

```

# Invocation : <prog name> <volume name>
#
import xml.etree.ElementTree as ETree
import sys
from subprocess import Popen, PIPE

glfsCmd = Popen(['gluster',
                'vol',
                'status',
                sys.argv[1],
                'detail',
                '--xml'], stdout=PIPE)

# cmdOut will be a string object
cmdOut = glfsCmd.communicate()[0]

# Parse the string, making an xml object
xmlRoot = ETree.fromstring(cmdOut)

# Return a list of 'sizeTotal' elements
brickSize = xmlRoot.findall('./sizeTotal')

# Return a list of 'sizeFree' elements
brickFree = xmlRoot.findall('./sizeFree')

# Just count the number of 'path' elements in
# the XML to indicate the number of bricks in the
# volume
numBricks = len(xmlRoot.findall('./path'))

# Loop through each brickSize element, forming a
# new list of values, that are then sum'd
rawTotal = sum([float(thisBrick.text) for thisBrick in
brickSize])

rawFree = sum([float(thisBrick.text) for thisBrick in
brickFree])

pctUsed = ((rawTotal-rawFree)/rawTotal)

print "\nVolume Name: " + sys.argv[1]
print "Number of Bricks %5d" % (numBricks)
print "Raw Volume Size %5.02f (GB)" %
(rawTotal/1024**3)
print "Raw Free %5.02f (GB)" %
(rawFree/1024**3)
print "%5 Used %5.02f\n" % (pctUsed)

```

TroubleShooting

Gluster uses the following log locations to record events and activity within the cluster

geo-replication

/var/log/glusterfs/geo-replication/*

self-heal operations

/var/log/glusterfs/glustershd.log

NFS access

/var/log/glusterfs/nfs.log

Upgrading to v3.4

Your Version	Upgrade Overview*
3.3	"Cold", or rolling upgrade supported. Once complete, an upgrade of all native clients is recommended.
3.x	Downtime is required due to changes in the location of config files and xlators.

*Further detail is available on gluster.org

Recommended Configuration Limits

Max Number of peers in a cluster	64
Clients per Volume	1000
Max Bricks per Node	4
Max bricksize (TB)	100

Useful Links

Web

<http://www.gluster.org>
<http://forge.gluster.org>

IRC Channel

<irc.gnu.org#gluster>

Mail Lists

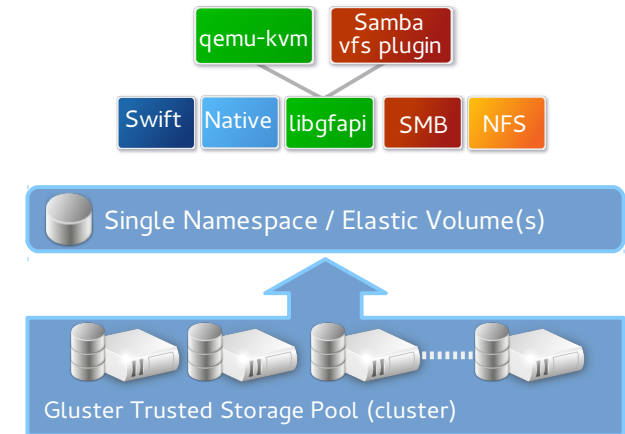
gluster-users@gluster.org
gluster-devel@nongnu.org



DRAFT

Gluster 3.4 Pocket Reference

Architectural Overview



Configuration Overview



Managing Cluster Membership

Adding a node

```
peer probe <node name>
```

Removing a node

```
peer detach <node name>
```

Querying status of the cluster (2 options)

```
peer status
pool list
```

Configuring Bricks

Bricks should be configured with the LVM for future flexibility and enhanced management. The following steps prepare an empty disk for use as a gluster brick, using /dev/sdb as an example device.

1. pvcreate /dev/sdb
2. vgcreate <vg_name> /dev/sdb
3. lvcreate -n <lv_name> -l 100%PVS \
<vg_name> /dev/sdb
4. mkfs.xfs -i size=512 <lv_path>

Once the LV is prepared, update fstab. Additional steps may be necessary if the disk device is a RAID LUN to ensure the device is aligned with the geometry of the underlying RAID group.

Managing Volumes

The process for creating a volume

1. Ensure bricks are available
2. vol create <vol-name> ...
3. vol set <vol-name> <key> <value>
4. vol start <vol-name>

Use "vol help" for the complete syntax

Expanding a volume

Distributed volumes may be expanded by any number of bricks, but replicated volumes must be expanded in units of the replication factor (i.e. if volume is a replica 2, expansion must be in multiples of 2 bricks/nodes)

```
> vol add-brick <vol-name> <brick list>
```

Shrinking a volume

To remove bricks from a volume you must use the 'start' parameter to avoid data loss!

```
> vol remove-brick <vol-name> <brick> start
```

Server Mount Options (fstab)

Filesystem	Option
xfs	allocsize=4096,inode64,logbufs=8,lqgbsize=256K,noatime

Client Mount Options (fstab)

fstype	Option	Req'd
glusterfs	_netdev	•
	backupvolfile-server=<node>	
cifs	_netdev,credentials=<file>	•
nfs	_netdev,vers=3,proto=tcp	•

Common Tuning Options

The following parameters are set via;

```
volume set <vol name> <key> <value>
```

Key	Value and Action
nfs.disable	'on' turns NFS off
auth.allow or auth.reject	Supply IP addresses to permit or explicitly deny access to a volume
Cluster.min-free-disk	% of free space to maintain across bricks
network.ping-timeout	Secs to wait before a node is declared 'dead'
user.cifs	'disable' turns Samba off
storage.owner-uid x or storage.owner-gid x	Where 'x' is 36 ... oVirt/RHEV 161 ... OpenStack Glance 165 ... OpenStack Cinder
cluster.eager-lock	on off Set to on to optimise lock useful for high write workloads

Using xattr's

Which bricks is my file stored on?

```
getfattr -d -e text -m . -n \<br>trusted.glusterfs.pathinfo <file_path_name>
```

Reusing a brick (after the volume is deleted)

```
setattr -x trusted.gfid <brick path><br>setfattr -x trusted.glusterfs.volume-id \ <brick path>
```

Cross Protocol Data Access

Although a gluster trusted pool can be configured to provide support for multiple protocols simultaneously, data access across these protocols has restrictions due to the locking semantics inherent in each protocol. The table below defines which protocols can be used concurrently on the same volume.

	SMB	NFS	Native	Object
SMB		✘	✔	✘
NFS	✘		✔	✘
Native	✔	✔		✔
Object	✘	✘	✔	

Scripting

Version 3.4 introduced the ability to generate command output in xml format using the --xml parameter. The example below shows how this can be used from python to form the basis of server side automation scripts.

The example below uses the python xml.etree.ElementTree module to parse the output. In this instance, the program prints the raw capacity values (size, used, pct used), with the brick count for a given volume.

```
#!/usr/bin/env python<br># Run the gluster command natively first to<br># understand the xml layout.<br>#
```