



# **Red Hat Gluster Storage 3.1**

## **3.1 Update 2 Release Notes**

Release Notes for Red Hat Gluster Storage - 3.1 Update 2

Edition 1

Last Updated: 2017-10-13



# Red Hat Gluster Storage 3.1 3.1 Update 2 Release Notes

---

Release Notes for Red Hat Gluster Storage - 3.1 Update 2  
Edition 1

Red Hat Gluster Storage Documentation Team  
Red Hat Customer Content Services

## Legal Notice

Copyright © 2015-2016 Red Hat Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

These release notes provide high-level coverage of the improvements and additions that have been implemented in Red Hat Gluster Storage 3.1 Update 2.

---

## Table of Contents

<b>CHAPTER 1. INTRODUCTION</b>	<b>3</b>
<b>CHAPTER 2. WHAT CHANGED IN THIS RELEASE?</b>	<b>4</b>
2.1. WHAT'S NEW IN THIS RELEASE?	4
2.2. DEPRECATED FEATURES	5
<b>CHAPTER 3. KNOWN ISSUES</b>	<b>6</b>
3.1. RED HAT GLUSTER STORAGE	6
3.2. RED HAT GLUSTER STORAGE CONSOLE	32
3.3. RED HAT GLUSTER STORAGE AND RED HAT ENTERPRISE VIRTUALIZATION INTEGRATION	38
3.4. RED HAT GLUSTER STORAGE AND RED HAT OPENSTACK INTEGRATION	38
<b>CHAPTER 4. TECHNOLOGY PREVIEWS</b>	<b>40</b>
4.1. RESTFUL VOLUME MANAGEMENT WITH HEKETI	40
4.2. REPLICATED VOLUMES WITH REPLICA COUNT GREATER THAN 3	40
4.3. STOP REMOVE BRICK OPERATION	40
4.4. READ-ONLY VOLUME	41
4.5. PNFS	41
4.6. NON UNIFORM FILE ALLOCATION	41
<b>APPENDIX A. REVISION HISTORY</b>	<b>42</b>



# 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 with Amazon Web Services (AWS), Microsoft Azure, or Google 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 for deploying scalable NAS in AWS, Microsoft Azure, and Google Cloud. This powerful storage server provides a highly available, scalable, virtualized, and centrally managed pool of storage for users of these public cloud providers.

## CHAPTER 2. WHAT CHANGED IN THIS RELEASE?

### 2.1. WHAT'S NEW IN THIS RELEASE?

This section describes the key features and enhancements in the Red Hat Gluster Storage 3.1 Update 2 release.

#### Technology Preview: RESTful Volume Management with Heketi

Heketi provides a RESTful management interface for managing Red Hat Gluster Storage volume life cycles. This interface allows cloud services like OpenStack Manila, Kubernetes, and OpenShift to dynamically provision Red Hat Gluster Storage volumes. For details about this technology preview, see the *Red Hat Gluster Storage 3.1 Administration Guide*

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Storage/3.1/html/Administration\\_Guide/ch06s02.html](https://access.redhat.com/documentation/en-US/Red_Hat_Storage/3.1/html/Administration_Guide/ch06s02.html).

#### Tiering

Red Hat Gluster Storage now provides the ability to automatically classify and migrate files based on how frequently those files are accessed. This allows frequently accessed files to be migrated to higher performing disks (the hot tier), and rarely accessed files to be stored on disks with lower performance (the cold tier). This enables faster response times, reduced latency, greater storage efficiency, and reduced deployment and operating costs. For more information about tiering, see the *Red Hat Gluster Storage 3.1 Administration Guide*: [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Storage/3.1/html/Administration\\_Guide/chap-Managing\\_Data\\_Tiering.html](https://access.redhat.com/documentation/en-US/Red_Hat_Storage/3.1/html/Administration_Guide/chap-Managing_Data_Tiering.html).

#### Writable Snapshots

Red Hat Gluster Storage snapshots can now be cloned and made writable by creating a new volume based on an existing snapshot. Clones are space efficient, as the cloned volume and original snapshot share the same logical volume back end, only consuming additional space as the clone diverges from the snapshot. For more information, see the *Red Hat Gluster Storage 3.1 Administration Guide*: [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Storage/3.1/html/Administration\\_Guide/](https://access.redhat.com/documentation/en-US/Red_Hat_Storage/3.1/html/Administration_Guide/).

#### Red Hat Gluster Storage for Containers

As of Red Hat Gluster Storage 3.1 Update 2, Red Hat Gluster Storage can now be set up in a container on either Red Hat Enterprise Linux Atomic Host 7.2 or Red Hat Enterprise Linux Server 7.2. Containers use the shared kernel concept and use system resources more efficiently than hypervisors. Containers rest on top of a single Linux instance and allow applications to use the same Linux kernel as the system that they are running on. This improves the overall efficiency of the system and reduces space consumption.

#### BitRot scrubber status

The BitRot scrubber command (**gluster volume bitrot VOLNAME scrub status**) can now display scrub process statistics and list identified corrupted files, allowing administrators to locate and repair corrupted files more easily. See the *Red Hat Gluster Storage 3.1 Administration Guide* for details: [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Storage/3.1/html/Administration\\_Guide/chap-Detecting\\_Data\\_Corruption.html](https://access.redhat.com/documentation/en-US/Red_Hat_Storage/3.1/html/Administration_Guide/chap-Detecting_Data_Corruption.html).

#### Samba Asynchronous I/O enabled by default

Red Hat Gluster Storage now supports and enables asynchronous I/O with Samba by default (**aio read size = 4096**). Asynchronous I/O can enable increased throughput on multi-threaded clients, or when multiple programs access the same share. This improves default performance for most



users of Samba and Red Hat Gluster Storage.

### Console Virtual Appliance

Red Hat Gluster Storage Console now provides a virtual appliance that can be used to quickly set up a pre-installed and partially configured Red Hat Gluster Storage Console. This also enables offline installation of the Red Hat Gluster Storage Console on virtual machines managed by Red Hat Enterprise Virtualization Management. See the *Red Hat Gluster Storage 3.1 Console Installation Guide* for details: [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Storage/3.1/html/Console\\_Installation\\_Guide/chap-Red\\_Hat\\_Storage\\_Console\\_Installation-OVA.html](https://access.redhat.com/documentation/en-US/Red_Hat_Storage/3.1/html/Console_Installation_Guide/chap-Red_Hat_Storage_Console_Installation-OVA.html)

## 2.2. DEPRECATED FEATURES

The following features are considered deprecated as of Red Hat Gluster Storage 3.1 Update 2. See each item for details about the likely removal timeframe of the feature.

### Hortonworks Data Platform (HDP)

Support for Hortonworks Data Platform (HDP) on Red Hat Gluster Storage integrated using the Hadoop Plug-In is deprecated as of Red Hat Gluster Storage 3.1 Update 2, and is unlikely to be supported in the next major release. Red Hat discourages further use of this plug-in for deployments where Red Hat Gluster Storage is directly used for holding analytics data for running in-place analytics. However, Red Hat Gluster Storage can be used as a general purpose repository for holding analytics data and as a companion store where the bulk of the data is stored and then moved to Hadoop clusters for analysis when necessary.

### CTDB 2.5

As of Red Hat Gluster Storage 3.1 Update 2, CTDB version 2.5 is no longer supported. To continue using CTDB in Red Hat Gluster Storage 3.1 Update 2 and later, upgrade to CTDB version 4, provided in the following channels and repositories:

- RHN channel for Red Hat Enterprise Linux 6: **rhel-x86\_64-server-6-rh-gluster-3-samba**
- RHN channel for Red Hat Enterprise Linux 7: **rhel-x86\_64-server-7-rh-gluster-3-samba**
- Subscription Management repository for Red Hat Enterprise Linux 6: **rh-gluster-3-samba-for-rhel-6-server-rpms**
- Subscription Management repository for Red Hat Enterprise Linux 7: **rh-gluster-3-samba-for-rhel-7-server-rpms**

## 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 Containers

##### BZ#1294776

When a container with one or more logical volumes bind-mounted as bricks is started in Atomic Host, the logical volumes are sometimes unmounted from Atomic Host during container start. This causes problems when the container is re-spawned.

**Workaround:** After starting the Red Hat Gluster Storage container, verify that the mount point is still mounted in the Atomic Host by checking the output of **df -h**. If the mount point is not mounted, ensure that it is configured in **/etc/fstab** and remount it on Atomic Host by running **mount -a**.

#### Issues related to Tiering

##### BZ#1294790 , BZ#1294808

Currently, the tier process performs a fix-layout operation on the entire volume every time it starts. Tier migration operations only begin after the fix-layout operation is complete. This means that in some circumstances, such as when extremely large amounts of data are present on the volume immediately before tiering is enabled, the fix-layout operation can take a long time to complete and prevents file promotion to the hot tier until after the fix-layout operation has completed.

##### BZ#1303298

When a **readdirp** call is performed on a USS (User Serviceable Snapshot) as part of a request to list the entries on a snapshot of a tiered volume, the USS provides the wrong stat for files in the cold tier. This results in incorrect permissions being applied to the mount point, and files appear to have **-----T** permissions.

**Workaround:** FUSE clients can work around this issue by applying any of the following options:

- **use-readdirp=no** (recommended)
- **attribute-timeout=0**
- **entry-timeout=0**

NFS clients can work around the issue by applying the **noac** option.

##### BZ#1300679

If the hot and cold tiers in a tiered volume have the same number of sub-volumes, the first group of files migrated in a single cycle is likely to be migrated to the same sub-volume on the hot tier rather than being distributed across multiple sub-volumes. This is particularly noticeable when the files that are candidates for migration exceed the number defined by **tier-max-files** or the size defined by **tier-max-mb**.

##### BZ#1302968

The defrag variable is not being reinitialized during glusterd restart. This means that if glusterd fails while the following processes are running, it does not reconnect to these processes after restarting:

- rebalance
- tier
- remove-brick

This results in these processes continuing to run without communicating with glusterd. Therefore, any operation that requires communication between these processes and glusterd fails.

**Workaround:** Stop or kill the rebalance, tier, or remove-brick process before restarting glusterd. This ensures that a new process is spawned when glusterd restarts.

#### BZ#1303045

When a tier is attached while I/O is occurring on an NFS mount, I/O pauses temporarily, usually for between 3 to 5 minutes. If I/O does not resume within 5 minutes, use the **gluster volume start volname force** command to resume I/O without interruption.

#### BZ#1273741

Files with hard links are not promoted or demoted on tiered volumes.

#### BZ#1305490

A race condition between tier migration and hard link creation results in the hard link operation failing with a 'File exists' error, and logging 'Stale file handle' messages on the client. This does not impact functionality, and file access works as expected.

This race occurs when a file is migrated to the cold tier after a hard link has been created on the cold tier, but before a hard link is created to the data on the hot tier. In this situation, the attempt to create a hard link on the hot tier fails. However, because the migration converts the hard link on the cold tier to a data file, and a linkto already exists on the cold tier, the links exist and work as expected.

#### BZ#1277112

When hot tier storage is full, write operations such as file creation or new writes to existing files fail with a 'No space left on device' error, instead of redirecting writes or flushing data to cold tier storage.

**Workaround:** If the hot tier is not completely full, it is possible to work around this issue by waiting for the next CTR promote/demote cycle before continuing with write operations.

If the hot tier does fill completely, administrators can copy a file from the hot tier to a safe location, delete the original file from the hot tier, and wait for demotion to free more space on the hot tier before copying the file back.

#### BZ#1278391

Migration from the hot tier fails when the hot tier is completely full because there is no space left to set the extended attribute that triggers migration.

#### BZ#1283507

Corrupted files can be identified for promotion and promoted to hot tier storage.

In rare circumstances, corruption can be missed by the BitRot scrubber. This can happen in two ways:

1. A file is corrupted before its checksum is created, so that the checksum matches the corrupted file, and the BitRot scrubber does not mark the file as corrupted.
2. A checksum is created for a healthy file, the file becomes corrupted, and the corrupted file is not compared to its checksum before being identified for promotion and promoted to the hot tier, where a new (corrupted) checksum is created.

When tiering is in use, these unidentified corrupted files can be 'heated' and selected for promotion to the hot tier. If a corrupted file is migrated to the hot tier, and the hot tier is not replicated, the corrupted file cannot be accessed or migrated back to the cold tier.

#### BZ#1283957

When volume status or volume tier status is requested for a tiered volume, the status of all nodes in the storage pool is listed as in progress, even when a node is not part of the tiered volume. This occurs because the tier daemon runs on all nodes of the trusted storage pool, and reports status for every volume in the trusted storage pool.

### Issues related to Snapshot

#### 1306917

When a User Serviceable Snapshot is enabled, attaching a tier succeeds, but any I/O operations in progress during the attach tier operation may fail with stale file handle errors.

**Workaround:** Disable User Serviceable Snapshots before performing **attach tier**. Once **attach tier** has succeeded, User Serviceable Snapshots can be enabled.

#### BZ#1309209

When a cloned volume is deleted, its brick paths (stored under `/run/gluster/snaps`) are not cleaned up correctly. This means that attempting to create a clone that has the same name as a previously cloned and deleted volume fails with a Commit failed message.

**Workaround:** After deleting a cloned volume, ensure that brick entries in `/run/gluster/snaps` are unmounted and deleted, and that their logical volumes are removed.

#### 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. To avoid this issue, ensure that you do not perform any file system operations on the snapshot that is about to be deleted.

#### BZ#1308837

When a tiered volume with quota enabled is snapshotted, and that snapshot is cloned, rebooting the node or restarting glusterd on the node can result in that node entering a peer rejected state. This occurs because the quota checksum is not being copied as part of the snapshot or clone operations.

#### Workaround:

1. Check glusterd logs for a quota checksum mismatch error, which looks similar to the following:

```
E [MSGID: 106012] [glusterd-  
utils.c:2845:glusterd_compare_friend_volume] 0-management: Cksums
```

```
of quota configuration of volume volname differ. local cksum =
1405646976, remote cksum = 0 on peer peername
```

2. If the volume with this error is cloned, edit the `/var/lib/glusterd/vols/volname/info` file for that volume and increase the value in the ***version*** field by one.
3. Restart glusterd on the node.

### 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 Kernel 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
```

Note that this can cause temporary performance degradation.

### 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 and use the following command to drop the VFS cache:

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

Then move back into the **.snaps** folder. Note that this command can cause temporary performance degradation.

### BZ#1170365

Virtual inode numbers are generated for all the files in the **.snaps** directory. Any hard links are assigned different inode numbers instead of the same inode number.

### BZ#1170502

When the User Serviceable Snapshot feature is enabled, 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 that mount the volume.

**BZ#1133861**

New snap bricks fails to start if the total snapshot brick count in a node goes beyond 1K. Until this bug is corrected, Red Hat recommends deactivating unused snapshots to avoid hitting the 1K limit.

**BZ#1126789**

If any node or **glusterd** service is down when snapshot is restored then any subsequent snapshot creation fails. Red Hat recommends not restoring a snapshot while the node or the **glusterd** service is unavailable.

**BZ#1139624**

While taking a snapshot of a gluster volume, Red Hat Gluster Storage creates another volume which is similar to the original volume. All volumes, including snapshot volumes, consume some memory when started. This can create an out of memory state when creating a snapshot on a system with low memory. Red Hat recommends deactivating unused snapshots to reduce the memory footprint of the system and avoid this issue.

**BZ#1129675**

Performing a snapshot restore while **glusterd** is not available in a cluster node or a node is unavailable results in the following errors:

- Executing the **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. Start **glusterd** by running 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 a large number of write or read operations are 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**

Executing the **gluster volume info** command displays system limits (**snap-max-hard-limit** and **snap-max-soft-limit**) instead of volume limits, and also displays snapshot auto-delete values.

**BZ#1111479**

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

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

**BZ#1092510**

If you create a snapshot when the rename of a directory is in progress (that is, renaming is 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 files and directories changes on 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. If you do not require Gluster to work with RDMA transport, these errors can be safely ignored.

**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. For 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#1266874

Rebalance operation tries to start the gluster volume before doing the actual rebalance. In most of the cases, volume is already in **Started** state. If the volume is already started and the volume start command fails, gdeploy assumes that volume has started and does not start the rebalance process.

**Workaround:** Rebalance in gdeploy is possible only for stopped volumes.

#### BZ#1110282

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

#### BZ#1140531

Extended attributes set on a file while it is being migrated during a rebalance operation are lost.

**Workaround:** Reset the extended attributes on the file once the migration is complete.

#### 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#1293634

Sync performance for geo-replicated storage is reduced when the master volume is tiered, resulting in slower geo-rep performance on tiered volumes.

#### BZ#1302320

During file promotion, the rebalance operation sets the sticky bit and suid/sgid bit. Normally, it removes these bits when the migration is complete. If `readdirp` is called on a file before migration completes, these bits are not removed, and remain applied on the client.

This means that, if `rsync` happens while the bits are applied, the bits remain applied to the file as it is synced to the destination, impairing accessibility on the destination. This can happen in any geo-replicated configuration, but the likelihood increases with tiering because the rebalance process is continuous.

#### BZ#1286587

When geo-replication is in use alongside tiering, bricks attached as part of a tier are incorrectly set to passive. If geo-replication is subsequently restarted, these bricks can become faulty.

**Workaround:** Stop geo-replication session prior to attaching or detaching bricks that are part of a tier.

#### To attach a tier:

1. Stop geo-replication:

```
# gluster volume geo-replication master_vol slave_host::slave_vol
stop
```

2. Attach the tier:

```
# gluster volume attach-tier master_vol replica 2
server1:/path/to/brick1 server2:/path/to/brick2
```

3. Restart geo-replication:

```
# gluster volume geo-replication master_vol slave_host::slave_vol
start
```

4. Verify that bricks in tier are available in geo-replication session:

```
# gluster volume geo-replication master_vol slave_host::slave_vol
status
```

#### To detach a tier:

1. Begin the tier detachment process:

```
# gluster volume detach-tier master_vol start
```

2. Ensure all data in that tier is synced to the slave:

```
# gluster volume geo-replication master_vol slave_host::slave_vol
config checkpoint now
```

3. Monitor checkpoint until displayed status is **checkpoint as of time of checkpoint creation is completed at time**.

```
# gluster volume geo-replication master_vol slave_host::slave_vol
status detail
```

4. Verify that detachment is complete:

```
# gluster volume detach-tier master_vol status
```

5. Stop geo-replication:

```
# gluster volume geo-replication master_vol slave_host::slave_vol
stop
```

6. Commit tier detachment:

```
# gluster volume detach-tier master_vol commit
```

7. Verify tier is detached:

```
# gluster volume info master_vol
```

8. Restart geo-replication:

```
# gluster volume geo-replication master_vol slave_host::slave_vol
start
```

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:

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

```
cp -a <brick_hptime_path> <backup_path>
```



#### NOTE

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

For example:

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

3. Run the following command to replace the **OLD** path in the htime file(s) with the new brick path, where *OLD\_BRICK\_PATH* is the brick path of the current volume, and *NEW\_BRICK\_PATH* is the brick path after snapshot restore.

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

For example:

```
find
/var/run/gluster/snaps/a4e2c4647cf642f68d0f8259b43494c0/brick0/b0/
.glusterfs/changelogs/htime/ -name 'HTIME.*' -print0 | \
xargs -0 sed -ci
's|/bricks/brick0/b0/|/var/run/gluster/snaps/a4e2c4647cf642f68d0f8
259b43494c0/brick0/b0/|g'
```

4. 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. Contact Red Hat Support for assistance with this issue.

## 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.

### 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#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.

### Issues related to NFS

- After you restart the NFS server, the unlock within the grace-period feature may fail and the locks held 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#1259402

When **vdsmd** and **abrt** are installed alongside each other, **vdsmd** overwrites **abrt** core dump configuration in **/proc/sys/kernel/core\_pattern**. This prevents NFS-Ganesha from generating core dumps.

**Workaround:** Disable core dumps in **/etc/vdsm/vdsm.conf** by setting **core\_dump\_enable** to **false**, and then restart the **abrt-ccpp** service:

```
# systemctl restart abrt-ccpp
```

### BZ#1257548

**nfs-ganesha** service monitor script which triggers IP failover runs periodically every 10 seconds. The ping-timeout of the glusterFS server (after which the locks of the unreachable client gets flushed) is 42 seconds by default. After an IP failover, some locks may not get cleaned by the glusterFS server process, hence reclaiming the lock state by NFS clients may fail.

**Workaround:** It is recommended to set the **nfs-ganesha** service monitor period interval (default 10sec) at least as twice as the Gluster server ping-timeout (default 42sec).

Hence, either you must decrease the network ping-timeout using the following command:

```
# gluster volume set <volname> network.ping-timeout <ping_timeout_value>
```

or increase nfs-service monitor interval time using the following commands:

```
# pcs resource op remove nfs-mon monitor
```

```
# pcs resource op add nfs-mon monitor interval=<interval_period_value>
timeout=<timeout_value>
```

### 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#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:**

1. Turn off the **nfs.disable** option for the volume:

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

2. 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#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**.

### 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#1306656

When management encryption is enabled, and a volume is started before **glusterd** has been started on all nodes in the cluster, the bricks on late-starting nodes are assigned different ports. This results in the bricks being inaccessible, as the new ports are blocked by the firewall.

**Workaround:** When management encryption is enabled, ensure **glusterd** is started on all nodes before starting volumes.

### BZ#1311362

Red Hat Gluster Storage 3.1 Update 2 adds a new directory (**`brick_path/.glusterfs/indices/dirty`**) to assist with internal maintenance. Version 3.1 Update 2 incorrectly expects this directory to be present when running commands, even on nodes with older Red Hat Gluster Storage versions, resulting in misleading output.

When a node with Red Hat Gluster Storage 3.1 Update 2 is used to run the **`gluster volume heal volname`** command on older nodes, the output of a **`gluster volume heal info`** command run from the new node contains the following message, even though all entries were processed:

```
Failed to process entries completely
```

**Workaround:** If not all nodes in your cluster have been updated to Red Hat Gluster Storage 3.1 Update 2, you can perform either of the following actions to work around the issue.

- Use older nodes to review heal info output.
- For each brick, check that the only index entry listed under the **`brick_path/.glusterfs/indices/xattrop`** directory is **`xattrop-*`**.

### BZ#1304585

When quota is disabled on a volume, a cleanup process is initiated to clean up the extended attributes used by quota. If this cleanup process is still in progress when quota is re-enabled, extended attributes for the newly enabled quota can be removed by the cleanup process. This has negative effects on quota accounting.

### BZ#1306907

During an inode forget operation, files under the quarantine directory are removed. The inode forget operation is called during the unlinking of a file, and when the inode table's LRU (Least Recently Used) cache size exceeds 16 KB. This means that, when a corrupted file is not accessed for a long time, and the LRU cache exceeds 16 KB, the corrupted file will be removed from the quarantine directory. This results in the corrupted file not being shown in BitRot status output, even though the corrupted file has not been deleted from the volume itself.

### 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#1260779

In a distribute-replicate volume, the **getfattr -n replica.split-brain-status <path-to-dir>** command on mount-point might report that the directory is not in split-brain even though it is.

**Workaround:** To know the split-brain status of a directory, run the following command:

```
# gluster v heal <volname> info split-brain
```

### 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:**

1. You can either remove the brick directories or remove the glusterfs-related extended attributes.
2. 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 **/var/lib/glusterd/glusterfind** directory.

### 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 clean up the session directory, for example, `/var/lib/glusterd/glusterfind/SESSION/VOLNAME`.

#### 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 has not 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 will then correctly populate.

#### 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#1297442

Currently, attempting to run the `gluster volume get volname user.option` command fails because the volume get command does not display user option values in its output.

**Workaround:** Run the `gluster volume info volname` command on the same volume to see the value of any user options.

#### 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>**, check if the file's **gfid** handle is not deleted by running the following command and checking whether the file appears in the output, for example, **<brick-path>/[.glusterfs/58/48/5848899c-b6da-41d0-95f4-64ac85c87d3f](#)**.

```
# find <brick-path>/glusterfs -type f -links 1
```

If the file appears in the output of this command, delete the file using the following command.

```
# 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
```

[illegible]

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.

1. Delete `trusted.ec.size`.

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

2. 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 Red Hat Gluster Storage Server

**BZ#1306667**

If server-side quorum is enabled, and the quorum conditions are not met, starting a volume should fail. Currently, executing **gluster volume start** incorrectly succeeds even when quorum conditions are not met. However, because stopping the volume is correctly dependent on quorum conditions being met, this means that attempts to stop the volume fail while quorum is enabled.

**Workaround:**

1. Disable server-side quorum:

```
# gluster volume reset volname cluster.server-quorum-type
```

2. Stop the volume.

3. Re-enable server-side quorum.

```
# gluster volume set volname cluster.server-quorum-type server
```

**BZ#1298955**

When Red Hat Gluster Storage is set up with a server version of 3.1 Update 1 and a client version of 3.0 Update 4, attempting to set any option with the **volume set** command fails with the following error:

```
volume set: failed: One or more connected clients cannot support the
feature being set. These clients need to be upgraded or disconnected
before running this command again
```

When operating correctly, this restriction is in place to prevent newer features from being enabled on a volume when the clients in use cannot support the feature. Currently, the restriction check is incorrect, and will prevent even valid, supported options from being set.

**Workaround:** Upgrade all clients to the same version of Red Hat Gluster Storage as the server.

**BZ#1266824**

After an ISO installation, the **ntpd** service does not start by default on Red Hat Enterprise Linux 7. The server is out of sync with the rest of the cluster. This is visible if there is a huge difference between the current date and the system time.

**Workaround:** You must configure the **ntpd** service manually after installation. Execute the following commands to enable and start the **ntpd** service:

```
# systemctl enable ntpd
```

```
# systemctl start ntpd
```

**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#1300572

Due to a bug in the Linux CIFS client, SMB2.0+ connections from Linux to Red Hat Gluster Storage currently will not work properly. SMB1 connections from Linux to Red Hat Gluster Storage, and all connections with supported protocols from Windows continue to work.

**Workaround:** If practical, restrict Linux CIFS mounts to SMB version 1. The simplest way to do this is to not specify the **vers** mount option, since the default setting is to use only SMB version 1. If restricting Linux CIFS mounts to SMB1 is not practical, disable asynchronous I/O in Samba by setting **aio read size** to 0 in the **smb.conf** file. Disabling asynchronous I/O is not generally recommended and may negatively impact performance on other clients.

### BZ#1282452

Attempting to upgrade to ctdb version 4 fails when ctdb2.5-debuginfo is installed, because the ctdb2.5-debuginfo package currently conflicts with the samba-debuginfo package.

**Workaround:** Manually remove the ctdb2.5-debuginfo package before upgrading to ctdb version 4. If necessary, install samba-debuginfo after the upgrade.

### 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.

## Issues related to SELinux

### BZ#1294762

When the Red Hat Gluster Storage Container is deployed on Red Hat Enterprise Atomic Host, SELinux policy labels the `/var/log/glusterfs` directory as `svirt_sandbox_file_t`. Logrotate cannot run on files with this label, and logs AVC denials when log rotation is attempted on files in `/var/log/glusterfs`. This means that Red Hat Gluster Storage logs cannot currently be rotated, and could potentially fill up and consume a large amount of storage as a result. Correcting this requires updates to the `selinux-policy` package. In the meantime, you can work around this issue by resetting the label of `/var/log/glusterfs` after the host volume is bind mounted inside the container.

#### Workaround:

1. Start the container and bind mount the host volume:

```
# docker run ... -v /var/log/glusterfs:/var/log/glusterfs:z ...  
image_name  
# docker exec -it container_id /bin/bash
```

2. In the container, run the following command to manually apply the appropriate SELinux label.

```
# chcon -Rt glusterd_log_t /var/log/glusterfs
```

Note that this workaround cannot persist to subsequent docker runs, and must be performed for each docker run.

### BZ#1256635

Red Hat Gluster Storage does not currently support SELinux Labeled mounts.

On a FUSE mount, SELinux cannot currently distinguish file systems by subtype, and therefore cannot distinguish between different FUSE file systems (BZ#1291606). This means that a client-specific policy for Red Hat Gluster Storage cannot be defined, and SELinux cannot safely translate client-side extended attributes for files tracked by Red Hat Gluster Storage.

A workaround is in progress for NFS-Ganesha mounts as part of BZ#1269584. When complete, BZ#1269584 will enable Red Hat Gluster Storage support for NFS version 4.2, including SELinux Labeled support.

### BZ#1290514 , BZ#1292781

Current SELinux policy prevents the use of the **ctdb enablescript** and **ctdb disablescript** commands.

**Workaround:** Instead of running **ctdb disablescript script**, run **chmod -x /etc/ctdb/events.d/script** as the root user. Instead of running **ctdb enablescript script**, run **chmod +x /etc/ctdb/events.d/script** as the root user.

### BZ#1291194 , BZ#1292783

Current SELinux policy prevents ctdb's 49.winbind event script from executing smbcontrol. This can create inconsistent state in winbind, because when a public IP address is moved away from a node, winbind fails to drop connections made through that IP address.

## General issues

### BZ#1303125

The defrag variable is not being reinitialized during glusterd restart. This means that if glusterd fails while the following processes are running, it does not reconnect to these processes after restarting:

- rebalance
- tier
- remove-brick

This results in these processes continuing to run without communicating with glusterd. Additionally, glusterd does not retain the `decommission_is_in_progress` flag that is set to indicate that the rebalance process is running.

If glusterd fails and restarts on a node where remove-brick was triggered and the rebalance process is not yet complete, but the rebalance process on other nodes has already completed, then the remove-brick commit operation succeeds because glusterd cannot identify that there is an ongoing rebalance operation on the node. This can result in data loss.



**Workaround:**

- Stop or kill the **rebalance** process before restarting **glusterd**. This ensures that a new **rebalance** process is spawned when **glusterd** restarts.
- On the node on which **glusterd** restarted, check the status of the remove-brick process. Only execute the **remove-brick commit** command when **remove-brick status** shows that data migration is complete.

**BZ#1290653**

When the **gluster volume status all tasks** command is executed, messages like the following are recorded in the **glusterd** log.

```
Failed to aggregate response from node/brick
```

This error is logged erroneously, and can be safely ignored.

**GFID mismatches cause errors**

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#1260119**

**glusterfind** command must be executed from one node of the cluster. If all the nodes of cluster are not added in **known\_hosts** list of the command initiated node, then **glusterfind create** command hangs.

**Workaround:** Add all the hosts in peer including local node to **known\_hosts**.

**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
```

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.

**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](#)

**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 images as the required ownership is not available.



**Workaround:** Before migration, power off the VMs. When migration is complete, restore the ownership of the VM Disk Image (107:107) and start the VMs.

### Concurrent volume and peer management

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 older versions 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#969020

Renaming a file during remove-brick operation may cause the file not to get migrated from the removed brick.

**Workaround:** Check the removed brick for any files that might not have been migrated and copy those to the gluster volume before decommissioning the brick.

#### 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. For example, when probing node B with a 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:

1. To allow **insecure** connections to a volume, run the following command:

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

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

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

- 3. Restart **glusterd** process using the following command:

```
# service glusterd restart
```

- 4. 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#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.

### Issues related to Red Hat Gluster Storage AMI

#### BZ#1267209

The `redhat-storage-server` package is not installed by default in a Red Hat Gluster Storage Server 3 on Red Hat Enterprise Linux 7 AMI image.

**Workaround:** It is highly recommended to manually install this package using `yum`.

```
# yum install redhat-storage-server
```

The `redhat-storage-server` package primarily provides the `/etc/redhat-storage-release` file, and sets the environment for the storage node.

## 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** expects this key in a dictionary which is 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

### BZ#1303566

When a user selects the auto-start option in the **Create Geo-replication Session** user interface, the `use_meta_volume` option is not set. This means that the geo-replication session is started without a metadata volume, which is not a recommended configuration.

**Workaround:** After session start, go to the geo-replication options tab for the master volume and set the `use_meta_volume` option to `true`.

### 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#990108**

Resetting the `user.cifs` option using the **Create Volume** operation on the **Volume Options** tab on the Red Hat Gluster Storage Console reports a failure.

**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 will 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:

1. When the `current_scheduler` value is set as oVirt in Gluster.
2. 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 Gluster 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:**

1. Ensure that **/etc/fstab** entries are removed when the Logical Volumes are deleted from system.
2. 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#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:**

1. The graph with y axis legend as **%(Total: \*\* GiB/Tib)** is the graph for total space used.

2. 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.

**BZ#1134319**

When run on versions higher than Firefox 17, the Red Hat Storage Console login page displays a browser incompatibility warning.

## 3.3. RED HAT GLUSTER STORAGE AND RED HAT ENTERPRISE VIRTUALIZATION INTEGRATION

**BZ#1293412**

When self-heal is required after an inode refresh, the file operation that triggered the heal process is served after the heal completes. This means that large files, such as virtual machine images, become unresponsive until the heal is complete and the file operation returns.

**Workaround:** Disable client-side self-heal to prevent file operations from blocking file access. Pending heals will still be tracked by the self-heal daemon.

```
# gluster volume set volname cluster.data-self-heal off
```

**All images in data center displayed regardless of context**

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

**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, 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 this 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 6.7, 7.1, and 7.2 are also considered technology preview features in Red Hat Gluster Storage 3.1. For more information on the technology preview features available in Red Hat Enterprise Linux 6.7, see the *Technology Previews* chapter of the *Red Hat Enterprise Linux 6.7 Technical Notes*: [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/html/6.7\\_Technical\\_Notes/technology-previews.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/6.7_Technical_Notes/technology-previews.html)

### 4.1. RESTFUL VOLUME MANAGEMENT WITH HEKETI

Heketi provides a RESTful management interface for managing Red Hat Gluster Storage volume life cycles. This interface allows cloud services like OpenStack Manila, Kubernetes, and OpenShift to dynamically provision Red Hat Gluster Storage volumes. For details about this technology preview, see the *Red Hat Gluster Storage 3.1 Administration Guide*

### 4.2. 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, see the *Red Hat Gluster Storage 3.1 Administration Guide*:

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Storage/3.1/html/Administration\\_Guide/sect-Creating\\_Replicated\\_Volumes.html](https://access.redhat.com/documentation/en-US/Red_Hat_Storage/3.1/html/Administration_Guide/sect-Creating_Replicated_Volumes.html)

### 4.3. 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, see the *Red Hat Gluster Storage 3.1 Administration Guide*:

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Storage/3.1/html/Administration\\_Guide/sect-Shrinking\\_Volumes.html#Stopping\\_a\\_remove-brick\\_Operation](https://access.redhat.com/documentation/en-US/Red_Hat_Storage/3.1/html/Administration_Guide/sect-Shrinking_Volumes.html#Stopping_a_remove-brick_Operation).

## 4.4. 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.

## 4.5. 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 the *Red Hat Gluster Storage 3.1 Administration Guide*:

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Storage/3.1/html/Administration\\_Guide/sect-NFS.html#sect-NFS\\_Ganesha](https://access.redhat.com/documentation/en-US/Red_Hat_Storage/3.1/html/Administration_Guide/sect-NFS.html#sect-NFS_Ganesha)

## 4.6. 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.

## APPENDIX A. REVISION HISTORY

<b>Revision 3.1-23</b>	<b>Tue Apr 05 2016</b>	<b>Laura Bailey</b>
Updating to remove incorrectly listed known issue BZ#1226995.		
<b>Revision 3.1-22</b>	<b>Fri Feb 26 2016</b>	<b>Laura Bailey</b>
Publishing for RHGS 3.1 Update 2 release. Minor corrections to known issue and feature descriptions.		
<b>Revision 3.1-17</b>	<b>Fri Feb 19 2016</b>	<b>Laura Bailey</b>
Corrected known issue summaries. Added to and corrected new feature summaries. Updated intro to include additional public cloud vendors.		
<b>Revision 3.1-14</b>	<b>Mon Feb 15 2016</b>	<b>Laura Bailey</b>
Corrected summary for known issue BZ#1283507, BZ#1294762.		
<b>Revision 3.1-9</b>	<b>Mon Feb 08 2016</b>	<b>Laura Bailey</b>
Added known issue descriptions for BZ#1300679, BZ#1303298, and BZ#1303566.		
<b>Revision 3.1-8</b>	<b>Thu Feb 04 2016</b>	<b>Laura Bailey</b>
Added known issues BZ#1302968 and BZ#1300679.		
<b>Revision 3.1-7</b>	<b>Fri Jan 29 2016</b>	<b>Laura Bailey</b>
Noted deprecation of CTDB 2.5 (BZ#1283114).		
<b>Revision 3.1-6</b>	<b>Thu Jan 28 2016</b>	<b>Laura Bailey</b>
Added direct links to referenced sections.		
<b>Revision 3.1-5</b>	<b>Wed Jan 27 2016</b>	<b>Laura Bailey</b>
Added known issues new to version 3.1.2. Noted deprecation of HDP (BZ#1300976). Added new features and technology previews.		
<b>Revision 3.1-2</b>	<b>Thu Jan 21 2016</b>	<b>Laura Bailey</b>
Removing old known issues.		
<b>Revision 3.1-1</b>	<b>Tue Nov 10 2015</b>	<b>Laura Bailey</b>
Initial creation of release notes for Red Hat Gluster Storage 3.1 Update 2 release.		