**Week 11**

**Creating grafana**

gcloud container clusters list

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

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

kubectl get nodes

kubectl create namespace monitoring

kubectl get ns

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

helm repo update

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

kubectl get secrets -n monitoring

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

**Access Grafana and port forwarding**

kubectl port-forward svc/kube-prometheus-stack-grafana -n monitoring 3001:80

**Grafana Credentials:-**

**user name:** admin

**Password:** prom-operator

**week 12**

**Creating Prometheus**

gcloud container clusters list

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

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

kubectl get nodes

kubectl create namespace monitoring

kubectl get ns

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

helm repo update

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

kubectl get secrets -n monitoring

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

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

**week 9**

**cluster creation:-**

gcloud container clusters list

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

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

kubectl get nodes

**Create the pods**

kubectl run --image tomcat webserver

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

kubectl get pods -o wide

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**

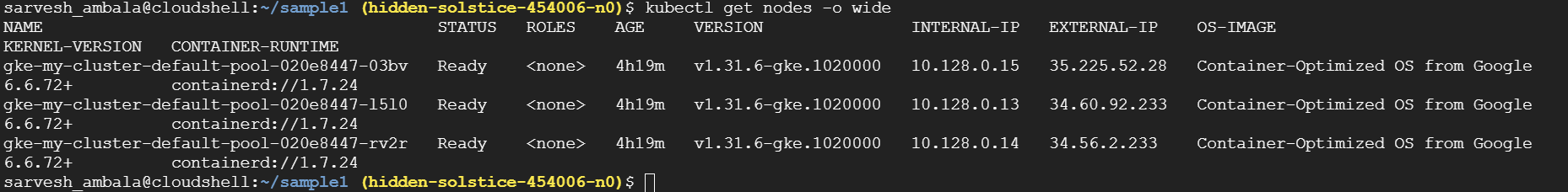
**How to open the port**

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**