Volume Support with GlusterFS

This will walk you through setting up GlusterFS backed by Kubernetes Persistent Volume Claims.

Get Started on GCE

More details: Labs Workbench on GKE

1. https://console.cloud.google.com and sign up for GCE (free up to $300 or 365 days)
2. Expand top-left menu and choose Container Engine > Container clusters
3. Click "Create New Cluster"
4. Select cluster parameters and click "Create"
5. Wait for your cluster to come online
6. Click "Connect" next to your cluster and copy the kubeconfig line into your clipboard
7. Open a shell using the button at the top-right
8. Paste the kubeconfig info from your clipboard

You should now be ready to start playing around with Kubernetes on GCE.

Running GlusterFS

The general process will be as follows:

1. git clone https://github.com/bodom0015/gluster -b NDS-785 && cd gluster/templates/
2. kubectl create -f claim1.json -f claim2.json
3. kubectl get pvc,pv
4. kubectl create -f gluster.svc.yaml -f gluster.rc.yaml
5. export ENDPOINT_1=$(kubectl get ep glusterfs-cluster | grep -v ENDPOINTS | awk '{print $2}' | awk 'BEGIN { FS=","; }{print $1}' | awk 'BEGIN { FS=":"; }{print $1}')
6. export ENDPOINT_2=$(kubectl get ep glusterfs-cluster | grep -v ENDPOINTS | awk '{print $2}' | awk 'BEGIN { FS=","; }{print $2}' | awk 'BEGIN { FS=":"; }{print $1}')
7. kubectl exec -it `kubectl get pod | grep glfs-server-1 | awk '{print $1}` sh
8. gluster peer probe <endpoint ip>
9. gluster volume create $VOLNAME $VOLSPEC <endpoint 1 from above>:/media/brick0 <endpoint 2 from above>:/media/brick0
10. gluster volume start global

You are now running a 2-brick replicated GlusterFS setup on top of Kubernetes.

Test Case

To test out this shared filesystem, run two busybox YAMLs mounting the same GlusterFS volume:

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: busybox1
  namespace: default
spec:
  containers:
  - image: busybox
    command:
      - sleep
      - "3600"
    volumeMounts:
    - name: "glfs"
      mountPath: "/var/glfs/global"
    imagePullPolicy: IfNotPresent
    name: busybox
  volumes:
  - name: glfs
    glusterfs:
      endpoints: glusterfs-cluster
      path: global
  restartPolicy: Always
---
apiVersion: v1
kind: Pod
metadata:
  name: busybox2
  namespace: default
spec:
  containers:
  - image: busybox
    command:
      - sleep
      - "3600"
    volumeMounts:
    - name: "glfs"
      mountPath: "/var/glfs/global"
    imagePullPolicy: IfNotPresent
    name: busybox
  volumes:
  - name: glfs
    glusterfs:
      endpoints: glusterfs-cluster
      path: global
  restartPolicy: Always
```

1. kubectl create -f busybox.yaml
2. kubectl get pods -o wide
3. kubectl exec -it busybox1 sh
4. echo 'Hello, GLFS!' >> /var/glfs/global/testing && exit
5. kubectl exec -it busybox2 cat /var/glfs/global/testing