

K8s StatefulSets 中的 NFS 应用

AUTHOR: 彭玲 TIME: 2022/2/24

K8s StatefulSets 中的 NFS 应用

StatefulSet 中 NFS 应用

前置条件

NFS 外部驱动

StorageClass 对象

StatefulSet 部署

yaml 配置清单

应用

结果

Pod 正常

PVC 正常

PV 正常

PVC 名称

PVC 信息 (自动绑定 PV)

PV 信息 (`spec.nfs.path`)

NFS 共享目录

清理工作

1. 删除 StatefulSet

2. 删除 PVC

3. 删除 PV

清理 NFS 共享目录

StatefulSet 中 NFS 应用

前置条件

部署 StatefulSet 之前，先部署外部的 NFS provisioner 以动态创建 PV。

NFS 外部驱动

NFS 外部驱动 关键配置 StorageClass 及 nfs 对应的 daemon 如下：

```
1 anxin@node38:~/pengling/k8s/kubesphere$ vi storageclass.yaml
2
3 ---
4 kind: Deployment
5 apiVersion: apps/v1
6 metadata:
7   name: nfs-client-provisioner
8 spec:
9   selector:
10    matchLabels:
11     app: nfs-client-provisioner
12   replicas: 1
```

```

13 strategy:
14   type: Recreate
15 template:
16   metadata:
17     labels:
18       app: nfs-client-provisioner
19   spec:
20     serviceAccountName: nfs-client-provisioner
21     containers:
22       - name: nfs-client-provisioner
23         image: quay.io/external_storage/nfs-client-provisioner:latest
24         imagePullPolicy: IfNotPresent
25         volumeMounts:
26           - name: nfs-client
27             mountPath: /persistentvolumes
28         env:
29           - name: PROVISIONER_NAME
30             value: fuseim.pri/ifs
31           - name: NFS_SERVER
32             value: 10.8.30.38 # 此处修改为 nfs 服务器 ip
33           - name: NFS_PATH
34             value: /data/backups # 这里为 nfs 共享目录
35     volumes:
36       - name: nfs-client
37         nfs:
38           server: 10.8.30.38 # 此处修改为 nfs 服务器 ip
39           path: /data/backups # 这里为 nfs 共享目录
40 ---
41 apiVersion: storage.k8s.io/v1
42 kind: StorageClass
43 metadata:
44   name: nfs-storage
45   annotations:
46     storageclass.kubernetes.io/is-default-class: "true"
47   provisioner: fuseim.pri/ifs # 必须和 Deployment 的 PROVISIONER_NAME 匹配
48   reclaimPolicy: Retain

```

StorageClass 对象

查看集群中 StorageClass 对象，在 StatefulSet 示例中使用 `nfs-storage` 默认存储类（不能使用 `localhostpath`）。

```

1 anxin@node38:~$ kubectl get sc -A
2 NAME                                PROVISIONER                                RECLAIMPOLICY    VOLUMEBINDINGMODE
3 localhostpath                       kubernetes.io/no-provisioner              Retain           Immediate
4 nfs-storage (default)                fuseim.pri/ifs                             Retain           Immediate

```

使用 `kubectl edit` 修改默认存储类注解 `storageclass.kubernetes.io/is-default-class`：

```

1  anxin@node38:~$ kubectl edit sc localhostpath
2
3  apiVersion: storage.k8s.io/v1
4  kind: StorageClass
5  metadata:
6    annotations:
7      kubectl.kubernetes.io/last-applied-configuration: |
8        {"apiVersion":"storage.k8s.io/v1","kind":"StorageClass","metadata":
9        {"annotations":{"storageclass.kubernetes.io/is-default-
10       class":"true"},"name":"localhostpath"},"provisioner":"kubernetes.io/no-
11       provisioner","reclaimPolicy":"Retain","volumeBindingMode":"Immediate"}
12       storageclass.kubernetes.io/is-default-class: "false" # true: 默认存储类
13  ...
14  storageclass.storage.k8s.io/localhostpath edited

```

StatefulSet 部署

yaml 配置清单

下面例子中：

- 名为 `nginx` 的 Headless Service 用来控制网络域名。
- 名为 `web` 的 StatefulSet 有一个 Spec，它表明将在独立的 Pod 中启动 3 个 `nginx` 容器副本 (replicas)。
- `volumeClaimTemplates` 将使用 `PersistentVolumes` 驱动提供的 `PersistentVolumes` 来提供稳定的存储。

```

1  # 该 yaml 中 service 可以去掉（不应该这样做），仅保留 StatefulSet 配置清单
2  anxin@node38:~/pengling/k8s/controller-demo$ vi statefulset-demo.yaml
3
4  apiVersion: v1
5  kind: Service
6  metadata:
7    name: nginx
8    namespace: julin
9    labels:
10     app: nginx
11  spec:
12    ports:
13     - port: 80
14      name: web
15    clusterIP: None # Headless Service
16    selector:
17     app: nginx
18  ---
19  apiVersion: apps/v1
20  kind: StatefulSet
21  metadata:
22    name: web
23    namespace: julin
24  spec:
25    selector:
26     matchLabels:
27     app: nginx
28    serviceName: "nginx" # serviceName 是管理这个 StatefulSet 的服务名称

```

```

29 replicas: 3 # by default is 1
30 template:
31   metadata:
32     labels:
33       app: nginx # has to match .spec.selector.matchLabels
34   spec:
35     terminationGracePeriodSeconds: 10
36     containers:
37     - name: nginx
38       image: nginx
39       ports:
40       - containerPort: 80
41         name: web
42       volumeMounts:
43       - name: www
44         mountPath: /usr/share/nginx/html
45   volumeClaimTemplates:
46   - metadata:
47     name: www
48     namespace: julin # PVC 命名空间 (可选)
49     spec:
50     accessModes: [ "ReadWriteOnce" ]
51     storageClassName: "nfs-storage" # 由 PersistentVolume Provisioner 提供
52     resources:
53     requests:
54     storage: 1Gi

```

应用

```

1 anxin@node38:~/pengling/k8s/controller-demo$ kubectl apply -f statefulset-
  demo.yaml
2 service/nginx created
3 statefulset.apps/web created

```

结果

- 查看 Service, StatefulSet 和其管理的 Pod

```

1 anxin@node38:~$ kubectl get all -n julin
2 NAME                READY    STATUS    RESTARTS   AGE
3 pod/web-0           1/1     Running  0          39m
4 pod/web-1           1/1     Running  0          38m
5 pod/web-2           1/1     Running  0          38m
6
7 NAME                TYPE          CLUSTER-IP    EXTERNAL-IP   PORT(S)    AGE
8 service/nginx       ClusterIP     None          <none>         80/TCP     39m
9
10 NAME                READY    AGE
11 statefulset.apps/web 3/3     39m

```

- 查看 PVC 和 PV

```

1 anxin@node38:~$ kubectl get pvc -l app=nginx -n julin
2 NAME          STATUS    VOLUME          CAPACITY   ACCESS MODES
   STORAGECLASS AGE
3 www-web-0     Bound    pvc-2c3bbea9-0d0c-41c7-9f4c-c1e8e7700652  1Gi        RWO
   nfs-storage  47m
4 www-web-1     Bound    pvc-8f7bedec-0d67-4b61-a8aa-028d351b60db  1Gi        RWO
   nfs-storage  46m
5 www-web-2     Bound    pvc-57ddf3f4-4729-42db-8037-e980e511fc34  1Gi        RWO
   nfs-storage  46m
6
7 anxin@node38:~$ kubectl get pv
8 NAME          CAPACITY   ACCESS MODES   RECLAIM POLICY
9 STATUS CLAIM          STORAGECLASS  REASON  AGE
10 pvc-2c3bbea9-0d0c-41c7-9f4c-c1e8e7700652  1Gi        RWO          Retain
   Bound julin/www-web-0  nfs-storage  47m
11 pvc-57ddf3f4-4729-42db-8037-e980e511fc34  1Gi        RWO          Retain
   Bound julin/www-web-2  nfs-storage  46m
   pvc-8f7bedec-0d67-4b61-a8aa-028d351b60db  1Gi        RWO          Retain
   Bound julin/www-web-1  nfs-storage  46m

```

Pod 正常

```

1 anxin@node38:~$ kubectl describe po web-0 -n julin
2 Name:          web-0
3 Namespace:     julin
4 Priority:      0
5 Node:         node37/10.8.30.37
6 Start Time:   Tue, 08 Feb 2022 09:15:16 +0800
7 Labels:       app=nginx
8               controller-revision-hash=web-67bb74dc
9               statefulset.kubernetes.io/pod-name=web-0
10 Annotations:  <none>
11 Status:       Running # 1. Pod 阶段
12 IP:          10.244.1.37
13 IPs:
14   IP:        10.244.1.37
15 Controlled By: StatefulSet/web # Pod 的控制器
16 Containers:
17   nginx:
18     Container ID:
19     docker://1e9fd82612c13b9b79594a9fc5527ebb5a62bdc43f7df95a97bfb29bd8f0fff73
20     Image:       nginx
21     Image ID:    docker-
22     pullable://nginx@sha256:2834dc507516af02784808c5f48b7cbe38b8ed5d0f4837f16e78
23     d00deb7e7767
24     Port:       80/TCP
25     Host Port:  0/TCP
26 State:        Running # 2. 容器状态
27   Started:     Tue, 08 Feb 2022 09:16:24 +0800
28   Ready:       True # 容器已就绪
29   Restart Count: 0
30 Environment:  <none>
31 Mounts:
32   /usr/share/nginx/html from www (rw) # 卷挂载
33   /var/run/secrets/kubernetes.io/serviceaccount from default-token-qf8cn
34   (ro)
35 Conditions: # 3. Pod 状况

```

```

32 | Type                Status
33 | Initialized          True
34 | Ready               True
35 | ContainersReady     True
36 | PodScheduled        True # Pod 已经被调度到某节点
37 | Volumes:
38 |   www:
39 |     Type:           PersistentVolumeClaim (a reference to a PersistentVolumeClaim
in the same namespace)
40 |     ClaimName:     www-web-0 # PVC name
41 |     ReadOnly:      false
42 |     default-token-qf8cn:
43 |       Type:        Secret (a volume populated by a Secret)
44 |       SecretName: default-token-qf8cn
45 |       optional:    false
46 | QoS Class:         BestEffort
47 | Node-Selectors:    <none>
48 | Tolerations:       node.kubernetes.io/not-ready:NoExecute for 300s
49 |                   node.kubernetes.io/unreachable:NoExecute for 300s
50 | Events:            <none>

```

PVC 正常

```

1 | anxin@node38:~$ kubectl describe pvc www-web-0 -n julin
2 | Name:          www-web-0
3 | Namespace:    julin
4 | StorageClass: nfs-storage
5 | Status:       Bound # 已绑定
6 | Volume:       pvc-2c3bbea9-0d0c-41c7-9f4c-c1e8e7700652 # PV name
7 | Labels:       app=nginx
8 | Annotations:  pv.kubernetes.io/bind-completed: yes
9 |               pv.kubernetes.io/bound-by-controller: yes
10 |              volume.beta.kubernetes.io/storage-provisioner: fuseim.pri/ifs
11 | Finalizers:   [kubernetes.io/pvc-protection]
12 | Capacity:    1Gi # 存储容量
13 | Access Modes: RWX # ReadWriteOnce
14 | VolumeMode:  Filesystem # 卷模式 (默认)
15 | Mounted By:  web-0 # Pod name
16 | Events:      <none>

```

PV 正常

```

1 | anxin@node38:~$ kubectl describe pv pvc-2c3bbea9-0d0c-41c7-9f4c-c1e8e7700652
2 | Name:          pvc-2c3bbea9-0d0c-41c7-9f4c-c1e8e7700652
3 | Labels:        <none>
4 | Annotations:   pv.kubernetes.io/provisioned-by: fuseim.pri/ifs #
provisioner
5 | Finalizers:    [kubernetes.io/pv-protection]
6 | StorageClass:  nfs-storage # StorageClass name
7 | Status:        Bound # 已绑定
8 | Claim:         julin/www-web-0 # PVC name
9 | Reclaim Policy: Retain # 回收策略: 保留数据, 需要手工处理
10 | Access Modes:  RWX # ReadWriteOnce
11 | VolumeMode:    Filesystem
12 | Capacity:     1Gi # 存储容量
13 | Node Affinity: <none>
14 | Message:

```

```
15 Source: # NFS 服务器
16   Type:      NFS (an NFS mount that lasts the lifetime of a pod)
17   Server:    10.8.30.38 # nfs 服务器 ip
18   Path:      /data/backups/julin-www-web-0-pvc-2c3bba9-0d0c-41c7-9f4c-
19             c1e8e7700652 # nfs 共享目录
20   ReadOnly:  false
Events:      <none>
```

PVC 名称

顺利运行 StatefulSet 后查看 pod 定义，发现自动生成如下内容：

```
1 anxin@node38:~$ kubectl edit po web-0 -n julin # Pod name: web-0
2
3 volumes:
4 - name: www
5   persistentVolumeClaim:
6     claimName: www-web-0 # 格式: <PVC 模板名称> + <Pod_Name>
```

其中，claimName 的格式为 volumeClaimTemplates 里面的 name + podname。

```
1 anxin@node38:~/pengling/k8s/controller-demo$ vi statefulset-demo.yaml
2
3 ---
4 apiVersion: apps/v1
5 kind: StatefulSet
6 metadata:
7   name: web
8   namespace: julin
9 spec:
10  ...
11  volumeClaimTemplates:
12  - metadata:
13    name: www # PVC 模板名称
```

PVC 信息 (自动绑定 PV)

查看 PVC 信息，自动绑定了一个名称随机的 pv (见 spec.volumeName 字段)。

```
1 anxin@node38:~$ kubectl edit pvc www-web-0 -n julin
2
3 apiVersion: v1
4 kind: PersistentVolumeClaim
5 metadata:
6   annotations:
7     pv.kubernetes.io/bind-completed: "yes"
8     pv.kubernetes.io/bound-by-controller: "yes"
9     volume.beta.kubernetes.io/storage-provisioner: fuseim.pri/ifs
10  creationTimestamp: "2022-02-08T01:15:14Z"
11  finalizers:
12  - kubernetes.io/pvc-protection
13  labels:
14    app: nginx
15    ...
16  spec:
17    accessModes:
```

```
18 - ReadWriteOnce
19 resources:
20   requests:
21     storage: 1Gi
22   storageClassName: nfs-storage # StorageClass name
23   volumeMode: Filesystem
24   volumeName: pvc-2c3bbea9-0d0c-41c7-9f4c-c1e8e7700652 # PV name
25 status:
26   accessModes:
27     - ReadWriteOnce
28   capacity:
29     storage: 1Gi
30   phase: Bound
```

PV 信息 (spec.nfs.path)

查看 PV 信息，可以看到自动在 nfs server 生成了一个目录 (见 spec.nfs.path)。

```
1 anxin@node38:~$ kubectl get pv pvc-2c3bbea9-0d0c-41c7-9f4c-c1e8e7700652 -o
  yaml
2
3 apiVersion: v1
4 kind: PersistentVolume
5 metadata:
6   annotations:
7     pv.kubernetes.io/provisioned-by: fuseim.pri/ifs
8   creationTimestamp: "2022-02-08T01:15:14Z"
9   finalizers:
10    - kubernetes.io/pv-protection
11    ...
12 spec:
13   accessModes:
14     - ReadWriteOnce
15   capacity:
16     storage: 1Gi
17   claimRef:
18     apiVersion: v1
19     kind: PersistentVolumeClaim
20     name: www-web-0
21     namespace: julin
22     resourceVersion: "11389720"
23     uid: 2c3bbea9-0d0c-41c7-9f4c-c1e8e7700652
24   nfs: # NFS 服务器
25     path: /data/backups/julin-www-web-0-pvc-2c3bbea9-0d0c-41c7-9f4c-
c1e8e7700652 # 自动生成了一个目录
26     server: 10.8.30.38
27   persistentVolumeReclaimPolicy: Retain
28   storageClassName: nfs-storage
29   volumeMode: Filesystem
30 status:
31   phase: Bound # 已绑定
```


NFS 共享目录

查看 NFS 服务器的 [共享目录](#)。

```
1 anxin@node38:~$ ll /data/backups # nfs 共享目录
2 drwxrwxrwx 2 nobody nogroup 4096 Feb 8 09:15 julin-www-web-0-pvc-2c3bbea9-
  0d0c-41c7-9f4c-c1e8e7700652/
3 drwxrwxrwx 2 nobody nogroup 4096 Feb 8 09:16 julin-www-web-1-pvc-8f7bedec-
  0d67-4b61-a8aa-028d351b60db/
4 drwxrwxrwx 2 nobody nogroup 4096 Feb 8 09:16 julin-www-web-2-pvc-57ddf3f4-
  4729-42db-8037-e980e511fc34/
```

清理工作

先删除 StatefulSet (Pod 由 StatefulSet 控制器管理), 再删除 PVC, 最后删除 PV。

1. 删除 StatefulSet

删除 StatefulSet 后, 其管理的 Pod 自动删除。

```
1 anxin@node38:~/pengling/k8s/controller-demo$ kubectl delete -f statefulset-
  demo.yaml
2 service "nginx" deleted
3 statefulset.apps "web" deleted
```

2. 删除 PVC

PVC 需要手动删除, 删除 PVC 之前需要先删除对应的 Pod (由 StatefulSet 控制器管理)。

```
1 anxin@node38:~$ kubectl delete pvc --all -n julin
2 persistentvolumeclaim "www-web-0" deleted
3 persistentvolumeclaim "www-web-1" deleted
4 persistentvolumeclaim "www-web-2" deleted
```

3. 删除 PV

PV 需要手动删除, 删除 PV 之前需要先删除对应的 PVC。

```
1 anxin@node38:~$ kubectl delete pv --all
2 persistentvolume "pvc-2c3bbea9-0d0c-41c7-9f4c-c1e8e7700652" deleted
3 persistentvolume "pvc-57ddf3f4-4729-42db-8037-e980e511fc34" deleted
4 persistentvolume "pvc-8f7bedec-0d67-4b61-a8aa-028d351b60db" deleted
```

清理 NFS 共享目录

如果不再需要, 可以清理 NFS 共享目录 (/data/backups) 下为 PV 自动生成的目录。

```
1 # 删除以 julin-www- 开头的目录
2 anxin@node38:~$ rm -rf /data/backups/julin-www-* # `julin` 为命名空间, `www` 为
  PVC 模板名称
```

