**Kubernetes command**

**reference :- https://phoenixnap.com/kb/install-minikube-on-ubuntu**

**https://gitlab.com/nanuchi/youtube-tutorial-series/-/blob/master/basic-kubectl-commands/cli-commands.md**

**https://www.youtube.com/watch?v=X48VuDVv0do**

### **kubectl commands**

kubectl get nodes

kubectl get pod

kubectl get services

kubectl create deployment nginx-depl --image=nginx

kubectl get deployment

kubectl get replicaset

kubectl edit deployment nginx-depl

### debugging

kubectl logs {pod-name}

kubectl exec -it {pod-name} -- bin/bash

### create mongo deployment

kubectl create deployment mongo-depl --image=mongo

kubectl logs mongo-depl-{pod-name}

kubectl describe pod mongo-depl-{pod-name}

### delete deplyoment

kubectl delete deployment mongo-depl

kubectl delete deployment nginx-depl

### create or edit config file

vi nginx-deployment.yaml

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: nginx-deployment**

**labels:**

**app: nginx**

**spec:**

**replicas: 2**

**selector:**

**matchLabels:**

**app: nginx**

**template:**

**metadata:**

**labels:**

**app: nginx**

**spec:**

**containers:**

**- name: nginx**

**image: nginx:1.16**

**ports:**

**- containerPort: 8080**

:wq!

kubectl apply -f nginx-deployment.yaml

kubectl get pod

kubectl get deployment

vi nginx-service.yaml

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: nginx-service**

**spec:**

**selector:**

**app: nginx**

**ports:**

**- protocol: TCP**

**port: 80**

**targetPort: 8080**

:wq!

kubectl apply -f nginx-service.yaml

kubectl get service

### delete with config

kubectl delete -f nginx-deployment.yaml

#Metrics

kubectl top The kubectl top command returns current CPU and memory usage for a cluster’s pods or nodes, or for a particular pod or node if specified.

**Kubectl demo project using mongo and mongo-express:**

## to encrypt username and password to be used in secret file

echo -n 'admin' | base64

echo -n 'welcome' | base64

vimongo-secret.yaml

**apiVersion: v1**

**kind: Secret**

**metadata:**

**name: mongodb-secret**

**type: Opaque**

**data:**

**mongo-root-username: YWRtaW4=**

**mongo-root-password: d2VsY29tZQ==**

**Esc**

**:wq!**

**vi** mongo.yaml

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: mongodb-deployment**

**labels:**

**app: mongodb**

**spec:**

**replicas: 1**

**selector:**

**matchLabels:**

**app: mongodb**

**template:**

**metadata:**

**labels:**

**app: mongodb**

**spec:**

**containers:**

**- name: mongodb**

**image: mongo**

**ports:**

**- containerPort: 27017**

**env:**

**- name: MONGO\_INITDB\_ROOT\_USERNAME**

**valueFrom:**

**secretKeyRef:**

**name: mongodb-secret**

**key: mongo-root-username**

**- name: MONGO\_INITDB\_ROOT\_PASSWORD**

**valueFrom:**

**secretKeyRef:**

**name: mongodb-secret**

**key: mongo-root-password**

**---**

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: mongodb-service**

**spec:**

**selector:**

**app: mongodb**

**ports:**

**- protocol: TCP**

**port: 27017**

**targetPort: 27017**

Escape

:wq!

**vi mongo-configmap.yaml**

**apiVersion: v1**

**kind: ConfigMap**

**metadata:**

**name: mongodb-configmap**

**data:**

**database\_url: mongodb-service**

esc

:wq!

**vi mongo-express.yaml**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: mongo-express**

**labels:**

**app: mongo-express**

**spec:**

**replicas: 1**

**selector:**

**matchLabels:**

**app: mongo-express**

**template:**

**metadata:**

**labels:**

**app: mongo-express**

**spec:**

**containers:**

**- name: mongo-express**

**image: mongo-express**

**ports:**

**- containerPort: 8081**

**env:**

**- name: ME\_CONFIG\_MONGODB\_ADMINUSERNAME**

**valueFrom:**

**secretKeyRef:**

**name: mongodb-secret**

**key: mongo-root-username**

**- name: ME\_CONFIG\_MONGODB\_ADMINPASSWORD**

**valueFrom:**

**secretKeyRef:**

**name: mongodb-secret**

**key: mongo-root-password**

**- name: ME\_CONFIG\_MONGODB\_SERVER**

**valueFrom:**

**configMapKeyRef:**

**name: mongodb-configmap**

**key: database\_url**

**---**

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: mongo-express-service**

**spec:**

**selector:**

**app: mongo-express**

**type: LoadBalancer**

**ports:**

**- protocol: TCP**

**port: 8081**

**targetPort: 8081**

**nodePort: 30000**

esc

:wq!

### kubectl apply commands in order

kubectl apply -f mongo-secret.yaml

kubectl apply -f mongo.yaml

kubectl apply -f mongo-configmap.yaml

kubectl apply -f mongo-express.yaml

### kubectl get commands

kubectl get pod

kubectl get pod --watch

kubectl get pod -o wide

kubectl get service

kubectl get secret

kubectl get all | grep mongodb

### kubectl debugging commands

kubectl describe pod mongodb-deployment-xxxxxx

kubectl describe service mongodb-service

kubectl logs mongo-express-xxxxxx

### give a URL to external service in minikube

minikube service mongo-express-service

**Kubectl Namespaces: virtual cluster inside k8s cluster**

kubectl cluster-info

kubectl get namespace

kubectl create namespace my-namespace

kubectl get namespace

kubectl get ns

kubectl apply configmap

# **To change the active namespace**

brew install kubectx

kubens

kubens my-namespace

kubens

**Ingress: instead of service external IP, url is used to access the service.**

**Ref: https://kubernetes.io/docs/tasks/access-application-cluster/ingress-minikube/**

## [**Create a Minikube cluster**](https://gitlab.com/nanuchi/youtube-tutorial-series/-/blob/master/kubernetes-ingress/dashboard-ingress.yaml) **and Start**

**minikube start –driver=virtualbox**

## Enable the Ingress controller

minikube addons enable ingress

kubectl get pods -n ingress-nginx

## Deploy a hello, world app

kubectl create deployment web --image=gcr.io/google-samples/hello-app:1.0

kubectl expose deployment web --type=NodePort --port=8080

kubectl get service web

minikube service web --url

## Create an Ingress resource

vi example-ingress.yaml

**apiVersion: networking.k8s.io/v1**

**kind: Ingress**

**metadata:**

**name: example-ingress**

**annotations:**

**nginx.ingress.kubernetes.io/rewrite-target: /$1**

**spec:**

**rules:**

**- host: hello-world.info**

**http:**

**paths:**

**- path: /**

**pathType: Prefix**

**backend:**

**service:**

**name: web**

**port:**

**number: 8080**

:wq!

kubectl apply -f https://k8s.io/examples/service/networking/example-ingress.yaml

kubectl get ingress

Add the following line to the bottom of the /etc/hosts file.

172.17.0.15 hello-world.info

curl hello-world.info

## **Create Second Deployment**

kubectl create deployment web2 --image=gcr.io/google-samples/hello-app:2.0

kubectl expose deployment web2 --port=8080 --type=NodePort

Edit the existing example-ingress.yaml and add the following lines:

vi example-ingress.yaml

**- path: /v2**

**pathType**: Prefix

**backend**:

**service**:

**name**: web2

**port**:

**number**: 8080

:wq!

kubectl apply -f example-ingress.yaml

## **Test Your Ingress**

curl hello-world.info

curl hello-world.info/v2