

Persistent Data Store in OpenShift using GlusterFS



Openshift Meetup Bangalore

1

PRESENTERS

Mohamed Ashiq Liyazudeen

*Associate Software Engineer @redhat,
@ IRC:ashiq on freenode,
Github: github.com/MohamedAshiqrh,
mailto: ashiq333@gmail.com*

Humble Devassy Chirammal

*Senior Software Engineer @redhat,
@ IRC : hchiramm on freenode,
Github : github.com/humblec,
mailto : humble.devassy@gmail.com
Website: humblec.com*

Openshift Meetup Bangalore

2

AGENDA

- OpenShift
- GlusterFS
- Persistent Storage
- Persistent Volume Framework
- Lifecycle of a Persistent Volume
- Q & A

Openshift Meetup Bangalore

3

OPENSIFT

- OpenShift Origin is a distribution of Kubernetes optimized for continuous application development and multi-tenant deployment.
- Built around a core of Docker container packaging and Kubernetes container cluster management, Origin is also augmented by application lifecycle management functionality and DevOps tooling.
- Origin provides a complete open source container application platform.

Openshift Meetup Bangalore

4

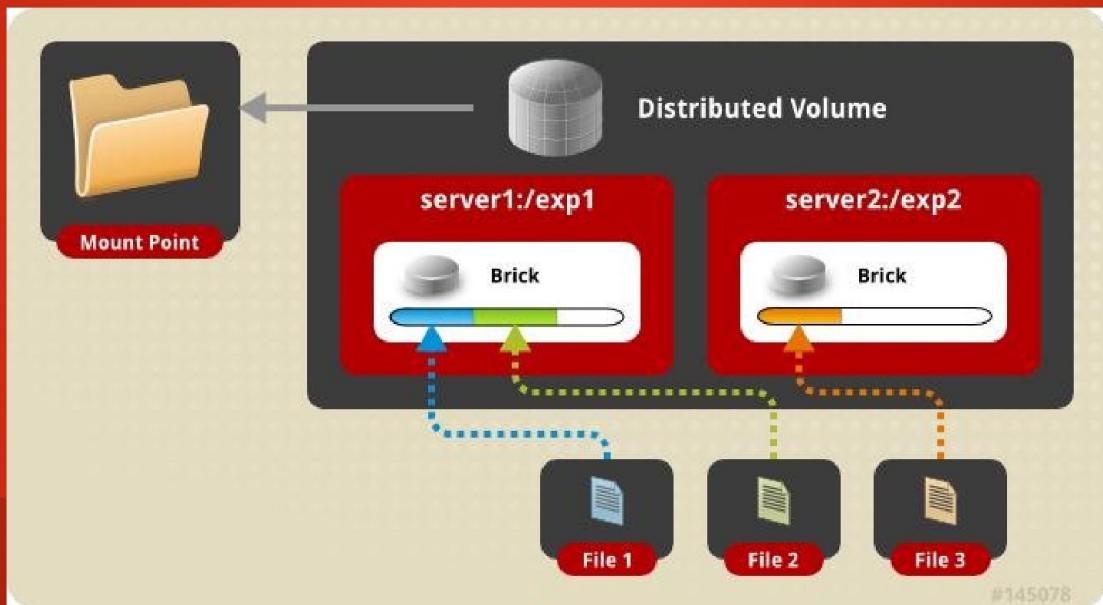
GLUSTERFS

- Open-source scale-out distributed file system.
- Aggregates various storage servers over network interconnects into one large parallel network file system.
- User space filesystem with no dedicated metadata server that helps us achieve scalability.

Openshift Meetup Bangalore

5

GLUSTERFS



Openshift Meetup Bangalore

6

PERSISTENT STORAGE

- Managing storage is a distinct problem from managing compute.
- In simple words, Containers in OpenShift Cluster need some storage which should be persistent even if the container goes down or no longer needed.

Openshift Meetup Bangalore

7

PERSISTENT VOLUME FRAMEWORK

- OpenShift leverages the Kubernetes persistent volume (PV) framework to allow administrators to provision persistent storage for a cluster.
- Using persistent volume claims (PVCs), developers can request PV resources without having specific knowledge of the underlying storage infrastructure.

Openshift Meetup Bangalore

8

PERSISTENT VOLUME(PV)

- PersistentVolume is a API object, which represents a piece of existing networked storage in the cluster that has been provisioned by an administrator.
- PVs are volume plug-ins like Volumes, but have a lifecycle independent of any individual pod that uses the PV.

Openshift Meetup Bangalore

9

PERSISTENT VOLUME CLAIM

- PersistentVolumeClaim is a API object, which represents a request for storage by a developer.
- It is similar to a pod in that pods consume node resources and PVCs consume PV resources.
- For example, pods can request specific levels of resources (e.g., CPU and memory), while PVCs can request specific storage capacity and access modes.

Openshift Meetup Bangalore

10

PV LIFECYCLE

- Provisioning – Admin creates a network storage and respective PV.
- Binding – PVC is received for the PV
- Using – PVC is used in a pod
- Releasing – Deleting PVC
- Reclaiming – Same PV for New PVC request
 - retained or recycled.

Openshift Meetup Bangalore

11

GLUSTERFS PERSISTENT VOLUME

- Service - Keeps the endpoint to be persistent or active.
- Endpoint – describes the access point of the storage(points to the GlusterFS cluster location.).
- Persistent Volume(PV) - where the administrator will define the gluster volume name, capacity of volume and access .
- Persistent Volume Claim(PVC) - where developer defines the type of storage as needed.

Openshift Meetup Bangalore

12

GLUSTERFS PERSISTENT VOLUME

- As prerequisite, all nodes in openshift cluster must have glusterfs-client installed.
- So the volume is initially mounted on the node where the pod is running and then its bind mounted to the container.

Openshift Meetup Bangalore

13

GlusterFS service

```
# cat gluster_pod/gluster-service.yaml
apiVersion: "v1"
kind: "Service"
metadata:
  name: "glusterfs-cluster"
spec:
  ports:
    - port: 1
# oc create -f gluster_pod/gluster-service.yaml
service "glusterfs-cluster" created
```

Openshift Meetup Bangalore

14

GlusterFS Endpoint

```
# cat gluster_pod/gluster-endpoints.yaml
```

```
apiVersion: v1
kind: Endpoints
metadata:
  name: glusterfs-cluster
subsets:
- addresses:
  - ip: 170.22.43.77
ports:
- port: 24007
```

```
# oc create -f gluster_pod/gluster-endpoint.yaml
```

Openshift Meetup Bangalore

15

GlusterFS PV

```
# cat gluster_pod/gluster-pv.yaml
```

```
apiVersion: "v1"
kind: "PersistentVolume"
metadata:
  name: "gluster-default-volume"
spec:
  capacity:
    storage: "8Gi"
  accessModes:
  - "ReadWriteMany"
  glusterfs:
    endpoints: "glusterfs-cluster"
    path: "gluster_vol"
    readOnly: false
  persistentVolumeReclaimPolicy: "Recycle"
```

```
# oc create -f gluster_pod/gluster-pv.yaml
```

Openshift Meetup Bangalore

16

GlusterFS PVClaim

```
# cat gluster_pod/gluster-pvc.yaml
apiVersion: "v1"
kind: "PersistentVolumeClaim"
metadata:
  name: "glusterfs-claim"
spec:
  accessModes:
    - "ReadWriteMany"
  resources:
    requests:
      storage: "8Gi"
```

```
# oc create -f gluster_pod/gluster-pvc.yaml
```

Openshift Meetup Bangalore

17

GlusterFS PVClaim

```
# oc get pv
NAME          LABELS   CAPACITY ACCESSMODES STATUS CLAIM   REASON AGE
gluster-default-volume 8Gi     RWX      Bound default/glusterfs-claim 5m
```

Openshift Meetup Bangalore

18

Attach the claim to a pod

```
# cat gluster_pod/gluster_pod.yaml
kind: Pod
apiVersion: v1
metadata:
  name: mypod
spec:
  containers:
    - name: mygluster
      image: ashiq/gluster-client
      command: ["/usr/sbin/init"]
  volumeMounts:
    - mountPath: "/home"
  name: gluster-default-volume
  volumes:
    - name: gluster-default-volume
  persistentVolumeClaim:
    claimName: glusterfs-claim
```

Openshift Meetup Bangalore

19

Verify the Claim binding!

```
# oc create -f gluster_pod/fedora_pod.yaml
pod "mypod" created
# oc get pods
NAME READY STATUS RESTARTS AGE
mypod 1/1 Running 0 1m
```

Openshift Meetup Bangalore

20

Examine the pod

```
#oc exec -it mypod /bin/bash  
[root@mypod /]# df -h | grep gluster_vol  
170.22.43.77:gluster_vol 35G 4.0G 31G 12% /home
```

Openshift Meetup Bangalore

21

Use this claim to App Pod

```
# cat gluster_pod/gluster_pod.yaml  
kind: Pod  
apiVersion: v1  
metadata:  
name: mypod  
spec:  
containers:  
- name: mygluster  
image: ashiq/gluster-client  
command: ["/usr/sbin/init"]  
volumeMounts:  
- mountPath: "/home"  
name: gluster-default-volume  
volumes:  
- name: gluster-default-volume  
persistentVolumeClaim:  
claimName: glusterfs-claim
```

Openshift Meetup Bangalore

22

Demo

Openshift Meetup Bangalore

23



Openshift Meetup Bangalore

24