Kubernetes StatefulSet – Step by Step Deployment & YAML Guide

This document explains StatefulSet in Kubernetes using simple words, with complete YAML files and a step-by-step deployment process suitable for beginners.

1. What is a StatefulSet?

A StatefulSet is used for applications that need stable pod names, stable network identity, and persistent storage. Each pod in a StatefulSet is unique and maintains its identity even after restarts.

Examples: MySQL, PostgreSQL, Redis, Kafka.

2. Key Characteristics

1. Pods have fixed names like web-0, web-1, web-2
2. Each pod gets its own persistent volume
3. Pods are created and deleted in order
4. Stable DNS names for pod-to-pod communication

3. Prerequisites

1. A running Kubernetes cluster (EKS, Minikube, etc.)
2. kubectl configured and working
3. A StorageClass available in the cluster

4. Step 1 – Create Namespace

```
kubectl create namespace demo-stateful
```

5. Step 2 – Headless Service YAML

```
apiVersion: v1
kind: Service
metadata:
  name: web-headless
  namespace: demo-stateful
spec:
  clusterIP: None
  selector:
    app: web
  ports:
    - port: 80
      name: http
```

This service provides stable DNS names for each pod.

6. Step 3 – StatefulSet YAML

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
  namespace: demo-stateful
spec:
  serviceName: web-headless
  replicas: 3
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
```

```
        - name: nginx
          image: nginx:1.27-alpine
          ports:
            - containerPort: 80
          volumeMounts:
            - name: data
              mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
    - metadata:
        name: data
      spec:
        accessModes:
          - ReadWriteOnce
        resources:
          requests:
            storage: 1Gi
        storageClassName: gp3
```

The pod name is created automatically using the StatefulSet name: web-0, web-1, web-2.

7. Step 4 – Apply the YAML

```
kubectl apply -f headless-service.yaml
kubectl apply -f statefulset.yaml
```

8. Verify Deployment

```
kubectl get pods -n demo-stateful
kubectl get pvc -n demo-stateful
kubectl get statefulset -n demo-stateful
```

9. Stable DNS Format

Each pod can be accessed using this DNS format:

```
<pod-name>.<headless-service-name>
```

Example: web-0.web-headless

10. Delete Pod and Observe Behavior

```
kubectl delete pod web-1 -n demo-stateful
```

The pod is recreated with the same name and the same persistent volume.

11. Scaling the StatefulSet

```
kubectl scale statefulset web --replicas=5 -n demo-stateful
kubectl scale statefulset web --replicas=2 -n demo-stateful
```

12. Cleanup

```
kubectl delete statefulset web -n demo-stateful
kubectl delete pvc --all -n demo-stateful
kubectl delete namespace demo-stateful
```

13. Summary

StatefulSet is essential for deploying databases and other stateful applications in Kubernetes. Use StatefulSets whenever data persistence and pod identity are required.