

Workbench storage redux

Once again, we're talking about the storage model in Workbench



NDS-776 - Design persistent volume claim support for shared storage

RESOLVED

. As of today, we're still running a very old Kubernetes (1.5.x) deployed via our obscure deploy-tools ansible process with a home-grown, in-cluster GlusterFS mounted to worker nodes and pods via hostPath. A long standing feature/requirement for Workbench has been the idea of a shared home directory across multiple pods/nodes for each user, which requires the ability to mount the filesystem read/write to many containers. The Kubernetes storage architecture, particularly support for persistent volume claims, is generally supported for read-write-once volumes.

With the advent of the Zonca method for deploying JupyterHub over Rook, we revisit Rook (Starting with 0.6). This at first looked quite promising – a straightforward helm based install provided near immediate support for PVCs (for the single-use volume scenario). Shared filesystem support proved less exciting. We spent cycles troubleshooting what amounts to an unclear setting in the examples (metadataPool replication number needs to match the number of storage nodes). There's also the 4.7 kernel requirement, which is a good thing, but multiple shared filesystems are still experimental. After some initial investigation we're back to the question of whether these complex methods are really any better than an NFS fileserver.

Here's what we know:

- We have Rook working via the Zonca method effectively for JupyterHub
 - <https://arxiv.org/pdf/1805.04781.pdf>
- We now have Rook shared storage via flexVolume thanks to Mike's work and the [Rook in Kubernetes#SharedStorageExample](#)
 - But this caused some hair pulling, and realization that under the covers Ceph is perhaps more complex than Gluster
- We know that AWS EFS supports NFSv4 (and probably SMB)
 - EFS appears to be the only ReadWriteMany storage option with PVC support
- We know that GCE Filers support NFS, SMB and Gluster
 - GCE has no built-in read-write-many storage option for Kubernetes
 - <https://cloud.google.com/solutions/filers-on-compute-engine>
- We know that AzureDisk supports SMB
- We've discussed two broad options
 - Try and get rook running everywhere (OpenStack, AWS, GCE, Azure)
 - Replace current GlusterFS with NFS and rely on existing file servers implementations

Getting rook running everywhere

- This is no problem for Openstack, but would require some work/thinking around how to provision different instance groups with attached storage, etc.

Using NFS everywhere:

- Should work for AWS via EFS or GCE via file server. Unclear whether NFS will work with AzureDisk.
 - See also GCE example for Jupyterhub using filer <https://cloud.google.com/solutions/using-tensorflow-jupyterhub-classrooms>
- Will require work to deploy fileserver via terraform for OpenStack.

- Things we should do in next 6 mos
- ISS
 - GCE / bus
 - Focus on ISDA
 - Focus on Earthquake live

- Review
- Lead meeting
 - private git repo
 - Flack + Rinc.

EC2 ← EFS /efs
/efs/ userfiles-pvc/...

1. Always NFS (except maybe Azure)

2. Provision NFS Server
 Provision Nfs server
 gluster server - NFS
 EFS...
 GCE...
 ? route filesystem → NFS

provision NFS
 - manual
 - terraform
 nfs server
 to talk NFS
 Test on AWS

Volumes:
 -PVC

POD

Volume Driver

Volumes:
 FlexVolume
 NFS
 ;

