

# Workbench storage

This page documents the preliminary requirements and design for volume management in the NDS Labs Workbench system.

## See also

- <https://github.com/nds-org/gluster>
- <https://opensource.ncsa.illinois.edu/jira/browse/NDS-186> (first comment)

## Requirements

- **Kubernetes support:** Since the workbench is based on Kubernetes, storage options must be supported or supportable in the Kubernetes framework. This means that the storage can be mounted in a Pod/container.
  - Support standard Kubernetes volume semantics and practices for multi-tenant clusters
- **Distributed:** Storage solution must support distribution across multiple nodes.
- **Quotas:** Storage solution must support per-project quotas
- **Scalability:** Storage solution must be scalable
- **Reliability:** Storage solution must be reliable, or support a balance in the reliability/performance trade-off
- **Performance:** Storage solution must be performant, or support a balance in the performance/reliability trade-off
- **Security:** Security between groups/users/projects in storage should follow standard practices and policies - i.e. mirror standard linux user/group models

## Storage options

We have explored the following storage options:

Storage	Pros	Cons
Local storage	Supported by Kubernetes  Simple  Useful for single-node or development configurations	Not distributed  Not scalable  Does not support quotas
NFS	Supported by Kubernetes  Distributed	Not scalable  Performance requires tuning
OpenStack Cinder	Supportable by Kubernetes (with changes)	Requires change to Kubernetes  Limits potential run-times to OpenStack
GlusterFS	Supported by Kubernetes for in-pod storage  Scalable and adaptable - growth, performance, and reliability	Not supported directly on cloud-oriented OS's

## Creating a GlusterFS server cluster

The basic steps for creating a GlusterFS cluster on Nebula include:

1. Allocate GlusterFS server VMs
2. Allocate OpenStack volumes, one volume per GFS node (e.g., 500GB per node)
3. For each VM/volume
  - a. Initialize physical volume
  - b. Create volume group, logical volumes, and bricks.
  - c. Mount bricks to server
4. Start the gluster server container
  - a. Initialize gluster FS servers
  - b. Create gluster volumes
5. Mount volumes to GFS VM
6. Configure storage: physical volumes, volume group, logical volumes, bricks

## Managing volumes for projects:

Currently, we are planning to allocate one volume per project, but we should anticipate the need for per-pod as it will likely be needed. Volumes will be expandable by the admin and deleted by the admin tools when the project is deleted

- Create volume:
  - gluster create
  - gluster start
- Delete volume
  - gluster stop
  - gluster delete
- Expanding volume
  - ?

## Expanding cluster-wide storage:

Cluster-wide storage is expanded by adding allocating, formatting, and configuring additional physical volumes.

## Quotas

With per-project volumes, quotas will be enforced by the volume size

## Use cases:

- Site administrator can deploy (and re-deploy) cluster storage system
- Cluster administrator can create, expand, and delete per-project volumes
- Project administrators can deploy services that use project storage
- Project administrators can view project storage quotas and usage
- Project administrators can request storage increases