2017 China Kubernetes End User Conference KEUC

kubernetes 中国用户大会 2017

2017.10.15 / 中国 · 杭州

Kubernetes 存储技术介绍及在云计算平台 Azure 上的实现

张夏 微软上海开源技术组

Github: andyzhangx

主办： CLOUD NATIVE COMPUTING FOUNDATION caicloud 才云

Kubernetes持久化存储(PV)技术介绍

k8s持久化存储(PV)在Azure上的实现

如何实现持久化存储(PV)插件

主办：
CLOUD NATIVE COMPUTING FOUNDATION
caicloud 才云
K8S 中国社区

Kubernetes持久化存储(PV)技术介绍

PersistentVolume(PV)

提供网络存储资源 by admin

PersistentVolumeClaim (PVC)

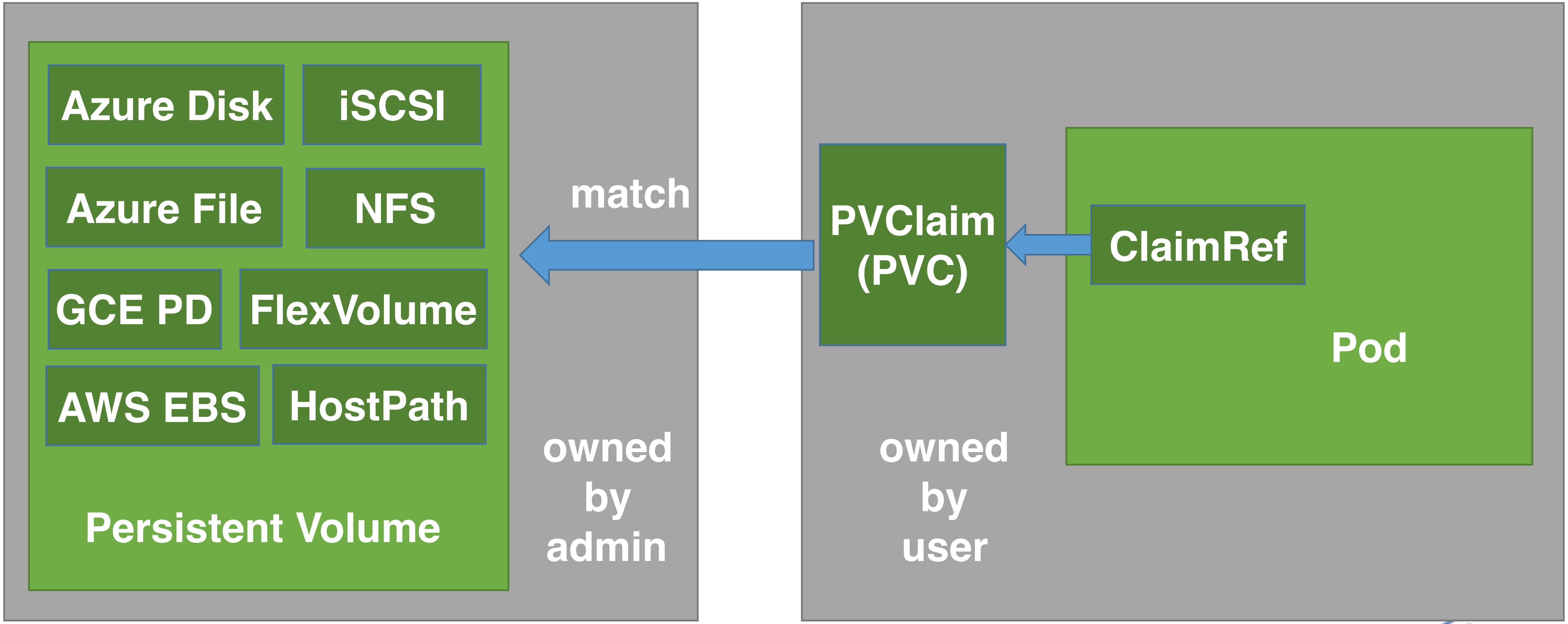
请求存储资源 by user

StorageClass

封装不同类型存储 by admin

动态创建PV

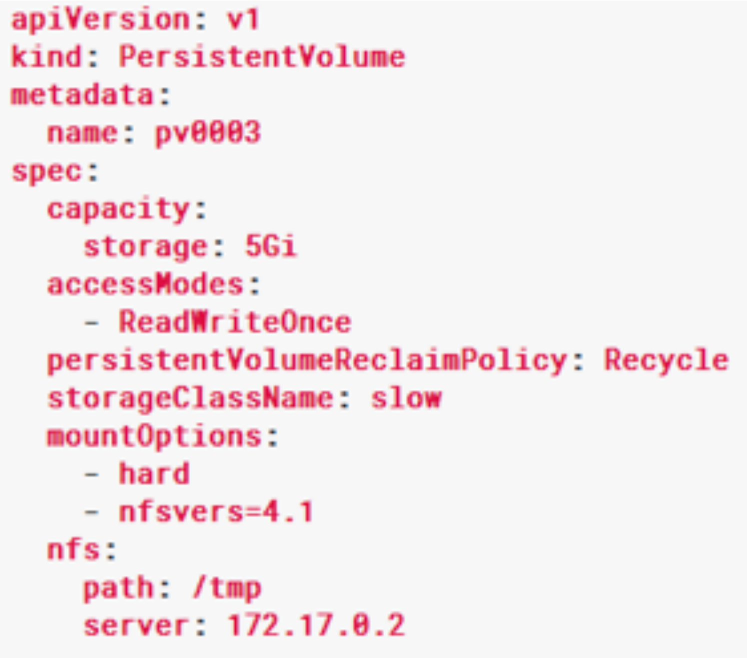
PV & PVC


Azure Disk
iSCSI
Azure File
NFS
GCE PD
FlexVolume
AWS EBS
HostPath
Persistent Volume
owned by admin
match
PVClaim (PVC)
ClaimRef
Pod
owned by user

PersistentVolume(PV) 持久化卷

Spec

- Capacity 容量
  - storage size
- Access Modes 访问模式
  - RWO, RWX, ROX
- Class 存储类
- Reclaim Policy 回收策略
  - Retain, Recycle, Delete
- Mount Options 加载选项



```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0003
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: slow
  mountOptions:
    - hard
    - nfsvers=4.1
  nfs:
    path: /tmp
    server: 172.17.0.2
```

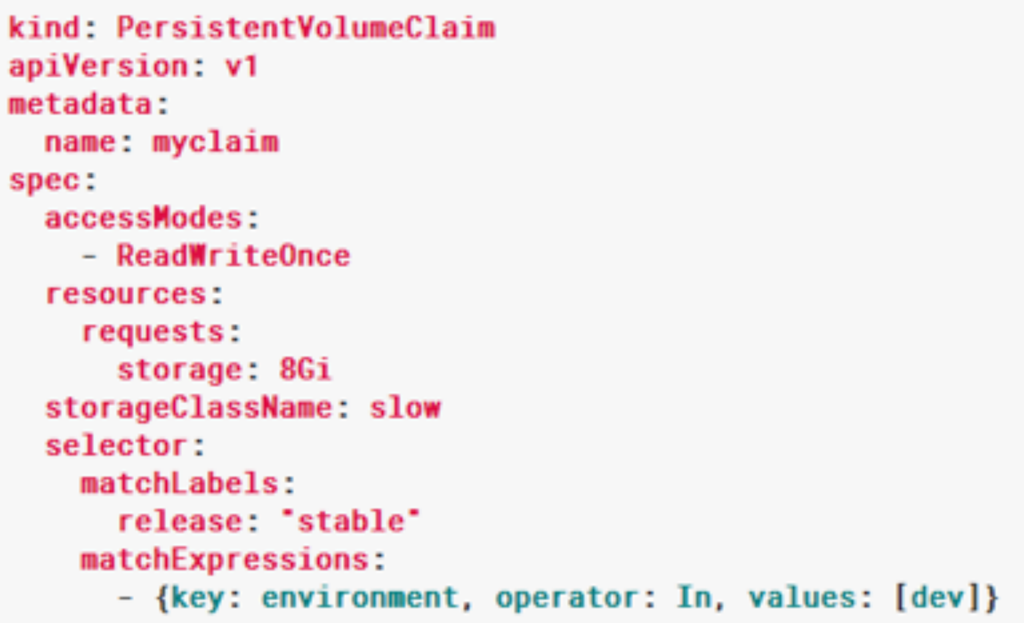
Status

- Available, Bound, Released, Failed

PersistentVolumeClaim(PVC) 持久化卷请求

Spec

- Access Modes 访问模式
  - RWO, RWX, ROX
- Resources 请求资源
  - storage size
- Class 存储类
- Selector 匹配
  - match label



```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: myclaim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 8Gi
  storageClassName: slow
  selector:
    matchLabels:
      release: "stable"
    matchExpressions:
      - {key: environment, operator: In, values: [dev]}
```

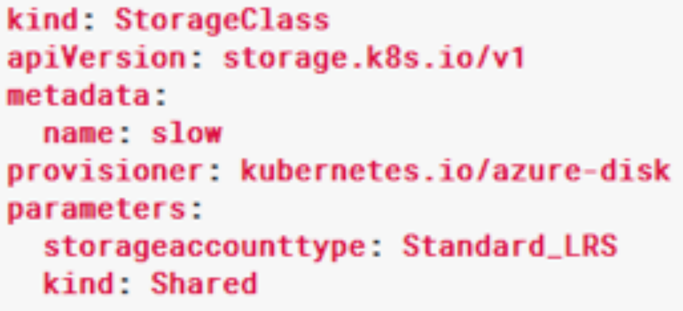
Status

- Pending, Bound

StorageClass 存储类型

Provisioner

- Internal(In-Tree) Provisioner
- External(Out-of-Tree) Provisioner



```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: slow
provisioner: kubernetes.io/azure-disk
parameters:
  storageaccounttype: Standard_LRS
  kind: Shared
```

Reclaim Policy 回收策略

Parameters 参数

当前kubernetes提供的PV Plugin

Volume Plugin	Internal Provisioner	Config Example
AWSElasticBlockStore	✓	AWS
AzureFile	✓	Azure File
AzureDisk	✓	Azure Disk
CephFS	-	-
Cinder	✓	OpenStack Cinder
FC	-	-
FlexVolume	-	-
Flocker	✓	-
GCEPersistentDisk	✓	GCE
Glusterfs	✓	Glusterfs
iSCSI	-	-
PhotonPersistentDisk	✓	-
Quobyte	✓	Quobyte
NFS	-	-
RBD	✓	Ceph RBD
VsphereVolume	✓	vSphere
PortworxVolume	✓	Portworx Volume

Kubernetes持久化存储(PV)技术介绍

k8s持久化存储(PV)在Azure上的实现

如何实现持久化存储(PV)插件

主办：

AzureDisk

通过在Node上加载Azure磁盘方式实现

Access Mode:

ReadWriteOnce(RWO)

AzureFile

通过在Node上加载Azure共享文件（SMB 2.1, 3.0 ）实现

Access Mode:

ReadWriteOnce(RWO), ReadWriteMany(RWX), ReadOnlyMany(ROX)

如何使用Azure Disk Volume

1. 创建Azure Disk StorageClass
2. 创建PersistentVolumeClaim(PVC)
3. 创建Pod时使用该PVC

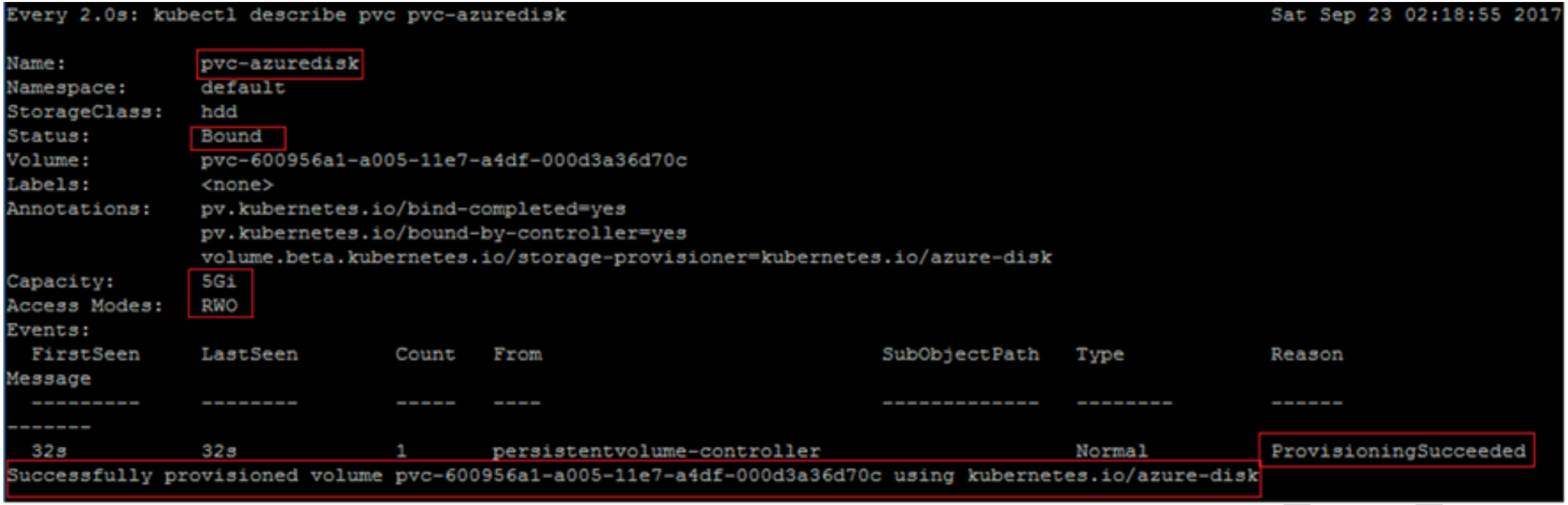
使用Azure Disk Volume

```
kind: StorageClass
apiVersion: v1
metadata:
  name: hdd
provisioner: kubernetes.io/azure-disk
parameters:
  storageaccounttype: Standard_LRS
  kind: Shared
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-azuredisk
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
  storageClassName: hdd
```

使用Azure Disk Volume

1. kubectl create -f storageclass-azuredisk.yaml
2. kubectl create -f pvc-azuredisk.yaml

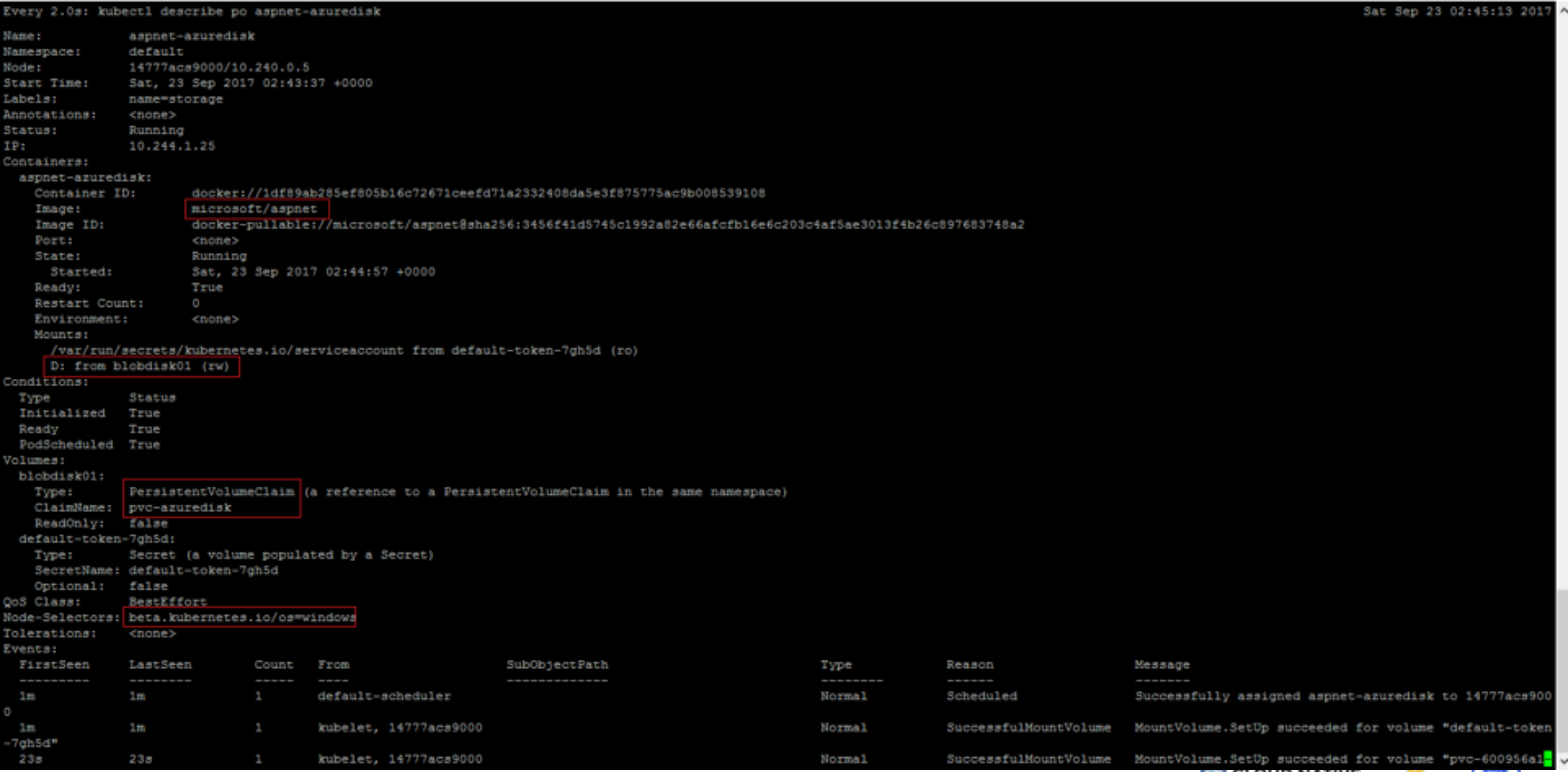


```
Every 2.0s: kubectl describe pvc pvc-azuredisk                                                              Sat Sep 23 02:18:55 2017

Name:           pvc-azuredisk
Namespace:      default
StorageClass:   hdd
Status:         Bound
Volume:         pvc-600956a1-a005-11e7-a4df-000d3a36d70c
Labels:         <none>
Annotations:    pv.kubernetes.io/bind-completed=yes
                pv.kubernetes.io/bound-by-controller=yes
                volume.beta.kubernetes.io/storage-provisioner=kubernetes.io/azure-disk
Capacity:       5Gi
Access Modes:   RWO
Events:
  FirstSeen     LastSeen        Count   From                            SubObjectPath   Type            Reason
Message
  ---------     --------        -----   ----                            -------------   --------        ------
-------
  32s           32s             1       persistentvolume-controller                     Normal          ProvisioningSucceeded
Successfully provisioned volume pvc-600956a1-a005-11e7-a4df-000d3a36d70c using kubernetes.io/azure-disk
```

创建一个基于Azure Disk Volume的Windows POD

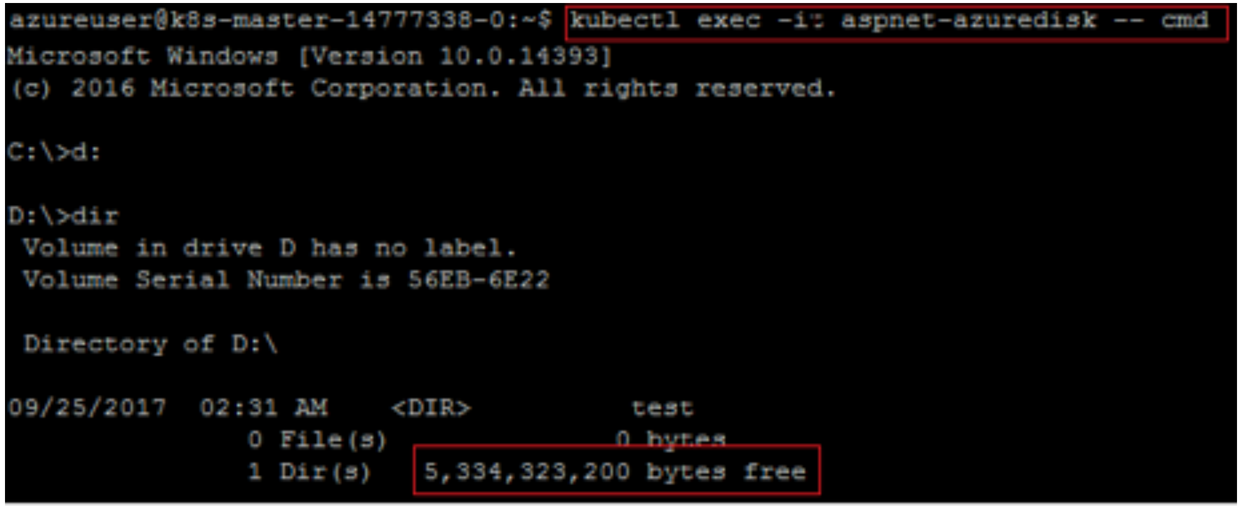
```
kind: Pod
apiVersion: v1
metadata:
  name: aspnet-azuredisk
spec:
  containers:
  - image: microsoft/aspnet
    name: aspnet-azuredisk
    volumeMounts:
    - name: blobdisk01
      mountPath: 'D:'
  nodeSelector:
    beta.kubernetes.io/os: windows
  volumes:
  - name: blobdisk01
    persistentVolumeClaim:
      claimName: pvc-azuredisk
```



```
Every 2.0s: kubectl describe po aspnet-azuredisk                                                                                                                                                        Sat Sep 23 02:45:13 2017

Name:           aspnet-azuredisk
Namespace:      default
Node:           14777acs9000/10.240.0.5
Start Time:     Sat, 23 Sep 2017 02:43:37 +0000
Labels:         name=storage
Annotations:    <none>
Status:         Running
IP:             10.244.1.25
Containers:
  aspnet-azuredisk:
    Container ID:       docker://1df89ab285ef805b16c72671ceefd71a2332408da5e3f875775ac9b008539108
    Image:              microsoft/aspnet
    Image ID:           docker-pullable://microsoft/aspnet@sha256:3456f41d5745c1992a82e66afcfb16e6c203c4af5ae3013f4b26c897683748a2
    Port:               <none>
    State:              Running
      Started:          Sat, 23 Sep 2017 02:44:57 +0000
    Ready:              True
    Restart Count:      0
    Environment:        <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-7gh5d (ro)
      D: from blobdisk01 (rw)
Conditions:
  Type          Status
  Initialized   True
  Ready         True
  PodScheduled  True
Volumes:
  blobdisk01:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  pvc-azuredisk
    ReadOnly:   false
  default-token-7gh5d:
    Type:       Secret (a volume populated by a Secret)
    SecretName: default-token-7gh5d
    Optional:   false
QoS Class:      BestEffort
Node-Selectors: beta.kubernetes.io/os=windows
Tolerations:    <none>
Events:
  FirstSeen     LastSeen        Count   From                    SubObjectPath                           Type            Reason                  Message
  ---------     --------        -----   ----                    -------------                           --------        ------                  -------
  1m            1m              1       default-scheduler                                               Normal          Scheduled               Successfully assigned aspnet-azuredisk to 14777acs900
0
  1m            1m              1       kubelet, 14777acs9000                                           Normal          SuccessfulMountVolume   MountVolume.SetUp succeeded for volume "default-token
-7gh5d"
  23s           23s             1       kubelet, 14777acs9000                                           Normal          SuccessfulMountVolume   MountVolume.SetUp succeeded for volume "pvc-600956a1
```

使用Azure Disk Volume



```
azureuser@k8s-master-14777338-0:~$ kubectl exec -it aspnet-azuredisk -- cmd
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\>d:

D:\>dir
 Volume in drive D has no label.
 Volume Serial Number is 56EB-6E22

 Directory of D:\

09/25/2017  02:31 AM    <DIR>          test
               0 File(s)              0 bytes
               1 Dir(s)  5,334,323,200 bytes free
```

AzureDisk持久化存储实现细节

On Kubernetes **Master (Controller Manager)**

- 在k8s 集群所在的资源组中创建一块Azure Disk
- 将新创建的Azure Disk添加到kubernetes node上

On Kubernetes **Node (Kubelet)**

- 循环检测新磁盘
- 将检测到的新磁盘格式化并链接到kubernetes目录下
- 创建pod时使用volume映射，将该目录映射到用户指

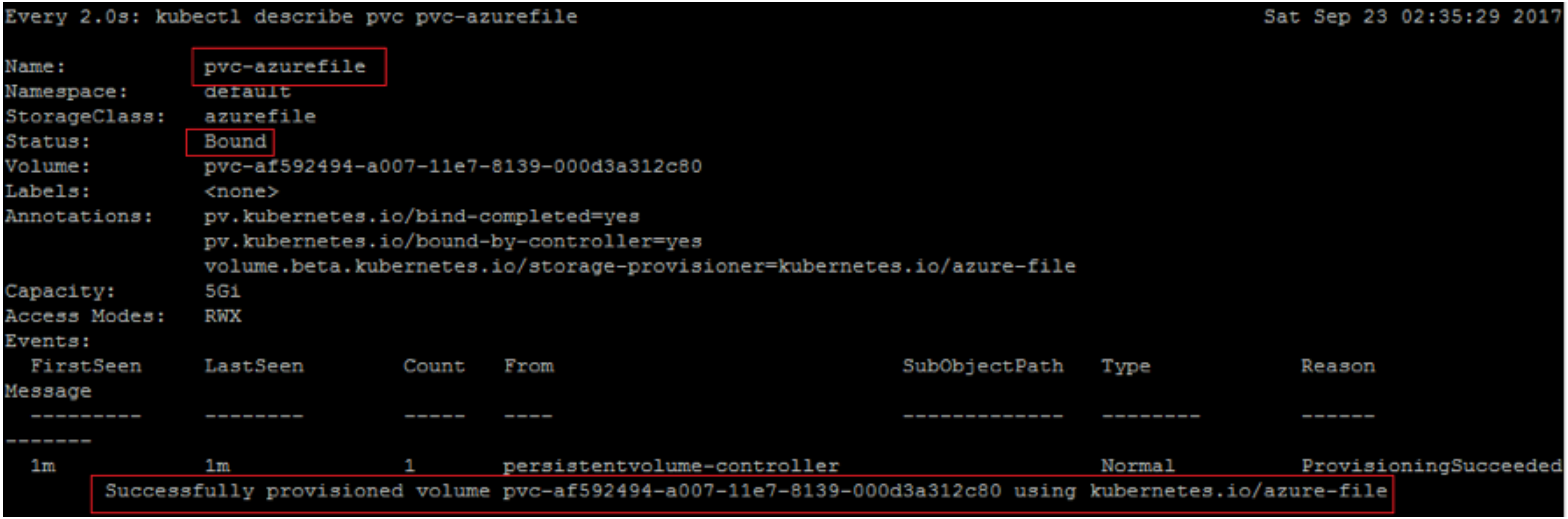
使用Azure File Volume

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: azurefile
provisioner: kubernetes.io/azure-file
parameters:
  skuName: Standard_LRS
  location: westus
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-azurefile
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 5Gi
  storageClassName: azurefile
```

使用Azure File Volume

1. kubectl create -f storageclass-azurefile.yaml
2. kubectl create -f pvc-azurefile.yaml

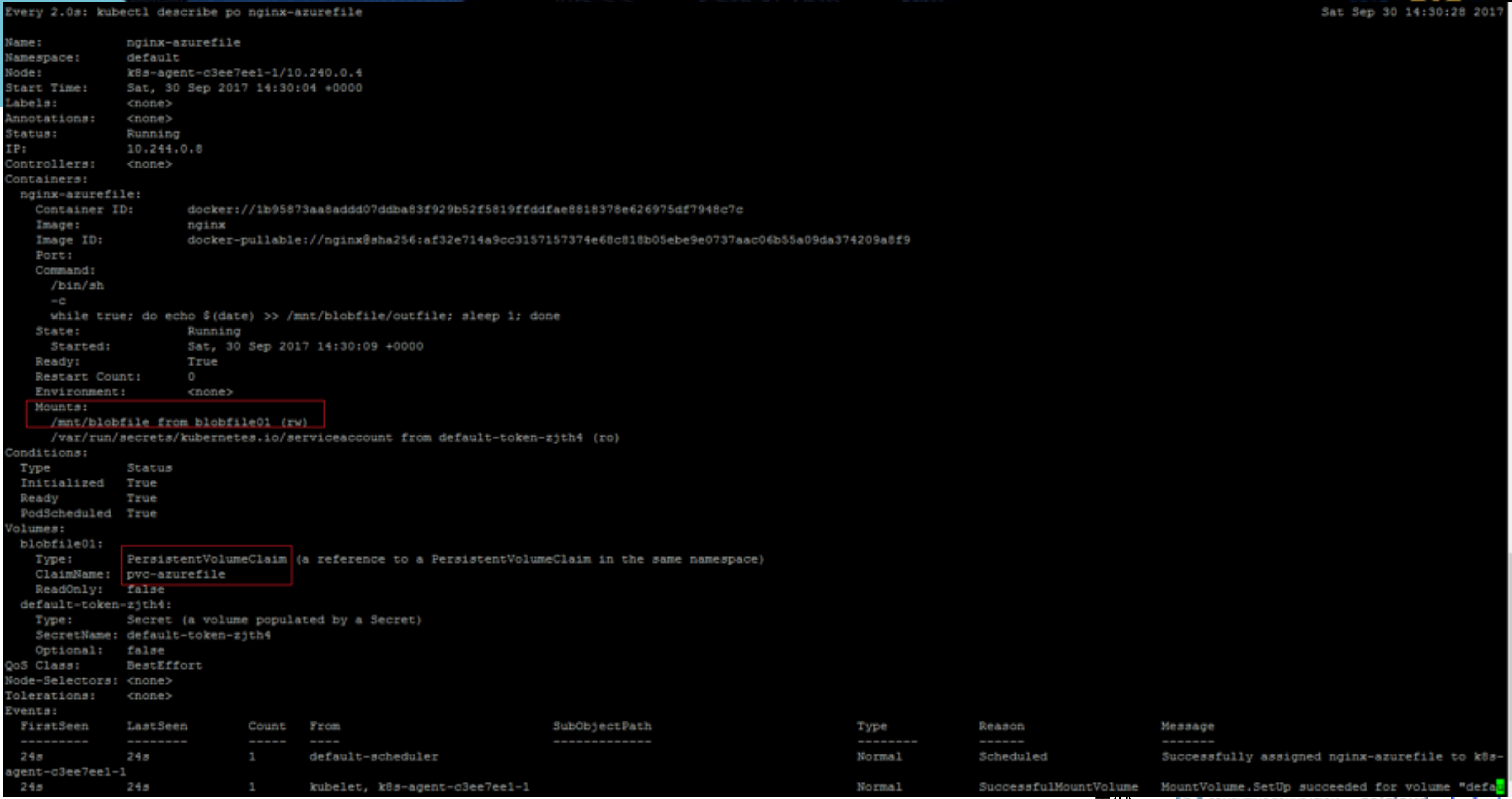


```
Every 2.0s: kubectl describe pvc pvc-azurefile                                                       Sat Sep 23 02:35:29 2017

Name:           pvc-azurefile
Namespace:      default
StorageClass:   azurefile
Status:         Bound
Volume:         pvc-af592494-a007-11e7-8139-000d3a312c80
Labels:         <none>
Annotations:    pv.kubernetes.io/bind-completed=yes
                pv.kubernetes.io/bound-by-controller=yes
                volume.beta.kubernetes.io/storage-provisioner=kubernetes.io/azure-file
Capacity:       5Gi
Access Modes:   RWX
Events:
  FirstSeen     LastSeen        Count   From                            SubObjectPath   Type            Reason
Message
  ---------     --------        -----   ----                            -------------   --------        ------
-------
  1m            1m              1       persistentvolume-controller                     Normal          ProvisioningSucceeded
        Successfully provisioned volume pvc-af592494-a007-11e7-8139-000d3a312c80 using kubernetes.io/azure-file
```

创建一个基于Azure File Volume 的 pod

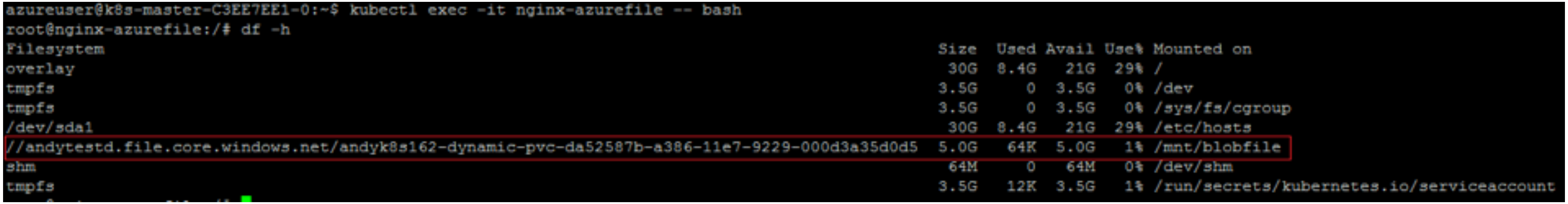
```
kind: Pod
apiVersion: v1
metadata:
  name: nginx-azurefile
spec:
  containers:
  - image: nginx
    name: nginx-azurefile
volumeMounts:
    - name: azurefile01
      mountPath: "/mnt/blobfile"
  volumes:
  - name: azurefile01
    persistentVolumeClaim:
      claimName: pvc-azurefile
```



```
Every 2.0s: kubectl describe po nginx-azurefile                                                                                                                                          Sat Sep 30 14:30:28 2017

Name:           nginx-azurefile
Namespace:      default
Node:           k8s-agent-c3ee7ee1-1/10.240.0.4
Start Time:     Sat, 30 Sep 2017 14:30:04 +0000
Labels:         <none>
Annotations:    <none>
Status:         Running
IP:             10.244.0.8
Controllers:    <none>
Containers:
  nginx-azurefile:
    Container ID:       docker://1b95873aa8addd07ddba83f929b52f5819ffddfae8818378e626975df7948c7c
    Image:              nginx
    Image ID:           docker-pullable://nginx@sha256:af32e714a9cc3157157374e68c818b05ebe9e0737aac06b55a09da374209a8f9
    Port:
    Command:
      /bin/sh
      -c
      while true; do echo $(date) >> /mnt/blobfile/outfile; sleep 1; done
    State:              Running
      Started:          Sat, 30 Sep 2017 14:30:09 +0000
    Ready:              True
    Restart Count:      0
    Environment:        <none>
    Mounts:
      /mnt/blobfile from blobfile01 (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-zjth4 (ro)
Conditions:
  Type          Status
  Initialized   True
  Ready         True
  PodScheduled  True
Volumes:
  blobfile01:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  pvc-azurefile
    ReadOnly:   false
  default-token-zjth4:
    Type:       Secret (a volume populated by a Secret)
    SecretName: default-token-zjth4
    Optional:   false
QoS Class:      BestEffort
Node-Selectors: <none>
Tolerations:    <none>
Events:
  FirstSeen     LastSeen        Count   From                            SubObjectPath                           Type            Reason                  Message
  ---------     --------        -----   ----                            -------------                           --------        ------                  -------
  24s           24s             1       default-scheduler                                                       Normal          Scheduled               Successfully assigned nginx-azurefile to k8s-
agent-c3ee7ee1-1
  24s           24s             1       kubelet, k8s-agent-c3ee7ee1-1                                           Normal          SuccessfulMountVolume   MountVolume.SetUp succeeded for volume "defa
```

使用Azure File Volume



```
azureuser@k8s-master-C3EE7EE1-0:~$ kubectl exec -it nginx-azurefile -- bash
root@nginx-azurefile:/# df -h
Filesystem                                                                                     Size  Used Avail Use% Mounted on
overlay                                                                                         30G  8.4G   21G  29% /
tmpfs                                                                                          3.5G     0  3.5G   0% /dev
tmpfs                                                                                          3.5G     0  3.5G   0% /sys/fs/cgroup
/dev/sda1                                                                                       30G  8.4G   21G  29% /etc/hosts
//andytestd.file.core.windows.net/andyk8s162-dynamic-pvc-da52587b-a386-11e7-9229-000d3a35d0d5  5.0G   64K  5.0G   1% /mnt/blobfile
shm                                                                                             64M     0   64M   0% /dev/shm
tmpfs                                                                                          3.5G   12K  3.5G   1% /run/secrets/kubernetes.io/serviceaccount
```

AzureFile持久化存储实现细节

On Kubernetes **Master (Controller Manager)**

- 在k8s cluster所在的资源组中创建Azure File(SMB share)

On Kubernetes **Node (Kubelet)**

- 取得azure file账号信息
- 将创建的Azure File 使用cifs协议加载到kubernetes目录下

Azure Disk & File Volume Demo

https://github.com/**andyzhangx**/Demo/tree/master/windows/azuredisk

https://github.com/**andyzhangx**/Demo/tree/master/linux/azurefile

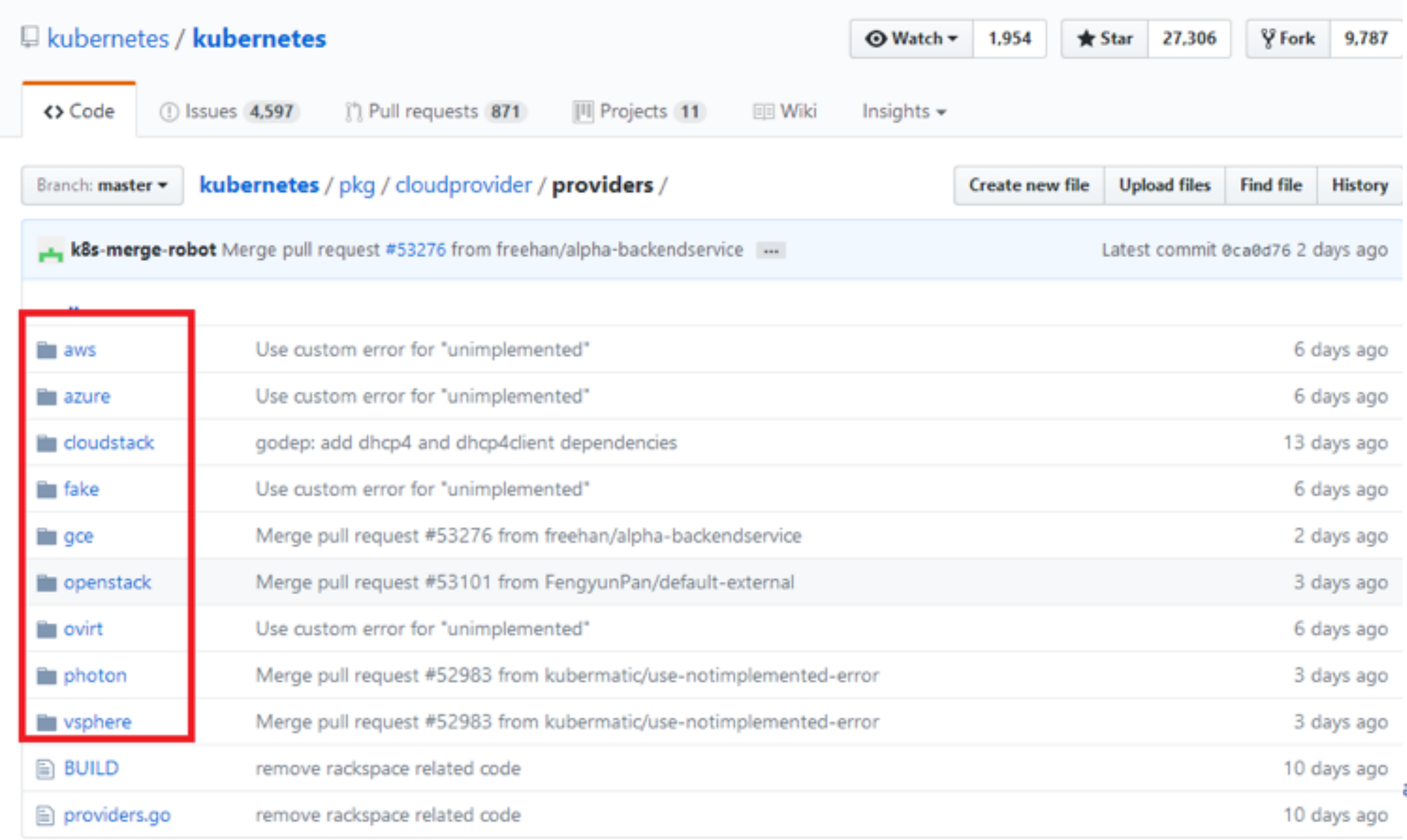
Kubernetes持久化存储(PV)技术介绍

k8s持久化存储(PV)在Azure上的实现

如何实现持久化存储(PV)插件

主办：CLOUD NATIVE COMPUTING FOUNDATION caicloud 才云 中国社区

实现新的Internal Cloud Provider?

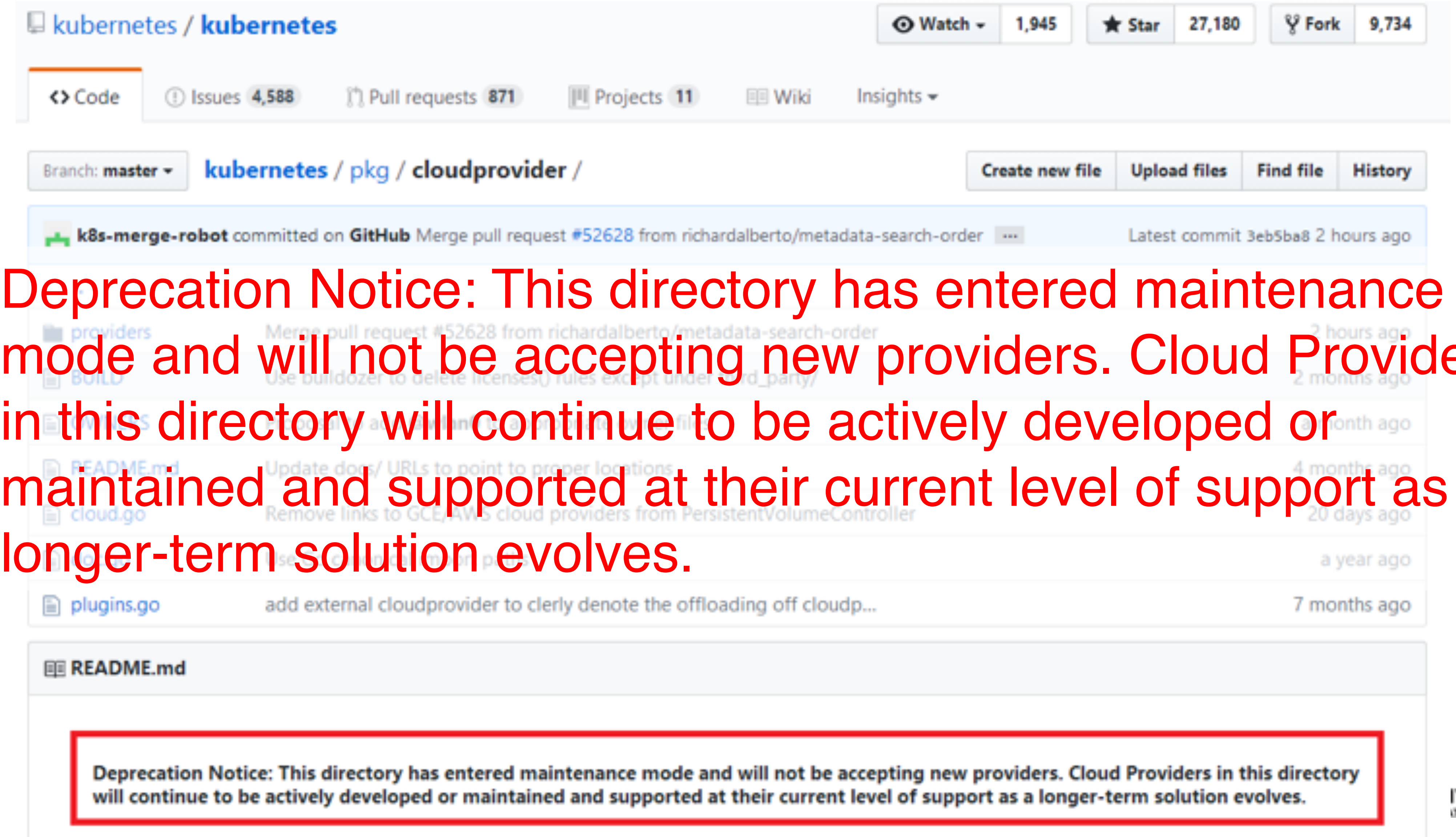


kubernetes / pkg / cloudprovider / providers /

k8s-merge-robot Merge pull request #53276 from freehan/alpha-backendservice … Latest commit 0ca0d76 2 days ago

..		
aws	Use custom error for "unimplemented"	6 days ago
azure	Use custom error for "unimplemented"	6 days ago
cloudstack	godep: add dhcp4 and dhcp4client dependencies	13 days ago
fake	Use custom error for "unimplemented"	6 days ago
gce	Merge pull request #53276 from freehan/alpha-backendservice	2 days ago
openstack	Merge pull request #53101 from FengyunPan/default-external	3 days ago
ovirt	Use custom error for "unimplemented"	6 days ago
photon	Merge pull request #52983 from kubermatic/use-notimplemented-error	3 days ago
vsphere	Merge pull request #52983 from kubermatic/use-notimplemented-error	3 days ago
BUILD	remove rackspace related code	10 days ago
providers.go	remove rackspace related code	10 days ago

实现新的Internal Cloud Provider?



kubernetes / pkg / cloudprovider /

k8s-merge-robot committed on GitHub Merge pull request #52628 from richardalberto/metadata-search-order ··· Latest commit 3eb5ba8 2 hours ago

providers	Merge pull request #52628 from richardalberto/metadata-search-order	2 hours ago
BUILD	Use buildozer to delete licenses() rules except under third_party/	2 months ago
OWNERS	[illegible]	a month ago
README.md	Update docs/ URLs to point to proper locations	4 months ago
cloud.go	Remove links to GCE/AWS cloud providers from PersistentVolumeController	20 days ago
[illegible]	[illegible]	a year ago
plugins.go	add external cloudprovider to clerly denote the offloading off cloudp...	7 months ago

README.md

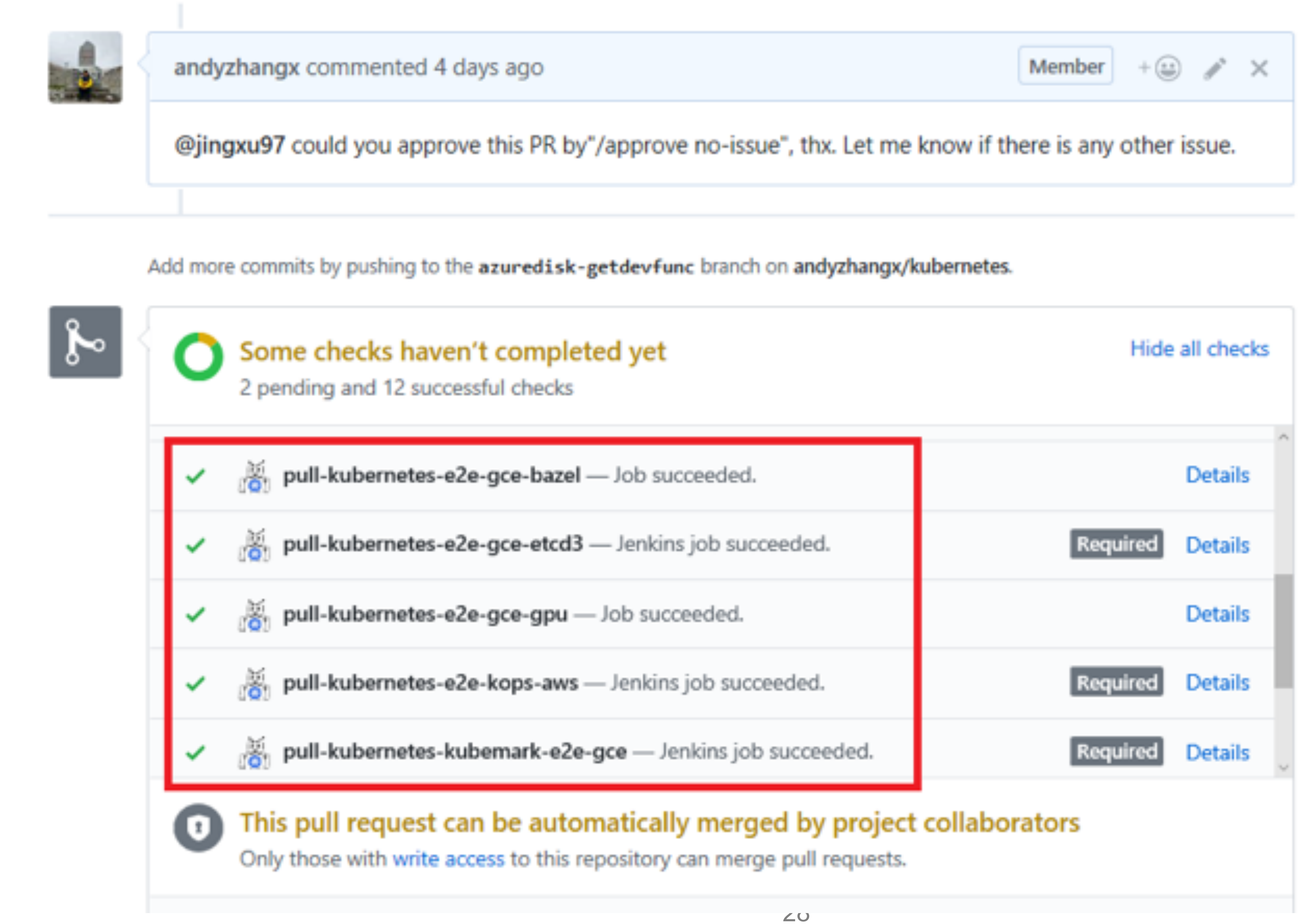
Deprecation Notice: This directory has entered maintenance mode and will not be accepting new providers. Cloud Providers in this directory will continue to be actively developed or maintained and supported at their current level of support as a longer-term solution evolves.

Deprecation Notice: This directory has entered maintenance mode and will not be accepting new providers. Cloud Providers in this directory will continue to be actively developed or maintained and supported at their current level of support as a longer-term solution evolves.

实现新的Internal Cloud Provider?

Inter

进行快速迭代



andyzhangx commented 4 days ago

@jingxu97 could you approve this PR by"/approve no-issue", thx. Let me know if there is any other issue.

Add more commits by pushing to the `azuredisk-getdevfunc` branch on andyzhangx/kubernetes.

Some checks haven't completed yet

2 pending and 12 successful checks

✓ pull-kubernetes-e2e-gce-bazel — Job succeeded. Details

✓ pull-kubernetes-e2e-gce-etcd3 — Jenkins job succeeded. Required Details

✓ pull-kubernetes-e2e-gce-gpu — Job succeeded. Details

✓ pull-kubernetes-e2e-kops-aws — Jenkins job succeeded. Required Details

✓ pull-kubernetes-kubemark-e2e-gce — Jenkins job succeeded. Required Details

This pull request can be automatically merged by project collaborators

Only those with write access to this repository can merge pull requests.

实现新的Internal Cloud Provider?

Proposal: refactor Cloud Provider out of Kubernetes Core

将cloud provider相关代码迁移到云厂商各自repo

计划 (https://github.com/kubernetes/features/issues/88)

1.7 alpha

1.8,1.9 beta

1.10 Stable: 将移除cloud provider相关代码

Standalone **Azure** cloud provider#50752

如何实现Persistent Volume插件

- Flexvolume插件
- External(Out-of-Tree) Provisioner
  - https://github.com/kubernetes-incubator/external-storage

flexvolume 插件需要实现的主要接口

init – initialize the driver

WaitForAttach - get device info

Attach - attach device to the node

MountDevice - mount (&format device) in file system in the node

UnmountDevice - unmount device in file system in the node

Detach - detach device from the node

如何使用

在各个k8s node及master指定目录安装flexvolume 插件

使用nfs flexvolume插件

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-nfs
spec:
  containers:
  - name: nginx-nfs
    image: nginx
    volumeMounts:
    - name: test
      mountPath: /data
volumes:
  - name: test
    flexVolume:
      driver: "k8s/nfs"
      fsType: "nfs"
      options:
        server: "172.16.0.25"
        share: "share"
```

实现Out-of-Tree Provisioner

需要实现的主要接口

provision

delete

main() – run as a pod in k8s

在kubernetes集群中以POD方式运行

1. 基于out-of-tree provisioner code生成docker image
2. 在k8s集群中运行一个基于该provisioner image的POD
3. 为POD设置RBAC权限访问k8s集群
4. 使用out-of-tree provisioner: 生成StorageClass和PVC

如何实现Persistent Volume插件

- Internal(in-tree) Provisioner
- Flexvolume插件
- External(Out-of-Tree) Provisioner
  - https://github.com/kubernetes-incubator/external-storage

2017 China Kubernetes End User Conference
kubernetes 中国用户大会
KEUC
2017
2017.10.15 / 中国 · 杭州

Q&A

Thank you for your time

主办： CLOUD NATIVE COMPUTING FOUNDATION caicloud 才云