How to deploy StatefulSets in Kubernetes (K8s)?

StatefulSets will represent the set of pods with different (unique), persistent identities, and elastic hostnames (stable). It makes you assure about the ordering of scaling and deployments

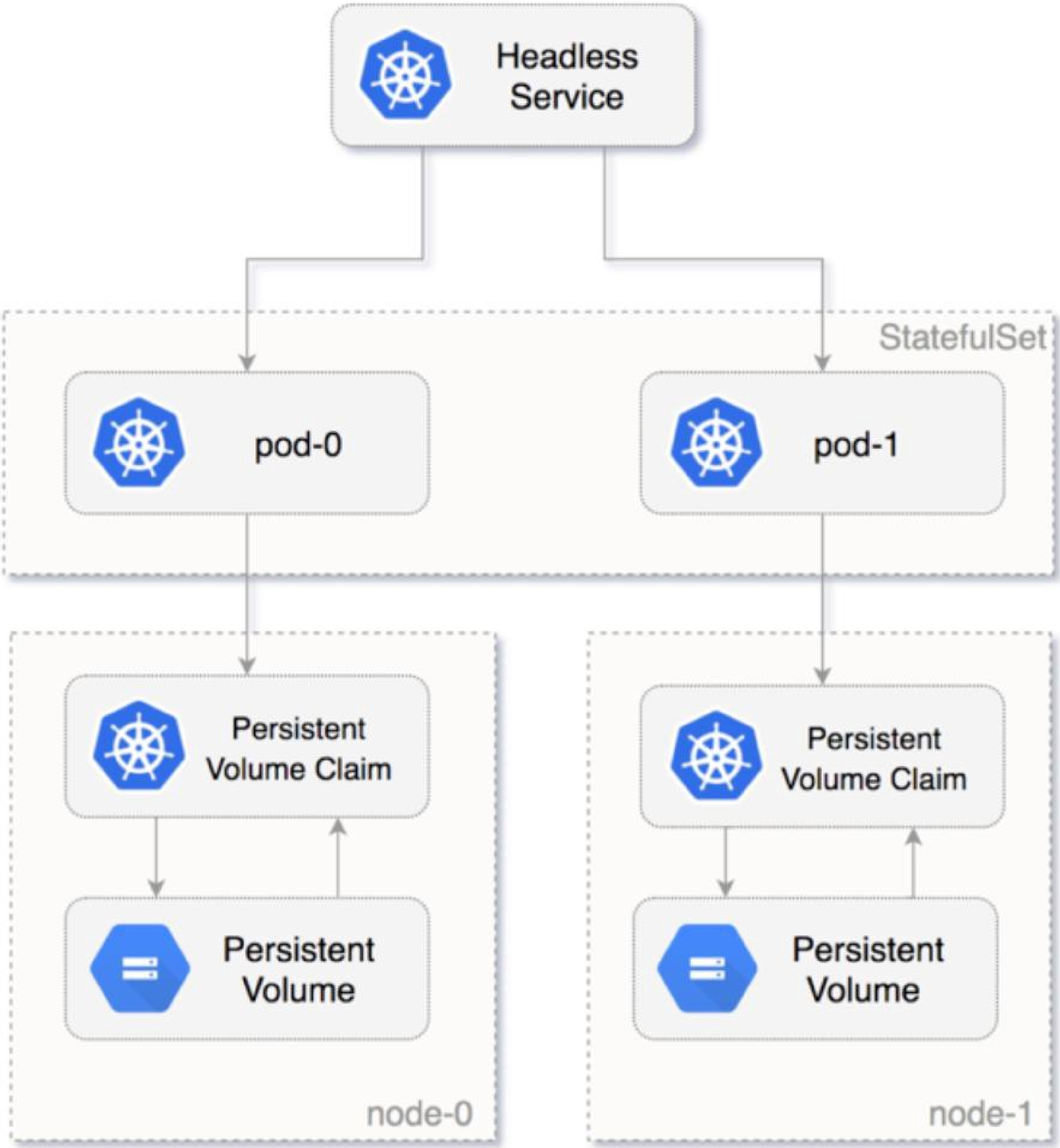
StatefulSets are valuable for applications that require one or more of the following:

1. Stable, unique network identifiers
2. Stable, persistent storage
3. Ordered, graceful deployment and scaling
4. Ordered, graceful deletion and termination

If an application doesn’t require any stable identifiers or ordered deployment, deletion, or scaling, you should deploy your application with a controller such as Deployments or ReplicaSets that provides a set of stateless replicas.

StatefulSet Components

1. A Headless Service
2. A StatefulSet
3. A PersistentVolume



Below are manifests of a Service, StatefulSet, and Persistent volume:

```
---
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
  - port: 80
    name: web
  clusterIP: None
  selector:
    app: nginx
```

2

```
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
  selector:
    matchLabels:
      app: nginx
  serviceName: "nginx"
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
          name: web
        volumeMounts:
        - name: www
          mountPath: /usr/share/nginx/html
      volumes:
      - name: www
        persistentVolumeClaim:
          claimName: myclaim
```

3

```
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: myclaim
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 8Gi
```

Create and manage StatefulSets

Create a stateful set

```
kubectl create -f statefulset.yaml
service "nginx" created
statefulset.apps "web" created
```

It will create three Pods named web-0,web-1,web-2

```
kubectl get podsNAME      READY     STATUS    RESTARTS   AGEweb-0     1
/1       Running   0          1mweb-1     1/1       Running
0          46sweb-2     1/1       Running   0          18s
```

Check services in Kubernetes

```
kubectl get svc nginxNAME      TYPE        CLUSTER-IP   EXTERNAL-IP
PORT(S)   AGEnginx     ClusterIP   None         < none >      80/TCP
2m
```

List your stateful sets:

```
kubectl get statefulsets
```

Get details of a stateful set:

```
kubectl describe statefulset web
```

Edit a stateful set:

```
kubectl edit statefulset web
```

Scaling a stateful set:

Scaling a StatefulSet refers to increasing or decreasing the number of replicas.

Scale-up a stateful set:

```
kubectl scale statefulset web --replicas=5
statefulset.apps "web" scaled
```

Check and get pods

```
kubectl get pods -l app=nginxNAME      READY     STATUS    RESTARTS
AGEweb-0     1/1       Running   0          11mweb-1     1/1
Running   0          10mweb-2     1/1       Running   0          10mweb-
3     1/1       Running   0          33sweb-4     1/1       Running
0          19s
```

Scale down a stateful set:

```
kubectl scale statefulset web --replicas=2
statefulset.apps "web" scaled
```

Check the scaling down of pods

```
kubectl get pods -w -l app=nginxNAME      READY     STATUS
RESTARTS   AGEweb-0     1/1       Running       0          13mweb-1
1/1       Running       0          12mweb-2     0/1       Terminating
0          12mweb-3     0/1       Terminating   0          2mweb-4     0
/1       Terminating   0          1m
```

Check again :

```
kubectl get pods -l app=nginxNAME      READY     STATUS    RESTARTS
AGEweb-0     1/1       Running   0          13mweb-1     1/1
Running   0          12m
```

Delete a stateful set

```
kubectl delete statefulset web
statefulset.apps "web" deleted
```

You must delete the Service manually.

```
kubectl delete service nginx
service "nginx" deleted
```

Limitations:

1. StatefulSet was a beta resource before 1.9 and not available in any Kubernetes release before 1.5.
2. The storage for a given Pod must either be provisioned by a PersistentVolume Provisioner based on the requested storage class or pre-provisioned by an admin.
3. Deleting and/or scaling a StatefulSet down will not delete the volumes associated with the StatefulSet. This is done to ensure data safety, which is generally more valuable than an automatic purge of all related StatefulSet resources.
4. StatefulSets currently require a Headless Service to be responsible for the network identity of the Pods. You are responsible for creating this Service.
5. StatefulSets do not provide any guarantees on the termination of pods when a StatefulSet is deleted. To achieve ordered and graceful termination of the pods in the StatefulSet, it is possible to scale the StatefulSet down to 0 before deletion.