1.1 StatefulSet

1.1.1 创建 StatefulSet

步骤 1　创建一个文件夹用于存放 StatefulSet 相关文件。

```
[root@k8s-master labfile]# mkdir /labfile/statefulset
[root@k8s-master labfile]# cd /labfile/statefulset/
```

步骤 2　在/nfs 文件夹下创建 pv1，pv2，pv3 三个文件夹，用于给 pv 使用。

```
[root@k8s-master stateful]# cd /nfs
[root@k8s-master nfs]# mkdir pv1 pv2 pv3
```

步骤 3　创建三个 PV，使用 PV 文件样例如下。PV2 和 PV3 的配置文件中需要修改 metadata.name 项和 spec.nfs.path 项。

```
[root@k8s-master stateful]# vim mypv1.yaml
```

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mypv1
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: my-sc
  nfs:
    path: /nfs/pv1
    server: 127.0.0.1
```

步骤 4　创建三个 PV

```
[root@k8s-master stateful]# kubectl apply -f mypv1.yaml
persistentvolume/mypv1 created
[root@k8s-master stateful]# kubectl apply -f mypv2.yaml
persistentvolume/mypv2 created
[root@k8s-master stateful]# kubectl apply -f mypv3.yaml
persistentvolume/mypv3 created
```

步骤 5　创建 StatefulSet 的 yaml 文件

```
[root@k8s-master stateful]# vim stateful.yaml
```

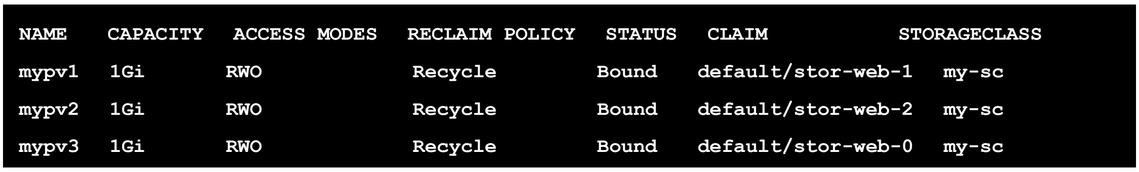
```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
  selector:
    matchLabels:
      app: nginx
  serviceName: nginx
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      terminationGracePeriodSeconds: 10
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
          name: web
        volumeMounts:
        - name: stor
          mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
  - metadata:
      name: stor
    spec:
      accessModes:
        - ReadWriteOnce
      storageClassName: my-sc
      resources:
        requests:
          storage: 1Gi
```

步骤 6 创建 statefulSet

```
[root@k8s-master stateful]# kubectl apply -f stateful.yaml
```

步骤 7 查看 pv 和 pvc

```
[root@k8s-master stateful]# kubectl get pv
```



```
NAME    CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM              STORAGECLASS
mypv1   1Gi        RWO            Recycle          Bound    default/stor-web-1   my-sc
mypv2   1Gi        RWO            Recycle          Bound    default/stor-web-2   my-sc
mypv3   1Gi        RWO            Recycle          Bound    default/stor-web-0   my-sc
```

```
[root@k8s-master stateful]# kubectl get pvc
```

```
NAME          STATUS    VOLUME    CAPACITY    ACCESS MODES    STORAGECLASS    AGE
stor-web-0    Bound     mypv3     1Gi         RWO             my-sc           100s
stor-web-1    Bound     mypv1     1Gi         RWO             my-sc           98s
stor-web-2    Bound     mypv2     1Gi         RWO             my-sc           95s
```

步骤 8 查看 statefulset 状态和事件

```
[root@k8s-master stateful]# kubectl get statefulset
```

```
NAME   READY   AGE
web    3/3     2m45s
```

详细信息

```
[root@k8s-master stateful]# kubectl describe statefulset
```

回显中可以看到事件信息：

```
Events:
  Type    Reason            Age    From                      Message
  ----    ------            ----   ----                      -------
  Normal  SuccessfulCreate  3m20s  statefulset-controller  create Claim stor-web-0 Pod web-0 in StatefulSet web success
  Normal  SuccessfulCreate  3m20s  statefulset-controller  create Pod web-0 in StatefulSet web successful
  Normal  SuccessfulCreate  3m18s  statefulset-controller  create Claim stor-web-1 Pod web-1 in StatefulSet web success
  Normal  SuccessfulCreate  3m18s  statefulset-controller  create Pod web-1 in StatefulSet web successful
  Normal  SuccessfulCreate  3m15s  statefulset-controller  create Claim stor-web-2 Pod web-2 in StatefulSet web success
  Normal  SuccessfulCreate  3m15s  statefulset-controller  create Pod web-2 in StatefulSet web successful
```

步骤 9 创建 headless 服务

```
[root@k8s-master stateful]# vim headless.yaml
```

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

```
[root@k8s-master stateful]# kubectl apply -f headless.yaml
```

步骤 10 创建一个 clientpod，用于测试 headless 服务（和之前在 service 章节创建的测试 pod 一致）

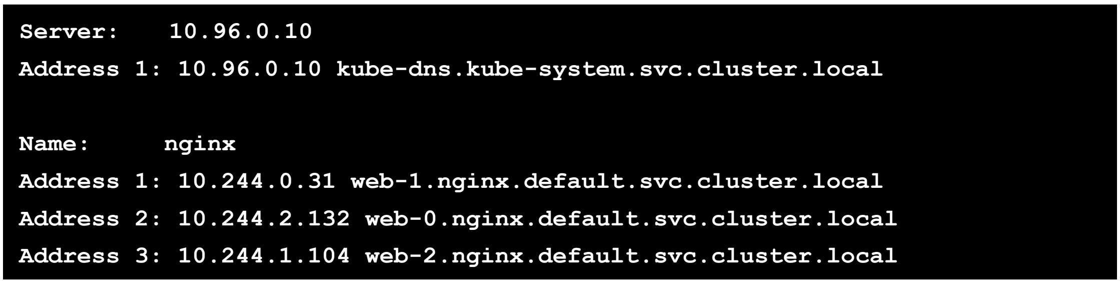
```
[root@k8s-master servicefile]# vim client.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: clientpod
spec:
  containers:
    - name: clientpod
      image: busybox:1.28.3
      args:
      - /bin/sh
      - -c
      - sleep 30000
```

```
[root@k8s-master servicefile]# kubectl apply -f client.yaml
```

步骤 11 进入 clientpod，查看 statefulset 服务的 nslookup 信息。可以看到，每个 pod 被用 web-n.nginx 命名。

```
[root@k8s-master servicefile]# kubectl exec -it clientpod /bin/sh
/ # nslookup nginx
```



```
Server:    10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name:      nginx
Address 1: 10.244.0.31 web-1.nginx.default.svc.cluster.local
Address 2: 10.244.2.132 web-0.nginx.default.svc.cluster.local
Address 3: 10.244.1.104 web-2.nginx.default.svc.cluster.local
```

1.1.2 StatefulSet 常见操作

步骤 1 进入 statefulSet 中 pod 的持久存储目录（/usr/share/nginx/html），创建一个文件。

```
[root@k8s-master servicefile]# kubectl exec -it web-0 /bin/sh
# cd /usr/share/nginx/html
# cat << EOF > hello.file
> this is hello web-0
> EOF
```

步骤 2 删除 web-0 pod，可以看到 statefulSet 自动创建了一个新的 web-0 pod。

```
[root@k8s-master servicefile]# kubectl delete pod web-0
pod "web-0" deleted
```

```
[root@k8s-master servicefile]# kubectl get pod
NAME        READY   STATUS    RESTARTS   AGE
clientpod    1/1     Running   0          19m
web-0       1/1     Running   0          23s
web-1       1/1     Running   0          46m
web-2       1/1     Running   0          46m
```

步骤 3 进入 clientpod，验证能否用原来的域名访问 web-0

```
[root@k8s-master servicefile]# kubectl exec -it clientpod /bin/sh
/ # nslookup web-0.nginx
Server:    10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name:      web-0.nginx
Address 1: 10.244.2.134 web-0.nginx.default.svc.cluster.local
```

步骤 4 进入 web-0 pod，查看保持在持久化存储上的文件是否存在。

```
[root@k8s-master servicefile]# kubectl exec -it web-0 /bin/sh
# cat /usr/share/nginx/html/hello.file
this is hello web-0
```

步骤 5 升级 statefulSet，将原有 statefulSet.yaml 中 nginx:1.7.9 的镜像版本变更为 nginx:1.9.1

```
[root@k8s-master stateful]# cp stateful.yaml stateful-new.yaml
[root@k8s-master stateful]# vim stateful-new.yaml
```

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
  selector:
    matchLabels:
      app: nginx
  serviceName: nginx
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      terminationGracePeriodSeconds: 10
      containers:
      - name: nginx
        image: nginx:1.9.1
        ports:
        - containerPort: 80
          name: web
        volumeMounts:
        - name: stor
          mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
  - metadata:
      name: stor
    spec:
      accessModes:
        - ReadWriteOnce
      storageClassName: my-sc
      resources:
        requests:
          storage: 1Gi
```

步骤 6 应用更新并查看系统运行状态，可以看到系统从 web-2 Pod 开始更新的全过程。

```
[root@k8s-master stateful]# kubectl apply -f stateful-new.yaml && kubectl get
pod -w
```