Gluster in Kubernetes

Michael Adam <obnox@redhat.com>

Vault conference
2017-03-23

Persistent Storage for Containers with Gluster in Containers

(Container Native Storage)

Agenda

- Gluster
- Kubernetes
- Dynamic Provisioning with Gluster
- Demos
- Roadmap

Gluster

Gluster

- Software-defined storage
- Scale-out file storage
- Highly available
- Easy to set up
- Easy to administer
- Very flexible
- Access:
  - Native fuse POSIX file system
  - NFS
  - SMB
  - iscsi (on file) (new)
  - Object: S3 / swift via gluster-swift (new)

Gluster

- https://gluster.org
- https://github.com/gluster

Gluster

- Volumes composed of local FS directories (bricks)
- Different “durability” types: *replicate*, *distribute*, *disperse* (ec), …
- Flexibility and feature-richness due to architecture of a stack of translators
- Example of features:
  - Encryption
  - Snapshots (user-serviceable)
  - Geo-replication
  - Quota
  - …
- Layout of multiple daemons for bricks, glusterd, quota, ...

Storage use cases in OpenShift

- Persistent storage for application containers
- Registry

Kubernetes

Kubernetes

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

- https://kubernetes.io
- Containers (docker)
- Orchestration / deployment / scaling
- Cluster
- “Apps” (applications)
- Flavor: OpenShift (distribution) origin / Red Hat OpenShift Container Platform

Kubernetes and Storage

- Containers: stateless, ephemeral in nature
  - Bringing up and down loses state
- Apps need persistent storage:
  - Configuration
  - Application data (websites…)
  - Databases …
- Storage needs to be available on all (kubernetes) nodes

Kubernetes - lingo and concepts

- pod: group of one or more containers that form an entity, smallest unit
- persistent volume (PV): to be mounted by application pod
- provisioner: to provide PVs upon request
- mount plugin: mechanism to mount the PV, referenced in PV
- persistent volume claim (PVC): mechanism for a user to request a PV
- Access types for volumes:
  - RWO - read write once (single node)
  - RWX - read write many (multiple nodes)
  - ROX - read only many (multiple nodes)
- flavors of provisioning: dynamic and static

Dynamic Provisioning (since 1.4) - in general

- a storage class (SC):
  - Created by admin
  - describes the storage
  - references a (dynamic) provisioner
- PVC (by user): references SC
- provisioner from SC: creates PV of requested size / type / …
- PV is bound to PVC
- user can mount the PV (by PVC) in application pod

Dynamic Provisioning with Gluster

Components

- Kubernetes
  - dynamic GlusterFS provisioner
  - GlusterFS mount plugin
- Heketi
  - high-level service interface for gluster volume lifecycle management
- Gluster:
  - one or more glusterfs clusters
  - running hyper-converged in Kubernetes (“container native storage”)
  - Can also run externally
- Gk-deploy:
  - tool to deploy gluster and heketi into an existing Kubernetes cluster

PV Creation: glusterfs dynamic provisioner

- PVC (created by user) references the glusterfs provisioner
  - glusterfs provisioner extracts details from PVC
  - provisioner tells heketi to create a volume of given size and type
    - heketi looks for a gluster cluster that can satisfy this request
    - if found, heketi tells the gluster instance to create the volume
      - gluster creates a volume
    - Heketi hands volume back to provisioner
  - provisioner creates PV and puts the gluster volume details into it
  - provisioner puts glusterfs as the mount plugin into the PV
  - Provisioner returns PV to the caller
- PVC is bound to the PV and can later be used in a pod by the user

GlusterFS mount plugin

- the OpenShift HOST has glusterfs-client installed
- the host mounts the gluster volume
- the gluster mount of the host is bind-mounted into the application container

About heketi

- high-level service interface for managing the lifecycle of gluster volumes
- RESTful API and cli ("heketi-cli")
- manages one or several gluster clusters
- can create, expand, delete volumes (more coming)
- hides nitty gritty details of volume creation from caller
- just takes size and desired durability type
    - (currently only replicate is supported in CNS)
- automatically finds cluster and disks to satisfy the request
- stores its state in a database (currently Bolt)
- https://github.com/heketi/heketi

WARNING

In a heketi-managed cluster, don’t mess with the volumes manually!

(will be removed in future version…)

About the heketi container

- single container
- can move in the cluster
- database needs to be persisted
  - ⇒ currently stored in a gluster volume

About the gluster containers

- Privileged
- Use disks from host
- Use network from host
- Tied to the nodes
- DaemonSet

How to set it all up? gk-deploy

- Set it all up in a single command
- project / community: https://github.com/gluster/gluster-kubernetes
- takes topology file to describe disk devices, gluster nodes and heketi
- deploys the gluster cluster (upon request)
  - gluster is deployed as a DaemonSet
- deploys heketi pod

Demos

Demos

- gk-deploy: https://asciinema.org/a/5apn5yv7rryqa0hpjozq0s06v
- Heketi: https://asciinema.org/a/9cluxpf9weuyq6oqhmd3v7r0c
- DP: https://asciinema.org/a/amyldm9lp8sxfqc89eogymx0x

Roadmap

Roadmap

- 1.5
  - GlusterFS as registry backend (OpenShift)
  - Improved day-2-day maintenance (remove disk …)
- 1.6
  - Improved RWO support with gluster-block provisioner (iscsi)
  - Scalability improvements
- 1.7+
  - Support for S3-object access from pods
  - Possibly Gluster with S3 as improved backend for registry

Questions?

More Questions? ⇒ Red Hat booth

Michael Adam <obnox@redhat.com>