Phase 5

Total classes : 10

Day 10 : 15 Feb 2024

Docker is use to create containerization application.

Using the docker we can create image which is responsible to run application with help of container. Application can be java, python, node js, angular , react js.

If we want run only one container is Docker is good option.

But if we want to run multiple containers and those containers are communicating with each others.

Like

Spring boot with MySQL Database

Angular with spring boot with mysql database container.

If we want to start, stop or restart these container we can use docker-compose.

Docker-compose is responsible to run multi container using docker compose file.

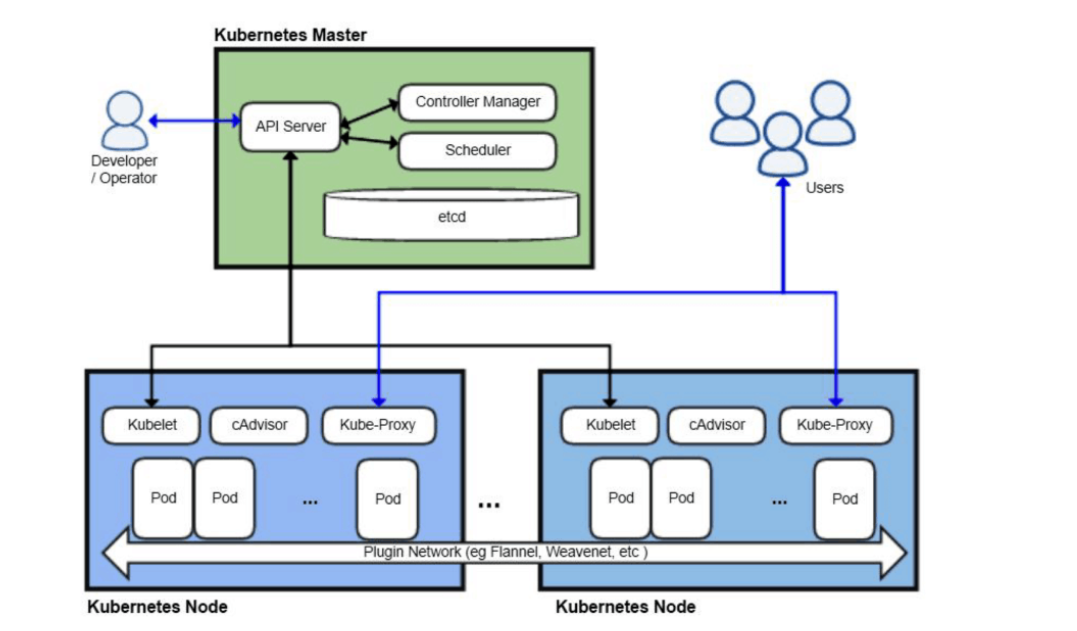
But if want to run 100 container as well as we want maintain container life cycle.

On demand we can scale up or down, Scheduling, container health, if any container go down we need backup of that container. To maintain all container life we need some tool ie orchestration tool. It is a type tool which is responsible to maintain the container life cycle.

Generally we can run all container in same machine.

Docker compose is responsible to run more than one container but all container must be run in same machine or same domain and it can manage may 5 to 10 container.

1. Kubernetes : it is part of google.
2. Docker Swarm : it is part of docker.



Node : Node is physical machine or device which can be identity by using domain name or IP Address.

EC2 instance ---🡪 Master Node ( we install all required Kubernetes software which provided us cluster environment allow to configure more than one worker node).

Cluster : it is collection of machine or nodes which connected to each others.

In Every work node we can create pods.

In Kubernetes we can’t communicate to container directly. Container wrap by pods. Pods is a collection of more than one container.

In one node ie machine we can create more than one pods. Each pods contains more than one container. Each container is link with one image which is responsible to run the application.

If we want to run Kubernetes cluster environment

We need

1. kubeadmn : software which provide multi cluster environment.
2. minikube : single cluster
3. docker desktop : single cluster

if we want to interact with master node ie local machine then Kubernetes provide kubectl (kube control) command line interface.

kubectl cluster-info

kubectl config current-context

kubectl config get-contexts

kubectl get nodes

kubectl get pods

kubectl create deployment myangularapp --image=akashkale/my-angular-app:v1

kubectl get deployment

kubectl get pods

delete the deployment

kubectl delete deployment myangularapp

Kubernetes deployment responsible to maintain the life of more than one pods. Each pods contains one or more container. Each container is responsible to run the application.

We can’t access container from pods directly we need expose as a service to access the container outside a pods.

kubectl get service

kubectl expose deployment myangularapp1 --name=ser1

--type=NodePort --port=80

This service expose in cluster environment but outside pods.

kubectl get services

if we want to expose this application outside cluster environment

then we need to runt command as

kubectl port-forward --address 0.0.0.0 service/ser1 81:80

1. Spring boot application : Create simple rest api
2. This project push in git repository
3. Login to AWS account
4. Then create EC2 instance
5. Then install java, git, jenkin in Ec2 instance.
6. Using Jenkin pipeline or jobs we run java jar file or docker image
7. Then we can access that application using <http://IpAddress:port>