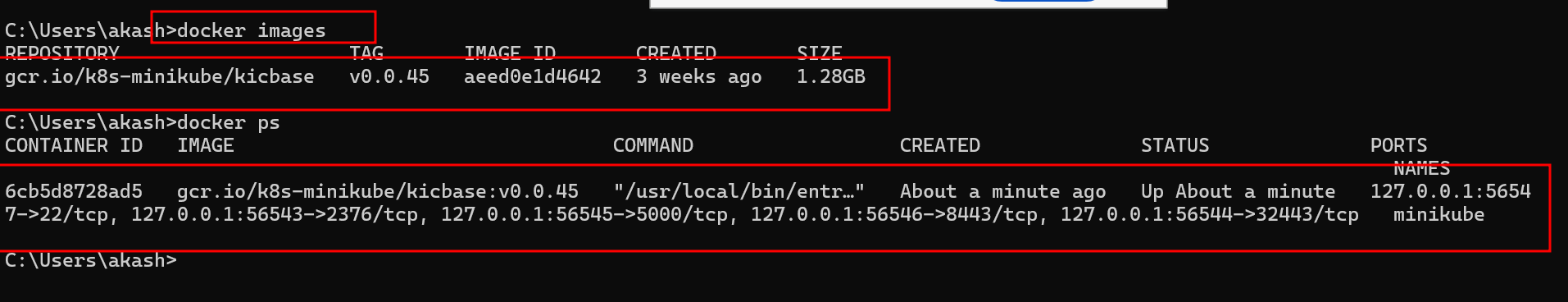
Minikube : minikube is one of the tool which provide single node cluster for Kubernetes. It is a GUI base.

After downloaded minikube please installed minikube

Verify minikube running or not using command as

minikube or minikube version

minikube start this command is use to start my minikube cluster environment.



After start successfully please write below command to open minikube dashboard

>kubectl cluster-info single cluster

This command provide us cluster information

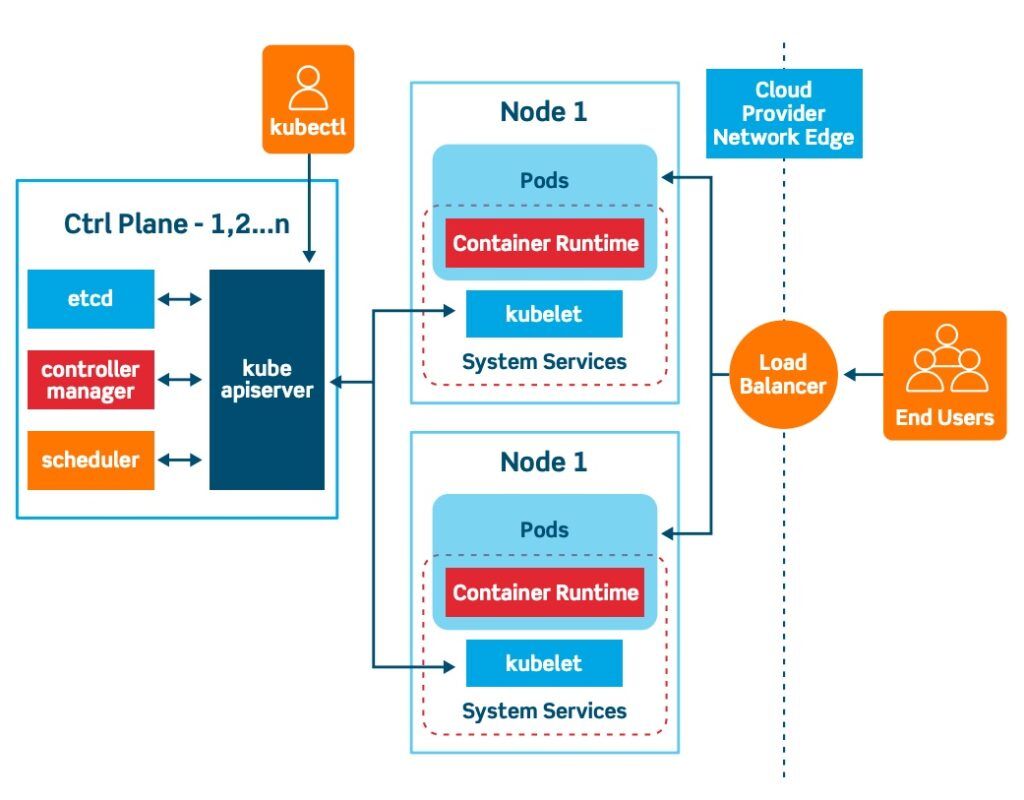
->kubectl get node

It provide us our machine details. This machine running cluster information ie control plane ie master node or main node.

We are only master node as well as worker node.

In real time using kubeadm we can run master node

Kubectl init to join more than one node to deploy the docker image.



In Kubernetes we can deploy or interact container directly. We need to use pods. Pods wrap one or more than one container and each container responsible to run application ie mysql or spring boot.

using pods we can run more than one container. We can specify optionally how much CPU and memory for each container required.

To find the pods information we need to run the command as

kubectl get pods

this command provide us pods details.

We can create the pods using ,yml or using commands. Ie imperative or declarative manner.

kubectl create deployment my-deployment --image=akashkale/my-spring:1.111

kubectl get deployment

kubectl get pods

deployment provide extra layer to created or provide configuration to run more than one container using pods.

To delete deployment using command

Kubectl delete deployment hello-pods

Kubectl delete pods podsname

We can create more than one replica ie pods for same using kubectl commands.

kubectl create deployment hello-more-deployment --image=akashkale/my-spring:1.111 --replicas=10

Pods are part of Kubernetes cluster environment. We can’t access those pods outside cluster environment. We can access those pods within a cluster environment with private ip address.

To allow to access these application outside cluster environment we need to expose our pods as a service.

This we can expose using command or using yml file.

Using command we expose our service

Type of service

LoadBalancer

Cluster-Ipd

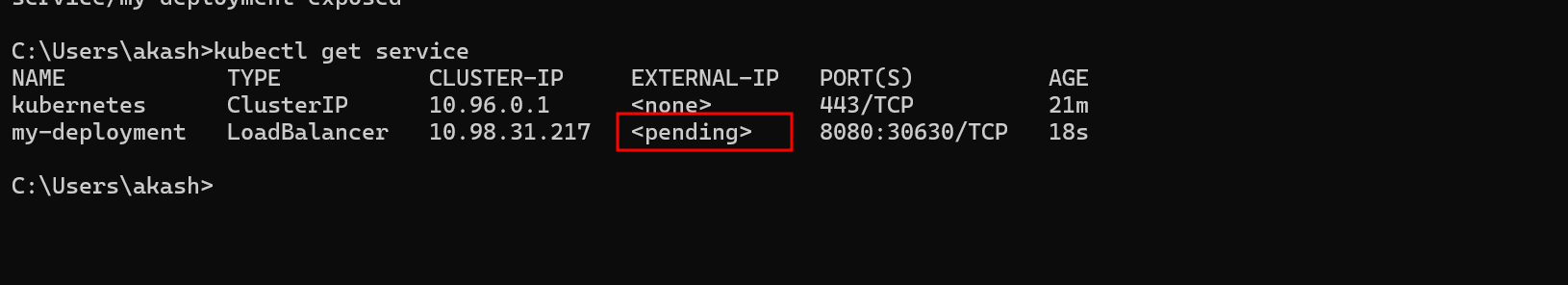
NodePort

kubectl expose deployment deployment\_name --type=LoadBalancer --port=8080

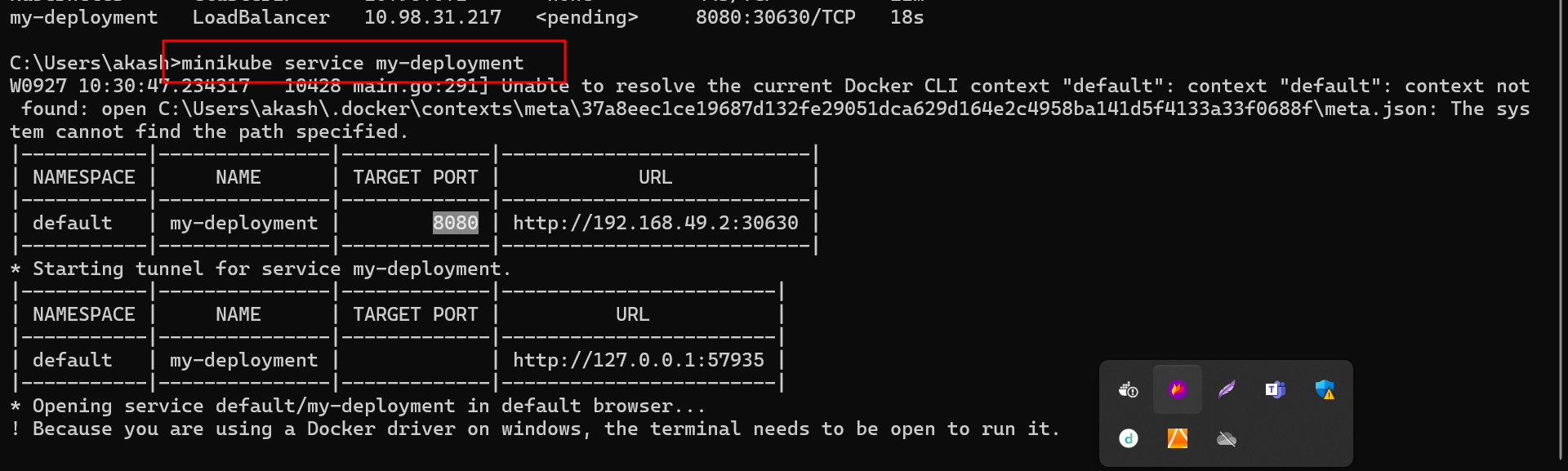
kubectl expose deployment my-deployment --type=LoadBalancer --port=8080

kubectl get service

in minikube cluster



Extra command in minikube cluster we need to write to expose our application outside environment.



**Delete service**

kubectl delete service my-deployment

**creating service with user defined name**

kubectl expose deployment my-deployment --name my-deployment-service --type=LoadBalancer --port=8080

**kubectl get service**

it will give service name

run the application using minikube command with help of service on browser.

minikube service my-deployment-service

creating the pods using yml file

pods.yml

apiVersion: v1

kind: Pod

metadata:

name: my-server

labels:

app: my-pod

spec:

containers:

- name: web-server-container

image: akashkale/my-simple-kuberneties:1.0

kubectl apply -f pods.yml

command to create the namespace using yml file

kubectl create namespace dev

or

**namespace.yml**

apiVersion: v1

kind: Namespace

metadata:

name: dev

**deployment.yml under the namespace as dev**

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-server

labels:

app: my-server

namespace: dev

spec:

replicas: 3

selector:

matchLabels:

app: my-server

template:

metadata:

labels:

app: my-server

spec:

containers:

- name: web-server-container

image: akashkale/my-simple-kuberneties:1.0

**kubectl apply -f deployment.yml**

creating service for this deploy

**deployment-service.yml**

apiVersion: v1

kind: Service

metadata:

name: simple-app-service

labels:

app: my-server

namespace: dev

spec:

type: LoadBalancer

selector:

app: my-server

ports:

- port: 80

nodePort: 30001

targetPort: 80

protocol: TCP

**kubectl get deployment --namespace dev**

**kubectl get service --namespace dev**

**minikube service simple-app-service --namespace dev**

git

maven

gradle

docker

docker compose

Kubernetes

Docker desktop

Minikube is one of the tools provide us cluster ie single node cluster.

Kubeadmin

Using kubectl command

Clouds like AWS or Azure or Google

Ansible

Terra form

application.properties file

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=${SPRING\_DATASOURCE\_URL}

spring.datasource.username=${SPRING\_DATASOURCE\_USERNAME}

spring.datasource.password=${MYSQL\_ROOT\_PASSWORD}

spring.jpa.hibernate.ddl-auto=update

**mysql-deployment.yml**

**kubectl apply -f mysql-deployment.yml**

**spring-boot-deployment.yml**

**kubectl apply -f spring-boot-deployment.yml**

**kubectl apply -f spring-boot-deployment.yml**