Kubernetes Notes

Pods

A **Pod** is the basic building block of Kubernetes. A pod represents a single instance of a running process within the cluster. It can encapsulate one or more containers that share the same network and storage resources.

1. Create a Pod using the `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 the default namespace

kubectl get pods

In all namespaces

kubectl get pods -A

For a specific namespace

kubectl get pods -n kube-system

For a specific Pod

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

5. Execute any command inside a Pod (Inside Pod OS)

kubectl exec <pod-name> -- <command>

ReplicaSet

YAML Definition for ReplicaSet

```
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: <image>
        ports:
          - containerPort: 8080
```

1. Create ReplicaSet using YAML file

kubectl create -f rs-test.yml
kubectl apply -f rs-test.yml
kubectl replace -f rs-test.yml # Completely modify Pod Template

2. View ReplicaSets

```
kubectl get replicasets
kubectl get rs
kubectl get rs -o wide
kubectl get rs <replica-set-name> -o json
kubectl get rs <replica-set-name> -o yaml
```

3. View ReplicaSet Description

```
kubectl describe rs <replica-set-name>
```

4. Modify a ReplicaSet

```
kubectl edit rs <replica-set-name>
# Change replicas count, then press (ESC):wq
kubectl apply -f rs-test.yml
```

5. Scale a ReplicaSet

```
kubectl scale replicaset <replicaset-name> --replicas=<desired-replica-count>
```

6. Delete a ReplicaSet

```
kubectl delete rs <replica-set-name>
kubectl delete -f rs-test.yml
```

Deployment

YAML Definition for Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      apptype: web-backend
```

```
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - name: my-app
        image: <image>
        ports:
        - containerPort: 7070
```

1. Create Deployment using YAML file

```
kubectl create -f web-deploy.yml
kubectl apply -f web-deploy.yml
kubectl replace -f web-deploy.yml  # Completely modify Pod Template
```

2. View Deployments

```
kubectl get deployments
kubectl get deploy
kubectl get deploy -o wide
kubectl get deploy <deployment-name> -o json
kubectl get deploy <deployment-name> -o yaml
```

3. View Deployment Description

```
kubectl describe deploy <deployment-name>
```

4. Modify a Deployment

```
kubectl edit deploy <deployment-name>
# Change replicas count, then press (ESC):wq
kubectl apply -f web-deploy.yml
```

5. Scale a Deployment

```
kubectl scale deploy <deployment-name> --replicas=<desired-replica-count>
```

6. Delete Deployment

kubectl delete deploy <deployment-name>
kubectl delete -f web-deploy.yml

Service

YAML Definition for Service

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-service
    type: backend-app
spec:
  type: NodePort
  ports:
   - targetPort: 7070
     port: 7070
     nodePort: 30002
  selector:
    apptype: web-backend
```

Useful Kubernetes Commands

kubectl create deployment webnginx2 --image=nginx:latest --replicas=1
kubectl scale deploy <deployment-name> --replicas=<desired-replica-count>