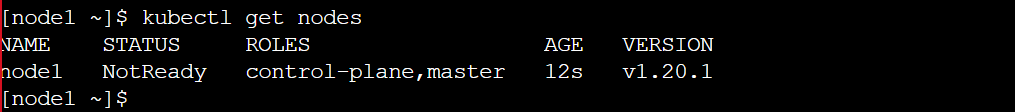
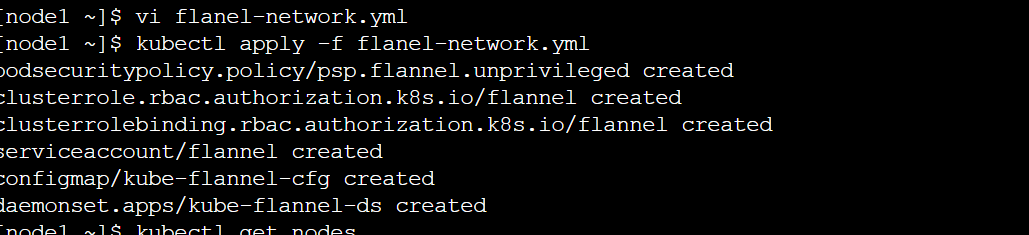
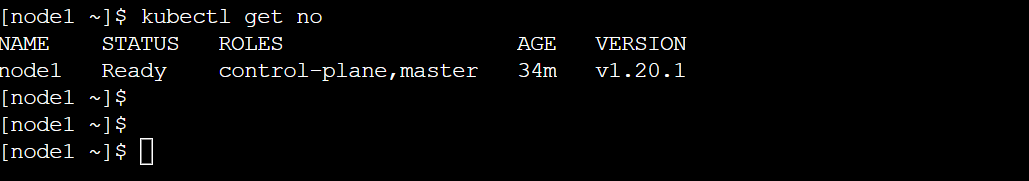
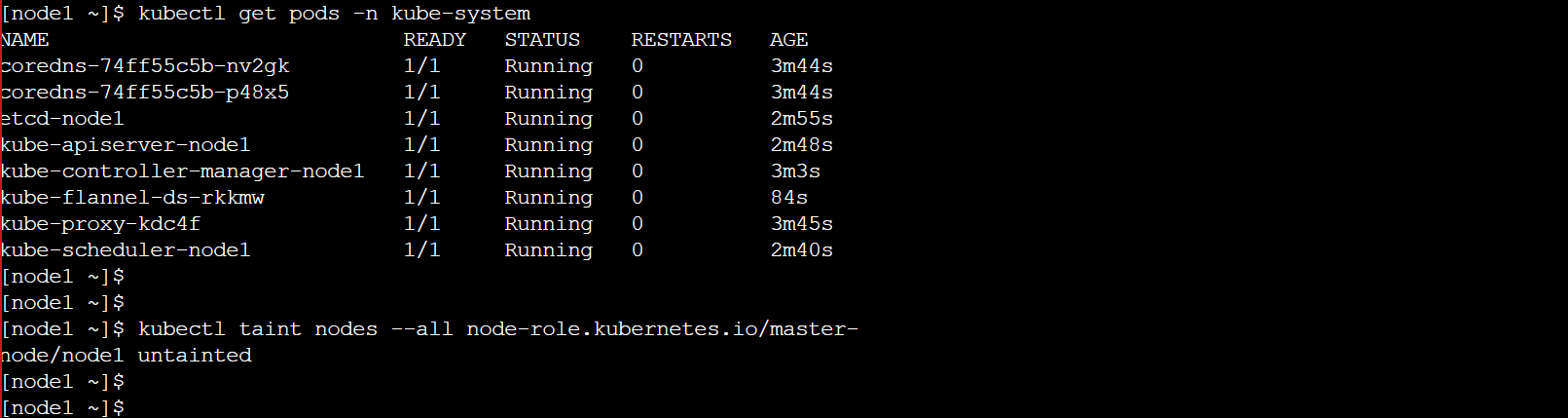
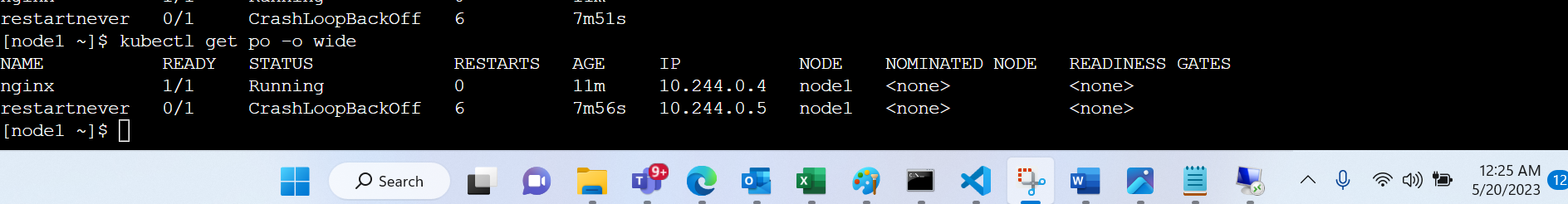
Task. 1: Setup Single node cluster with calico/flannel networking and pod-network-cidr=10.244.0.0/16 using kubeadm. Make sure your cluster is healthy before you proceed.







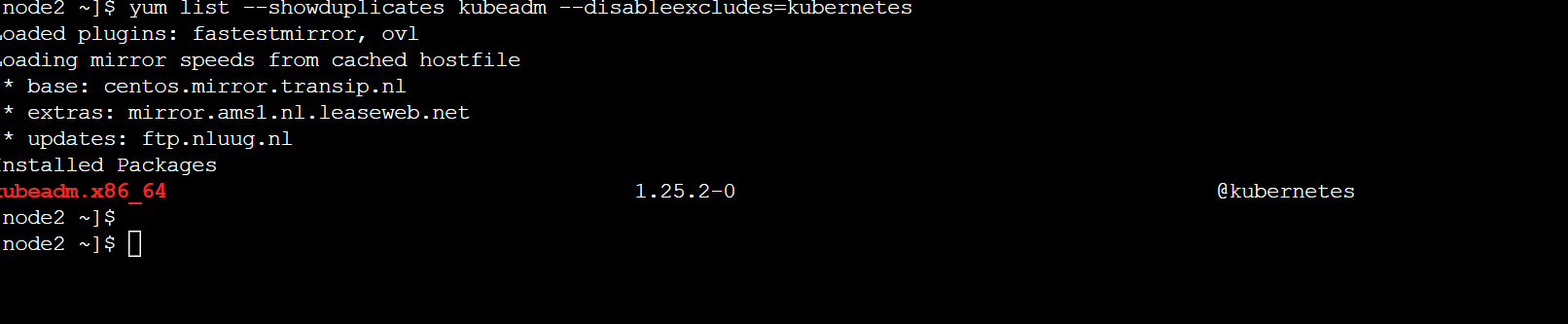


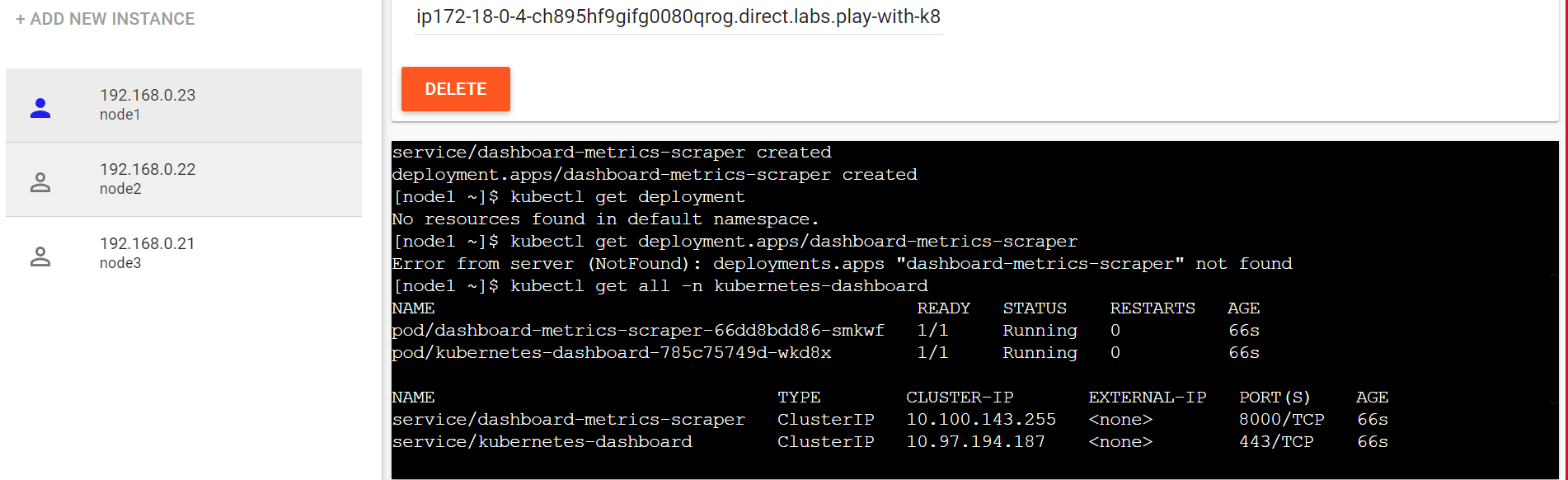


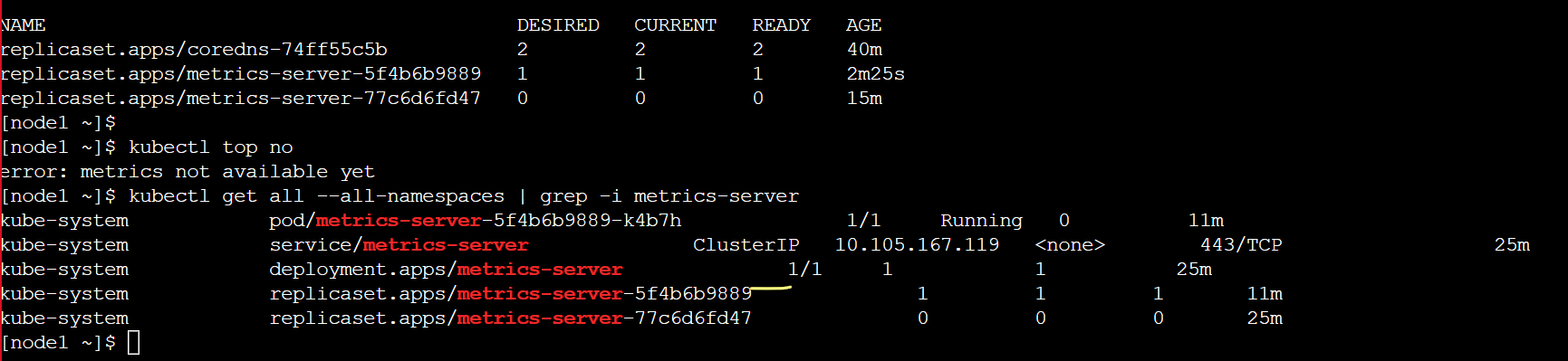
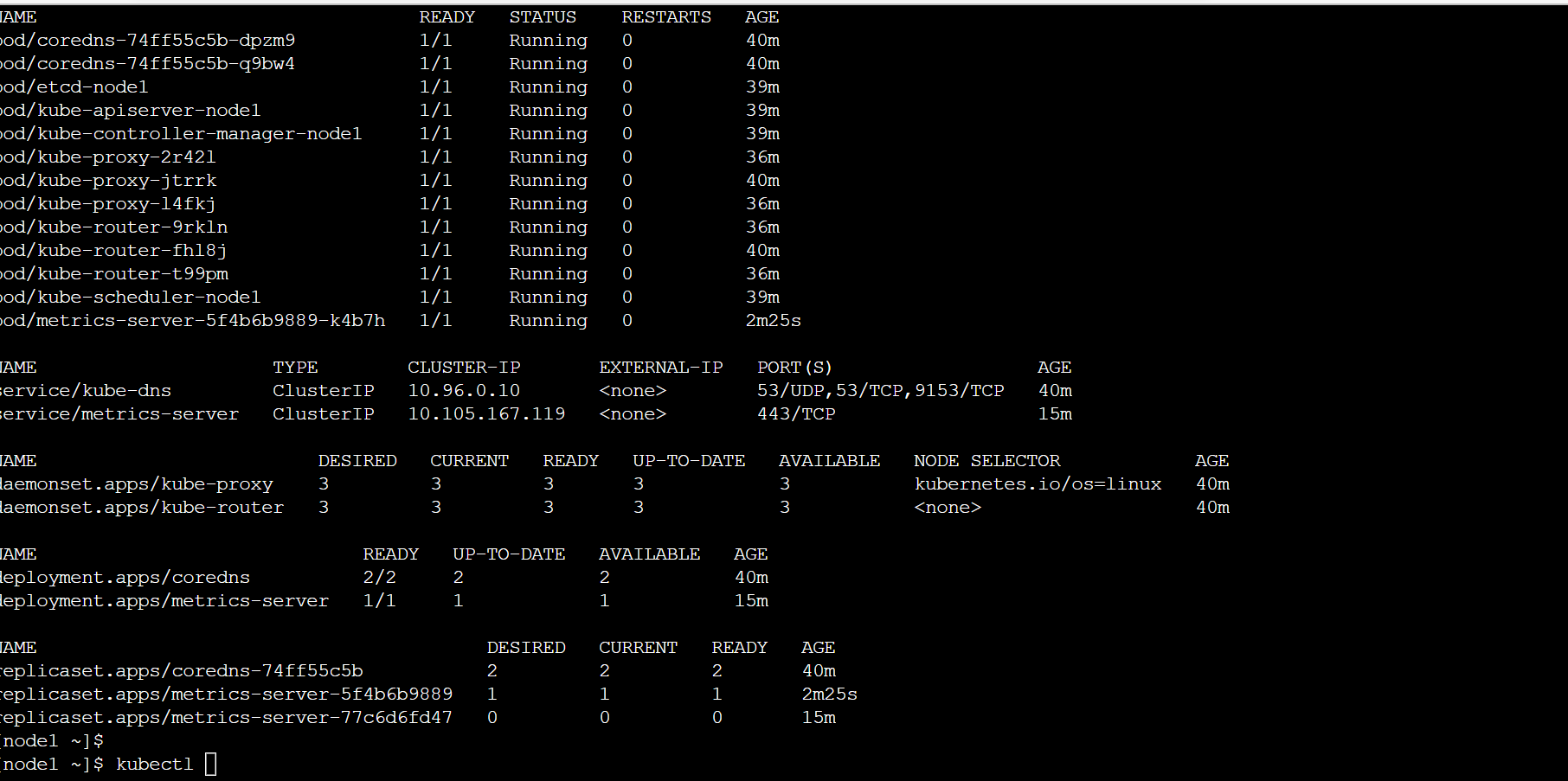
Task 13:

yum install -y kubeadm-1.19.0-0 --disableexcludes=Kubernetes

yum list --showduplicates kubeadm --disableexcludes=kubernetes

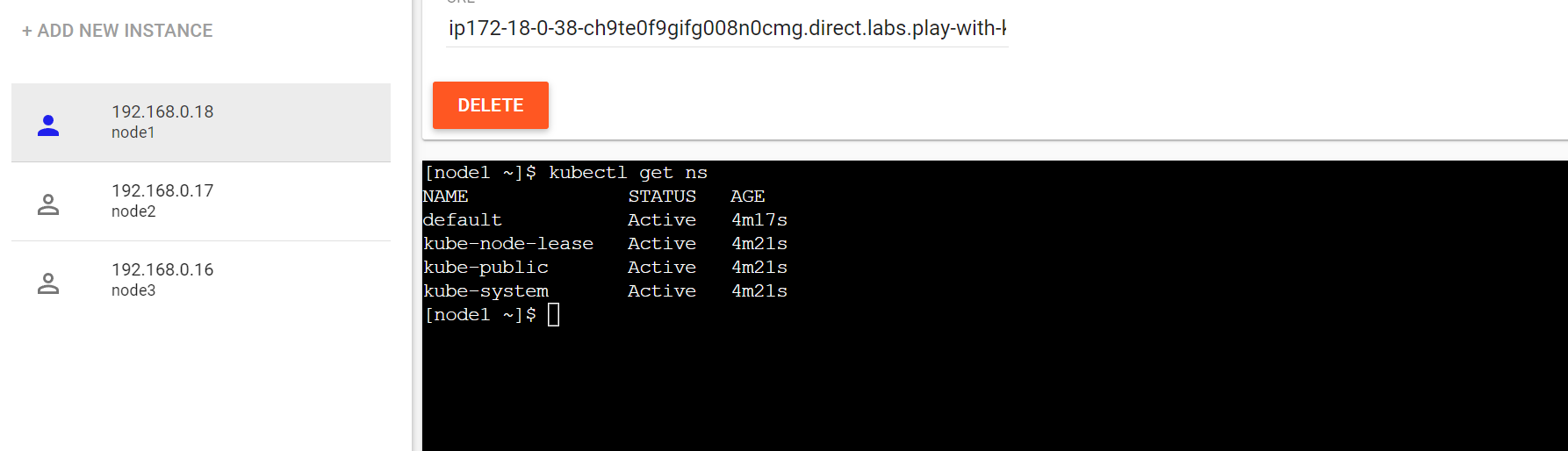




Task 6:

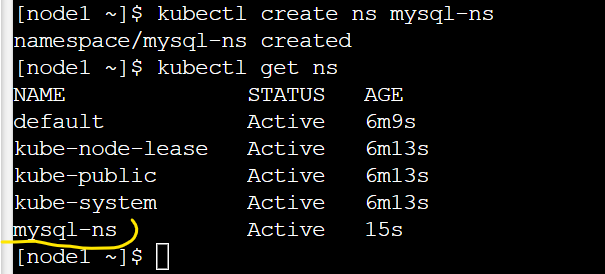
>>> kubectl get no



>>> kubectl create ns mysql-ns

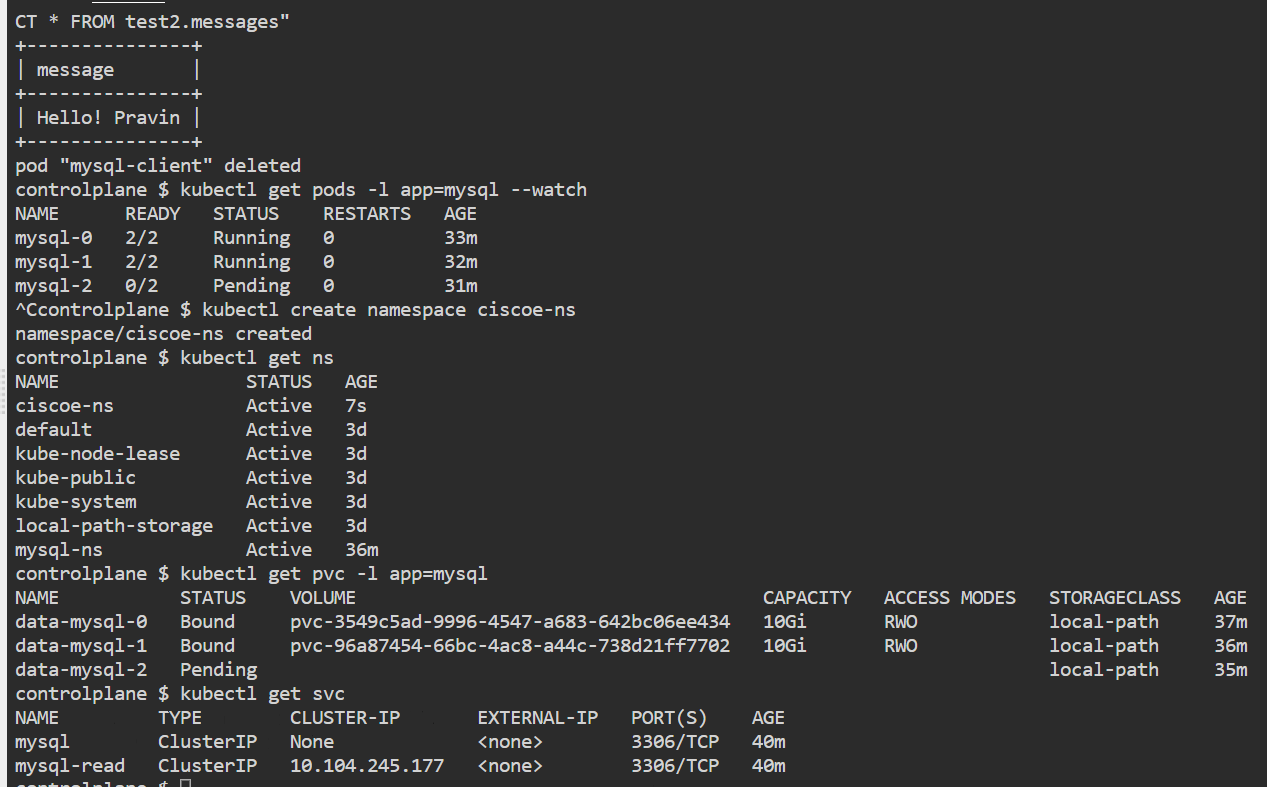
### make mysql-ns default

>>>



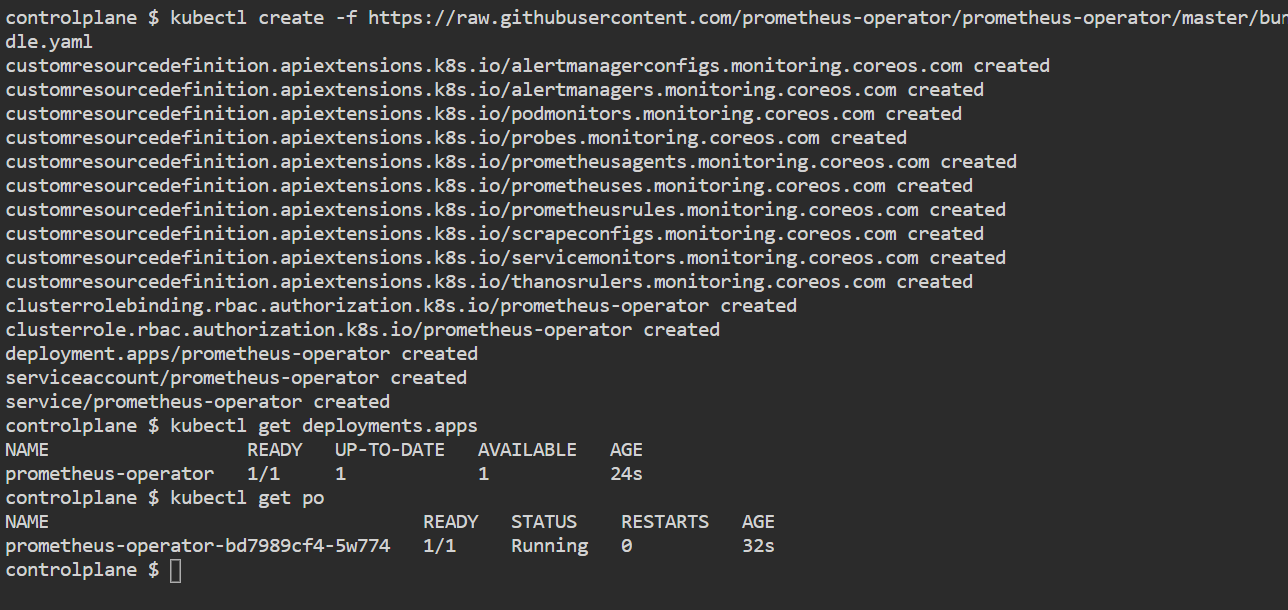
>>> kubectl apply -f mysql-cm.yml

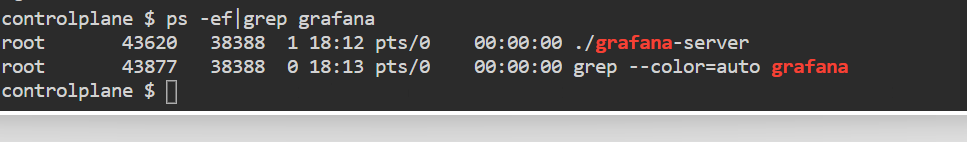
>>> kubectl get cm



**Task 4: Install Prometheus and Grafana. Check the link at https://github.com/coreos/kube-prometheus and follow the steps. Please make sure that you are able to see the metrics of the pods, nodes, etc. Check Grafana is working.**

>>> kubectl create -f https://raw.githubusercontent.com/prometheus-operator/prometheus-operator/master/bundle.yaml



Text

Description automatically generated with medium confidence

**Task 15:** Docker, Kubernetes TODO application -- Digital Ocean (Deploy application as per online course mentioned in below link)

<https://www.digitalocean.com/community/tutorials/webinar-series-deploying-and-scaling-microservices-in-kubernetes>

mkdir /todo-app

cd /todo-app

git clone https://github.com/janakiramm/todo-app.git

### write Docker file

Vi Dockerfile

FROM node:slim

LABEL maintainer = "pravin.ade@doamin.com"

RUN mkdir -p /usr/src/app

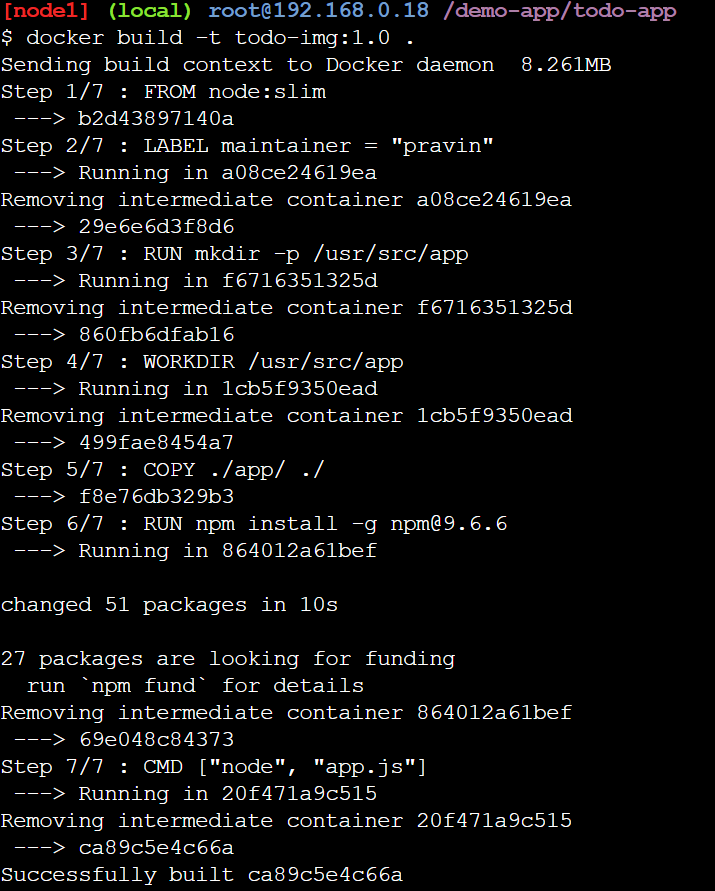
WORKDIR /usr/src/app

COPY ./app/ ./

RUN npm install -g npm@9.6.6

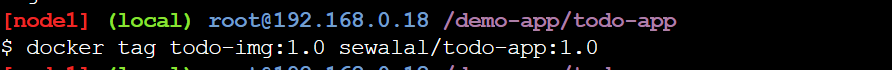
CMD ["node", "app.js"]

docker build -t todo-img:1.0 .



docker login

Text

Description automatically generated

----------------------------------- Deploy MongoDB Pod in Kubernetes

>>> nano db-pod.yaml

apiVersion: v1

kind: Pod

metadata:

name: db

labels:

name: mongo

app: todoapp

spec:

containers:

- image: mongo

name: mongo

ports:

- name: mongo

containerPort: 27017

volumeMounts:

- name: mongo-storage

mountPath: /data/db

volumes:

- name: mongo-storage

hostPath:

path: /data/db

>>> kubectl create -f db-pod.yaml

>>> nano db-service.yaml

apiVersion: v1

kind: Service

metadata:

name: db

labels:

name: mongo

app: todoapp

spec:

selector:

name: mongo

type: ClusterIP

ports:

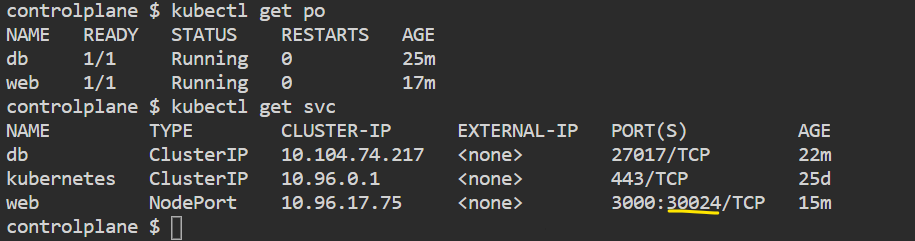
- name: db

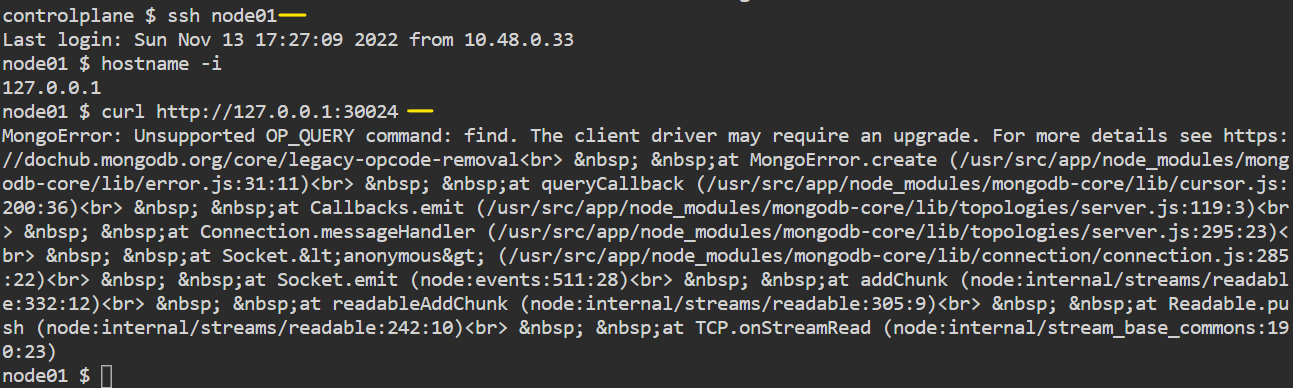
port: 27017

targetPort: 27017

>>> kubectl create -f db-service.yaml

>>> kubectl get svc





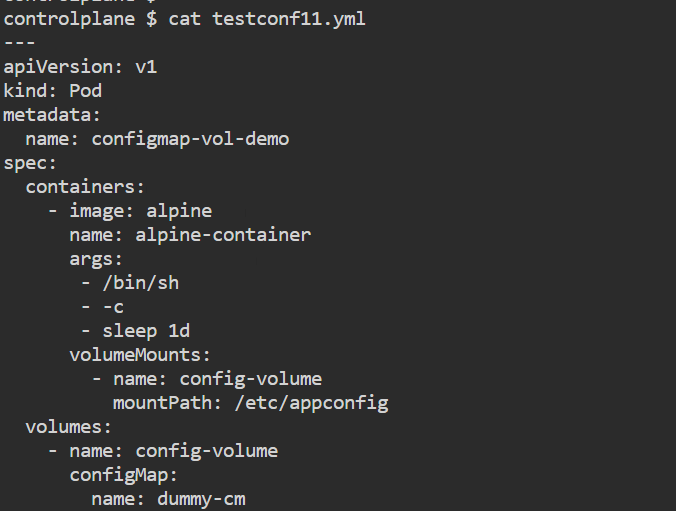
**ConfigMap**

config maps can be mounted to the pods as volumes as well.

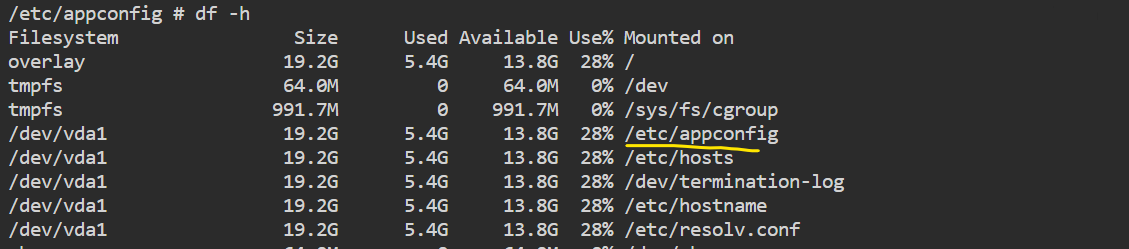
>>> kubectl create configmap example-cm --from-literal=welcome-msg="Hello! I am config map"

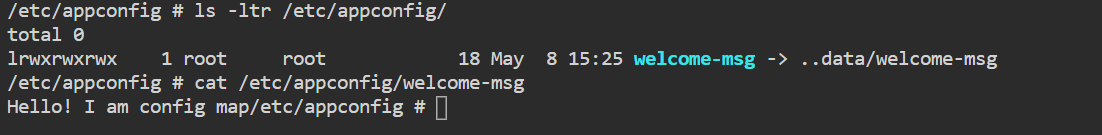
>>> kubectl get cm

>>> kubectl



Kubectl exec -it podname /bin/bash/sh



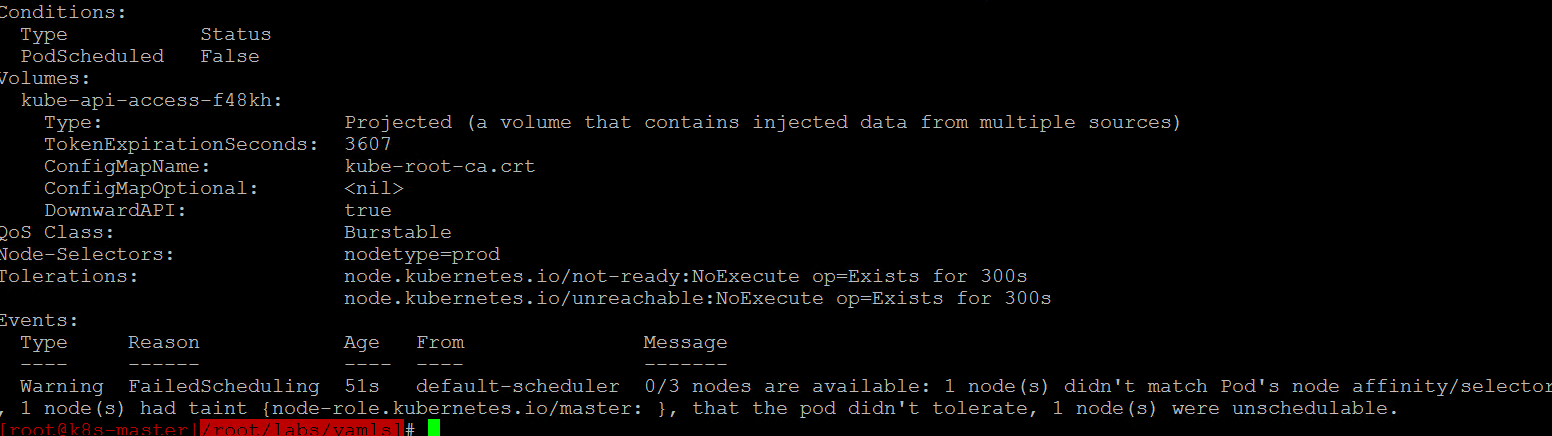


Kubeconf as env variable

### kubectl rollout undo deployment Jenkins-deployment

#### cordon and drain

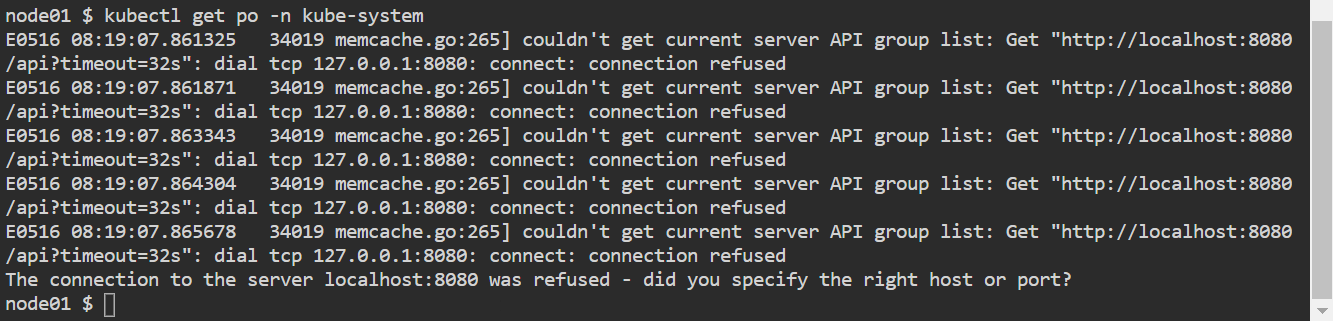
cordon 1st and trying to run simple-webapp.



##### Ingress errors

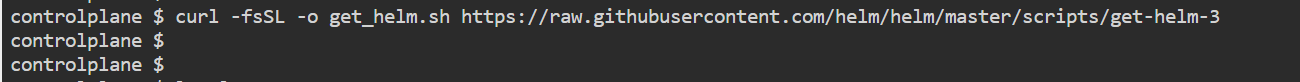
A screenshot of a computer

Description automatically generated



Q. 10 Now install helm v 3 on your node and make sure that helm is working fine

--- curl -fsSL -o get\_helm.sh <https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3>



--- chmod 700 get\_helm.sh

A picture containing text, screenshot, font

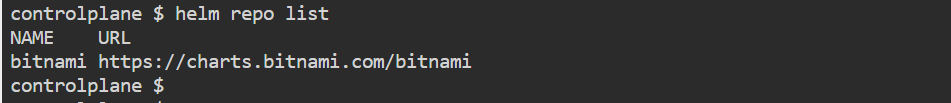
Description automatically generated

>>> sudo sh ./get\_helm.sh

### to add helm repo

helm repo add bitnami <https://charts.bitnami.com/bitnami>

helm repo list



## install mysql from bitnami/repo

Helm installs.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

**Task 13: Install kubernetes cluster using kubeadm use version 1.25.x and use calico CNI and make sure cluster is healthy. Once cluster is running fine upgrade the K8S cluster to version 1.26.y**

upgrade cluster from 1.20.1 to 1.20.15-

---- To list the available kubeadm version in the official repository:

>>> yum list --showduplicates kubeadm --disableexcludes=kubernetes

############ To create token

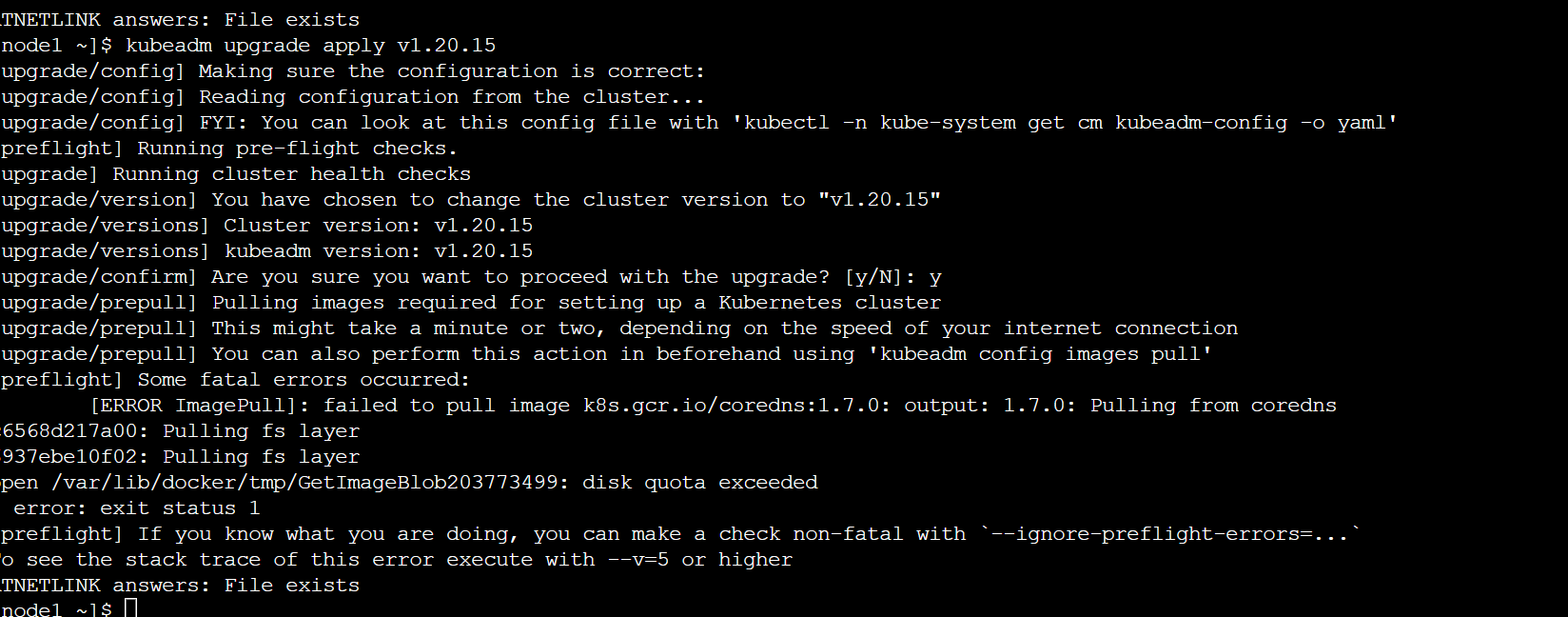
>>> kubeadm token create --print-join-command

kubeadm join 192.168.0.23:6443 --token absojx.xa0vh0s2xxbwklf7 --discovery-token-ca-cert-hash sha256:5344dfe4aef02eb7a079cc75b80c7c1accb0d49e591bff9eb2a4ae5843d99a5a

---- cordone node

>>> kubectl drain controller.example.com --ignore-daemonsets

>>> create upgrade plan | suggest us to upgrade versions of kubeadm



ResourceQuota:

Switch ns : snap install kubectx –classic

Kubens test

A screenshot of a computer

Description automatically generated

ImagePullSecret

Doker login

Get ---- ./docker/config.json

