使用glusterfs做持久化存储

安装glusterfs

我们直接在物理机上使用yum安装，如果你选择在kubernetes上安装，请参考：https://github.com/gluster/gluster-kubernetes/blob/master/docs/setup-guide.md

所有节点

```
# 先安装 gluster 源
[root@vlnx251101 ~]# yum install centos-release-gluster -y

# 安装 glusterfs 组件
[root@vlnx251101 ~]# yum install glusterfs glusterfs-server glusterfs-fuse glusterfs-rdma glusterfs-geo-replication glusterfs-devel

## 创建 glusterfs 目录
[root@vlnx251101 ~]# mkdir /opt/glusterd

## 修改 glusterd 目录
[root@vlnx251101 ~]# sed -i 's/var\/lib/opt/g' /etc/glusterfs/glusterd.vol

# 启动 glusterfs
[root@vlnx251101 ~]# systemctl start glusterd.service ; systemctl enable glusterd.service ;  systemctl status glusterd.service
```

配置 glusterfs

```
# 配置 hosts
[root@vlnx251101 ~]# vim /etc/hosts

192.168.251.101 vlnx251101.zyg.com vlnx251101
192.168.251.102 vlnx251102.zyg.com vlnx251102
192.168.251.103 vlnx251103.zyg.com vlnx251103


# 开放端口
$ iptables -I INPUT -p tcp --dport 24007 -j ACCEPT


# 创建存储目录
[root@vlnx251101 ~]# mkdir /opt/gfs_data

# 添加节点到 集群
# 执行操作的本机不需要probe 本机
[root@vlnx251101 ~]# gluster peer probe vlnx251102.zyg.com
[root@vlnx251101 ~]# gluster peer probe vlnx251103.zyg.com

# 查看集群状态
[root@vlnx251101 ~]# gluster peer status
Number of Peers: 2
Hostname: vlnx251102.zyg.com
Uuid: 4db5fd67-fae7-4b6c-8474-6ca7a1486b62
State: Peer in Cluster (Connected)
Hostname: vlnx251103.zyg.com
Uuid: 5bcbd0b5-12e6-4d6d-8587-55ef43f63133
State: Peer in Cluster (Connected)
```

配置 volume

GlusterFS中的volume的模式有很多中，包括以下几种：

- **分布卷（默认模式）**：即DHT, 也叫 分布卷: 将文件已hash算法随机分布到 一台服务器节点中存储。
- **复制模式**：即AFR, 创建volume 时带 replica x 数量: 将文件复制到 replica x 个节点中。
- **条带模式**：即Striped, 创建volume 时带 stripe x 数量： 将文件切割成数据块，分别存储到 stripe x 个节点中 (类似raid 0)。
- **分布式条带模式**：最少需要4台服务器才能创建。 创建volume 时 stripe 2 server = 4 个节点： 是DHT 与 Striped 的组合型。
- **分布式复制模式**：最少需要4台服务器才能创建。 创建volume 时 replica 2 server = 4 个节点：是DHT 与 AFR 的组合型。
- **条带复制卷模式**：最少需要4台服务器才能创建。 创建volume 时 stripe 2 replica 2 server = 4 个节点： 是 Striped 与 AFR 的组合型。
- **三种模式混合**： 至少需要8台 服务器才能创建。 stripe 2 replica 2 , 每4个节点 组成一个 组。

因为我们只有三台主机，在此我们使用默认的**分布卷模式。请勿在生产环境上使用该模式，容易导致数据丢失。**

```
# 创建分布卷
[root@vlnx251101 ~]# gluster volume create k8s-volume transport tcp vlnx251101.zyg.com:/opt/gfs_data vlnx251102.zyg.com:/opt/gfs_data vlnx251103.zyg.com:/opt/gfs_data force

# 查看volume状态
```

```
[root@vlnx251101 ~]# gluster volume info

Volume Name: k8s-volume
Type: Distribute
Volume ID: 67ad9716-e854-4621-967b-2b9f268b6c45
Status: Created
Snapshot Count: 0
Number of Bricks: 3
Transport-type: tcp
Bricks:
Brick1: vlnx251101.zyg.com:/opt/gfs_data
Brick2: vlnx251102.zyg.com:/opt/gfs_data
Brick3: vlnx251103.zyg.com:/opt/gfs_data
Options Reconfigured:
transport.address-family: inet
nfs.disable: on

# 启动 分布卷
[root@vlnx251101 ~]# gluster volume start k8s-volume
```

Glusterfs调优

```
# 开启 指定 volume 的配额
$ gluster volume quota k8s-volume enable

# 限制 指定 volume 的配额
$ gluster volume quota k8s-volume limit-usage / 1TB

# 设置 cache 大小, 默认32MB
$ gluster volume set k8s-volume performance.cache-size 4GB
```

```
# 设置 io 线程，太大会导致进程崩溃
$ gluster volume set k8s-volume performance.io-thread-count 16

# 设置 网络检测时间，默认42s
$ gluster volume set k8s-volume network.ping-timeout 10

# 设置 写缓冲区的大小，默认1M
$ gluster volume set k8s-volume performance.write-behind-window-size 1024MB
```

Kubernetes中配置glusterfs

kubernetes安装客户端

```
# 在所有 k8s node 中安装 glusterfs 客户端
[root@vlnx251101 ~]# yum install -y glusterfs glusterfs-fuse


# 配置 hosts
[root@vlnx251101 ~]# vim /etc/hosts

192.168.251.101 vlnx251101.zyg.com vlnx251101
192.168.251.102 vlnx251102.zyg.com vlnx251102
192.168.251.103 vlnx251103.zyg.com vlnx251103
```

因为我们glusterfs跟kubernetes集群复用主机，因为此这一步可以省去。

配置 endpoints

```
# 修改 endpoints.json ，配置 glusters 集群节点ip
# 每一个 addresses 为一个 ip 组

[root@vlnx251101 glusterfs]# vim glusterfs-endpoints.json
{
  "kind": "Endpoints",
  "apiVersion": "v1",
  "metadata": {
    "name": "glusterfs-cluster"
  },
  "subsets": [
    {
      "addresses": [
        {
          "ip": "192.168.251.101"
        }
      ],
      "ports": [
        {
          "port": 1990
        }
      ]
    }
  ]
}
```

```
# 导入 glusterfs-endpoints.json

[root@vlnx251101 glusterfs]# kubectl create -f glusterfs-endpoints.json

# 查看 endpoints 信息
[root@vlnx251101 glusterfs]# kubectl get endpoints
NAME                ENDPOINTS                AGE
glusterfs-cluster   192.168.251.101:1990   8s
kubernetes          192.168.251.101:6443   3d
```

配置 service

```
# service.json 里面查找的是 enpointes 的名称与端口
[root@vlnx251101 glusterfs]# vim glusterfs-service.json
{
  "kind": "Service",
  "apiVersion": "v1",
  "metadata": {
    "name": "glusterfs-cluster"
  },
  "spec": {
    "ports": [
      {"port": 1990}
    ]
  }
}
```

```
# 导入 glusterfs-service.json
[root@vlnx251101 glusterfs]# kubectl create -f glusterfs-service.json

# 查看 service 信息
[root@vlnx251101 glusterfs]# kubectl get service
NAME                TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
glusterfs-cluster   ClusterIP   10.254.134.71   <none>        1990/TCP   8s
kubernetes          ClusterIP   10.254.0.1      <none>        443/TCP    3d
```

创建测试 pod

```
[root@vlnx251101 glusterfs]# vim glusterfs-pod.json
{
    "apiVersion": "v1",
    "kind": "Pod",
    "metadata": {
        "name": "glusterfs"
    },
    "spec": {
        "containers": [
            {
                "name": "glusterfs",
                "image": "k8s.gcr.io/pause:latest",
```

```
                "volumeMounts": [
                    {
                        "mountPath": "/mnt/glusterfs",
                        "name": "glusterfsvol"
                    }
                ]
            }
        ],
        "volumes": [
            {
                "name": "glusterfsvol",
                "glusterfs": {
                    "endpoints": "glusterfs-cluster",
                    "path": "k8s-volume",
                    "readOnly": true
                }
            }
        ]
    }
}
```

```
# 导入 glusterfs-pod.json
[root@vlnx251101 glusterfs]# kubectl create -f glusterfs-pod.json
```

```
# 查看 pods 状态
[root@vlnx251101 glusterfs]# kubectl get pods
NAME          READY     STATUS    RESTARTS   AGE
glusterfs     1/1       Running   0          1m
```

```
# 查看 pods 所在 node
[root@vlnx251101 glusterfs]# kubectl describe pod/glusterfs

# 登陆 node 物理机，使用 df 可查看挂载目录
[root@vlnx251103 ~]# df -Th

192.168.251.101:k8s-volume fuse.glusterfs   51G   32G   20G  61% /var/lib/kubelet/pods/438dab96-9f61-11e8-b407-000c29526d85/volumes/kubernetes.io~glusterfs/glusterfsvol
```

配置PersistentVolume

PersistentVolume（PV）和 PersistentVolumeClaim（PVC）是kubernetes提供的两种API资源，用于抽象存储细节。管理员关注于如何通过pv提供存储功能而无需关注用户如何使用，同样的用户只需要挂载PVC到容器中而不需要关注存储卷采用何种技术实现。

PVC和PV的关系跟pod和node关系类似，前者消耗后者的资源。PVC可以向PV申请指定大小的存储资源并设置访问模式。

PV属性

- storage容量
- 读写属性：分别为ReadWriteOnce：单个节点读写；

  ReadOnlyMany：多节点只读 ； ReadWriteMany：多节点读写

```
[root@vlnx251101 glusterfs]# vim glusterfs-pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: gluster-dev-volume
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteMany
  glusterfs:
    endpoints: "glusterfs-cluster"
    path: "k8s-volume"
    readOnly: false

# 导入PV
[root@vlnx251101 glusterfs]# kubectl create -f glusterfs-pv.yaml


# 查看 pv

[root@vlnx251101 glusterfs]# kubectl get pv
NAME                  CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM     STORAGECLASS   REASON   AGE
gluster-dev-volume    1Gi        RWX            Retain           Available                                   20s
```

PVC属性

- 访问属性与PV相同
- 容量：向PV申请的容量 <= PV总容量

配置PVC

```
[root@vlnx251101 glusterfs]# vim glusterfs-pvc.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: glusterfs-nginx
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi

# 导入 pvc
[root@vlnx251101 glusterfs]# kubectl create -f glusterfs-
pvc.yaml

# 查看 pvc
[root@vlnx251101 glusterfs]# kubectl get pvc
NAME                STATUS    VOLUME                  CAPACITY
  ACCESS MODES    STORAGECLASS    AGE
glusterfs-nginx    Bound     gluster-dev-volume
1Gi         RWX                                 11s
```

创建 nginx deployment 挂载 volume

```
[root@vlnx251101 glusterfs]# vim nginx-deployment.yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-dm
spec:
  replicas: 2
  template:
    metadata:
      labels:
        name: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
          imagePullPolicy: IfNotPresent
          ports:
            - containerPort: 80
          volumeMounts:
            - name: gluster-dev-volume
              mountPath: "/usr/share/nginx/html"
```

```
      volumes:
      - name: gluster-dev-volume
        persistentVolumeClaim:
          claimName: glusterfs-nginx
```

```
# 导入 deployment
[root@vlnx251101 glusterfs]# kubectl create -f nginx-deployment.yaml
```

```
# 查看 deployment
[root@vlnx251101 glusterfs]# kubectl get pods | grep nginx-dm
nginx-dm-867cb67894-7qcqd   1/1       Running   0          13s
nginx-dm-867cb67894-vpnd5   1/1       Running   0          13s
```

```
# 查看 挂载
[root@vlnx251101 glusterfs]# kubectl exec -it nginx-dm-867cb67894-7qcqd -- df -h | grep k8s-volume
192.168.251.101:k8s-volume   51G   32G   20G  61% /usr/share/nginx/html
```

```
# 创建文件 测试
```

```
[root@vlnx251101 glusterfs]# kubectl exec -it nginx-dm-
867cb67894-7qcqd -- touch /usr/share/nginx/html/index.html

[root@vlnx251101 glusterfs]# kubectl exec -it nginx-dm-
867cb67894-7qcqd -- ls -l
/usr/share/nginx/html/index.html
-rw-r--r-- 1 root root 0 Aug 14 01:47
/usr/share/nginx/html/index.html



# 验证 glusterfs
# 因为我们使用分布卷，所以可以看到某个节点中有文件

[root@vlnx251101 glusterfs]# ls /opt/gfs_data/
[root@vlnx251102 ~]# ls /opt/gfs_data/
index.html
[root@vlnx251103 ~]# ls /opt/gfs_data/


[root@vlnx251102 ~]# echo 123 > /opt/gfs_data/index.html


[root@vlnx251101 glusterfs]# kubectl exec -it nginx-dm-
867cb67894-7qcqd -- cat  /usr/share/nginx/html/index.html
123
```