**MongoDB StatefulSet**

**MongoDB Stateful-Set**

---

apiVersion: v1

kind: Service

metadata:

name: mongo

namespace: webapps

labels:

app: mongo

spec:

ports:

- port: 27017

name: mongo

clusterIP: None

selector:

app: mongo

---

apiVersion: apps/v1

kind: StatefulSet

metadata:

name: mongo

namespace: webapps

spec:

serviceName: "mongo"

replicas: 3

selector:

matchLabels:

app: mongo

template:

metadata:

labels:

app: mongo

spec:

initContainers:

- name: init-mongo

image: mongo:7.0

command:

[

"bash",

"-c",

"if [[ $(hostname) == \*-0 ]]; then \

echo 'Starting MongoD temporarily in init container...'; \

mkdir -p /data/db; \

mongod --replSet rs0 --bind\_ip\_all --dbpath /data/db --fork --logpath /var/log/mongodb.log; \

until mongosh --eval 'print(\"waiting for startup\")'; do sleep 5; done; \

mongosh --eval 'rs.initiate({\_id: \"rs0\", members: [ \

{\_id: 0, host: \"mongo-0.mongo.webapps.svc.cluster.local:27017\"}, \

{\_id: 1, host: \"mongo-1.mongo.webapps.svc.cluster.local:27017\"}, \

{\_id: 2, host: \"mongo-2.mongo.webapps.svc.cluster.local:27017\"} \

]})'; \

mongod --shutdown --dbpath /data/db; \

else echo 'Skipping RS init on non-zero pod'; fi"

]

volumeMounts:

- name: mongo-persistent-storage

mountPath: /data/db

containers:

- name: mongo

image: mongo:7.0

command:

- mongod

- "--replSet"

- rs0

- "--bind\_ip\_all"

ports:

- containerPort: 27017

volumeMounts:

- name: mongo-persistent-storage

mountPath: /data/db

livenessProbe:

exec:

command:

- mongosh

- --eval

- "db.adminCommand('ping')"

initialDelaySeconds: 10

periodSeconds: 10

timeoutSeconds: 2

failureThreshold: 5

readinessProbe:

exec:

command:

- mongosh

- --eval

- "db.adminCommand('ping')"

initialDelaySeconds: 5

periodSeconds: 5

timeoutSeconds: 2

failureThreshold: 3

volumeClaimTemplates:

- metadata:

name: mongo-persistent-storage

spec:

accessModes: ["ReadWriteOnce"]

storageClassName: "ebs-sc" # Change based on your cloud or local setup

resources:

requests:

storage: 5Gi

**Check If Sync Is there or not**

kubectl run -it --rm mongo-client -n webapps --image=mongo:7.0 -- bash

mongosh "mongodb://mongo-0.mongo.webapps.svc.cluster.local:27017"

rs.initiate({

\_id: "rs0",

members: [

{ \_id: 0, host: "mongo-0.mongo.webapps.svc.cluster.local:27017" },

{ \_id: 1, host: "mongo-1.mongo.webapps.svc.cluster.local:27017" },

{ \_id: 2, host: "mongo-2.mongo.webapps.svc.cluster.local:27017" }

]

})

kubectl exec -it -n webapps mongo-1 -- mongosh --eval "rs.status()"