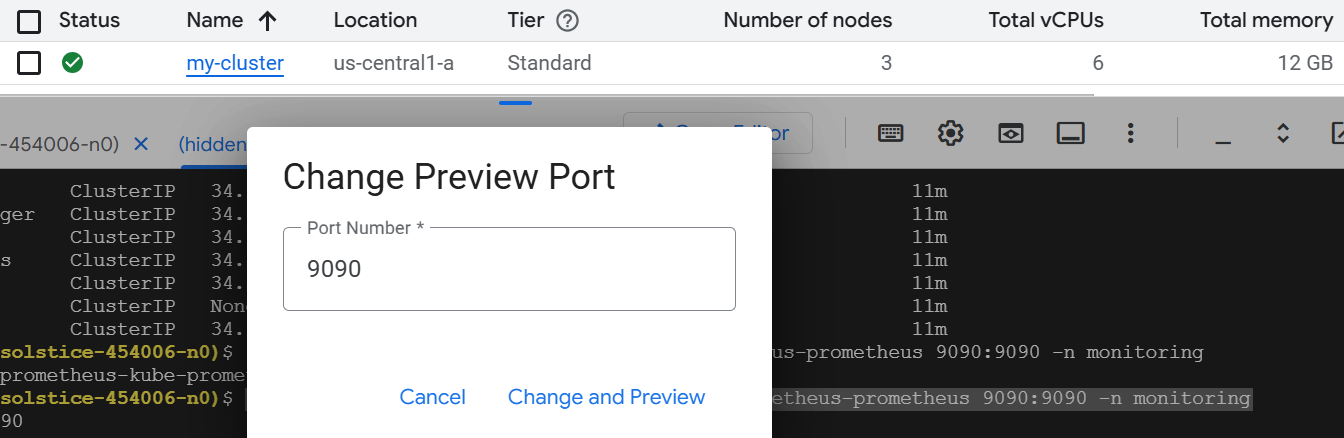
Click on the webpreview



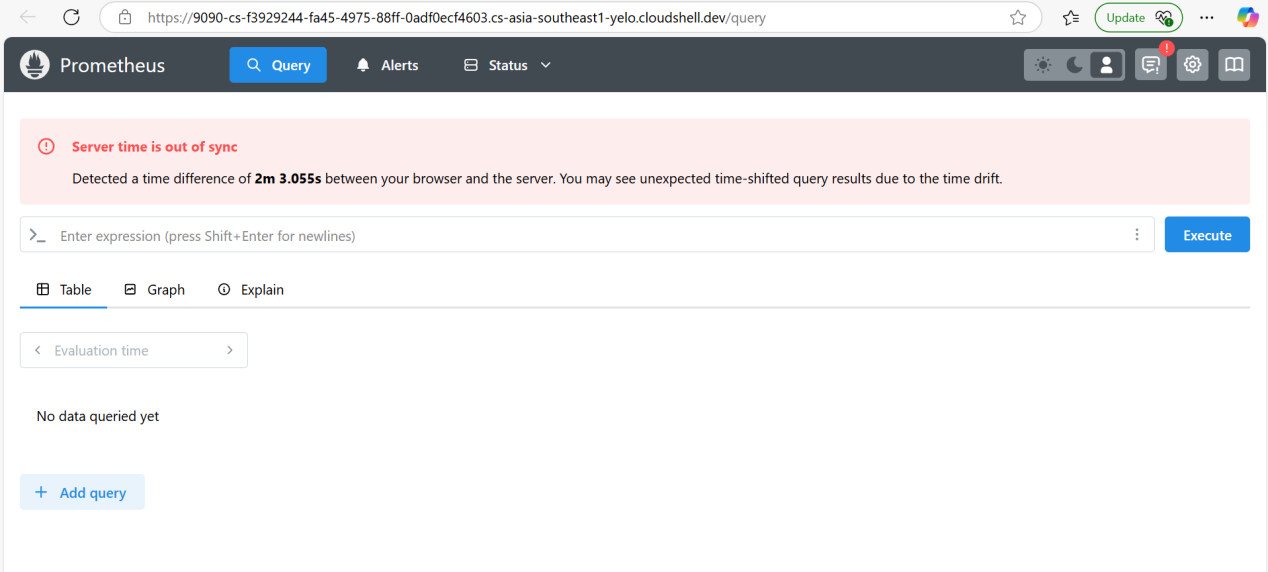
Change port no to 9090



Click on change and preview



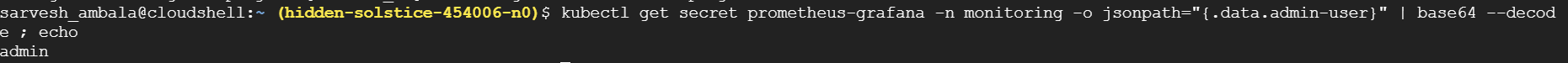
Now u can able to see prometheus in the browser



**ACCESS GRAFANA :**

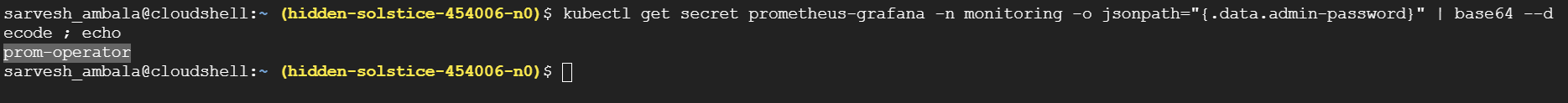
$ kubectl get secret prometheus-grafana -n monitoring -o jsonpath="{.data.admin-user}" | base64 --decode ; echo

If you run the above command u can see the username for grafana (**admin**)



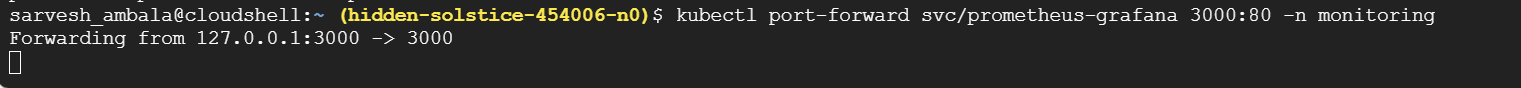
$ kubectl get secret prometheus-grafana -n monitoring -o jsonpath="{.data.admin-password}" | base64 --decode ; echo

If you run the abvoe command u can see the password for grafana (**prom-operator**)

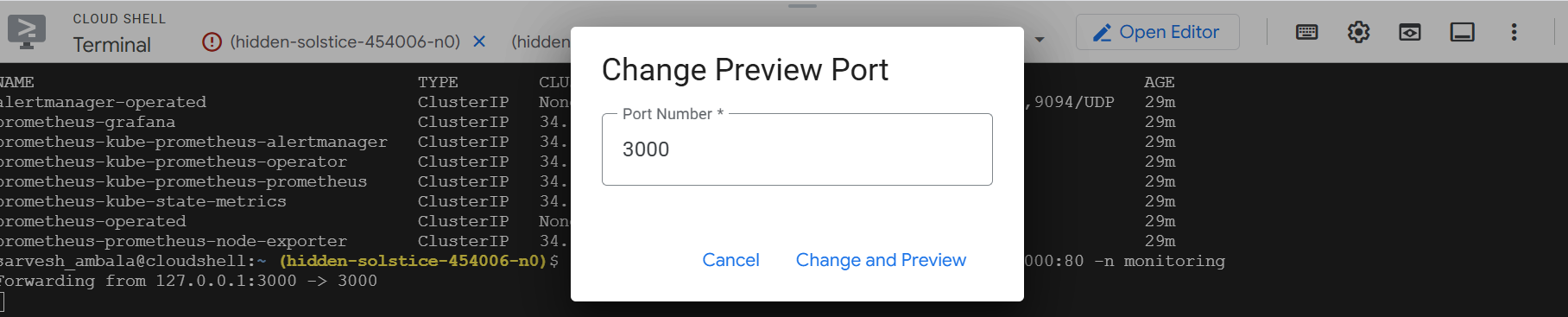


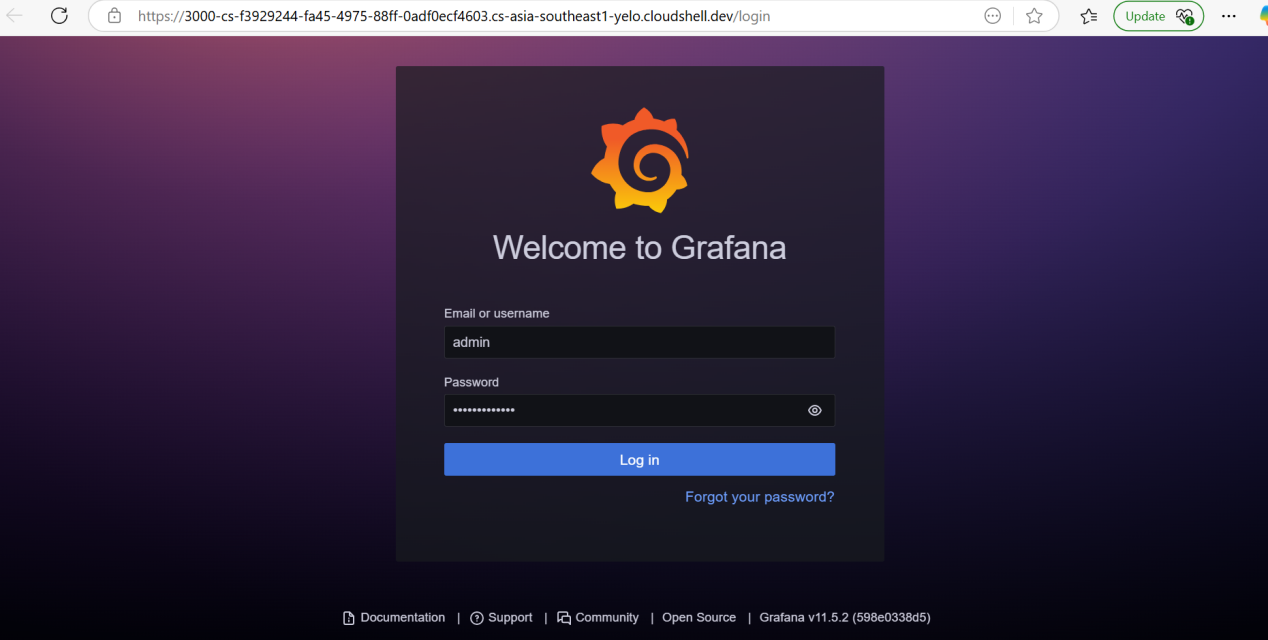
PORT FORWARDING

$ kubectl port-forward svc/prometheus-grafana 3000:80 -n monitoring



Click on the web preview give the port no 3000 and click on change and preview u can see the grafana





You can login with admin and prom-operator

