363.1 GlusterFS Storage Clusters (weight: 5)

Weight	5
Description	Candidates should be able to manage and maintain a GlusterFS storage cluster.

Key Knowledge Areas:

- Understand the architecture and components of GlusterFS
- Manage GlusterFS peers, trusted storge pools, bricks and volumes
- Mount and use an existing GlusterFS
- Configure high availability aspects of GlusterFS
- Scale up a GlusterFS cluster
- Replace failed bricks
- Recover GlusterFS from a physical media failure
- Restore and verify the integrity of a GlusterFS cluster after an outage
- Awareness of GNFS (Ganesha NFS)

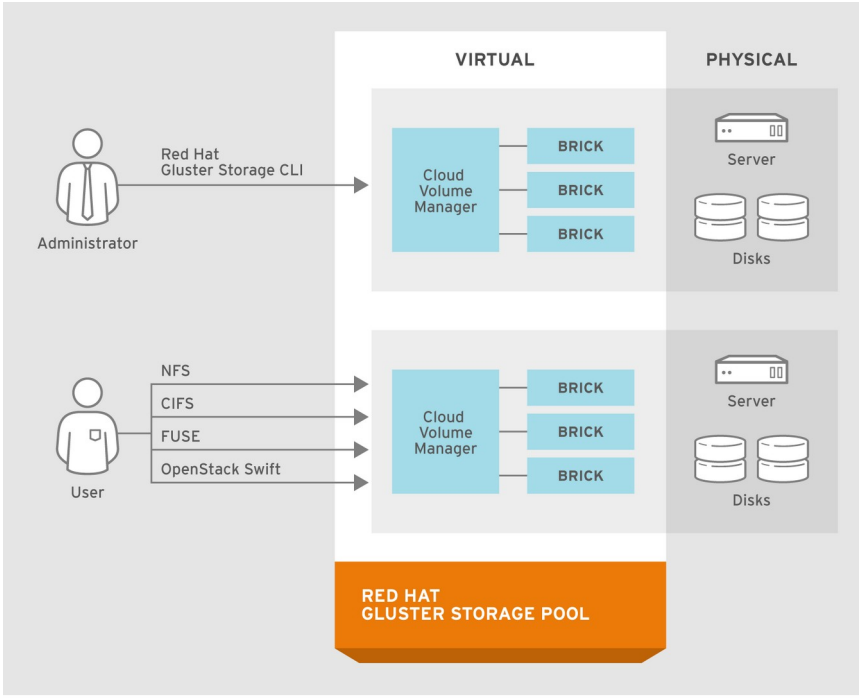
Partial list of the used files, terms and utilities:

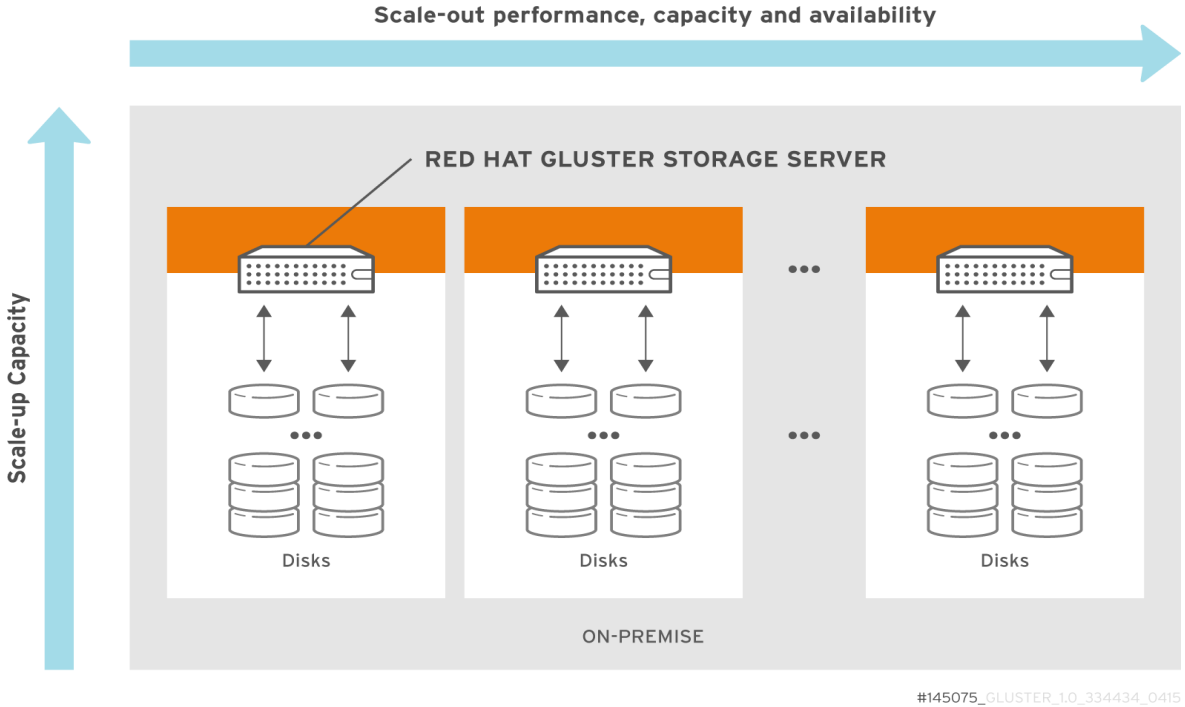
- gluster (including relevant subcommands)

→→ Conceptos Preliminares.

Arquitectura GlusterFS.

En el núcleo del diseño de Red Hat Gluster Storage se encuentra un método completamente nuevo de diseñar la arquitectura del almacenamiento. El resultado es un sistema que tiene una inmensa escalabilidad, es altamente resistente y ofrece extraordinarias actuación. En un sistema de escalamiento horizontal, uno de los mayores desafíos es realizar un seguimiento de las ubicaciones lógicas y físicas de datos y metadatos. La mayoría de los sistemas distribuidos resuelven este problema creando un servidor de metadatos para rastrear la ubicación de datos y metadatos. A medida que los sistemas tradicionales agregan más archivos, más servidores o más discos, el servidor central de metadatos se convierte en un cuello de botella de rendimiento, así como un punto central de falla. A diferencia de otras soluciones de almacenamiento tradicionales, Red Hat Gluster Storage no necesita un servidor de metadatos, y localiza archivos algorítmicamente usando un algoritmo de hash elástico. Esta arquitectura de servidor sin metadatos asegura un mejor rendimiento, escalabilidad lineal y confiabilidad.


VIRTUAL
PHYSICAL
Red Hat
Gluster Storage CLI
Administrator
Cloud
Volume
Manager
BRICK
BRICK
BRICK
Server
Disks
NFS
CIFS
FUSE
OpenStack Swift
User
Cloud
Volume
Manager
BRICK
BRICK
BRICK
Server
Disks
RED HAT
GLUSTER STORAGE POOL
#153460_GLUSTER_1.0_334434_0415


Scale-out performance, capacity and availability
Scale-up Capacity
RED HAT GLUSTER STORAGE SERVER
Disks
Disks
Disks
ON-PREMISE
#145075_GLUSTER_1.0_334434_0415

Conceptos de Almacenamiento.

→ Brick

The glusterFS basic unit of storage, represented by an export directory on a server in the trusted storage pool. A brick is expressed by combining a server with an export directory in the following format:

SERVER:EXPORT

→ Volume

A volume is a logical collection of bricks. Most of the Red Hat Gluster Storage management operations happen on the volume.

→ Translator

A translator connects to one or more subvolumes, does something with them, and offers a subvolume connection.

→ Subvolume

A brick after being processed by at least one translator.

→ Volfile

Volume (vol) files are configuration files that determine the behavior of your Red Hat Gluster Storage trusted storage pool. At a high level, GlusterFS has three entities, that is, **Server**, **Client** and **Management daemon**. Each of these entities have their own volume files. Volume files for servers and clients are generated by the management daemon upon creation of a volume.

Server and Client Vol files are located in **/var/lib/glusterd/vols/VOLNAME** directory.

The management daemon vol file is named as **glusterd.vol** and is located in **/etc/glusterfs/** directory.

→ glusterd

glusterd is the glusterFS Management Service that must run on all servers in the trusted storage pool.

→ Cluster

A trusted pool of linked computers working together, resembling a single computing resource. In Red Hat Gluster Storage, a cluster is also referred to as a trusted storage pool.

→ Client

The machine that mounts a volume (this may also be a server).

→ File System

A method of storing and organizing computer files. A file system organizes files into a database for the storage, manipulation, and retrieval by the computer's operating system.

→ Distributed File System

A file system that allows multiple clients to concurrently access data which is spread across servers/bricks in a trusted storage pool. Data sharing among multiple locations is fundamental to all distributed file systems.

→ Virtual File System (VFS)

VFS is a kernel software layer that handles all system calls related to the standard Linux file system. It provides a common interface to several kinds of file systems.

→ POSIX

Portable Operating System Interface (for Unix) (POSIX) is the name of a family of related standards

specified by the IEEE to define the application programming interface (API), as well as shell and utilities interfaces, for software that is compatible with variants of the UNIX operating system. Red Hat Gluster Storage exports a fully POSIX compatible file system.

→ Metadata

Metadata is data providing information about other pieces of data.

→ FUSE

Filesystem in User space (FUSE) is a loadable kernel module for Unix-like operating systems that lets non-privileged users create their own file systems without editing kernel code. This is achieved by running file system code in user space while the FUSE module provides only a "bridge" to the kernel interfaces.

→ Geo-Replication

Geo-replication provides a continuous, asynchronous, and incremental replication service from one site to another over Local Area Networks (LAN), Wide Area Networks (WAN), and the Internet.

→ N-way Replication

Local synchronous data replication that is typically deployed across campus or Amazon Web Services Availability Zones.

→ Petabyte

A petabyte is a unit of information equal to one quadrillion bytes, or 1000 terabytes. The unit symbol for the petabyte is PB. The prefix peta- (P) indicates a power of 1000:

1 PB = 1,000,000,000,000,000 B = 1000^5 B = 10^15 B.

The term "pebibyte" (PiB), using a binary prefix, is used for the corresponding power of 1024.

→ RAID

Redundant Array of Independent Disks (RAID) is a technology that provides increased storage reliability through redundancy. It combines multiple low-cost, less-reliable disk drives components into a logical unit where all drives in the array are interdependent.

→ RRDNS

Round Robin Domain Name Service (RRDNS) is a method to distribute load across application servers. RRDNS is implemented by creating multiple records with the same name and different IP addresses in the zone file of a DNS server.

→ Server

The machine (virtual or bare metal) that hosts the file system in which data is stored.

→ Block Storage

Block special files, or block devices, correspond to devices through which the system moves data in the form of blocks. These device nodes often represent addressable devices such as hard disks, CD-ROM drives, or memory regions. As of Red Hat Gluster Storage 3.4 and later, block storage supports only OpenShift Container Storage converged and independent mode use cases. Block storage can be created and configured for this use case by using the gluster-block command line tool. For moreinformation, see Container-Native Storage for OpenShift Container Platform.

→ Scale-Up Storage

Increases the capacity of the storage device in a single dimension. For example, adding additional disk capacity in a trusted storage pool.

→ Scale-Out Storage

Increases the capability of a storage device in single dimension. For example, adding more systems of the same size, or adding servers to a trusted storage pool that increases CPU, disk capacity, and throughput for the trusted storage pool.

→ Trusted Storage Pool

A storage pool is a trusted network of storage servers. When you start the first server, the storage pool consists of only that server.

→ Namespace

An abstract container or environment that is created to hold a logical grouping of unique identifiers or symbols. Each Red Hat Gluster Storage trusted storage pool exposes a single namespace as a POSIX mount point which contains every file in the trusted storage pool.

→ User Space

Applications running in user space do not directly interact with hardware, instead using the kernel to moderate access. User space applications are generally more portable than applications in kernel space. glusterFS is a user space application.

→ Hashed subvolume

A Distributed Hash Table Translator subvolume to which the file or directory name is hashed to.

→ Cached subvolume

A Distributed Hash Table Translator subvolume where the file content is actually present. For directories, the concept of cached-subvolume is not relevant. It is loosely used to mean subvolumes which are not hashed-subvolume.

→ Linkto-file

For a newly created file, the hashed and cached subvolumes are the same. When directory entry operations like rename (which can change the name and hence hashed subvolume of the file) are performed on the file, instead of moving the entire data in the file to a new hashed subvolume, a file is created with the same name on the newly hashed subvolume. The purpose of this file is only to act as a pointer to the node where the data is present. In the extended attributes of this file, the name of the cached subvolume is stored. This file on the newly hashed-subvolume is called a linkto-file. The linkto file is relevant only for non-directory entities.

→ Directory Layout

The directory layout helps determine where files in a gluster volume are stored.

When a client creates or requests a file, the DHT translator hashes the file's path to create an integer.

Each directory in a gluster subvolume holds files that have integers in a specific range, so the hash

of any given file maps to a specific subvolume in the gluster volume. The directory layout determines which integer ranges are assigned to a given directory across all subvolumes.

Directory layouts are assigned when a directory is first created, and can be reassigned by running a rebalance operation on the volume. If a brick or subvolume is offline when a directory is created, it will not be part of the layout until after a rebalance is run. You should rebalance a volume to recalculate its directory layout after bricks are added to the volume.

→ Fix Layout

A command that is executed during the rebalance process.

The **rebalance** process itself comprises of two stages:

1. Fixes the layouts of directories to accommodate any subvolumes that are added or removed. It also heals the directories, checks whether the layout is non-contiguous, and persists the layout in extended attributes, if needed. It also ensures that the directories have the same attributes across all the subvolumes.

2. Migrates the data from the cached-subvolume to the hashed-subvolume.

→ Topología

 # cat /etc/hosts

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
## Servers glusterFS.
192.168.10.161      gluster01.cadilinea.lan      gluster01
192.168.10.162      gluster02.cadilinea.lan      gluster02
192.168.10.163      gluster03.cadilinea.lan      gluster03
```

 # lsblk /dev/vd[b-z]

```
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vdb  252:16   0  20G  0 disk
vdc  252:32   0  20G  0 disk
vdd  252:48   0  20G  0 disk
```

→ Sincronización Horaria.

Todos los hosts deben estar sincronizados previamente → NTP chronyc por ejemplo.

 # systemctl status chronyd.service

→ Instalación.

En todos los nodos:

 # dnf install glusterfs-server -y

 # systemctl enable --now glusterd

→ bash-completion para gluster

cd /etc/bash_completion.d/

./gluster

(Es necesario reiniciar el Sistema).

 # gluster --version

glusterfs 8.4

Repository revision: git://git.gluster.org/glusterfs.git

Copyright (c) 2006-2016 Red Hat, Inc. <https://www.gluster.org/>

GlusterFS comes with ABSOLUTELY NO WARRANTY.

It is licensed to you under your choice of the GNU Lesser

General Public License, version 3 or any later version (LGPLv3

or later), or the GNU General Public License, version 2 (GPLv2),

in all cases as published by the Free Software Foundation.

 # firewall-cmd --permanent --add-service=glusterfs

 # firewall-cmd --reload

 # mkdir /bricks/repVol-01/ -p

 # mkdir /mnt/repVol-01/ -p

En el nodo gluster01:

gluster01 ~ # gluster peer probe gluster02

peer probe: success

gluster01 ~ # gluster peer probe gluster03

peer probe: success

gluster01 ~ # gluster peer status

Number of Peers: 2

Hostname: gluster02

Uuid: 9812eb50-35dc-4527-99d7-f7ba56321cf6

State: Peer in Cluster (Connected)

Hostname: gluster03

Uuid: 57e79ea9-9dc6-4ee2-8337-c4a20fc30d29

State: Peer in Cluster (Connected)

==> **Volúmenes en Réplica** → **NO es Necesario Rebalancear**

gluster01 ~ # gluster volume create repVol-01 replica 3 gluster01:/bricks/repVol-01/data gluster02:/bricks/repVol-01/data gluster03:/bricks/repVol-01/data force

volume create: repVol-01: success: please start the volume to access data

gluster01 ~ # gluster volume start repVol-01

volume start: repVol-01: success

gluster01 ~ # gluster volume info

Volume Name: repVol-01

Type: Replicate

Volume ID: e0c050f8-7876-45c6-80ed-cd6a4a10a5a6

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 3 = 3

Transport-type: tcp

Bricks:

Brick1: gluster01:/bricks/repVol-01/data

Brick2: gluster02:/bricks/repVol-01/data

Brick3: gluster03:/bricks/repVol-01/data

Options Reconfigured:

storage.fips-mode-rchecksum: on

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

→ **Montaje de glusterfs (Cliente):**

gluster01 ~ # mount.glusterfs gluster01:/repVol-01 /mnt/repVol-01/

gluster01 ~ # df -hT

S.ficheros Tipo Tamaño Usados Disp Uso% Montado en

…

gluster01:/repVol-01 fuse.glusterfs 15G 2,5G 13G 17% /mnt/repVol-01/

gluster01 ~ # touch /mnt/repVol-01/Desde_gluster01.txt

Probamos en otro nodo:

gluster02 ~ # mount.glusterfs gluster02:/repVol-01 /mnt/repVol-01

gluster02 ~ # ls /mnt/

Desde_gluster01.txt

→ **Eliminamos nodo → gluster03**

gluster01 ~ # gluster volume info repVol-01

Volume Name: repVol-01

Type: Replicate

Volume ID: e0c050f8-7876-45c6-80ed-cd6a4a10a5a6

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 3 = 3

Transport-type: tcp

Bricks:

Brick1: gluster01:/bricks/repVol-01/data

Brick2: gluster02:/bricks/repVol-01/data

Brick3: gluster03:/bricks/repVol-01/data

Options Reconfigured:

storage.fips-mode-rchecksum: on

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

gluster01 ~ # gluster volume set repVol-01 cluster.force-migration disable

volume set: success

gluster01 ~ # gluster volume info repVol-01

Volume Name: repVol-01

Type: Replicate

Volume ID: e0c050f8-7876-45c6-80ed-cd6a4a10a5a6

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 3 = 3

Transport-type: tcp

Bricks:

Brick1: gluster01:/bricks/repVol-01/data

Brick2: gluster02:/bricks/repVol-01/data

Brick3: gluster03:/bricks/repVol-01/data

Options Reconfigured:

cluster.force-migration: disable

storage.fips-mode-rchecksum: on

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

gluster01 ~ # gluster volume remove-brick repVol-01 replica 2 gluster03:/bricks/repVol-01/data start

Replica 2 volumes are prone to split-brain. Use Arbiter or Replica 3 to avoid this. See: http://docs.gluster.org/en/latest/Administrator%20Guide/Split%20brain%20and%20ways%20to%20deal%20with%20it/.

Do you still want to continue?

(y/n) **y**

It is recommended that remove-brick be run with cluster.force-migration option disabled to prevent possible data corruption. Doing so will ensure that files that receive writes during migration will not be migrated and will need to be manually copied after the remove-brick commit operation. Please check the value of the option and update accordingly.

Do you want to continue with your current cluster.force-migration settings? (y/n) y

volume remove-brick start: failed: Migration of data is not needed when reducing replica count. Use the 'force' option

gluster01 ~ # gluster volume remove-brick repVol-01 replica 2 gluster03:/bricks/repVol-01/data force

Remove-brick force will not migrate files from the removed bricks, so they will no longer be available on the volume.

Do you want to continue? (y/n) **y**

volume remove-brick commit force: success

gluster01 ~ # gluster volume info repVol-01

Volume Name: repVol-01

Type: Replicate

Volume ID: e0c050f8-7876-45c6-80ed-cd6a4a10a5a6

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 2 = 2

Transport-type: tcp

Bricks:

Brick1: gluster01:/bricks/repVol-01/data

Brick2: gluster02:/bricks/repVol-01/data

Options Reconfigured:

cluster.force-migration: disable

storage.fips-mode-rchecksum: on

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

→ **Añadimos de nuevo el nodo** → **gluster03**

gluster01 ~ # gluster volume add-brick repVol-01 replica 3 gluster03:/bricks/repVol-01/data force

volume add-brick: success

gluster01 ~ # gluster volume info repVol-01

Volume Name: repVol-01

Type: Replicate

Volume ID: e0c050f8-7876-45c6-80ed-cd6a4a10a5a6

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 3 = 3

Transport-type: tcp

Bricks:

Brick1: gluster01:/bricks/repVol-01/data

Brick2: gluster02:/bricks/repVol-01/data

Brick3: gluster03:/bricks/repVol-01/data

Options Reconfigured:

cluster.force-migration: disable

storage.fips-mode-rchecksum: on

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

→ Comprobamos su salud, y posible split-brain → heal

gluster01 ~ # gluster volume heal repVol-01

Launching heal operation to perform index self heal on volume repVol-01 has been successful

Use heal info commands to check status.

gluster01 ~ # gluster volume heal repVol-01 info

Brick gluster01:/bricks/repVol-01/data

Status: Connected

Number of entries: 0

Brick gluster02:/bricks/repVol-01/data

Status: Connected

Number of entries: 0

Brick gluster03:/bricks/repVol-01/data

Status: Connected

Number of entries: 0

gluster01 ~ # gluster volume heal repVol-01 info split-brain

Brick gluster01:/bricks/repVol-01/data

Status: Connected

Number of entries in split-brain: 0

Brick gluster02:/bricks/repVol-01/data

Status: Connected

Number of entries in split-brain: 0

Brick gluster03:/bricks/repVol-01/data

Status: Connected

Number of entries in split-brain: 0

→ **Ajustes 'split-brain', y otros recomendados** → **'repVol-01'** → **'gluster volume info repVol-01'**

gluster01 ~ # gluster volume set repVol-01 features.barrier-timeout 2m

gluster01 ~ # gluster volume set repVol-01 features.uss enable

gluster01 ~ # gluster volume set repVol-01 cluster.self-heal-daemon on

gluster01 ~ # gluster volume set repVol-01 cluster.entry-self-heal on

gluster01 ~ # gluster volume set repVol-01 cluster.metadata-self-heal on

gluster01 ~ # gluster volume set repVol-01 cluster.data-self-heal on

gluster01 ~ # gluster volume bitrot repVol-01 enable

gluster01 ~ # gluster volume set repVol-01 features.barrier enable

gluster01 ~ # gluster volume set repVol-01 auth.allow 192.168.10.*

gluster01 ~ # gluster volume set repVol-01 cluster.quorum-type auto

gluster01 ~ # gluster volume set repVol-01 cluster.server-quorum-type server

gluster01 ~ # gluster volume set all cluster.enable-shared-storage enable

gluster01 ~ # gluster volume set all cluster.server-quorum-ratio 51%

gluster01 ~ # gluster snapshot config auto-delete enable

→ Verificamos las opciones:

gluster01 ~ # gluster volume info repVol-01

…

Options Reconfigured:

cluster.server-quorum-type: server

cluster.quorum-type: auto

auth.allow: 192.168.10.*

features.barrier: enable

features.scrub: Active

features.bitrot: on

cluster.data-self-heal: on

cluster.metadata-self-heal: on

cluster.entry-self-heal: on

cluster.self-heal-daemon: on

features.uss: enable

features.barrier-timeout: 2m

cluster.force-migration: disable

storage.fips-mode-rchecksum: on

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

auto-delete: enable

cluster.server-quorum-ratio: 51%

cluster.enable-shared-storage: enable

gluster01 ~ # gluster volume info gluster_shared_storage

Volume Name: gluster_shared_storage

Type: Replicate

Volume ID: 5e497220-f30b-4254-b523-e8102e6f086c

Status: Started

Snapshot Count: 0

Number of Bricks: 1 x 3 = 3

Transport-type: tcp

Bricks:

Brick1: gluster02:/var/lib/glusterd/ss_brick

Brick2: gluster03:/var/lib/glusterd/ss_brick

Brick3: gluster01.cadilinea.lan:/var/lib/glusterd/ss_brick

Options Reconfigured:

storage.fips-mode-rchecksum: on

transport.address-family: inet

nfs.disable: on

performance.client-io-threads: off

auto-delete: enable

cluster.server-quorum-ratio: 51%

cluster.enable-shared-storage: enable

==> **Volúmenes Distribuidos** → **Es necesario Rebalancear !**

 # mkdir /bricks/distVol-02/ -p

 # mkdir /mnt/distVol-02/ -p

gluster01 ~ # gluster peer status

Number of Peers: 2

Hostname: gluster02

Uuid: 9812eb50-35dc-4527-99d7-f7ba56321cf6

State: Peer in Cluster (Connected)

Hostname: gluster03

Uuid: 57e79ea9-9dc6-4ee2-8337-c4a20fc30d29

State: Peer in Cluster (Connected)

gluster01 ~ # gluster volume create distVol-02 gluster01:/bricks/distVol-02/data gluster02:/bricks/distVol-02/data gluster03:/bricks/distVol-02/data force

volume create: distVol-02: success: please start the volume to access data

gluster01 ~ # gluster volume start distVol-02

volume start: distVol-02: success

gluster01 ~ # mount.glusterfs gluster01:/distVol-02 /mnt/distVol-02

gluster01 ~ # df -hT

…

gluster01:/distVol-02 fuse.glusterfs 45G 7,5G 38G 17% /mnt/distVol-02

gluster01:/repVol-01 fuse.glusterfs 15G 2,5G 13G 17% /mnt/repVol-01

→ **Eliminamos el nodo** → **'gluster03'**

gluster01 ~ # gluster volume info distVol-02

Volume Name: distVol-02

Type: Distribute

Volume ID: 3199e497-c9ee-4402-b275-ab8f56f87f01

Status: Started

Snapshot Count: 0

Number of Bricks: 3

Transport-type: tcp

Bricks:

Brick1: gluster01:/bricks/distVol-02/data

Brick2: gluster02:/bricks/distVol-02/data

Brick3: gluster03:/bricks/distVol-02/data

Options Reconfigured:

storage.fips-mode-rchecksum: on

transport.address-family: inet

nfs.disable: on

gluster01 ~ # gluster volume remove-brick distVol-02 gluster03:/bricks/distVol-02/data start

It is recommended that remove-brick be run with cluster.force-migration option disabled to prevent possible data corruption. Doing so will ensure that files that receive writes during migration will not be migrated and will need to be manually copied after the remove-brick commit operation. Please check the value of the option and update accordingly.

Do you want to continue with your current cluster.force-migration settings? (y/n) y

volume remove-brick start: success

ID: 847da82e-1cb3-47f0-b043-f5a78d76739e

gluster01 ~ # gluster volume remove-brick distVol-02 gluster03:/bricks/distVol-02/data status

Node	Rebalanced-files	size	scanned	failures	skipped	status	run time in h:m:s
gluster03	1	0Bytes	1	0	0	completed	0:00:00

gluster01 ~ # gluster volume remove-brick distVol-02 gluster03:/bricks/distVol-02/data commit

volume remove-brick commit: success

Check the removed bricks to ensure all files are migrated.

If files with data are found on the brick path, copy them via a gluster mount point before re-purposing the removed brick.

gluster01 ~ # gluster volume info distVol-02

Volume Name: distVol-02

Type: Distribute

Volume ID: 3199e497-c9ee-4402-b275-ab8f56f87f01

Status: Started

Snapshot Count: 0

Number of Bricks: 2

Transport-type: tcp

Bricks:

Brick1: gluster01:/bricks/distVol-02/data

Brick2: gluster02:/bricks/distVol-02/data

Options Reconfigured:

performance.client-io-threads: on

storage.fips-mode-rchecksum: on

transport.address-family: inet

nfs.disable: on

→ **Añadimos de nuevo el nodo** → **'gluster03'**

gluster01 ~ # gluster volume add-brick distVol-02 gluster03:/bricks/distVol-02/data force

volume add-brick: success

gluster01 ~ # gluster volume rebalance distVol-02 start

volume rebalance: distVol-02: success: Rebalance on distVol-02 has been started successfully. Use rebalance status command to check status of the rebalance process.

ID: 1ee5315f-5272-4ea5-ac11-49dcc4bbbb33

gluster01 ~ # gluster volume rebalance distVol-02 status

Node	Rebalanced-files	size	scanned	failures	skipped	status	run time in h:m:s
gluster02	0	0Bytes	0	0	0	completed	0:00:00
gluster03	0	0Bytes	1	0	0	completed	0:00:00
localhost	1	0Bytes	1	0	0	completed	0:00:00

volume rebalance: distVol-02: success

==> NFS-Ganesha → **'distVol-02'**

gluster01 ~ # gluster volume info distVol-02 | grep nfs

nfs.disable: on

gluster01 ~ # gluster volume get repVol-01 nfs.disable

Option Value

------ -----

nfs.disable on

gluster01 ~ # dnf install nfs-ganesha-gluster

gluster01 ~ # mv /etc/ganesha/ganesha.conf /etc/ganesha/ganesha.conf.original

gluster01 ~ # vim /etc/ganesha/ganesha.conf

```
NFS_CORE_PARAM {
   # possible to mount with NFSv3 to NFSv4 Pseudo path
   mount_path_pseudo = true;
   # NFS protocol
   Protocols = 3,4;
}
EXPORT_DEFAULTS {
   # default access mode
   Access_Type = RW;
}
EXPORT {
   # uniq ID
   Export_Id = 101;
   # mount path of Gluster Volume
   Path = "/distVol-02";
   FSAL {
      # any name
```

```
        name = GLUSTER;
        # hostname or IP address of this Node
        hostname="192.168.10.161";
        # Gluster volume name
        volume="distVol-02";
    }
    # config for root Squash
    Squash="No_root_squash";
    # NFSv4 Pseudo path
    Pseudo="/Ganesha-Gluster_Distribuido";
    # allowed security options
    SecType = "sys";
}
LOG {
    # default log level
    Default_Log_Level = WARN;
}
```

gluster01 ~ # systemctl start nfs-ganesha

gluster01 ~ # showmount -e

Export list for gluster01.cadilinea.lan:

/Ganesha-Gluster_Distribuido (everyone)

gluster01 ~ # firewall-cmd --add-service=nfs --permanent

gluster01 ~ # firewall-cmd --reload

gluster01 ~ # dnf install nfs-utils

gluster01 ~ # mkdir /mnt/NFS-Ganesha

gluster01 ~ # mount.nfs4 gluster01:/Ganesha-Gluster_Distribuido /mnt/NFS-Ganesha/

gluster01 ~ # df -hT

```
S.ficheros                      Tipo         Tamaño Usados  Disp Uso% Montado en
…
gluster01:/distVol-02           fuse.glusterfs  45G   7,6G   38G  17% /mnt/distVol-02
gluster01:/repVol-01            fuse.glusterfs  15G   2,6G   13G  17% /mnt/repVol-01
gluster01.cadilinea.lan:/gluster_shared_storage fuse.glusterfs  15G   2,6G   13G  17% /run/gluster/shared_storage
gluster01:/Ganesha-Gluster_Distribuido nfs4          45G   7,6G   38G  17% /mnt/NFS-Ganesha
```

BIBLIOGRAFIA :

https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.5/html/administration_guide/index

https://www.server-world.info/en/note?os=Fedora_33&p=glusterfs&f=1