

Stateful Applications

» StatefulSets

Overview

This exercise requires at least two worker nodes.

Before we begin, confirm that you have a storage class called 'general' for aws-ebs.

```
kubectl get storageclasses
```

Create a StatefulSet manifest file for an nginx web service with a shared volume.

```

cat >> web-stateful-set.yaml <<EOF
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
    - port: 80
      name: web
  clusterIP: None
  selector:
    app: nginx
---
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: web
spec:
  serviceName: "nginx"
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      terminationGracePeriodSeconds: 10
      containers:
        - name: nginx
          image: gcr.io/google_containers/nginx-slim:0.8
          ports:
            - containerPort: 80
              name: web
          volumeMounts:
            - name: www
              mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
    - metadata:
        name: www
        annotations:
          volume.beta.kubernetes.io/storage-class: general
      spec:
        accessModes: [ "ReadWriteOnce" ]

```

```
accessModes: [ReadWriteOnce]
resources:
  requests:
    storage: 1Gi
EOF
```

Create the Stateful Set.

```
kubectl create -f web-stateful-set.yaml
```

Verify the headless service was created. Observe the headless service does not have a Cluster-IP and is only accessible via DNS.

```
kubectl get service nginx
```

Verify the StatefulSet was created.

```
kubectl get statefulset web
```

View all pods in the StatefulSet and take note of the ordinal pod names.

```
kubectl get pods -l app=nginx
```

Execute nslookups inside a temporary pod to verify dns resolution to

```
kubectl run -it --image=busybox dns-test --restart=Never --rm /bin/sh
nslookup web-0.nginx
nslookup web-1.nginx
nslookup nginx
exit
```

To show volume persistence despite pod deletion, let's write some files to the website on the volume and verify we can see the result.

```
kubectl exec web-0 -- sh -c 'echo hi there > /usr/share/nginx/html/index.html'
kubectl exec web-1 -- sh -c 'curl nginx'
```

Delete the StatefulSet pods.

```
kubectl delete pod -l app=nginx
```

Examine the output of the `kubectl get` command. You will see the pods terminate in sequential order. They will then get re-created in sequential order.

```
kubectl get pod -w -l app=nginx
```

Scale the StatefulSet up to 6 replicas.

```
kubectl scale statefulset web --replicas=6
```

Examine the output of the `kubectl get` command.

```
kubectl get pods -w -l app=nginx
```

Observe how the number of persistent volume claims increases.

```
kubectl get pvc
```

Scale it back down to 3 replicas.

```
kubectl scale statefulset web --replicas=3
```

Examine the output of the `kubectl get` command.

```
kubectl get pod -w -l app=nginx
```

Verify the website remains the same.

```
kubectl exec web-1 -- sh -c 'curl nginx'
```

Clean Up

Delete the headless StatefulSet service.

```
kubectl delete service nginx
```

Delete the StatefulSet.

```
kubectl delete statefulset web
```

Delete the volumes

```
kubectl delete pvc -l app=nginx
```