Red Hat Gluster Storage 3.1
3.1 Release Notes

Release Notes for Red Hat Gluster Storage - 3.1

Red Hat Gluster Storage Documentation Team

Red Hat Gluster Storage 3.1 3.1 Release Notes

Release Notes for Red Hat Gluster Storage - 3.1

Red Hat Gluster Storage Documentation Team
Red Hat Customer Content Services

Legal Notice

Abstract

This Release Notes provides high-level coverage of the improvements and additions that have been implemented in Red Hat Gluster Storage 3.1.

Table of Contents

Chapter 1. Introduction

Red Hat Gluster Storage is a software only, scale-out storage solution that provides flexible and agile unstructured data storage for the enterprise. Red Hat Gluster Storage provides new opportunities to unify data storage and infrastructure, increase performance, and improve availability and manageability to meet a broader set of the storage challenges and needs of an organization.

GlusterFS, a key building block of Red Hat Gluster Storage, is based on a stackable user space design and can deliver exceptional performance for diverse workloads. GlusterFS aggregates various storage servers over different network interfaces and connects them to form a single large parallel network file system. The POSIX compliant GlusterFS servers use XFS file system format to store data on disks. These servers be accessed using industry standard access protocols including Network File System (NFS) and Server Message Block SMB (also known as CIFS).

Red Hat Gluster Storage Servers for On-premises can be used in the deployment of private clouds or data centers. Red Hat Gluster Storage can be installed on commodity servers and storage hardware resulting in a powerful, massively scalable, and highly available NAS environment. Additionally, Red Hat Gluster Storage can be deployed in the public cloud using Red Hat Gluster Storage Server for Public Cloud, for example, within the Amazon Web Services (AWS) cloud. It delivers all the features and functionality possible in a private cloud or data center to the public cloud by providing massively scalable and high available NAS in the cloud.

Red Hat Gluster Storage Server for On-premises

Red Hat Gluster Storage Server for On-premises enables enterprises to treat physical storage as a virtualized, scalable, and centrally managed pool of storage by using commodity servers and storage hardware.

Red Hat Gluster Storage Server for Public Cloud

Red Hat Gluster Storage Server for Public Cloud packages GlusterFS as an Amazon Machine Image (AMI) for deploying scalable NAS in the AWS public cloud. This powerful storage server provides a highly available, scalable, virtualized, and centrally managed pool of storage for Amazon users.

Report a bug

Chapter 2. What's New in this Release?

This chapter describes the key features and major enhancements in the Red Hat Gluster Storage 3.1 release.

- **Dispersed Volumes and Distributed Dispersed Volumes**

  Dispersed volumes are based on erasure coding. Erasure coding is a method of data protection in which data is broken into fragments, expanded and encoded with redundant data pieces and stored across a set of different locations. This allows the recovery of the data stored on one or more bricks in case of failure. Dispersed volume requires less storage space when compared to a replicated volume.

  For more information, see sections *Creating Dispersed Volumes* and *Creating Distributed Dispersed Volumes* in the *Red Hat Gluster Storage Administration Guide*.

- **NFS Ganesha**

  NFS-Ganesha is now supported in highly available active-active environment. In a highly available active-active environment, if a NFS-Ganesha server that is connected to a NFS client running a particular application crashes, the application/NFS client is seamlessly connected to another NFS-Ganesha server without any administrative intervention. The highly available NFS-Ganesha cluster can be modified with the help of the **`ganesha-ha.sh`** script.

  For more information, see section *NFS Ganesha* in the *Red Hat Gluster Storage Administration Guide*.

- **Snapshot Enhancements**

  - **Snapshot Scheduler**

    Snapshot scheduler creates snapshots automatically based on the configured scheduled interval of time. The snapshots can be created every hour, a particular day of the month, particular month, or a particular day of the week based on the configured time interval.

  - **Snapshot Clone**

    You can now create a clone of a snapshot. This is a writable clone and behaves like a regular volume. A new volume can be created from a particular snapshot clone. **`Snapshot Clone is provided as a Technology Preview`**.

  For more information, see chapter *Managing Snapshots* in the *Red Hat Gluster Storage Administration Guide*.

- **SMB**

  With this this release, by upgrading Samba to version 4.1, the following enhancements are added to SMB:

  - Basic support for SMB version 3.0.0 including support for new ciphers for signing.
  - SMB 3 protocol encryption.
  - SMB 2.1 multi-credit (large MTU) operations.
  - SMB 2 offload copying using the COPYCHUNK mechanism.
  - The client tools now support SMB versions 2 and 3.

  For more information, see section *SMB* in the *Red Hat Gluster Storage Administration Guide*.

- **Network Encryption**

Red Hat Gluster Storage supports network encryption using TLS/SSL. Red Hat Gluster Storage uses TLS/SSL for authentication and authorization, in place of the home grown authentication framework used for normal connections. Red Hat Gluster Storage supports both I/O encryption and management (**`glusterd`**) encryption .

For more information, see chapter *Configuring Network Encryption in Red Hat Gluster Storage* in the *Red Hat Gluster Storage Administration Guide*.

- **Detecting Data Corruption with BitRot**

  BitRot detection is a technique used in Red Hat Gluster Storage to identify the silent corruption of data with no indication from the disk to the storage software layer when the error has occurred. BitRot also helps in catching backend tinkering of bricks, where the data is directly manipulated on the bricks without going through FUSE, NFS or any other access protocols. BitRot detection is exceptionally useful when using JBOD. A **bitrot** command scans all the bricks, detects bitrot issue and logs any bit rot errors on the underlying disks.

  For more information, see chapter *Detecting Data Corruption with BitRot* in the *Red Hat Gluster Storage Administration Guide*.

- **Glusterfind (Backup Hooks)**

  Glusterfind is a utility that provides the list of files that are modified between the previous backup session and the current period. This list of files can then be used by any industry standard backup application for backup. These files can be used for periodic antivirus scans too.

  For more information, see chapter *Red Hat Gluster Storage Utilities* in the *Red Hat Gluster Storage Administration Guide*.

- **pNFS**

  The Parallel Network File System (pNFS) is part of the NFS v4.1 protocol that allows compute clients to access storage devices directly and in parallel. **`pNFS is provided as a Technology Preview`**.

  For more information, see section *NFS Ganesha* in the *Red Hat Gluster Storage Administration Guide*.

- **SELinux Support**

  Red Hat Gluster Storage now supports SELinux in **`enabled`** mode. SELinux is supported both on client-side and server-side. You can choose to run SELinux in **`enabled`** or **`permissive`** mode.

  For more information, see chapter *Enabling SELinux* in the *Red Hat Gluster Storage Installation Guide*.

- **Tiering**

  Tiering improves the performance, and the compliance aspects in a Red Hat Gluster Storage environment. This is achieved by optimizing the placement of the most accessed files on the Faster Storage Medium (fast tier / hot tier) and placing less accessed data to Slower Storage Medium (slow tier / cold tier). It also serves as an enabling technology for other enhancements by combining cost-effective or archivally oriented storage for the majority of user data with high-performance storage to absorb the majority of I/O workload. **`Tiering is provided as a Technology Preview`**.

  For more information, see chapter *Managing Tiering* in the *Red Hat Gluster Storage Administration Guide*.

- **Enhancements in Red Hat Gluster Storage Console**

  - **Dashboard**

Dashboard displays an overview of all the entities in Red Hat Gluster Storage like Hosts, Volumes, Bricks, and Clusters. The Dashboard shows a consolidated view of the system and helps the administrator to know the status of the system.

For more information, see chapter *Dashboard Overview* in the *Red Hat Gluster Storage Console Administration Guide*.

- **Disk Provisioning**

  The list of storage devices can be viewed through Red Hat Gluster Storage Console and provisioned through the Console using Disk Provisioning feature. You can also create bricks through Red Hat Gluster Storage Console.

  For more information, see section *Managing Storage Devices* in the *Red Hat Gluster Storage Console Administration Guide*.

- **Logical Networks (Network Traffic Segregation)**

  Logical networks allow both connectivity and segregation. You can create a logical network for gluster storage communication to optimize network traffic between hosts and gluster bricks.

  For more information, see chapter *Logical Networks* in the *Red Hat Gluster Storage Console Administration Guide*.

- **Snapshot Management**

  Snapshot feature enables you to create point-in-time copies of Red Hat Gluster Storage volumes, which you can use to protect data. You can directly access read-only Snapshot copies to recover from accidental deletion, corruption, or modification of the data. Through Red Hat Gluster Storage Console, you can view the list of snapshots and snapshot status, create, delete, activate, deactivate and restore to a given snapshot.

  For more information, see chapter *Managing Snapshots* in the *Red Hat Gluster Storage Console Administration Guide*.

- **Geo-replication Management and Monitoring**

  Geo-replication provides a distributed, continuous, asynchronous, and incremental replication service from one site to another over Local Area Networks (LANs), Wide Area Networks (WANs), and the Internet. You can perform Geo-replication operations and also manage source and destination volumes through Red Hat Gluster Storage Console.

  For more information, see chapter *Managing Geo-replication* in the *Red Hat Gluster Storage Console Administration Guide*.

Chapter 3. Known Issues

This chapter provides a list of known issues at the time of release.

3.1. Red Hat Gluster Storage

Issues related to Snapshot

- BZ# 1201820

  When a snapshot is deleted, the corresponding file system object in the User Serviceable Snapshot is also deleted. Any subsequent file system access results in the **`snapshot`** daemon becoming unresponsive.

  Workaround: Ensure that you do not perform any file system operations on the snapshot that is about to be deleted.

- BZ# 1160621

  If the current directory is not a part of the snapshot, for example, **`snap1`**, then the user cannot enter the **`.snaps/snap1`** directory.

- BZ# 1169790

  When a volume is down and there is an attempt to access **`.snaps`** directory, a negative cache entry is created in the Kernal Virtual File System (VFS) cache for the **`.snaps`**directory. After the volume is brought back online, accessing the **`.snaps`** directory fails with an ENOENT error because of the negative cache entry.

  Workaround: Clear the kernel VFS cache by executing the following command:

  ```
  # echo 3 >/proc/sys/vm/drop_caches
  ```

- BZ# 1170145

  After the restore operation is complete, if restore a volume while you are in the **`.snaps`** directory, the following error message is displayed from the mount point - **`"No such file or directory"`**.

  Workaround:

  - Navigate to the parent directory of the **`.snaps`** directory.
  - Drop VFS cache by executing the following command:

    ```
    # echo 3 >/proc/sys/vm/drop_caches
    ```

  - Change to the **`.snaps`** folder.

- BZ# 1170365

  Virtual inode numbers are generated for all the files in the **`.snaps`** directory. If there are hard links, they are assigned different inode numbers instead of the same inode number.

- BZ# 1170502

On enabling the User Serviceable Snapshot feature, if a directory or a file by name **.snaps** exists on a volume, it appears in the output of the **ls -a** command.

- BZ# 1174618

  If the User Serviceable Snapshot feature is enabled, and a directory has a pre-existing **.snaps** folder, then accessing that folder can lead to unexpected behavior.

  Workaround: Rename the pre-existing **.snaps** folder with another name.

- BZ# 1167648

  Performing operations which involve client graph changes such as volume set operations, restoring snapshot etc eventually leads to out of memory scenarios for the client processes which mount the volume.

- BZ# 1133861

  New snap bricks fails to start if the total snapshot brick count in a node goes beyond 1K.

  Workaround: Deactivate unused snapshots.

- BZ# 1126789

  If any node or **glusterd** service is down when snapshot is restored then any subsequent snapshot creation fails.

  Workaround: Do not restore a snapshot, if node or **glusterd** service is down.

- BZ# 1139624

  While taking snapshot of a gluster volume, it creates another volume which is similar to the original volume. Gluster volume consumes some amount of memory when it is in started state, so as Snapshot volume. Hence, the system goes to out of memory state.

  Workaround: Deactivate unused snapshots to reduce the memory foot print.

- BZ# 1129675

  If **glusterd** is down in one of the nodes in cluster or if the node itself is down, then performing a snapshot restore operation leads to the inconsistencies:

  - Executing **gluster volume heal *vol-name* info** command displays the error message *Transport endpoint not connected*.
  - Error occurs when clients try to connect to glusterd service.

  Workaround: Perform snapshot restore only if all the nodes and their corresponding **glusterd** services are running.

  Restart **glusterd** service using the following command.

  ```
  # service glusterd start
  ```

- BZ# 1105543

  When a node with old snap entry is attached to the cluster, the old entries are propagated throughout the cluster and old snapshots which are not present are displayed.

  Workaround: Do not attach a peer with old snap entries.

- BZ# 1104191

  The **`snapshot`** command fails if snapshot command is run simultaneously from multiple nodes when high write or read operation is happening on the origin or parent volume.

  Workaround: Avoid running multiple snapshot commands simultaneously from different nodes.

- BZ# 1059158

  The **`NFS mount`** option is not supported for snapshot volumes.

- BZ# 1113510

  The output of **`gluster volume info`** information (**`snap-max-hard-limit`**, **`snap-max-soft-limit`**) even though the values that are not set explicitly and must not be displayed.

- BZ# 1111479

  Attaching a new node to the cluster while snapshot delete was in progress, deleted snapshots successfully but gluster snapshot list shows some of the snaps are still present.

  Workaround: Do not attach or detach new node to the trusted storage pool operation while snapshot is in progress.

- BZ# 1092510

  If you create a snapshot when the rename of directory is in progress (here, its complete on hashed sub-volume but not on all of the sub-volumes), on snapshot restore, directory which was undergoing rename operation will have same GFID for both source and destination. Having same GFID is an inconsistency in DHT and can lead to undefined behavior.

  In DHT, a rename (source, destination) of directories is done first on hashed sub-volume and if successful, then on rest of the sub-volumes. At this point in time, if you have both source and destination directories present in the cluster with same GFID - destination on hashed sub-volume and source on rest of the sub-volumes. A parallel lookup (on either source or destination) at this time can result in creation of directories on missing sub-volumes - source directory entry on hashed and destination directory entry on rest of the sub-volumes. Hence, there would be two directory entries - source and destination - having same GFID.

- BZ# 1112250

  Probing/detaching a new peer during any snapshot operation is not supported.

- BZ# 1236149

  If a node/brick is down, the **`snapshot create`** command fails even with the force option.

- BZ# 1240227

  LUKS encryption over LVM is currently not supported.

- BZ# 1236025

  The time stamp of the files/dirs changes when one executes a snapshot restore, resulting in a failure to read the appropriate change logs. **`glusterfind pre`** fails with the following error: '**`historical changelogs not available`**'

  Existing glusterfind sessions fail to work after a snapshot restore.

  Workaround: Gather the necessary information from existing glusterfind sessions, remove the sessions, perform a snapshot restore, and then create new glusterfind sessions.

- BZ# 1160412

  During the update of the *glusterfs-server* package, warnings and fatal errors appear on-screen by **`librdmacm`** if the machine does not have an RDMA device.

  Workaround: You may safely ignore these errors if the configuration does not require Gluster to work with RDMA transport.

- BZ# 1246183

  User Serviceable Snapshots is not supported on Erasure Coded (EC) volumes.

Issues related to Nagios

- BZ# 1136207

  Volume status service shows *All bricks are Up* message even when some of the bricks are in unknown state due to unavailability of **`glusterd`** service.

- BZ# 1109683

  When a volume has a large number of files to heal, the **`volume self heal info`** command takes time to return results and the nrpe plug-in times out as the default timeout is 10 seconds.

  Workaround:

  In **`/etc/nagios/gluster/gluster-commands.cfg`** increase the timeout of nrpe plug-in to 10 minutes by using the -t option in the command.

  Example: $USER1$/gluster/check_vol_server.py $ARG1$ $ARG2$ -o self-heal -t 600

- BZ# 1094765

  When certain commands invoked by Nagios plug-ins fail, irrelevant outputs are displayed as part of performance data.

- BZ# 1107605

  Executing **`sadf`** command used by the Nagios plug-ins returns invalid output.

  Workaround: Delete the datafile located at **`/var/log/sa/saDD`** where DD is current date. This deletes the datafile for current day and a new datafile is automatically created and which is usable by Nagios plug-in.

- BZ# 1107577

  The Volume self heal service returns a WARNING when there unsynchronized entries are present in the volume, even though these files may be synchronized during the next run of self-heal process if **`self-heal`** is turned on in the volume.

- BZ# 1121009

  In Nagios, CTDB service is created by default for all the gluster nodes regardless of whether CTDB is enabled on the Red Hat Gluster Storage node or not.

- BZ# 1089636

  In the Nagios GUI, incorrect status information is displayed as *Cluster Status OK : None of the Volumes are in Critical State*, when volumes are utilized beyond critical level.

- BZ# 1111828

  In Nagios GUI, Volume Utilization graph displays an error when volume is restored using its snapshot.

- BZ# 1236997

  Bricks with an **UNKNOWN** status are not considered as **DOWN** when volume status is calculated. When the glusterd service is down in one node, brick status changes to **UNKNOWN** while the volume status remains OK. You may think the volume is up and running when bricks may not be running. You are not able to detect the correct status.

  Workaround: You are notified when gluster is down and when bricks are in an **UNKNOWN** state.

- BZ# 1240385

  When the **configure-gluster-nagios** command tries to get the IP Address and FLAGs for all network interfaces in the system, the following error is displayed:

  ERROR:root:unable to get ipaddr/flags for *nic-name*: [Errno 99] Cannot assign requested address when there is an issue while retrieving IP Address/Fags for a NIC.

  However, the command actually succeeded and configured the nagios correctly.

Issues related to Rebalancing Volumes

- BZ# 1110282

  Executing **rebalance status** command, after stopping rebalance process, fails and displays a message that the rebalance process is not started.

- BZ# 960910

  After executing **rebalance** on a volume, running the **rm -rf** command on the mount point to remove all of the content from the current working directory recursively without being prompted may return *Directory not Empty* error message.

- BZ# 862618

  After completion of the rebalance operation, there may be a mismatch in the failure counts reported by the **gluster volume rebalance status** output and the rebalance log files.

- BZ# 1039533

  While Rebalance is in progress, adding a brick to the cluster displays an error message, **failed to get index** in the gluster log file. This message can be safely ignored.

- BZ# 1064321

  When a node is brought online after rebalance, the status displays that the operation is completed, but the data is not rebalanced. The data on the node is not rebalanced in a remove-brick rebalance operation and running commit command can cause data loss.

  Workaround: Run the **rebalance** command again if any node is brought down while rebalance is in progress, and also when the rebalance operation is performed after remove-brick operation.

- BZ# 1237059

The rebalance process on a distributed-replicated volume may stop if a brick from a replica pair goes down as some operations cannot be redirected to the other available brick. This causes the rebalance process to fail.

- BZ# 1245202

  When rebalance is run as a part of **`remove-brick`** command, some files may be reported as split-brain and, therefore, not migrated, even if the files are not split-brain.

  Workaround: Manually copy the files that did not migrate from the bricks into the Gluster volume via the mount.

Issues related to Geo-replication

- BZ# 1102524

  The Geo-replication worker goes to faulty state and restarts when resumed. It works as expected when it is restarted, but takes more time to synchronize compared to resume.

- BZ# 987929

  While the **`rebalance`** process is in progress, starting or stopping a Geo-replication session results in some files not get synced to the slave volumes. When a Geo-replication sync process is in progress, running the **`rebalance`** command causes the Geo-replication sync process to stop. As a result, some files do not get synced to the slave volumes.

- BZ# 1029799

  Starting a Geo-replication session when there are tens of millions of files on the master volume takes a long time to observe the updates on the slave mount point.

- BZ# 1027727

  When there are hundreds of thousands of hard links on the master volume prior to starting the Geo-replication session, some hard links are not getting synchronized to the slave volume.

- BZ# 984591

  After stopping a Geo-replication session, if the files synced to the slave volume are renamed then when Geo-replication starts again, the renamed files are treated anew, (without considering the renaming) and synced on to the slave volumes again. For example, if 100 files were renamed, you would find 200 files on the slave side.

- BZ# 1235633

  Concurrent **`rmdir`** and **`lookup`** operations on a directory during a recursive remove may prevent the directory from being deleted on some bricks. The recursive remove operation fails with **`Directory not empty`** errors even though the directory listing from the mount point shows no entries.

  Workaround: Unmount the volume and delete the contents of the directory on each brick. If the affected volume is a geo-replication slave volume, run **`stop geo-rep session`** before deleting the contents of the directory on the bricks.

- BZ# 1238699

  The Changelog History API expects brick path to remain the same for a session. However, on snapshot restore, brick path is changed. This causes the History API to fail and geo-rep to change to **`Faulty`**.

  Workaround: To resolve this issue, perform the following steps:

- After the snapshot restore, ensure the master and slave volumes are stopped.
- Backup the **`htime`** directory (of master volume).

```
cp -a <brick_htime_path> <backup_path>
```

Note

Using **`-a`** option is important to preserve extended attributes.

For example:

```
cp -a
/var/run/gluster/snaps/a4e2c4647cf642f68d0f8259b43494c0/brick0/b0/.g
lusterfs/changeslogs/htime  /opt/backup_htime/brick0_b0
```

- Run the following command to replace the **`OLD`** path in the htime file(s) with the new brick path:

```
find <new_brick_htime_path> - name 'HTIME.*' -print0  | \
xargs -0 sed -ci 's|<OLD_BRICK_PATH>|<NEW_BRICK_PATH>|g'
```

where *OLD_BRICK_PATH* is the brick path of the current volume, and*NEW_BRICK_PATH* is the brick path after snapshot restore. For example:

```
find
/var/run/gluster/snaps/a4e2c4647cf642f68d0f8259b43494c0/brick0/b0/.g
lusterfs/changelogs/htime/ -name 'HTIME.*' -print0  | \
xargs -0 sed -ci
's|/bricks/brick0/b0/|/var/run/gluster/snaps/a4e2c4647cf642f68d0f825
9b43494c0/brick0/b0/|g'
```

- Start the Master and Slave volumes and Geo-replication session on the restored volume. The status should update to **`Active`**.

- BZ# 1240333

  Concurrent rename and lookup operations on a directory can cause both old and new directories to be "healed." Both directories will exist at the end of the operation and will have the same GFID. Clients might be unable to access some of the contents of the directory.

  Workaround: Contact Red Hat Support Services.

Issues related to Self-heal

- BZ# 1063830

  Performing add-brick or remove-brick operations on a volume having replica pairs when there are pending self-heals can cause potential data loss.

  Workaround: Ensure that all bricks of the volume are online and there are no pending self-heals. You can view the pending heal info using the command **`gluster volume heal volname info`**.

- BZ# 1230092

When you create a replica 3 volume, client quorum is enabled and set to **auto** by default. However, it does not get displayed in **gluster volume info**.

- BZ# 1233608

  When **cluster.data-self-heal**, **cluster.metadata-self-heal** and **cluster.entry-self-heal** are set to **off** (through volume set commands), the Gluster CLI to resolve split-brain fails with **File not in split brain** message (even though the file is in split brain).

- BZ# 1240658

  When files are accidentally deleted from a brick in a replica pair in the back-end, and **gluster volume heal *VOLNAME* full** is run, then there is a chance that the files may not get healed.

  Workaround: Perform a lookup on the files from the client (mount). This triggers the heal.

- BZ# 1173519

  If you write to an existing file and go over the **_AVAILABLE_BRICK_SPACE_**, the write fails with an I/O error.

  Workaround: Use the **cluster.min-free-disk** option. If you routinely write files up to *n*GB in size, then you can set min-free-disk to an *m*GB value greater than *n*.

  For example, if your file size is 5GB, which is at the high end of the file size you will be writing, you might consider setting min-free-disk to 8 GB. This ensures that the file will be written to a brick with enough available space (assuming one exists).

```
# gluster v set _VOL_NAME_ min-free-disk 8GB
```

Issues related to replace-brick operation

- After the **gluster volume replace-brick *VOLNAME* Brick *New-Brick* commit force** command is executed, the file system operations on that particular volume, which are in transit, fail.

- After a replace-brick operation, the stat information is different on the NFS mount and the FUSE mount. This happens due to internal time stamp changes when the **replace-brick** operation is performed.

Issues related to Directory Quota

- BZ# 1021466

  After setting Quota limit on a directory, creating sub directories and populating them with files and renaming the files subsequently while the I/O operation is in progress causes a quota limit violation.

- BZ# 998791

  During a file rename operation if the hashing logic moves the target file to a different brick, then the rename operation fails if it is initiated by a non-root user.

- BZ# 1020713

  In a distribute or distribute replicate volume, while setting quota limit on a directory, if one or more bricks or one or more replica sets respectively, experience downtime, quota is not enforced on those bricks or replica sets, when they are back online. As a result, the disk usage exceeds the quota limit.

  Workaround: Set quota limit again after the brick is back online.

- BZ# 1032449

In the case when two or more bricks experience a downtime and data is written to their replica bricks, invoking the quota list command on that multi-node cluster displays different outputs after the bricks are back online.

Issues related to NFS

- After you restart the NFS server, the unlock within the grace-period feature may fail and the locks help previously may not be reclaimed.
- **fcntl** locking (NFS Lock Manager) does not work over IPv6.
- You cannot perform NFS mount on a machine on which glusterfs-NFS process is already running unless you use the NFS mount **`-o nolock`** option. This is because glusterfs-nfs has already registered NLM port with portmapper.
- If the NFS client is behind a NAT (Network Address Translation) router or a firewall, the locking behavior is unpredictable. The current implementation of NLM assumes that Network Address Translation of the client's IP does not happen.
- **`nfs.mount-udp`** option is disabled by default. You must enable it to use posix-locks on Solaris when using NFS to mount on a Red Hat Gluster Storage volume.
- If you enable the **`nfs.mount-udp`** option, while mounting a subdirectory (exported using the **`nfs.export-dir`** option) on Linux, you must mount using the **`-o proto=tcp`** option. UDP is not supported for subdirectory mounts on the GlusterFS-NFS server.
- For NFS Lock Manager to function properly, you must ensure that all of the servers and clients have resolvable hostnames. That is, servers must be able to resolve client names and clients must be able to resolve server hostnames.

Issues related to NFS-Ganesha

- BZ# 1224250

  Same epoch values on all the NFS-Ganesha heads results in NFS server sending **`NFS4ERR_FHEXPIRED`** error instead of **`NFS4ERR_STALE_CLIENTID`** or **`NFS4ERR_STALE_STATEID`** after failover. This results in NFSv4 clients not able to recover locks after failover.

  Workaround: To use NFSv4 locks, specify different epoch values for each NFS-Ganesha head before setting up the NFS-Ganesha cluster.

- BZ# 1226874

  If NFS-Ganesha is started before you set up an HA cluster, there is no way to validate the cluster state and stop NFS-Ganesha if the set up fails. Even if the HA cluster set up fails, the NFS-Ganesha service continues running.

  Workaround: If HA set up fails, run service nfs-ganesha stop on all nodes in the HA cluster.

- BZ# 1227169

  Executing the **`rpcinfo -p`** command after stopping nfs-ganesha displays NFS related programs.

  Workaround: Use **`rpcinfo -d`** on each of the NFS related services listed in **`rpcifnfo -p`** . Alternatively, restart the **`rpcbind`** service using the following command:

  ```
  #service rpcbind restart
  ```

- BZ# 1228196

  If you have less than three nodes, pacemaker shuts down HA.

  Workaround: To restore HA, add a third node with **`ganesha-ha.sh --add $path-to-config $node $virt-ip`** .

- BZ# 1233533

  When the **`nfs-ganesha`** option is turned **`off`**, gluster NFS may not restart automatically.. The volume may no longer be exported from the storage nodes via a nfs-server.

  Workaround:

  - Turn off the **`nfs.disable`** option for the volume:

    ```
    gluster volume set volume name nfs.disable off
    ```

  - Restart the volume:

    ```
    gluster volume start volume name force
    ```

- BZ# 1235597

  On the nfs-ganesha server IP, **`showmount`** does not display a list of the clients mounting from that host.

- BZ# 1236017

  When a server is rebooted, services such as **`pcsd`** and **`nfs-ganesha`** do not start by default. **`nfs-ganesha`** won't be running on the rebooted node, so it won't be part of the HA-cluster.

  Workaround: Manually restart the services after a server reboot.

- BZ# 1238561

  Although **`DENY`** entries are handled in **`nfs4_setfacl`**, they cannot be stored directly in the backend (**`DENY`** entry cannot convert in POSIX ACL). **`DENY`** entries won't display in **`nfs4_getfacl`**. If the permission bit is not set in **`ALLOW`** entry, it is considered as **`DENY`**.

  

  Note

  Use minimal required permission for **`EVERYONE@Entry`**, otherwise it will result in undesired behavior of **`nfs4_acl`**.

- BZ# 1240258

  When files and directories are created on the mount point with root squash enabled for **`nfs-ganesha`**, executing **`ls`** command displays **`user:group as 4294967294:4294967294`** instead of **`nfsnobody:nfsnobody`**. This is because the client maps only 16 bit unsigned representation of -2 to **`nfsnobody`** whereas 4294967294 is 32 bit equivalent of -2.

  This is currently a limitation in upstream **`nfs-ganesha`**.

- BZ# 1240502

  Delete-node logic does not remove the VIP of the deleted node from **`ganesha-ha.conf`**. The VIP exists

even after the node is deleted from the HA cluster.

Workaround: Manually delete the entry if it is not required for subsequent operations.

- BZ# 1241436

  The output of the **`refresh-config`** option is not meaningful.

  Workaround: If the output displays as follows, '**`method return sender=:1.61 -> dest=:1.65 reply_serial=2`**', consider it successful.

- BZ# 1242148

  When ACLs are enabled, if you rename a file, an error is thrown on nfs4 mount. However, the operation is successful. It may take a few seconds to complete.

- BZ# 1246007

  NFS-Ganesha export files are not copied as part of snapshot creation. As a result, snapshot restore will not work with NFS-Ganesha.

Issues related to Object Store

- The GET and PUT commands fail on large files while using Unified File and Object Storage.

  Workaround: You must set the ***`node_timeout=60`*** variable in the proxy, container, and the object server configuration files.

Issues related to Red Hat Gluster Storage Volumes:

- BZ# 986090

  Currently, the Red Hat Gluster Storage server has issues with mixed usage of hostnames, IPs and FQDNs to refer to a peer. If a peer has been probed using its hostname but IPs are used during add-brick, the operation may fail. It is recommended to use the same address for all the operations, that is, during peer probe, volume creation, and adding/removing bricks. It is preferable if the address is correctly resolvable to a FQDN.

- BZ# 852293

  The management daemon does not have a rollback mechanism to revert any action that may have succeeded on some nodes and failed on the those that do not have the brick's parent directory. For example, setting the **`volume-id`** extended attribute may fail on some nodes and succeed on others. Because of this, the subsequent attempts to recreate the volume using the same bricks may fail with the error *brickname or a prefix of it is already part of a volume*.

  Workaround:

  - You can either remove the brick directories or remove the glusterfs-related extended attributes.
  - Try creating the volume again.

- BZ# 913364

  An NFS server reboot does not reclaim the file LOCK held by a Red Hat Enterprise Linux 5.9 client.

- BZ# 1030438

On a volume, when read and write operations are in progress and simultaneously a rebalance operation is performed followed by a remove-brick operation on that volume, then the **`rm -rf`** command fails on a few files.

- BZ# 1224064

  Glusterfind is a independent tool and is not integrated with glusterd. When a Gluster volume is deleted, respective glusterfind session directories/files for that volume persist.

  Workaround: Manually, delete the Glusterfind session directory in each node for the Gluster volume in the following directory **`/var/lib/glusterd/glusterfind`**

- BZ# 1224153

  When a brick process dies, BitD tries to read from the socket used to communicate with the corresponding brick. If it fails, BitD logs the failure to the log file. This results in many messages in the log files, leading to the failure of reading from the socket and an increase in the size of the log file.

- BZ# 1224162

  Due to an unhandled race in the RPC interaction layer, brick down notifications may result in corrupted data structures being accessed. This can lead to NULL pointer access and segfault.

  Workaround: When the **`Bitrot`** daemon (bitd) crashes (segfault), you can use **`volume start`** ***`VOLNAME`*** **`force`** to restart **`bitd`** on the node(s) where it crashed.

- BZ# 1224880

  If you delete a gluster volume before deleting the Glusterfind session, then the Glusterfind session can't be deleted. A new session can't be created with same name.

  Workaround: In all the nodes that were part of the volume before you deleted it, manually cleanup the session directory in: /**`var/lib/glusterd/glusterfind/`*****`SESSION/VOLNAME`***

- BZ# 1226995

  Using brick up time to calculate the next scrub time results in premature filesystem scrubbing:

  - Brick up-time: T Next scrub time (frequency hourly): T + 3600 seconds
  - After 55 minutes (T + 3300 seconds), the scrub frequency is changed to daily. Therefore, the next scrub would happen at (T + 86400 seconds) rather than (current_time + 86400 seconds).

- BZ# 1227672

  A successful scrub of the filesystem (objects) is required to see if a given object is clean or corrupted. When a file gets corrupted and a scrub hasn't been run on the filesystem, there is a good chance of replicating corrupted objects in cases when the brick holding the good copy was offline when I/O was performed.

  Workaround: Objects need to be checked on demand for corruption during healing.

- BZ# 1231150

  When you set diagnostic.client-log-level DEBUG, and then reset the **`diagnostic.client-log-level`** option, DEBUG logs continue to appear in log files. INFO log level is enabled by default.

  Workaround: Restart the volume using **`gluster volume start`** ***`VOLNAME`*** **`force`**, to reset log level defaults.

- BZ# 1233213

  If you run a **`gluster volume info --xml`** command on a newly probed peer without running any other gluster volume command in between, brick UUIDs will appear as null ('00000000-0000-0000-0000-000000000000').

  Workaround: Run any volume command (excluding **`gluster volume list`** and **`gluster volume get`**) before you run the info command. Brick UUIDs correctly populate.

- BZ# 1236153

  The shared storage Gluster command accepts only the **`cluster.enable-shared-storage`** key. It should also accept the **`enable-shared-storage`** key.

- BZ# 1236503

  Disabling **`cluster.enable-shared-storage`** results in the deletion of any volume named **`gluster_shared_storage`** even if it is a pre-existing volume.

- BZ# 1237022

  If you have a cluster with more than one node and try to perform a peer probe from a node that is not part of the cluster, the peer probe fails without a meaningful notification.

- BZ# 1241314

  The **`volume get VOLNAME enable-shared-storage`** option always shows as disabled, even when it is enabled.

  Workaround:`gluster volume info VOLNAME` command shows the correct status of the **`enable-shared-storage`** option.

- BZ# 1241336

  When an Red Hat Gluster Storage node is shut down due to power failure or hardware failure, or when the network interface on a node goes down abruptly, subsequent gluster commands may time out. This happens because the corresponding TCP connection remains in the **`ESTABLISHED`** state.

  You can confirm this by executing the following command: **`ss -tap state established '( dport = :24007 )' dst IP-addr-of-powered-off-RHGS-node`**

  Workaround: Restart **`glusterd`** service on all other nodes.

- BZ# 1223306

  `gluster volume heal VOLNAME info` shows stale entries, even after the file is deleted. This happens due to a rare case when the *gfid-handle* of the file is not deleted.

  Workaround: On the bricks where the stale entries are present, for example, **`<gfid:5848899c-b6da-41d0-95f4-64ac85c87d3f>`**, perform the following steps:

  - 1) Check if the file's **`gfid`** handle is not deleted.

    `# find <brick-path>/.glusterfs -type f -links 1` and check if the file **`<brick-path>/.glusterfs/58/48/5848899c-b6da-41d0-95f4-64ac85c87d3f`** appears in the output.

  - If it appears in the output, you must delete that file.

```
# rm <brick-path>/.glusterfs/58/48/5848899c-b6da-41d0-95f4-64ac85c87d3f
```

- BZ# 1224180

  In some cases, operations on the mount displays error: **`Input/Output error`** instead of **`Disk quota exceeded`** message after the quota limit is exceeded.

- BZ# 1244759

  Sometimes gluster volume heal *VOLNAME* info shows some symlinks which need to be healed for hours.

  To confirm this issue, the files must have the following extended attributes:

```
# getfattr -d -m. -e hex -h /path/to/file/on/brick | grep trusted.ec

Example output:
trusted.ec.dirty=0x3000
trusted.ec.size=0x3000
trusted.ec.version=0x30000000000000000000000000000001
```

  The first four digits must be **`3000`** and the file must be a symlink/softlink.

  Workaround: Execute the following commands on the files in each brick and ensure to stop all operations on them.

  - **`trusted.ec.size`** must be deleted.

```
# setfattr -x trusted.ec.size /path/to/file/on/brick
```

  - First 16 digits must have '0' in both **`trusted.ec.dirty`** and **`trusted.ec.version`** attributes and the rest of the 16 digits should remain as is. If the number of digits is less than 32, then use '0 s as padding.

```
# setfattr -n trusted.ec.dirty -v 0x00000000000000000000000000000000
/path/to/file/on/brick
# setfattr -n trusted.ec.version -v
0x00000000000000000000000000000001 /path/to/file/on/brick
```

Issues related to POSIX ACLs:

- Mounting a volume with **`-o acl`** can negatively impact the directory read performance. Commands like recursive directory listing can be slower than normal.

- When POSIX ACLs are set and multiple NFS clients are used, there could be inconsistency in the way ACLs are applied due to attribute caching in NFS. For a consistent view of POSIX ACLs in a multiple client setup, use the **-o noac** option on the NFS mount to disable attribute caching. Note that disabling the attribute caching option could lead to a performance impact on the operations involving the attributes.

Issues related to Samba

- BZ# 1013151

  Accessing a Samba share may fail, if GlusterFS is updated while Samba is running.

  Workaround: On each node where GlusterFS is updated, restart Samba services after GlusterFS is updated.

- BZ# 994990

  When the same file is accessed concurrently by multiple users for reading and writing. The users trying to write to the same file will not be able to complete the write operation because of the lock not being available.

  Workaround: To avoid the issue, execute the command:

  ```
  # gluster volume set VOLNAME storage.batch-fsync-delay-usec 0
  ```

- BZ# 1031783

  If Red Hat Gluster Storage volumes are exported by Samba, NT ACLs set on the folders by Microsoft Windows clients do not behave as expected.

- BZ# 1164778

  Any changes performed by an administrator in a Gluster volume's share section of **`smb.conf`** are replaced with the default Gluster hook scripts settings when the volume is restarted.

  Workaround: The administrator must perform the changes again on all nodes after the volume restarts.

General issues

- If files and directories have different GFIDs on different back-ends, the glusterFS client may hang or display errors.

  Contact Red Hat Support for more information on this issue.

- BZ# 1030962

  On installing the Red Hat Gluster Storage Server from an ISO or PXE, the **`kexec-tools`** package for the **`kdump`** service gets installed by default. However, the ***`crashkernel=auto`*** kernel parameter required for reserving memory for the **`kdump`** kernel, is not set for the current kernel entry in the bootloader configuration file, **`/boot/grub/grub.conf`**. Therefore the **`kdump`** service fails to start up with the following message available in the logs.

  ```
  kdump: No crashkernel parameter specified for running kernel
  ```

  Workaround: After installing the Red Hat Gluster Storage Server, the***`crashkernel=auto`***, or an appropriate ***`crashkernel=sizeM`*** kernel parameter can be set manually for the current kernel in the bootloader configuration file. After that, the Red Hat Gluster Storage Server system must be rebooted, upon which the memory for the **`kdump`** kernel is reserved and the **`kdump`** service starts successfully. Refer to the following link for more information on Configuring kdump on the Command Line

  Additional information: On installing a new kernel after installing the Red Hat Gluster Storage Server, the ***`crashkernel=auto`*** kernel parameter is successfully set in the bootloader configuration file for the newly added kernel.

- BZ# 1058032

  While migrating VMs, libvirt changes the ownership of the guest image, unless it detects that the image is on a shared filesystem and the VMs can not access the disk imanges as the required ownership is not available.

  Workaround: Perform the steps:

  - Power-off the VMs before migration.

- After migration is complete, restore the ownership of the VM Disk Image (107:107)
- Start the VMs after migration.

- The glusterd service crashes when volume management commands are executed concurrently with peer commands.

- BZ# 1130270

  If a 32 bit Samba package is installed before installing Red Hat Gluster Storage Samba package, the installation fails as Samba packages built for Red Hat Gluster Storage do not have 32 bit variants

  Workaround:Uninstall 32 bit variants of Samba packages.

- BZ# 1139183

  The Red Hat Gluster Storage 3.0 version does not prevent clients with versions older Red Hat Gluster Storage 3.0 from mounting a volume on which rebalance is performed. Users with versions older than Red Hat Gluster Storage 3.0 mounting a volume on which rebalance is performed can lead to data loss.

  Workaround: You must install latest client version to avoid this issue.

- BZ# 1127178

  If a replica brick goes down and comes up when **`rm -rf`** command is executed, the operation may fail with the message *Directory not empty*.

  Workaround: Retry the operation when there are no pending self-heals.

- BZ# 1007773

  When **`remove-brick start`** command is executed, even though the graph change is propagated to the NFS server, the directory inodes in memory are not refreshed to exclude the removed brick. Hence, new files that are created may end up on the removed-brick.

  Workaround: If files are found on the removed-brick path after **`remove-brick commit`**, copy them via a gluster mount point before re-purposing the removed brick.

- BZ# 1120437

  Executing **`peer-status`** command on probed host displays the IP address of the node on which the peer probe was performed.

  Example: When probing a peer, node B with hostname from node A, executing **`peer status`** command on node B, displays IP address of node A instead of its hostname.

  Workaround: Probe node A from node B with hostname of node A. For example, execute the command: **`# gluster peer probe HostnameA`** from node B.

- BZ# 1122371

  The NFS server process and gluster **`self-heal`** daemon process restarts when gluster daemon process is restarted.

- BZ# 1110692

  Executing **`remove-brick status`** command, after stopping remove-brick process, fails and displays a message that the remove-brick process is not started.

- BZ# 1123733

Executing a command which involves glusterd-glusterd communication **`gluster volume status`** immediately after one of the nodes is down hangs and fails after 2 minutes with cli-timeout message. The subsequent command fails with the error message *Another transaction in progress* for 10 mins (frame timeout).

Workaround: Set a non-zero value for *ping-timeout* in **`/etc/glusterfs/glusterd.vol`** file.

- BZ# 1136718

  The AFR self-heal can leave behind a partially healed file if the brick containing AFR self-heal source file goes down in the middle of heal operation. If this partially healed file is migrated before the brick that was down comes online again, the migrated file would have incorrect data and the original file would be deleted.

- BZ# 1139193

  After **`add-brick`** operation, any application (like git) which attempts **`opendir`** on a previously present directory fails with **`ESTALE/ENOENT`** errors.

- BZ# 1141172

  If you rename a file from multiple mount points, there are chances of losing the file. This issue is witnessed since **`mv`** command sends unlinks instead of renames when source and destination happens to be hard links to each other. Hence, the issue is in mv, distributed as part of **`coreutils`** in various Linux distributions.

  For example, if there are parallel renames of the form (mv a b) and (mv b a) where a and b are hard links to the same file, because of the above mentioned behavior of mv, unlink (a) and unlink (b) would be issued from both instances of **`mv`**. This results in losing both the links a and b and hence the file.

- BZ# 979926

  When any process establishes a TCP connection with **`glusterfs`** servers of a volume using port **`> 1023`**, the server rejects the requests and the corresponding file or management operations fail. By default, **`glusterfs`** servers treat ports **`> 1023`** as unprivileged.

  Workaround: To disable this behavior, enable **`rpc-auth-allow-insecure`** option on the volume using the steps given below:

  - To allow **`insecure`** connections to a volume, run the following command:

    ```
    #gluster volume set VOLNAME rpc-auth-allow-insecure on
    ```

  - To allow **`insecure`** connections to glusterd process, add the following line in **`/etc/glusterfs/glusterd.vol`** file:

    ```
    option rpc-auth-allow-insecure on
    ```

  - Restart **`glusterd`** process using the following command:

    ```
    # service glusterd restart
    ```

  - Restrict connections to trusted clients using the following command:

    ```
    #gluster volume set VOLNAME auth.allow IP address
    ```

- BZ# 1139676

  Renaming a directory may cause both source and target directories to exist on the volume with the same GFID and make some files in these directories not visible from the mount point. The files will still be present on the bricks.

  Workaround: The steps to fix this issue are documented in: https://access.redhat.com/solutions/1211133

- BZ# 1030309

  During directory creations attempted by geo-replication, though an **`mkdir`** fails with **`EEXIST`**, the directory might not have a complete layout for sometime and the directory creation fails with **`Directory exists`** message. This can happen if there is a parallel **`mkdir`** attempt on the same name. Till the other **`mkdir`** completes, layout is not set on the directory. Without a layout, entry creations within that directory fails.

  Workaround: Set the layout on those sub-volumes where the directory is already created by the parallel **`mkdir`** before failing the current **`mkdir`** with **`EEXIST`**.

  

  Note

  This is not a complete fix as the other **`mkdir`** might not have created directories on all sub-volumes. The layout is set on the sub-volumes where directory is already created. Any file or directory names which hash to these sub-volumes on which layout is set, can be created successfully.

- BZ# 1238067

  In rare instances, glusterd may crash when it is stopped. The crash is due to a race between the clean up thread and the running thread and doesn't impact functionality. The clean up thread releases URCU resources while a running thread continues to try to access it, which results in a crash.

- BZ# 1238171

  When an inode is unlinked from the backend (bricks) directly, the corresponding in-memory inode is not cleaned on subsequent lookup. This causes the recovery procedure using healing damons (such as AFR/EC self-heal) to not function as expected as the in-memory inode structure represents a corrupted backend object.

  Workaround: A patch is available. The object could still be recoverable when the inode is forgotten (due to memory pressure or brick restart). In such cases, accessing the object would trigger a successful self-heal and recover it.

- BZ# 1241385

  Due to a code bug, the output prefix was not considered when updating the path of deleted entries. The output file/dir name will not have an output prefix.

Issues related to Red Hat Gluster Storage AMI

- BZ# 1250821

  In the Red Hat Gluster Storage 3.1 on Red Hat Enterprise Linux 7 AMI, the Red Hat Enterprise Linux 7 server base repo is disabled by default. We must manually enable the repo to receive package updates from the Red Hat Enterprise Linux 7 server base repo.

  Workaround: To enable the repo manually, run the following command:

```
yum-config-manager --enable rhui-REGION-rhel-server-releases
```

Once enabled, the AMI will receive package updates from Red Hat Enterprise Linux 7 server base repo.

3.1.1. Issues Related to Upgrade

- BZ# 1247515

  As part of the tiering feature, a new dictionary key value pair was introduced to send the number of bricks in the hot-tier. So **`glusterd`** will expect this key in a dictionary which was sent to other peers during the data exchange. Since one of the node runs Red Hat Gluster Storage 2.1, this key value pair is not sent which causes **`glusterd`** running on Red Hat Gluster Storage 3.1 to complain about the missing key value pair from the peer data.

  Workaround: No functionality issues. An error is displayed in **`glusterd`** logs.

3.2. Red Hat Gluster Storage Console

Issues related to Red Hat Gluster Storage Console

- BZ# 1246047

  If a logical network is attached to the interface with boot protocol DHCP, the IP address is not assigned to the interface on saving network configuration, if DHCP server responses are slow.

  Workaround: Click **`Refresh Capabilities`** on the **`Hosts`** tab and the network details are refreshed and the IP address is correctly assigned to the interface.

- BZ#1164662

  The **`Trends`** tab in the Red Hat Gluster Storage Console appears to be empty after the ovirt engine restarts. This is due to the Red Hat Gluster Storage Console UI-plugin failing to load on the first instance of restarting the ovirt engine.

  Workaround: Refresh (F5) the browser page to load the **`Trends`** tab.

- BZ#1167305

  The **`Trends`** tab on the Red Hat Gluster Storage Console does not display the thin-pool utilization graphs in addition to the brick utilization graphs. Currently, there is no mechanism for the UI plugin to detect if the volume is provisioned using the thin provisioning feature.

- BZ#1167572

  On editing the cluster version in the Edit Cluster dialog box on the Red Hat Gluster Storage Console, the compatible version field gets loaded with the highest available compatibility version by default, instead of the current version of the cluster.

  Workaround: Select the correct version of the cluster in the **`Edit Cluster`** dialog box before clicking on the **`OK`** button.

- BZ# 1054366

  In Internet Explorer 10, while creating a new cluster with Compatibility version 3.3, the **Host** drop down list does not open correctly. Also, if there is only one item, the drop down list gets hidden when the user clicks on it.

- BZ# 1053395

  In Internet Explorer, while performing a task, an error message Unable to evaluate payload is displayed.

- BZ# 1056372

  When no migration is occurring, incorrect error message is displayed for the **stop migrate** operation.

- BZ# 1048426

  When there are more entries in rebalance status and remove-brick status window, the column names scrolls up along with the entries while scrolling the window.

  Workaround: Scroll up the rebalance status and remove-brick status window to view the column names.

- BZ# 1053112

  When large sized files are migrated, the stop migrate task does not stop the migration immediately but only after the migration is complete.

- BZ# 1040310

  If the Rebalance Status dialog box is open in the Red Hat Gluster Storage Console while Rebalance is being stopped from the Command Line Interface, the status is currently updated as *Stopped*. But if the Rebalance Status dialog box is not open, the task status is displayed as *Unknown* because the status update relies on the gluster Command Line Interface.

- BZ# 838329

  When incorrect create request is sent through REST api, an error message is displayed which contains the internal package structure.

- BZ# 1049863

  When Rebalance is running on multiple volumes, viewing the brick advanced details fails and the error message could not fetch brick details, please try again later is displayed in the **`Brick Advanced Details`** dialog box.

- BZ# 1024184

  If there is an error while adding bricks, all the "." characters of FQDN / IP address in the error message wil be replaced with "_" characters.

- BZ# 975399

  When Gluster daemon service is restarted, the host status does not change to UP from Non-Operational immediately in the Red Hat Gluster Storage Console. There would be a 5 minute interval for auto-recovery operations which detect changes in Non-Operational hosts.

- BZ# 971676

  While enabling or disabling Gluster hooks, the error message displayed if all the servers are not in UP state is incorrect.

- BZ# 1057122

  While configuring the Red Hat Gluster Storage Console to use a remote database server, on providing either **`yes`** or **`no`** as input for ***`Database host name validation`*** parameter, it is considered as **`No`**.

- BZ# 1042808

When remove-brick operation fails on a volume, the Red Hat Gluster Storage node does not allow any other operation on that volume.

Workaround: Perform *commit* or *stop* for the failed remove-brick task, before another task can be started on the volume.

- BZ# 1060991

  In Red Hat Gluster Storage Console, Technology Preview warning is not displayed for stop remove-brick operation.

- BZ# 1057450

  Brick operations like adding and removing a brick from Red Hat Gluster Storage Console fails when Red Hat Gluster Storage nodes in the cluster have multiple FQDNs (Fully Qualified Domain Names).

  Workaround: Host with multiple interfaces should map to the same FQDN for both Red Hat Gluster Storage Console and gluster peer probe.

- BZ# 1038663

  Framework restricts displaying delete actions for collections in RSDL display.

- BZ# 1061677

  When Red Hat Gluster Storage Console detects a **`remove-brick`** operation which is started from Gluster Command Line Interface, engine does not acquire lock on the volume and Rebalance task is allowed.

  Workaround: Perform **`commit`** or **`stop`** on **`remove-brick`** operation before starting Rebalance.

- BZ# 1046055

  While creating volume, if the bricks are added in root partition, the error message displayed does not contain the information that **`Allow bricks in root partition and re-use the bricks by clearing xattrs`** option needs to be selected to add bricks in root partition.

  Workaround: Select **`Allow bricks in root partition and re-use the bricks by clearing xattrs`** option to add bricks in root partition.

- BZ# 1060991

  In Red Hat Gluster Storage Console UI, Technology Preview warning is not displayed for stop remove-brick operation.

- BZ# 1066130

  Simultaneous start of Rebalance on volumes that span same set of hosts fails as gluster daemon lock is acquired on participating hosts.

  Workaround: Start Rebalance again on the other volume after the process starts on first volume.

- BZ# 1200248

  The **`Trends`** tab on the Red Hat Gluster Storage Console does not display all the network interfaces available on a host. This limitation is because the Red Hat Gluster Storage Console **`ui-plugin`** does not have this information.

  Workaround: The graphs associated with the hosts are available in the Nagios UI on the Red Hat Gluster Storage Console.You can view the graphs by clicking the **`Nagios home`** link

- BZ# 1224724

  The **Volume** tab loads before the dashboard plug-in is loaded. When the dashboard is set as the default tab, the volume sub-tab remains on top of dashboard tab.

  Workaround: Switch to a different tab and the sub-tab is removed.

- BZ# 1225826

  In Firefox-38.0-4.el6_6, check boxes and labels in **Add brick** and **Remove Brick** dialog boxes are misaligned.

- BZ# 1228179

  gluster volume set help-xml does not list the **config.transport** option in the UI.

  Workaround: Type the option name instead of selecting it from the drop-down list. Enter the desired value in the value field.

- BZ# 1231723

  Storage devices with disk labels appear as locked on the storage devices sub-tab. When a user deletes a brick by removing lv, vg, pv and partition, the storage device appears with the lock symbol and the user is unable to create a new brick from the storage device.

  Workaround: Using the CLI, manually create a partition. Click Sync on the **Storage Device** sub-tab under the host shows the created partition in the UI. The partition appears as a free device that can be used to create a brick through the Red Hat Gluster Storage Console GUI.

- BZ# 1231725

  Red Hat Gluster Storage Console cannot detect bricks that are created manually using the CLI and mounted to a location other than **/rhgs**. Users must manually type the brick directory in the **Add Bricks** dialog box.

  Workaround: Mount bricks in the **/rhgs** folder, which are detected automatically by Red Hat Gluster Storage Console.

- BZ# 1232275

  Blivet provides only partial device details on any major disk failure. The Storage Devices tab does not show some storage devices if the partition table is corrupted.

  Workaround: Clean the corrupted partition table using the **dd** command. All storage devices are then synced to the UI.

- BZ# 1233592

  The **Force Remove** checkbox appears in the **Remove GeoReplication** window even if it is unnecessary. Even if you use the force option, it is the equivalent of using w/o force as the option is not available in the Gluster CLI to remove a geo-replication session.

- BZ# 1232575

  When performing a search on a specific cluster, the volumes of all clusters that have a name beginning with the selected cluster name are returned.

- BZ# 1234445

The task-id corresponding to the previously performed retain/stop remove-brick is preserved by engine. When a user queries for remove-brick status, it passes the bricks of both the previous remove-brick as well as the current bricks to the status command. The UI returns the error **`Could not fetch remove brick status of volume`**.

In Gluster, once a remove-brick has been stopped, the status can't be obtained.

- BZ# 1235559

  The same audit log messages is used in two cases:

  - When the current_scheduler value is set as oVirt in Gluster.
  - When the current_scheduler value is set as oVirt in Gluster.

  The first message should be corrected to mention that the flag is set successfully to oVirt in the CLI.

- BZ# 1236410

  While syncing snapshots created from the CLI, the engine populates the creation time, which is returned from the Gluster CLI. When you create a snapshot from the UI, the engine current time is marked as the creation time in the engine DB. This leads to a mismatch between creation times for snapshots created from the engine and the CLI.

- BZ# 1238244

  Upgrade is supported from Red Hat Gluster Storage 3.0 to 3.1, but you cannot upgrade from Red Hat Gluster Storage 2.1 to 3.1.

  Workaround: Reinstall Red Hat Gluster Storage 3.1 on existing deployments of 2.1 and import existing clusters. Refer to the *Red Hat Guster Storage Console Installation Guide* for further information.

- BZ# 1238332

  When the console doesn't know that **`glusterd`** is not running on the host, removal of a brick results in an undetermined state (question mark). When **`glusterd`** is started again, the brick remains in an undetermined state. The volume command shows status as **`not started`** but the **`remove-brick`** status command returns null in the status field.

  Workaround: Stop/commit remove-brick from the CLI.

- BZ# 1238540

  When you create volume snapshots, time zone and time stamp details are appended to the snapshot name. The engine passes only the prefix for the snapshot name. If master and slave clusters of a geo-replication session are in different time zones (or sometimes even in the same time zone), the snapshot names of the master and slave are different. This causes a restore of a snapshot of the master volume to fail because the slave volume name does not match.

  Workaround: Identify the respective snapshots for the master and slave volumes and restore them separately from the gluster CLI by pausing the geo-replication session.

- BZ# 1240627

  There is a time out for a VDSM call from the **`oVirt`** engine. Removing 256 snapshots from a volume causes the engine to time out during the call. UI shows a network error as the command timed out. However, the snapshots were deleted successfully.

  Workaround: Delete the snapshots in smaller chunks using the **`Delete`** option, which supports the deletion of multiple snapshots at once.

- BZ# 1242128

  Deleting a gluster volume does not remove the **/etc/fstab** entries for the bricks. A Red Hat Enterprise Linux 7 system may fail to boot if the mount fails for any entry in the **/etc/fstab** file. If the LVs corresponding to the bricks are deleted but not the respective entry in **/etc/fstab**, then the system may not boot.

  Workaround:

  - Ensure that **/etc/fstab** entries are removed when the Logical Volumes are deleted from system.
  - If the system fails to boot, start it in emergency mode, use your root password, remount '/' in rw, edit fstab, save, and then reboot.

- BZ# 1242442

  Restoring a volume to a snapshot changes the volume to use the snapshot bricks mounted at **/var/run/gluster/snaps/**. However, it does not remove the **/etc/fstab** entries for the original brick. This could cause a Red Hat Enterprise Linux 7 system to fail to boot.

  Workaround:

  - Ensure that **/etc/fstab** entries are removed when the Logical Volumes are deleted from system.
  - If the system fails to boot, then start the system in emergency mode, use the root password, remount '/' in rw, edit fstab, save it, and then reboot.

- BZ# 1243443

  Unable to resolve Gluster hook conflicts when there are three conflicts: Content + Status + Missing

  Workaround: Resolve the Content + Missing hook conflict before resolving the Status conflict.

- BZ# 1243537

  Labels do not show enough information for the Graphs shown on the **Trends** tab. When you select a host in the system tree and switch to the **Trends** tab, you will see two graphs for the mount point '/': one graph for the total space used and another for the inodes used on the disk.

  Workaround:

  - The graph with y axis legend as %(Total: ** GiB/Tib) is the graph for total space used.
  - The graph with y axis legend as %(Total: *number*) is the graph for inode usage.

- BZ# 1244507

  If the meta volume is not already mounted, snapshot schedule creation fails as it needs meta volume to be mounted so that CLI based scheduling can be disabled.

  Workaround: If meta volume is available, mount it from the CLI, and then create the snapshot schedule in the UI.

- BZ# 1246038

  Selection of the Gluster network role is not persistent when changing multiple fields. If you attach this logical network to an interface, it is ignored when you add bricks.

  Workaround: Reconfigure the role for the logical network.

3.3. Red Hat Gluster Storage and Red Hat Enterprise Virtualization Integration

Issues related to Red Hat Enterprise Virtualization and Red Hat Gluster Storage Integration

- In the case that the Red Hat Gluster Storage server nodes and the Red Hat Enterprise Virtualization Hypervisors are present in the same data center, the servers of both types are listed for selection when you create a virtual machine or add a storage domain. Red Hat recommends that you create a separate data center for the Red Hat Gluster Storage server nodes.

3.4. Red Hat Gluster Storage and Red Hat OpenStack Integration

Issues related to Red Hat OpenStack and Red Hat Gluster Storage integration

- BZ# 1004745

  If a replica pair is down while taking a snapshot of a Nova instance on top of a Cinder volume hosted on a Red Hat Gluster Storage volume, the snapshot process may not complete as expected.

- BZ# 980977 and BZ# 1017340

  If storage becomes unavailable, the volume actions fail with **`error_deleting`** message.

  `Workaround`: Run **`gluster volume delete VOLNAME force`** to forcefully delete the volume.

- BZ# 1062848

  When a nova instance is rebooted while rebalance is in progress on the Red Hat Gluster Storage volume, the root file system will be mounted as read-only after the instance comes back up. Corruption messages are also seen on the instance.

Chapter 4. Technology Previews

This chapter provides a list of all available Technology Preview features in Red Hat Gluster Storage 3.1 release.

Technology Preview features are currently not supported under Red Hat Gluster Storage subscription services, may not be functionally complete, and are generally not suitable for production environments. However, these features are included for customer convenience and to provide wider exposure to the feature.

Customers may find these features useful in a non-production environment. Customers are also free to provide feedback and functionality suggestions for a Technology Preview feature before it becomes fully supported. Errata will be provided for high-severity security issues.

During the development of a Technology Preview feature, additional components may become available to the public for testing. Red Hat intends to fully support Technology Preview features in the future releases.

Note

All Technology Preview features in Red Hat Enterprise Linux 6.5 release, will be considered as technology preview features in Red Hat Gluster Storage 3.1 release. For more information on technology preview features of Red Hat Enterprise Linux 6.5 release, see *Technology Previews* chapter of *Red Hat Enterprise Linux 6.5 Technical Notes*.

Report a bug

4.1. Tiering

Tiering feature improves the performance, and the compliance aspects in a Red Hat Gluster Storage environment. This is achieved by optimizing the placement of the most accessed files on the Faster Storage Medium (fast tier / hot tier) and placing less accessed data to slower storage medium (slow tier / cold tier). It also serves as an enabling technology for other enhancements by combining cost-effective or archivally oriented storage for the majority of user data with high-performance storage to absorb the majority of I/O workload. .

For more information, see chapter *Managing Tiering* in the *Red Hat Gluster Storage Administration Guide*.

Report a bug

4.2. gstatus Command

The **`gstatus`** command provides an easy-to-use, high-level view of the health of a trusted storage pool with a single command. It gathers information by executing the GlusterFS commands, to gather information about the statuses of the Red Hat Gluster Storage nodes, volumes, and bricks.

For more information, see section *gstatus command* in the *Red Hat Gluster Storage Administration Guide*.

Report a bug

4.3. Replicated Volumes with Replica Count greater than 3

The replicated volumes create copies of files across multiple bricks in the volume. It is recommended that you use replicated volumes in environments where high-availability and high reliability are critical. Creating replicated volumes with replica count more than 3 is under technology preview.

For more information, refer the section *Creating Replicated Volumes* in the *Red Hat Gluster Storage 3.1 Administration Guide*.

Report a bug

4.4. Stop Remove Brick Operation

You can stop a remove brick operation after you have opted to remove a brick through the Command Line Interface and Red Hat Gluster Storage Console. After executing a remove-brick operation, you can choose to stop the remove-brick operation by executing the **`remove-brick stop`** command. The files that are already migrated during remove-brick operation, will not be reverse migrated to the original brick.

For more information, refer the section *Stopping Remove Brick Operation* in the *Red Hat Gluster Storage 3.1 Administration Guide* and section *Stopping a Remove Brick Operation* in the *Red Hat Gluster Storage 3.1 Console Administration Guide.*

Report a bug

4.5. Read-only Volume

Red Hat Gluster Storage enables you to mount volumes with read-only permission. While mounting the client, you can mount a volume as read-only and also make the entire volume as read-only, which applies for all the clients using the **`volume set`** command.

Report a bug

4.6. Snapshot Clone

You can now create a clone of a snapshot. This is a writable clone and behaves like a regular volume. A new volume can be created from a particular snapshot clone.

For more information, see chapter *Managing Snapshots* in the *Red Hat Gluster Storage Administration Guide*.

4.7. pNFS

The Parallel Network File System (pNFS) is part of the NFS v4.1 protocol that allows compute clients to access storage devices directly and in parallel. .

For more information, see section *NFS Ganesha* in the *Red Hat Gluster Storage Administration Guide*.

4.8. Non Uniform File Allocation

When a client on a server creates files, the files are allocated to a brick in the volume based on the file name. This allocation may not be ideal, as there is higher latency and unnecessary network traffic for read/write operations to a non-local brick or export directory. NUFA ensures that the files are created in the local export directory of the server, and as a result, reduces latency and conserves bandwidth for that server accessing that file. This can also be useful for applications running on mount points on the storage server.

If the local brick runs out of space or reaches the minimum disk free limit, instead of allocating files to the local brick, NUFA distributes files to other bricks in the same volume if there is space available on those bricks. You must enable NUFA before creating any data in the volume.

Report a bug

Appendix A. Revision History

Revision 3.1-4	**Mon Aug 14 2015**	**Divya Muntimadugu**
Fixed BZ# 1253132		
Revision 3.1-2	**Wed Jul 1 2015**	**Divya Muntimadugu**
Version for 3.1 GA release		