Day 4:

Kubernetes and minikube

Notes:

1. Create a pod using run command

$ kubectl run <pod-name> --image=<image-name> --port=<container-port>

$ kubectl run my-pod --image=nginx --port=80

2. View all the pods

(In default namespace)

$ kubectl get pods

(In All namespace)

$ kubectl get pods -A

# For a specific namespace

$ kubectl get pods -n kube-system

# For a specific type

$ kubectl get pods <pod-name>

$ kubectl get pods <pod-name> -o wide

$ kubectl get pods <pod-name> -o yaml

$ kubectl get pods <pod-name> -o json

3. Describe a pod (View Pod details)

$ kubectl describe pod <pod-name>

$ kubectl describe pod my-pod

4. View Logs of a pod

$ kubectl logs <pod-name>

$ kubectl logs my-pod

Installation:

Kubernetes installation:

1. Go to <https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/>

2. curl -LO <https://dl.k8s.io/release/v1.32.0/bin/linux/amd64/kubectl>

3. sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

4. chmod +x kubectl

   mkdir -p ~/.local/bin

   mv ./kubectl ~/.local/bin/kubectl

5. kubectl version --client

Minikube installation:

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

2. curl -LO <https://github.com/kubernetes/minikube/releases/latest/download/minikube-linux-amd64>

3. sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64

4. minikube start

5. minikube status

6. kubectl get pod

7. kubeclt get deploy

8. kubectl get replica or rs or replicaaset

9. kubectl get pod -o wide

Yml files:

Pod.yml

apiVersion: v1

kind: Pod

metadata:

  name: my-pod

  labels:

      app: my-web-app

spec:

  containers:

    - name: nginx-container

      image: devasria/mysimplewebapplication:latest

      ports:

        - containerPort: 80

rs-test.yml

apiVersion: apps/v1

kind: ReplicaSet

metadata:

  name: my-rs

  labels:

    name: my-rs

spec:

  replicas: 4

  selector:

    matchLabels:

      apptype: web-backend

  template:

    metadata:

      labels:

        apptype: web-backend

    spec:

      containers:

      - name: my-app

        image: devasria/mysimplewebapplication:latest

        ports:

          - containerPort: 8080

Deploy.yml

apiVersion: apps/v1

kind: Deployment

metadata:

  name: my-deploy

  labels:

    name: my-deploy

spec:

  replicas: 5

  selector:

    matchLabels:

      apptype: web-backend

  strategy:

    type: RollingUpdate

  template:

    metadata:

      labels:

        apptype: web-backend

    spec:

      containers:

      - name: my-app

        image: devasria/mysimplewebapplication:latest

        ports:

              - containerPort: 7070

Namespace.yml

apiVersion: v1

kind: Pod

metadata:

  name: my-deploy

  namespace: mydeploy

spec:

  containers:

  - name: my-container

    image: nginx:latest



