Statefulset

A StatefulSet in Kubernetes is a workload API object used to manage stateful applications. It ensures:

- ❖ Unique, stable network identifiers for each Pod.
- ❖ Ordered, graceful deployment and scaling.
- ❖ Persistent storage using PersistentVolumeClaims.

Key features of stateful applications in Kubernetes include:

1. **Persistent Storage**: Data is preserved across pod restarts using PersistentVolumeClaims.
2. **Stable Network IDs**: Each pod has a unique, stable network identity.
3. **Ordered Deployment and Scaling**: Pods are deployed, scaled, and terminated in a defined order.

1. Create the yamlfile (vi stateful.yaml)

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
  serviceName: "nginx"
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: k8s.gcr.io/nginx-slim:0.8
        ports:
        - containerPort: 80
          name: web
        volumeMounts:
        - name: www
          mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
  - metadata:
      name: www
    spec:
      accessModes: [ "ReadWriteOnce" ]
      resources:
        requests:
          storage: 1Gi
      storageClassName: local-path
```

2. To check the pods:

Kubectl create –f stateful.yaml

```
controlplane $ kubectl create -f stateful.yaml
statefulset.apps/web created
```

3. To check the pods:

Kubectl get pods

```
controlplane $ kubectl get pods
NAME    READY   STATUS              RESTARTS   AGE
web-0   1/1     Running             0          22s
web-1   0/1     ContainerCreating   0          10s
```

4. To delete the Pods:

Kubectl delete po web-1

```
controlplane $ kubectl delete po web-1
pod "web-1" deleted
```

5. To verify the running pods:

Kubectl get pod

```
controlplane $ kubectl get pod
NAME    READY   STATUS    RESTARTS   AGE
web-0   1/1     Running   0          77s
web-1   1/1     Running   0          15s
web-2   1/1     Running   0          49s
```

6. To check the history status:

Kubectl rollout history statefulset web statefulset.apps/web

```
controlplane $ kubectl rollout history statefulset web
statefulset.apps/web
REVISION  CHANGE-CAUSE
1         <none>
```

7. To add and rollout methods:

Kubectl rollout history statefulset web –revision 1

```
controlplane $ kubectl rollout history statefulset web --revision 1
statefulset.apps/web with revision #1
Pod Template:
  Labels:       app=nginx
  Containers:
   nginx:
    Image:      k8s.gcr.io/nginx-slim:0.8
    Port:       80/TCP
    Host Port:  0/TCP
    Environment:        <none>
    Mounts:
      /usr/share/nginx/html from www (rw)
  Volumes:      <none>
  Node-Selectors:       <none>
  Tolerations:  <none>
```

8. To able the scaling Methods:

Kubectl scale statefulset web –replicas=5

```
controlplane $ kubectl scale statefulset web --replicas=5
statefulset.apps/web scaled
```

9. Check the scaling of pods to verify:

Kubectl get pod

```
controlplane $ kubectl get pod
NAME    READY   STATUS    RESTARTS   AGE
web-0   1/1     Running   0          15m
web-1   1/1     Running   0          14m
web-2   1/1     Running   0          14m
web-3   1/1     Running   0          35s
web-4   1/1     Running   0          28s
```

10. Over all the describe status:

Kubectl describe statefulsets.apps

```
controlplane $ kubectl describe statefulsets.apps
Name:               web
Namespace:          default
CreationTimestamp:  Mon, 29 Jul 2024 08:52:46 +0000
Selector:           app=nginx
Labels:             <none>
Annotations:        <none>
Replicas:           5 desired | 5 total
Update Strategy:    RollingUpdate
  Partition:        0
Pods Status:        5 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        k8s.gcr.io/nginx-slim:0.8
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
      /usr/share/nginx/html from www (rw)
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Volume Claims:
  Name:          www
  StorageClass:  local-path
  Labels:        <none>
  Annotations:   <none>
  Capacity:      1Gi
  Access Modes:  [ReadWriteOnce]
```

```
Events:
  Type    Reason                 Age                  From                  Message
  ----    ------                 ----                 ----                  -------
  Normal  SuccessfulCreate       23m                  statefulset-controller  create Claim www-web-0 Pod web-0 in StatefulSet web
success
  Normal  SuccessfulCreate       23m                  statefulset-controller  create Pod web-0 in StatefulSet web successful
  Normal  SuccessfulCreate       22m                  statefulset-controller  create Claim www-web-1 Pod web-1 in StatefulSet web
success
  Normal  SuccessfulCreate       22m                  statefulset-controller  create Claim www-web-2 Pod web-2 in StatefulSet web
success
  Normal  SuccessfulCreate       22m                  statefulset-controller  create Pod web-2 in StatefulSet web successful
  Normal  SuccessfulCreate       22m (x2 over 22m)    statefulset-controller  create Pod web-1 in StatefulSet web successful
  Normal  RecreatingTerminatedPod  22m (x7 over 22m)  statefulset-controller  StatefulSet default/web is recreating terminated Pod
 web-1
  Normal  SuccessfulDelete       22m (x7 over 22m)    statefulset-controller  delete Pod web-1 in StatefulSet web successful
  Normal  SuccessfulCreate       8m24s                statefulset-controller  create Claim www-web-3 Pod web-3 in StatefulSet web
success
  Normal  SuccessfulCreate       8m24s                statefulset-controller  create Pod web-3 in StatefulSet web successful
  Normal  SuccessfulCreate       8m17s                statefulset-controller  create Claim www-web-4 Pod web-4 in StatefulSet web
success
  Normal  SuccessfulCreate       8m17s                statefulset-controller  create Pod web-4 in StatefulSet web successful
```

11. To images only view the command:

Kubectl describe statefulsets.apps |grep Image

```
controlplane $ kubectl describe statefulsets.apps |grep Image
    Image:        k8s.gcr.io/nginx-slim:0.8
```

Need to Methods change the version:

1. vi stateful2.yaml

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web-state
spec:
  serviceName: "nginx"
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: registry.k8s.io/nginx-slim:0.21
        ports:
        - containerPort: 80
          name: web
        volumeMounts:
        - name: www
          mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
  - metadata:
      name: www
    spec:
      accessModes: [ "ReadWriteOnce" ]
      resources:
        requests:
          storage: 1Gi
      storageClassName: local-path
~
```

2. To check the pods:

Kubectl create –f stateful2.yaml

```
controlplane $ kubectl create -f stateful2.yaml
statefulset.apps/web-state created
```

3. To check the pods:

Kubectl get pods

```
controlplane $ kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
web-0         1/1     Running   0          39m
web-1         1/1     Running   0          38m
web-2         1/1     Running   0          38m
web-3         1/1     Running   0          24m
web-4         1/1     Running   0          24m
web-state-0   1/1     Running   0          3m47s
web-state-1   1/1     Running   0          3m37s
```

4. To delete the Pods:

Kubectl delete po web-1

```
controlplane $ kubectl delete po web-state-1
pod "web-state-1" deleted
```

5. To verify the running pods:

Kubectl get pod

```
controlplane $ kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
web-0         1/1     Running   0          41m
web-1         1/1     Running   0          40m
web-2         1/1     Running   0          41m
web-3         1/1     Running   0          27m
web-4         1/1     Running   0          26m
web-state-0   1/1     Running   0          6m15s
web-state-1   1/1     Running   0          52s
```

6. To check the history status:

Kubectl rollout history statefulset web-state statefulset.apps/web-state

```
controlplane $ kubectl rollout history statefulset web-state
statefulset.apps/web-state
REVISION  CHANGE-CAUSE
1         <none>
```

7. To able the scaling Methods:

Kubectl scale statefulset web –replicas=5

```
controlplane $ kubectl scale statefulset web-state --replicas 5
statefulset.apps/web-state scaled
```

8. Check the scaling of pods to verify:

Kubectl get pod

```
controlplane $ kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
web-0         1/1     Running   0          44m
web-1         1/1     Running   0          43m
web-2         1/1     Running   0          44m
web-3         1/1     Running   0          29m
web-4         1/1     Running   0          29m
web-state-0   1/1     Running   0          9m4s
web-state-1   1/1     Running   0          3m41s
web-state-2   1/1     Running   0          22s
web-state-3   1/1     Running   0          15s
web-state-4   1/1     Running   0          9s
```

10. Over all the describe status:

Kubectl describe statefulsets.apps

```
Name:               web-state
Namespace:          default
CreationTimestamp:  Mon, 29 Jul 2024 09:28:22 +0000
Selector:           app=nginx
Labels:             <none>
Annotations:        <none>
Replicas:           5 desired | 5 total
Update Strategy:    RollingUpdate
  Partition:        0
Pods Status:        5 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        registry.k8s.io/nginx-slim:0.21
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
      /usr/share/nginx/html from www (rw)
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Volume Claims:
  Name:          www
  StorageClass:  local-path
  Labels:        <none>
  Annotations:   <none>
  Capacity:      1Gi
  Access Modes:  [ReadWriteOnce]
```

```
Events:
  Type     Reason                 Age                  From                  Message
  ----     ------                 ----                 ----                  -------
  Normal   SuccessfulCreate       12m                  statefulset-controller  create Claim www-web-state-0 Pod web-state-0 in StatefulSet web-state success
  Normal   SuccessfulCreate       12m                  statefulset-controller  create Pod web-state-0 in StatefulSet web-state successful
  Normal   SuccessfulCreate       11m                  statefulset-controller  create Claim www-web-state-1 Pod web-state-1 in StatefulSet web-state success
  Normal   SuccessfulCreate       6m38s (x2 over 11m)  statefulset-controller  create Pod web-state-1 in StatefulSet web-state successful
  Normal   RecreatingTerminatedPod  6m38s (x9 over 6m38s)  statefulset-controller  StatefulSet default/web-state is recreating terminated Pod web-state-1
  Normal   SuccessfulDelete       6m38s (x8 over 6m38s)  statefulset-controller  delete Pod web-state-1 in StatefulSet web-state successful
  Warning  FailedDelete           6m38s                statefulset-controller  delete Pod web-state-1 in StatefulSet web-state failed error: pods "web-state-1"
not found
  Normal   SuccessfulCreate       3m19s                statefulset-controller  create Claim www-web-state-2 Pod web-state-2 in StatefulSet web-state success
  Normal   SuccessfulCreate       3m19s                statefulset-controller  create Pod web-state-2 in StatefulSet web-state successful
  Normal   SuccessfulCreate       3m12s                statefulset-controller  create Claim www-web-state-3 Pod web-state-3 in StatefulSet web-state success
```

11. To add and rollout methods:

Kubectl rollout history statefulset web –revision 1

```
controlplane $ kubectl rollout history statefulset web-state --revision 1
statefulset.apps/web-state with revision #1
Pod Template:
  Labels:       app=nginx
  Containers:
   nginx:
    Image:      registry.k8s.io/nginx-slim:0.21
    Port:       80/TCP
    Host Port:  0/TCP
    Environment:        <none>
    Mounts:
      /usr/share/nginx/html from www (rw)
  Volumes:      <none>
  Node-Selectors:       <none>
  Tolerations:  <none>
```

12. To verify the undo status:

Kubectl rollout undo statefulset web-state –to-revision=1

```
controlplane $ kubectl rollout undo statefulset web-state --to-revision=1
statefulset.apps/web-state skipped rollback (current template already matches revision 1)
```

13. To verify the undo status:

Kubectl rollout undo statefulset web-state –to-revision=1

```
controlplane $ kubectl rollout undo statefulset web --to-revision=1
statefulset.apps/web skipped rollback (current template already matches revision 1)
```