**Kubernetes Notes:**

1. **StatefulSets:**

With the help of StatefulSets user can create master-slave architecture. In normal case we create to create 2 relicas of pods which you to balance the load b/w the server but in case of StatefulSet we can create master-slave architecture.

With the help of Master-slave architecture data replicas became very easy. With write can be done on master and read can be done from master or slave servers.

Data b/w master – slaves remain in sync.

Using statefulset the hostname of pods defined by itself which will remain fixed even after the pod dies and restart with this help connectivity with the other pods become easier.

Like the below can we will be create mysql database with 2 sets.

* **We need to create PV.**

apiVersion: v1

kind: PersistentVolume

metadata:

name: mysql1

spec:

accessModes:

* ReadWriteOnce

storageClassName: -

capacity:

storage: 10Gi

hostPath:

path: /mysql1

----------------------

apiVersion: v1

kind: PersistentVolume

metadata:

name: mysql2

spec:

accessModes:

* ReadWriteOnce

storageClassName: -

capacity:

storage: 10Gi

hostPath:

path: /mysql2

* **Secrets for mysql file.**

cat <<EOF >./kustomization.yaml

secretGenerator:

- name: mysql-pass

literals:

- password=Thanku71

EOF

* **Mysql as StatefulSet and service files.**

**apiVersion**: v1

**kind**: Service

**metadata**:

**name**: mysql

**labels**:

**app**: mysql

**spec**:

**ports**:

- **port**: 3306

**selector**:

**app**: mysql

**clusterIP**: None

**Mysql as StatefulSets:**

apiVersion: apps/v1

**kind**: StatefulSet

**metadata**:

**labels**:

**name**: mysql-set

**app**: mysql

**spec**:

**selector**:

**matchLabels**:

**app**: mysql

serviceName: “mysql”

replicas: 2

**template**:

**metadata**:

**labels**:

**app:** mysql

**spec**:

**terminationGracePeriodSeconds**: 15 //This used to give pods 15 sec to shutdown before termination

**containers**:

- **image**: mysql:5.6

**name**: mysql

**ports**:

- **containerPort**: 3306

**volumeMounts**:

- **name**: mysql-pvc

**mountPath**: /var/lib/mysql

**env**:

- **name**: MYSQL\_ROOT\_PASSWORD

**valueFrom**:

**secretKeyRef**:

**name**: mysql-pass

**key**: password

**volumeClaimTemplates:**

- metadata:

name: mysql-pvc

spec:

accessModes:

- ReadWriteOnce

storageClassName: -

resources:

requests:

storage: 10Gi

nodeAffinity:

required:

nodeSelectorTerms:

- matchExpressions:

- key: kubernetes.io/hostname

operator: In

values:

- <values>