记k8s的一次pv&pvc创建过程

 发表于 2018-08-18 |  更新于 2018-10-04

今天尝试学习一下k8s的pv&pvc,并记录下实验的过程

**pv&pvc简介**

k8s提供了emptyDir,hostPath,rbd,cephfs等存储方式供容器使用,不过这些存储方式都有一个缺点:开发人员必须得知指定存储的相关配置信息,才能使用存储.例如要使用cephfs,Pod的配置信息就必须指明cephfs的monitor,user,selectFile等等,而这些应该是系统管理员的工作.对此,k8s提供了两个新的API资源:PersistentVolume,PersistentVolumeClaim

PV(PersistentVolume)是管理员已经提供好的一块存储.在k8s集群中,PV像Node一样,是一个资源

PVC(PersistentVolumeClaim)是用户对PV的一次申请.PVC对于PV就像Pod对于Node一样,Pod可以申请CPU和Memory资源,而PVC也可以申请PV的大小与权限

有了PersistentVolumeClaim,用户只需要告诉Kubernetes需要什么样的存储资源,而不必关心真正的空间从哪里分配,如何访问等底层细节信息;这些Storage Provider的底层信息交给管理员来处理,只有管理员才应该关心创建PersistentVolume的细节信息

**实验**

实验环境:

192.168.122.10:host1,Master  
192.168.122.20:host2,Node  
192.168.122.30:host3,Node

目的:使用nfs类型的PV,并使用Job验证PV创建是否创建成功

**1. 在host1上创建nfs服务器,用于提供存储**

安装nfs服务端:

|  |  |
| --- | --- |
| 1 2 3 4 | [root@host1 kube]# sudo apt install nfs-kernel-server ... Installed:  nfs-utils.x86\_64 1:1.3.0-0.54.el7 rpcbind.x86\_64 0:0.2.0-44.el7 |

创建用于nfs服务的存储文件夹:

|  |  |
| --- | --- |
| 1 | [root@host1 kube]# mkdir -p /data/nfs |

编辑/etc/exports文件:

|  |  |
| --- | --- |
| 1 2 | [root@host1 kube]# vim /etc/exports /data/nfs 192.168.122.0/24(rw,sync) |

开启nfs服务:

|  |  |
| --- | --- |
| 1 | [root@host1 kube]# systemctl start nfs-server.service |

检验是否开启成功:

|  |  |
| --- | --- |
| 1 2 3 | [root@host1 kube]# showmount -e Export list for host1: /data/nfs 192.168.122.0/24 |

**2. 创建pv**

编辑pv资源的配置文件:

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 | [kube@host1 ~]$ cat pv\_nfs.yml apiVersion: v1 kind: PersistentVolume metadata:  name: nfspv1 spec:  #指定pv的容量为1Gi  capacity:  storage: 1Gi  #指定访问模式  accessModes:  #pv能以readwrite模式mount到单个节点  - ReadWriteOnce  #指定pv的回收策略,即pvc资源释放后的事件.recycle(不建议,使用动态供给代替)删除pvc的所有文件  persistentVolumeReclaimPolicy: Recycle  #指定pv的class为nfs,相当于为pv分类,pvc将指定class申请pv  storageClassName: mynfs  #指定pv为nfs服务器上对应的目录  nfs:  path: /data/nfs  server: 192.168.122.10 |

应用该pv资源:

|  |  |
| --- | --- |
| 1 2 | [kube@host1 ~]$ kubectl apply -f pv\_nfs.yml  persistentvolume/nfspv1 created |

查看该pv资源:

|  |  |
| --- | --- |
| 1 2 3 | [kube@host1 ~]$ kubectl get pv NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE nfspv1 1Gi RWO Recycle Available mynfs 14s |

**3. 创建pvc**

编辑pvc资源配置文件:

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 | [kube@host1 ~]$ cat pvc\_nfs.yml kind: PersistentVolumeClaim apiVersion: v1 metadata:  name: nfspvc1 spec:  accessModes:  - ReadWriteOnce  resources:  requests:  storage: 1Gi  storageClassName: mynfs |

应用pvc配置文件:

|  |  |
| --- | --- |
| 1 2 | [kube@host1 ~]$ kubectl apply -f pvc\_nfs.yml persistentvolumeclaim/nfspvc1 created |

查看该pvc资源:

|  |  |
| --- | --- |
| 1 2 3 | [kube@host1 ~]$ kubectl get pvc NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE nfspvc1 Bound nfspv1 1Gi RWO mynfs 24s |

**4. 创建应用资源,使用pvc存储**

编辑应用资源配置文件:

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 | [kube@host1 ~]$ cat pvjob.yml apiVersion: batch/v1 kind: Job metadata:  name: pvjob spec:  template:  spec:  containers:  - name: bbox1  image: busybox  args:  - /bin/sh  - -c  - echo "hello pv" > /mydata/hello  volumeMounts:  - mountPath: "/mydata"  name: mydata  restartPolicy: Never  volumes:  - name: mydata  persistentVolumeClaim:  claimName: nfspvc1 |

该job将在nfs的volume创建一个hello文件,打印”hello pv”字符串

应用该Job资源:

|  |  |
| --- | --- |
| 1 2 | [kube@host1 ~]$ kubectl apply -f pvjob.yml job.batch/pvjob created |

查看host1上的/data/nfs目录,检查实验结果:

|  |  |
| --- | --- |
| 1 2 | [kube@host1 ~]$ cat /data/nfs/hello hello pv |

实验成功!

**排错**

在这次实验中出现了几个错误:

**1. 使用pvc存储的pod一直处于pending状态,无法启动**

|  |  |
| --- | --- |
| 1 2 3 4 5 6 | [kube@host1 ~]$ kubectl get jobs NAME READY STATUS RESTARTS AGE pvjob-47cwq 0/1 Pending 0 5s |

查看该pod的启动事件:

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 | [kube@host1 ~]$ kubectl describe job pvjob Name: pvjob-47cwq Namespace: default Node: host3/192.168.122.30 Status: Pending IP: Events:  Type Reason Age From Message  ---- ------ ---- ---- -------  Normal Scheduled 44s default-scheduler Successfully assigned default/liveness to host3  Warning FailedMount 44s kubelet, host3 MountVolume.SetUp failed for volume "nfspv1" : mount failed: exit status 32 Mounting command: systemd-run Mounting arguments: --description=Kubernetes transient mount for /var/lib/kubelet/pods/91ddd2ee-a2b8-11e8-a8f5-525400909158/volumes/kubernetes.io~nfs/nfspv1 --scope -- mount -t nfs 192.168.122.10:/data/nfs /var/lib/kubelet/pods/91ddd2ee-a2b8-11e8-a8f5-525400909158/volumes/kubernetes.io~nfs/nfspv1 Output: Running scope as unit run-24516.scope. mount: wrong fs type, bad option, bad superblock on 192.168.122.10:/data/nfs,  missing codepage or helper program, or other error  (for several filesystems (e.g. nfs, cifs) you might  need a /sbin/mount.<type> helper program)   In some cases useful info is found in syslog - try  dmesg | tail or so. |

注意到报错信息:wrong fs type, bad option, bad superblock on 192.168.122.10:/data/nfs,这是因为host3没有安装nfs-utils软件包,无法识别nfs类型的文件系统,也无法作为nfs的客户端使用

解决方案:安装nfs-utils软件包,删除之前创建失败的Job资源并重新创建

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 | [root@host3 ~]# apt-get install nfs-common ... Installed:  nfs-utils.x86\_64 1:1.3.0-0.54.el7  [kube@host1 ~]$ kubectl delete pod pvjob-47cwq pod "pvjob-47cwq" deleted  [kube@host1 ~]$ kubectl apply -f pvjob-47cwq job.batch/pvjob created |

可以看到,Job资源正常创建:

|  |  |
| --- | --- |
| 1 2 3 | [kube@host1 ~]$ kubectl get pod NAME READY STATUS RESTARTS AGE pvjob-47cwq 0/1 ContainerCreating 0 6s |

**2. 使用pvc的job执行失败**

|  |  |
| --- | --- |
| 1 2 3 4 5 | [kube@host1 ~]$ kubectl get pod NAME READY STATUS RESTARTS AGE pvjob-c25xg 0/1 Error 0 25s pvjob-db6fb 0/1 Error 0 32s pvjob-hf7px 0/1 Error 0 15s |

查看执行失败Job的log日志:

|  |  |
| --- | --- |
| 1 2 | [kube@host1 ~]$ kubectl logs pvjob-74g64 /bin/sh: can't create /mydata/hello: Permission denied |

解决方案:

Permission denied多见于普通用户执行高权限命令失败,不过busybox容器本身使用的就是root用户,因此不存在这个问题.在nfs中,nfs服务端没有权限访问挂载的目录也会导致这个问题

更改目录属主为nfsnobody:

|  |  |
| --- | --- |
| 1 | [kube@host1 ~]$ sudo chown nfsnobody /data/nfs |

删除原来Job并重新创建:

|  |  |
| --- | --- |
| 1 2 3 4 5 | [kube@host1 ~]$ kubectl delete jobs pvjob.yml job.batch "pvjob" deleted  [kube@host1 ~]$ kubectl apply -f pvjob.yml job.batch/pvjob created |

可以看到,Job执行成功:

|  |  |
| --- | --- |
| 1 2 3 | [kube@host1 ~]$ kubectl get pod NAME READY STATUS RESTARTS AGE pvjob-pct6b 0/1 Completed 0 8s |