* CB FSD - Integration and Deployment

Day 15 : 9 Aug 2024

Kubernetes : Kubernetes also known as K8s. It is an a open source platform designed to automate the deployment, scaling and managed containerized application. It is also known as a container management tools. It was originally developed by Google and it is part of CNCF (Cloud Native Computing Foundation).

To run the application using Docker we create image and we run the container. Every container responsible to run some application develop using any language.

If we want to run more than one container those container want to communicate with each others.

Using Docker-compose. In YML configuration file we can provide all container details like service name, networking details, volume etc.

In Docker compose all container must be running in same machine or node.

Docker compose doesn’t provide any features like scale up, scale down, backup, security, rollout, if any container go down to create new container etc.

Kubernetes also known as orchestration tool. This tool is responsible to maintain the life of the containers.

List of tool responsible to maintain the life the container or also known as orchestration tools.

1. Kubernetes
2. Docker swarm
3. GKE (Google Kubernetes Engine)
4. EKS (Amazon Elastic Kubernetes Service)
5. Azure container app
6. Open shift container platform

Kubernetes components

1. Cluster : A set of machine, either physical or virtual machine or device that run the Kubernetes Environment. In Kubernetes cluster environment we need at least one master node and one or more worker node.
2. Master node : Master node container all Kubernetes run time environment like control plane which is responsible for managing the state of the cluster. It includes component like api server (kube-apiserver), etcd(which hold key-value cluster data), kube-scheduler and controller manager (kube-controller-manager).

To run the Kubernetes cluster we need take help of

1. Docker desktop
2. Minikube

It provide single cluster environment

1. Kubeadm

It provide multi cluster environment. Generally we use in production environment.

We use minikube single cluster environment

First install minikube in local machine

<https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download>

Then start docker

After installed

minikube –version

then run the

minikube start

using docker images and docker ps verify images and running containers.

kubectl : it is a tool or command line tool use to interact with Kubernetes cluster environment. This tool must be present in worker node. That machine is responsible to deploy the application on cluster environment.

Please refer below website to download kubectl tool.

<https://kubernetes.io/docs/tasks/tools/>

kubectl version

kubectl cluster-info

**kubectl get nodes**

Pods : In Kubernetes Pods is the smallest and simplest unit in the Kubernetes object model that you can create and deploy. In Kubernetes we can run container directly. That container must be part of pods. Pods can contains one or more than one container.

Application or program 🡪 using docker file we create the image-🡪 images is responsible to run the container in docker environment. But in Kubernetes this container is part of pods.

We can create the pods using 2 ways

1. Imperative mode

We use command line tool to write the instruction. This option useful for testing the application.

1. Declarative mode

In this mode we use .yml. inside this file we write all configuration details which help to run the pods.

Kubectl command

Creating pods using imperative mode

kubectl run testpod --image=busybox -it

kubectl describe pods testpod1 check pods details.

Running the pods using declarative mode

**testpod.yml**

apiVersion: v1

kind: Pod

metadata:

name: testpod

labels:

app: testpod

spec:

containers:

- image: busybox

command:

- sleep

- "3600"

imagePullPolicy: IfNotPresent

name: busybox

restartPolicy: Always

kubectl apply -f testpod.yml

deployment : it is a Kubernetes resources that provide declarative or imperative mode to run more than one pods and we can update those pods with help of scale up and down etc.

creating deployment resource using imperative mode

kubectl create deployment my-web-deployment --image=akashkale/my-web:123

kubectl get deployment

kubectl get pods

creating deployment.yml file

by default all application we can access within a cluster environment.

But if we want to access this application outside cluster environment we need to expose as service.

How to expose deployment outside cluster using imperative mode

kubectl expose deployment my-web-deployment --type=LoadBalancer --port=80

minikube service my-web-deployment

**creating service using declarative mode**

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

to check the service

kubectl get service --namespace=dev

to run the service using minikube

minikube service simple-app-service --namespace=dev