Red Hat Gluster Storage 3.2

3.2 Release Notes

Release Notes for Red Hat Gluster Storage - 3.2

Edition 1

Last Updated: 2017-11-14

Red Hat Gluster Storage 3.2 3.2 Release Notes

Release Notes for Red Hat Gluster Storage - 3.2
Edition 1

Bhavana Mohan
Red Hat Customer Content Services.
bmohanra@redhat.com

Legal Notice

Abstract

These release notes provide high-level coverage of the improvements and additions that have been implemented in Red Hat Gluster Storage 3.2.

Table of Contents

CHAPTER 1. INTRODUCTION

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

GlusterFS, a key building block of Red Hat Gluster Storage, is based on a stackable user space design and can deliver exceptional performance for diverse workloads. GlusterFS aggregates various storage servers over different network interfaces and connects them to form a single large parallel network file system. The POSIX compatible 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 3.2 release.

Improved Performance with Compound File Operations

Administrators can now enable compound file operations on volumes with the **`cluster.use-compound-fops`** volume option. When this option is enabled, write transactions are compounded resulting in less network activity, improving performance.

For more information, see the *Red Hat Gluster Storage 3.2 Administration Guide* https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.2/html-single/administration_guide/#Configuring_Volume_Options.

md-cache Performance Enhancement

In order to improve the performance of directory operations of Red Hat Gluster Storage volumes, the maximum metadata (stat, xattr) caching time on the client side can now be increased to 10 minutes. This does not compromise the consistency of the cache. The change improves the performance of directory operations of Red Hat Gluster Storage volumes. Significant performance improvements can be achieved in the following workloads by enabling metadata caching:

- Listing of directories (recursive)
- Creating files
- Deleting files
- Renaming files

For more information, see the *Red Hat Gluster Storage 3.2 Administration Guide* https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.2/html-single/administration_guide/#sect-Directory_Operations

Parallel I/O for Dispersed Volumes

The new **`performance.client-io-threads`** volume option enables up to 16 threads to be used in parallel on dispersed (erasure-coded) volumes. Threads are created automatically based on client workload when this option is enabled.

For more information, see the *Red Hat Gluster Storage 3.2 Administration Guide* https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.2/html-single/administration_guide/#chap-Red_Hat_Storage_Volumes-Creating_Dispersed_Volumes_1.

Enhancements to Bitrot

The new **ondemand** option is introduced in this release. With this option, you can start the scrubbing process on demand. When you run **`gluster volume bitrot <VOLNAME> scrub ondemand`** command, the scrubber will start crawling the file system immediately.

For more information, see the *Red Hat Gluster Storage 3.2 Administration Guide* https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.2/html-single/administration_guide/#chap-Detecting_Data_Corruption.

Obtaining Node Information

The `get-state` command is introduced to obtain node information. The command writes information about the specified node to a specified file. Using the command line interface, external applications can invoke the command on all nodes of the trusted storage pool, and parse and collate the data obtained from all these nodes to get an easy-to-use and complete picture of the state of the trusted storage pool in a machine parseable format.

For more information, see the *Red Hat Gluster Storage 3.2 Administration Guide* https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.2/html-single/administration_guide/#obtaining_node_information.

Arbitrated Replicated Volumes

An arbitrated replicated volume, or arbiter volume, is a three-way replicated volume where every third brick is a special type of brick called an arbiter. Arbiter bricks do not store file data; they only store file names, structure, and metadata. The arbiter uses client quorum to compare this metadata with that of the other nodes to ensure consistency in the volume and prevent split-brain conditions. This maintains the consistency of three-way replication but requires far less storage.

For more information, see the *Red Hat Gluster Storage 3.2 Administration Guide*for details: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.2/html-single/administration_guide/#Creating_Arbitrated_Replicated_Volumes.

Multithreaded Self-heal for Erasure Coded Volume

With Red Hat Gluster Storage 3.2, multiple threads on every brick process can scan indices in parallel and trigger heal for those at the same time. It is supported on disperse and distribute-disperse volumes. Increasing the number of heals impacts I/O performance. The `disperse.shd-max-threads` volume option can be used to configure the number of entries that can be self healed in parallel on each disperse. The `disperse.shd-wait-qlength` volume option can be used to configure the maximum number of entries that must be kept in the queue for self-heal daemon threads to take up as soon as any of the threads are free to heal. This value should be changed based on how much memory the self-heal daemon process can use for keeping the next set of entries that need to be healed.

For more information, see the *Red Hat Gluster Storage 3.2 Administration Guide*for details: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.2/html-single/administration_guide/#Configuring_Volume_Options.

gdeploy Enhancements

The gdeploy tool automates system administration tasks such as creating bricks and setting up and mounting volumes. When setting up a fresh cluster, gdeploy could be the preferred choice of cluster setup, as manually executing numerous commands can be error prone. With the Red Hat Gluster Storage 3.2 release, gdeploy now provides NFS-Ganesha, Samba, and SSL setup support in volumes.

For more information, see the *Red Hat Gluster Storage 3.2 Administration Guide*for details: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.2/html-single/administration_guide/#chap-Red_Hat_Storage_Volumes-gdeploy .

glusterd Enhancements

The Red Hat Gluster Storage 3.2 release includes several bug fixes and enhancements for glusterd which now enables to configure larger number of volumes in a trusted storage pool.

Granular Entry Self-heal

When the glanular entry self-heal option is enabled, it stores more granular information about the entries which were created or deleted from a directory while a brick in a replica was down. This

helps in faster self-heal of directories, especially in use cases where directories with large number of entries are modified by creating or deleting entries. If this option is disabled, it only stores that the directory needs heal without information about what entries within the directories need to be healed, and thereby requires entire directory crawl to identify the changes.

For more information, see the *Red Hat Gluster Storage 3.2 Administration Guide*for details: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.2/html-single/administration_guide/#Configuring_Volume_Options .

NFS-Ganesha Enhancements

The NFS-Ganesha package is rebased to the upstream version 2.4.1, which provides several important bug fixes and enhancements. This rebase includes the following enhancements:

- cache_inode replaced with stackable FSAL_MDCACHE.
- support_ex FSAL API extensions to allow associating file descriptors or other FSAL specific information with state_t objects.
- abort() on ENOMEM rather than attempt to continue.
- Proper handling of NFS v3 (NLM) blocked locks.
- netgroup cache.
- Cache open owners.
- Various bug fixes, memory leaks and refcount issue resolution.

For more information, see the *Red Hat Gluster Storage 3.2 Administration Guide*for details: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.2/html-single/administration_guide/#sect-NFS_Ganesha

Geo-replication Enhancements

Resetting synchronization time while deleting the geo-replication session : The geo-replication delete command retains the information about the last synchronized time. Due to this, if the same geo-replication session is recreated, then the synchronization will continue from the time where it was left before deleting the session. For the geo-replication session to not maintain any details about the deleted session, the reset-sync-time option must be used with the delete command. Now, when the session is recreated, it starts synchronization from the beginning just like a new session.

Simplified Secure Geo-replication Setup : The internal mountbroker feature is now enhanced to set the necessary SELinux rules and permissions, create the required directory, and update glusterd.vol files while setting up the secure geo-replication slave. This simplifies the existing way of setting up secure geo-replication slave.

Geo-replication Changelog Log Level : You can now set the log level for the geo-replication changelog. The default log level is set to INFO.

For more information, see the *Red Hat Gluster Storage 3.2 Administration Guide*for details: https://access.redhat.com/documentation/en-us/red_hat_gluster_storage/3.2/html-single/administration_guide/#chap-Managing_Geo-replication

2.2. DEPRECATED FEATURES

The following features are considered deprecated as of Red Hat Gluster Storage 3.2. 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. NOTABLE BUG FIXES

This chapter describes bugs fixed in this release of Red Hat Gluster Storage that have significant impact on users.

BZ#1341934

When a file that had a hard link was marked as bad, the bad file marker was not being removed from the inode. This meant that self-heal attempted to open the file in order to heal it, the heal failed with an EIO error, and the hard link was not recovered. The bad file marker on the inode is now removed during lookup so that files with hard links recover successfully.

BZ#1359619

Previously if glusterd took more than 120 seconds to process 'gluster volume status all clients --xml' then the XML output that was returned was incomplete. This could result in gstatus failures when attempting to parse the incomplete result. This update ensures parseable results are returned.

BZ#1414663

The rpc.statd process runs as the rpcuser user by default. Previously, the /var/lib/nfs/statd directory was not owned by the rpcuser user or group. This meant that when the rpc.statd process started before NFS-Ganesha had started, the rpc.statd process was unable to read or write client state to the /var/lib/nfs/statd directory. This directory and its files are now created with rpcuser as the owning user and group so that the rpc.statd process can access and maintain client state.

BZ#1278336

Previously, during virtual IP failover, the TCP packets sent by a client and received by the server may be out of sequence because of previous failures to close the TCP socket. This could result in mount points becoming unresponsive. Portblock resource agents now 'tickle' TCP connections to ensure that packets are in sequence after failover.

BZ#1344675

Export configuration files were stored separately in each node. This could lead to situations where a volume was being exported differently on different nodes, and therefore developed inconsistencies after failing over to another node. Configuration details are now stored in a shared meta volume so that volumes are exported identically by all nodes.

BZ#1348949

NFS-Ganesha now places ganesha.conf, ganesha-ha.conf, and export configuration files in a shared storage directory to make it easier to manage export configuration details.

BZ#1348954

The HA_VOL_SERVER parameter in the ganesha-ha.conf file is no longer used by Red Hat Gluster Storage.

BZ#1333885

When a client attempted to connect using SSL and the connection failed, the client identifier was not part of the log message. The client identifier is now included in the log message to make it easier to determine which client was attempting to connect.

BZ#1340608

Red Hat Gluster Storage now provides support for Samba to enable Transport Layer Security (SSL) on a management connection between the smbd and glusterd services. Libgfapi now checks for the /var/lib/glusterd/secure-access file before making an RPC connection and enables SSL on the management connection if the file is present.

BZ#1347251

Online rolling upgrades were not possible from Red Hat Gluster Storage 3.1.x to 3.1.y (where y is more recent than x) because of client limitations. Red Hat Gluster Storage 3.2 enables online rolling upgrades from 3.2.x to 3.2.y (where y is more recent than x).

BZ#1352805

The thread pool limit for the rebalance process was static and set to 40. This meant that machines with more than 40 cores crashed when the rebalance process attempted to create more than 40 threads and access more memory than was allocated to the stack. The thread pool limit is now dynamic, and is determined based on the number of available cores.

BZ#1286572

Previously, the output of the 'gluster volume rebalance VOLNAME status' command attempted to display status details for fix-layout operations despite this operation not populating the output with details of rebalanced files, scanned files, failures, and size. The unpopulated sections have now been removed so that only status and elapsed time on each node are shown.

BZ#1294035

When a new brick was added, Red Hat Gluster Storage did not propagate permissions on the root directory of a brick during a heal operation. This resulted in errors because of incorrect permissions. This has been corrected so that permissions are correctly propagated.

BZ#1404989

Previously, the 'gluster volume add-brick' command failed on some nodes when a distributed volume was converted into a replicated volume, and the volume was not mounted, and no lookup had been performed. This could result in inconsistent data across gluster nodes. To avoid this situation, the 'gluster volume add-brick' command is no longer allowed when the replica count has increased and there any replica bricks are unavailable.

BZ#1405000

Previously, if the rm command was still running when the remove-brick rebalance operation was triggered, the rebalance operation aborted when it received the resulting ENOENT and ESTALE errors. These errors are now ignored so that the rebalance process can complete as expected in this situation.

BZ#1343320

After a connection failure, the Gluster FUSE client did not correctly clean up threads spawned to handle the failure, which resulted in a memory leak. This caused a crash when the process was killed to reclaim memory. Threads are now cleaned up correctly and the process is no longer killed, preventing the crash.

BZ#1337863

Previously, when SSL was enabled for a Red Hat Gluster Storage volume, thread cleanup code did

not always finish running before a new thread with the same ID was generated. When this happened, clients with I/O in progress experienced hung behavior. New threads are now generated after the detach flag is enabled during thread creation, so that these race conditions are avoided, and I/O no longer hangs for clients.

BZ#1400365

The maximum length for the slave user was set to the value of __POSIX_LOGIN_NAME_MAX (9 including the NULL byte). This meant that the glusterd service failed upon geo-replication mountbroker setup when the slave user length was more than 8 characters long. LOGIN_NAME_MAX is now used instead, allowing for up to 256 characters including the NULL byte. This prevents failure of the glusterd service in this situation.

BZ#1412883

Sometimes when a brick crashed, a race condition occurred that caused corruption of extended attributes on the htime file located at BRICKPATH/.glusterfs/changelogs/htime/HTIME.TIMESTAMP. This caused geo-replication to enter a "Faulty" state because of the resulting Changelog Exception. This has been corrected by ensuring that every time the brick processes come up, the extended attributes on the htime file are checked for corruption, and fixed if corruption exists.

BZ#1240333

Concurrent rename and lookup operations on a directory caused both old and new directories to be healed. At the end of the heal operation, both directories existed and had the same GFID. This meant that clients were sometimes unable to access the contents of the directory. The distributed hash table algorithm has been updated so that this issue no longer occurs.

BZ#1383898

Previously, when the gsyncd_template.conf file could not be read by the non-root account used in the geo-replication process, the default values were used instead. Because the default values do not include a complete path for the gluster binary, geo-replication fails. A check has been added to the process so that if the gsyncd_template.conf file cannot be read, an appropriate error is logged.

BZ#1205162

When a geo-replication session was deleted, the sync time attribute on the root directory of the brick was not reset to zero. This meant that when a new geo-replication session was created, the stale sync time attribute caused the sync process to ignore all files created up until the stale sync time, and start syncing from that time. A new reset-sync-time option has been added to the session delete command so that administrators can reset the sync time attribute is to zero if required.

BZ#1340756

Previously, when the rsync command failed with an error, geo-replication attempted to retrieve the error status after the child rsync process was already closed. This caused geo-replication to fail with an elines error. The elines attribute in the error object is now initialized correctly so that this failure does not occur.

BZ#1344826

When an rsync operation is retried, the geo-replication process attempted to clean up GFIDs from the rsync queue that were already unlinked during the previous sync attempt. This resulted in a KeyError. The geo-replication process now checks for the existence of a GFID before attempting to unlink a file and remove it from the rsync queue, preventing this failure.

BZ#1344908

Previously, enabling User Serviceable Snapshots (USS) caused a graph switch, which meant that when data was copied from the .snaps directory to the master volume, the first copy operation is not synchronized to the slave volume/s. The GFID access translator now correctly handles nameless lookups during the graph switch, and all data copied from the .snaps directory is correctly synced from the master volume to the slave volume/s.

BZ#1347625

If geo-replication status was requested after an upgrade but before glusterd was started again, an empty monitor.status was created and the session status was listed as 'Started'. This meant that when glusterd restarted, because monitor.status was empty, a fresh geo-replication session started instead of the previous session being resumed. This has been corrected so that an empty monitor.status results in an error, and geo-replication status is listed as 'Stopped' after an upgrade but before glusterd restarts.

BZ#1364422

If no changelog entries existed for a requested time range, the changelog history API did not return a distinguished error. This meant that applications (e.g geo-rep) using the history API assumed a general failure and re-attempted the operation. This could cause applications to loop. A specific error is now returned in this situation so that the application falls back to another mechanism like 'hybrid crawl' in geo-replication use cases.

BZ#1315544

Previously, when a NFS client unmounted all volumes, Red Hat Gluster Storage Native NFS server freed a structure that was still being used, which resulted in a segmentation fault on the server (use-after-free). The server now does not free the structure while the mount service is available, so the segmentation fault no longer occurs.

BZ#1337811

Previously, when 'showmount' was run, the structure of data passed from the mount protocol meant that the groupnodes defined in the nfs.rpc-auth-allow volume option were handled as a single string, which caused errors when the string of groupnodes was longer than 255 characters. This single string is now handled as a list of strings so that 'showmount' receives the correct number of hostnames.

BZ#1404996

As of Red Hat Gluster Storage 3.2, Gluster Native NFS server is disabled when creating a new volume. Administrators can either use NFS-Ganesha with new volumes, or continue using Native NFS server by enabling it manually. Existing volumes that use Gluster Native NFS are not modified in order to prevent disruptions during upgrade.

BZ#1351949

Previously, when the nominated volfile server in a Red Hat Gluster Storage cluster became unavailable, any configuration changes that would result in changes to the volfile were not passed to clients until the volfile server became available again. This update ensures that when a volfile server becomes unavailable, another server takes over the role of volfile server, so that clients do not need to wait to receive updates from the original volfile server.

BZ#1241314

The enable-shared-storage option always appeared disabled when the 'volume get VOLNAME enable-shared-storage' command was run, regardless of the actual state. This has been corrected so that the enable-shared-state option's state is shown accurately.

BZ#1263090

Previously, glusterd managed its portmap table so that ports that had previously been allocated to one daemon could not be reused by other daemons after the original daemon no longer used it, for example, after a brick was removed. This update ensures that ports can be reused by another daemon after they become available.

BZ#1306120

Previously, the glusterd log file was named etc-glusterfs-glusterd.vol.log. This update changes the file name to glusterd.log. The file is still found in the /var/log/glusterfs directory.

BZ#1336267

When a node is rebooted, each brick has its identity verified as it resumes operation. Previously, this was done with address resolution. However, this meant that when a cluster had a large number of bricks, there were a very large number of address lookups, which created contention and sometimes meant that bricks failed to restart. This update changes the brick verification method to use a brick's UUID rather than its address, which reduces contention and ensures that all brick processes restart after a reboot.

BZ#1340995

Previously, when glusterd was restarted, bricks were started even when server quorum was not met. This update ensures that bricks are stopped if server quorum is no longer met, or if server quorum is disabled, to ensure that bricks in maintenance are not started incorrectly.

BZ#1351732

Previously, when a server node was unavailable, the client details of the bricks on that node were not displayed when the 'gluster volume status VOLNAME clients' command was run. This has been corrected and client details are now displayed as expected.

BZ#1367472

Previously if a cluster was upgraded from Red Hat Gluster Storage 3.0.x to a version greater than or equal to 3.1.1, and any volumes had quotas enabled, attempts to run peer probe were rejected. This update ensures that peer probe can run as expected.

BZ#1379790

If a long running "pre" operation was terminated with a CTRL+C key at the keyboard, the local processes terminated, but remote processes continued to run until they had completed, because the "delete" command context could not affect remote processes. The remote command is now executed by forcing allocation of a terminal to the remote processes executed via ssh. With a terminal attached to the remote process, a local keyboard interrupt is propagated immediately to the remote processes so that they abort operations and terminate as desired by the user.

BZ#1318000

The glusterd service expected that all files in the /var/lib/glusterd/snaps/snapshot name directory were volumes. This meant that when a snapshot was taken of an NFS-Ganesha volume, glusterd interpreted the configuration file as an invalid volume and did not start. glusterd now starts correctly in this situation.

BZ#1374166

A file descriptor leak meant that a file that was removed from a volume exported using NFS-Ganesha was not removed from the underlying storage. The file descriptor leak has been corrected so that files removed from a mounted volume are also removed from the underlying storage.

BZ#1371475

Red Hat Gluster Storage now provides support for NFS-Ganesha to enable Transport Layer Security (SSL) on a management connection between the Ganesha and glusterd services. Libgfapi now checks for the /var/lib/glusterd/secure-access file before making an RPC connection and enables SSL on the management connection if the file is present.

BZ#1392895

Previously, when the pacemaker and corosync packages were updated, the update process did not remove the /etc/cluster/corosync.conf file. This meant that setup of high availability functionality failed after an upgrade. This file is now removed as part of upgrade so that setup succeeds.

BZ#1392299

In some situations, read operations were skipped by the io-cache translator, which led to a hung client mount. This has been corrected so that the client mount process works as expected for read operations.

BZ#1403840

Previously, split-brain resolution commands issued from the command line interface did not work when client-side self-heals were disabled, and returned incorrect output that indicated the file was not in split brain. This update ensures that split-brain resolution commands work regardless of whether client-side heal or the self-heal daemon are enabled.

BZ#1403180

When a directory is healed, the replication module creates the directory on the brick that needs healing, and then marks on the source brick that the newly created directory needs healing. Previously, there was a possibility for race conditions to develop if the source brick became unavailable after the creation of the new directory and before the directory was marked as requiring healing. In this situation, when I/O occurs in the new directory, the healing direction was reversed, leading to deletion of the files on the source brick. The order of operations has been reversed so that marking the source directory now occurs before directory creation. This ensures that race conditions do not develop.

BZ#1417177

Earlier, the split-brain resolution commands would erroneously resolve split-brains if two bricks that blamed each other were available, but a correct source brick was unavailable. This has now been corrected so that split-brain resolution commands will work only when all bricks are available and true split-brain conditions are present.

BZ#1284873

Enumerating large directories on a Samba client issued a large number of file operations to the gluster volume, slowing directory enumeration performance. Gluster caching has been improved to increase performance in this situation.

BZ#1412554

Premature requests to create shards before the .shard directory had been initialized caused virtual machines to pause when bricks were added because directory layout information was not yet available in memory. Shard creation now includes checks at various points to ensure that virtual machines operate normally.

BZ#1361759

Red Hat Gluster Storage now includes expedited demotion functionality to ensure that tiered volumes remain stable and available to clients when the hi-watermark value has been breached and continuing writes threaten to consume the remaining space on the hot tier before the hi-watermark breach event is triggered. Demotions are now triggered at most 5 seconds after a hi-watermark breach event. Administrators can use the cluster.tier-query-limit volume parameter to specify the number of records extracted from the heat database during a single run of the expedited demotion process. Note that there is still potential for the hot tier to become completely full, for example, if the sum of the disk space retained for metadata and the hi-watermark value are greater than the total size of the hot tier.

BZ#1427783

The metadata cache translator has been updated to improve Red Hat Gluster Storage performance when reading small files.

BZ#1388464

The 'gluster volume attach-tier' and 'gluster volume detach-tier' commands are considered deprecated in favor of the new commands, 'gluster volume tier VOLNAME attach' and gluster volume tier VOLNAME detach'. This update includes extra warning message output when the older commands are used in order to give users advance notice of the deprecation and eventual removal of the older commands.

BZ#1200927

It was found that glusterfs-server RPM package would write file with predictable name into world readable /tmp directory. A local attacker could potentially use this flaw to escalate their privileges to root by modifying the shell script during the installation of the glusterfs-server package.

BZ#1328191

Nagios expected and attempted to process performance data even when performance data had not yet been collected. This caused a crash in the Nagios plugin. This update ensures that Nagios checks whether data exists before attempting to process it, and displays an appropriate error message instead of crashing.

BZ#1351749

When Nagios retrieved live status information, this information was limited to a total of 8192 bytes. In clusters with large numbers of volumes, this meant that the data retrieved was not complete and therefore could not be parsed by Nagios. This resulted in a crash of the Nagios plugin. This update increases the allowed length for live status data in order to ensure that a complete result is retrieved and avoid the situation that led to the crash.

BZ#1372691

Previously, old RPMs included a -doc subpackage which was not included in the new RPMs. This -doc subpackage was not removed during an update. With this fix, the -doc subpackage is erased by an update.

BZ#1380695

Incorrect SELinux rules resulted in failed Gluster Native NFS functionality on Red Hat Enterprise Linux 6 based installations of Red Hat Gluster Storage. These rules have now been updated so that Gluster Native NFS works as expected.

BZ#1354661

Previously, SElinux denied binding to socket listener. This caused the ganesha.nfsd fail to start. With this fix the SElinux rules are updated and the issue is resolved.

BZ#1240258

Previously, NFS-ganesha mapped all anonymous users to uid 4294967294. This value is different from the nfsnobody value of 65534. With this fix all the anonymous uid and gid are mapped to nfsnobody by default.

BZ#1327195

On reboot, the NFS-ganesha export configuration for the volume were not copied from the online nodes. Due to this, the configuration for a volume in the NFS-ganesha cluster was out of sync. With this release this issue is fixed.

BZ#1331559

Previously, SELinux blocked the gluster brick processes to create non-regular socket files. Due to this, users were unable to create socket type files on gluster volume. With this fix, SELinux rules have been added to provide relevant permissions to gluster brick process and files of type socket can be created on nfs mount of gluster volumes.

BZ#1338068

Previously, there were few memory leaks while creating and removing files on a volume exported via NFS-Ganesha server. Due to this, NFS-Ganesha server might have gotten OOM killed, depending on the system memory limits. With this fix, the memory leaks issue has been addressed and the server shall no longer become unavailable while creating/removing large number of files.

BZ#1379329

Previously, there was an fd-leak on the file on which lock operations have been performed from a nfs-mount of the volume that is exported via NFS-Ganesha server. When that file is deleted, it was not removed from .glusterfs/unlink folder at the backend consuming memory. With this fix, all the files that are removed from the mount point shall be removed completely from the backend as well.

BZ#1377275

The gluster packages have been upgraded to upstream version (3.8.4), which provides various bug fixes and enhancements over the previous version.

CHAPTER 4. KNOWN ISSUES

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

4.1. RED HAT GLUSTER STORAGE

Issues related to glusterd

BZ#1400092

Performing add-brick to increase replica count while I/O is going on can lead to data loss.

Workaround: Ensure that increasing replica count is done offline, i.e. without clients accessing the volume.

BZ#1403767

On a multi node setup where NFS-Ganesha is configured, if the setup has multiple volumes and a node is rebooted at the same time as when volume is stopped, then, once the node comes up the volume status shows that volume is in started state where as it should have been stopped.

Workaround: Restarting the glusterd instance on the node where the volume status reflects **`started`** resolves the issue.

BZ#1417097

glusterd takes time to initialize if the setup is slow. As a result, by the time /etc/fstab entries are mounted, glusterd on the node is not ready to serve that mount, and the glusterd mount fails. Due to this, shared storage may not get mounted after node reboots.

Workaround: If shared storage is not mounted after the node reboots, check if glusterd is up and mount the shared storage volume manually.

BZ#1425681

Running volume rebalance/volume profile commands concurrently from all the nodes can cause one of the glusterd instance in a node to hold a volume lock for ever. Due to this, all the further commands on the same volume will fail with **`another transaction is in progress`** or **`locking failed`** error message. This is primarily seen when sosreport is executed on all the nodes at a same time.

Workaround: Restart the glusterd instance on the node where the stale lock exists.

BZ#1394138

If a node is deleted from the NFS-Ganesha HA cluster without performing umount, and then a peer detach of that node is performed, that volume is still accessible in /var/run/gluster/shared_storage/ location even after removing the node in the HA-Cluster.

Workaround: After a peer is detached from the cluster, manually unmount the shared storage on that peer.

BZ#1369420

AVC denial message is seen on port 61000 when glusterd is (re)started.

Workaround: Execute **`setsebool -P nis_enabled on`** and restart glusterd.

BZ#1395989

The export configurations is not deleted during volume delete and will still exist on shared storage.

Workaround: After performing volume delete, remove the file manually from the shared storage: `/var/run/gluster/shared_storage/nfs-ganesha/exports/export.<volname>.conf`

BZ#1400816

glusterd tries to create symlink "ganesh.conf" on every node of trusted storage pool. Symlink creation fails if the nfs-ganesha package is missing.

Workaround: Install nfs-ganesha package on all the nodes.

Issues related to gdeploy

BZ#1406403

If a VG already exists and if user tries to create another VG with the same name, gdeploy would extend it instead of failing.

Workaround: Ensure that a new VG name is used.

BZ#1417596

On Red Hat Enterprise Linux 6, PyYAML is not added as a dependency. Due to this, when gdeploy tries to import PyYAML it would exit as the package is not found.

Workaround: Install the PyYAML package from the repository.

BZ#1408926

Currently the `ssl_enable` option is part of the `volume` section. It is a site wide change. If more than one volume is used in the same configuration (and within the same set of servers) and `ssl_enable` is set in all the volume sections, then the ssl operation steps are performed multiple times. This causes the older volumes to fail to mount. Users will then not be able to set SSL automatically with a single line of configuration.

Workaround: If there are more than one volume on a node. Set the variable `enable_ssl` under one [volume] section and set the keys: '`client.ssl`', value: 'on'; '`server.ssl`', value: 'on';'auth.ssl-allow', value: <comma separated ssl hosts>

BZ#1418999

Deletion of a node from NFS Ganesha would fail, as the playbook `hosts' section was not pointed to the correct node.

Workaround: The node has to be deleted using the script: /usr/libexec/ganesha/ganesha-ha.sh.

Issues related to Arbiter Volumes

BZ#1387494

If the data bricks of the arbiter volume get filled up, further creation of new entries might succeed in the arbiter brick despite failing on the data bricks with ENOSPC and the application (client) itself receiving an error on the mount point. Thus the arbiter bricks might have more entries. Now when an rm -rf is performed from the client, if the readdir (as a part of rm -rf) gets served on the data

brick, it might delete only those entries and not the ones present only in the arbiter. When the rmdir on the parent dir of these entries comes, it won't succeed on the arbiter (errors out with ENOTEMPTY), leading to it not being removed from arbiter.

Workaround: If the deletion from the mount did not complain but the bricks still contain the directories, we would need to remove the directory and its associated gfid symlink from the back end. If the directory contains files, they (file + its gfid hardlink) would need to be removed too.

BZ#1388074

If some of the bricks of a replica or arbiter sub volume go down or get disconnected from the client while performing 'rm -rf', the directories may re-appear on the back end when the bricks come up and self-heal is over. When the user again tries to create a directory with the same name from the mount, it may heal this existing directory into other DHT subvols of the volume.

Workaround: If the deletion from the mount did not complain but the bricks still contain the directories, the directory and its associated gfid symlink must be removed from the back end. If the directory contains files, they (file + its gfid hardlink) would need to be removed too.

BZ#1361518

If a file create is wound to all bricks, and it succeeds only on arbiter, the application will get a failure. But during self-heal, the file gets created on the data bricks with arbiter marked as source. Since data self-heal can never happen from arbiter, 'heal-info' will list the entries forever.

Workaround: If 'gluster vol heal <volname> info` shows the pending heals for a file forever, then check if the issue is the same as mentioned above by

1. checking that trusted.afr.volname-client* xattrs are zero on the data bricks
2. checking that trusted.afr.volname-client* xattrs is non-zero on the arbiter brick *only* for the data part (first 4 bytes)

   For example:

   ```
   #getfattr -d -m . -e hex /bricks/arbiterbrick/file |grep
   trusted.afr.testvol*
   getfattr: Removing leading '/' from absolute path names
   trusted.afr.testvol-client-0=0x000000540000000000000000
   trusted.afr.testvol-client-1=0x000000540000000000000000
   ```

3. If it is in the above mentioned state, then delete the xattr:

   ```
   # for i in $(getfattr -d -m . -e hex /bricks/arbiterbrick/file
   |grep trusted.afr.testvol*|cut -f1 -d'='); do  setfattr -x $i
   file; done
   ```

Issues related to Distribute (DHT) Translator

BZ#1118770

There is no synchronization between mkdir and directory creation as part of self heal. This results in scenarios where rmdir or rename can proceed and remove the directory while mkdir is completed only on some subvolumes of DHT. Post completion of rmdir or rename, mkdir recreates the just removed or renamed directory with same gfid. Due to this, in the case of rename, both source and destination directories with the same gfid are present. In the case of rmdir, the directory can be

present on some subvols even after rmdir and it can be healed back. In both cases of rename or rmdir, the directory may not be visible on mount point and hence rm -rf of parent directory will fail with an error "Directory not empty"

Workaround: As a workaround, try the following steps:

1. If rm -rf <dir> fails with ENOTEMPTY for "dir", check whether "dir" contains any subdirectories on the bricks. If present, then delete them.

2. If post rename both the source and destination directories exist with the same gfid, then please contact redhat support for assistance.

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#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#1409474

A bug in the remove-brick code can cause file migration on some files with multiple hardlinks to fail. Files may be left behind on the removed brick. These will not be available on the gluster volume once the remove-brick operation is committed.

Workaround: Once the remove-brick operation is complete, check for any files left behind on the removed bricks and copy them to the volume via a mount point.

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#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.

Issues related to Replication (AFR)

BZ#1426128

In a replicate volume, if a gluster volume snapshot is taken when a create is in progress the file may be present in one brick of the replica and not the other on the snapshotted volume. Due to this,

when this snapshot is restored and a rm -rf is executed on a directory from the mount, it may fail with ENOTEMPTY.

Workaround: If you get an ENOTEMPTY during `rm -rf dir`, but `ls` of the directory shows no entries, check the backend bricks of the replica to verify if files exist on some bricks and not the other. Perform a stat of that file name from the mount so that it is healed to all bricks of the replica. Now when you do `` `rm -rf dir` ``, it should succeed.

Issues related to gNFS

BZ#1413910

From Red Hat Gluster Storage 3.2 onwards, for every volume the option nfs.disable will be explicitly set to either on or off. The snapshots which were created from 3.1.x or earlier does not have that volume option.

Workaround: Execute the following command on the volumes:

```
# gluster v set nfs.disable <volname> off
```

The restored volume will not be exported via gluster nfs.

Issues related to Tiering

BZ#1334262

If the **`gluster volume tier attach`** command times out, it could result in either of two situations. Either the volume does not become a tiered volume, or the tier daemon is not started.

Workaround: When the timeout is observed, follow these steps:

1. Check if the volume has become a tiered volume.
    - If not, then rerun attach tier.
    - If it has, then proceed with the next step.
2. Check if the tier daemons were created on each server.
    - If the tier daemons were not created, then execute the following command:

```
# gluster volume tier <volname> start
```

BZ#1303298

Listing the entries on a snapshot of a tiered volume shows incorrect permissions for some files. This is because the USS returns the stat information for the linkto files in the cold tier instead of the actual data file and these 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#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 works 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.

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.

Issues related to Snapshot

1403169

If NFS-ganesha was enabled while taking a snapshot, and during the restore of that snapshot it is disabled or shared storage is down, then the snapshot restore will fail.

1403195

Snapshot create might fail, if a brick has started but not all translators have initialized.

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#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#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#1394229

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#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#1059158

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

BZ#1118780

On restoring a snapshot which was created while the rename of a directory was in progress (the directory has been renamed on the hashed sub-volume but not on all of the sub-volumes), both the old and new directories will exist and have the same GFID. This can cause inconsistencies and issues accessing files in those directories.

In DHT, a rename (source, destination) of a directory is done first on the hashed sub-volume and if successful, on the remaining sub-volumes. At this point in time, both source and destination directories are present in the volume 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 these directories on the sub-volumes on which they do not yet exist- source directory entry on hashed and destination directory entry on the remaining sub-volumes. Hence, there would be two directory entries - source and destination - having the same GFID.

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#1246183

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

Issues related to Nagios

BZ#1327017

Log messages related to quorum being regained are missed by Nagios server as it is either shutdown or has communication issues with nodes. Due to this, if Cluster Quorum status was critical prior to connection issues, then it continues to remain so.

Workaround: Administrator can check the alert from the Nagios UI and once the quorum is regained, the plugin result can be manually changed using "Submit passive check result for this service" option from the service page

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.

Issues related to Rebalancing Volumes

BZ#1286074

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#1286126

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.

Issues related to Geo-replication

BZ#1393362

If a geo-replication session is created while gluster volume rebalance is in progress, then geo-replication may miss some files/directories sync to slave volume. This is caused because of internal movement of files due to rebalance.

Workaround: Do not create a geo-replication session if the master volume rebalance is in progress.

BZ#1344861

Geo-replication configuration changes when one or more nodes are down in the Master Cluster. Due to this, the nodes that are down will have the old configuration when the nodes are up.

Workaround: Execute the Geo-replication config command again once all nodes are up. With this, all nodes in Master Cluster will have same Geo-replication config options.

BZ#1293634

Sync performance for geo-replicated storage is reduced when the master volume is tiered, resulting in slower geo-replication 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#1102524

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

BZ#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_htime_path> <backup_path>
```

NOTE

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

For example:

```
cp -a
/var/run/gluster/snaps/a4e2c4647cf642f68d0f8259b43494c0/brick0/b0/
.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**.

Issues related to Self-heal

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#1240658

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

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

BZ#1173519

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

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

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

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

Issues related to replace-brick operation

- After the **gluster volume replace-brick *VOLNAME* Brick *New-Brick* commit force** command is executed, the file system operations on that particular volume, which are in transit, fail.
- After a replace-brick operation, the stat information is different on the NFS mount and the FUSE mount. This happens due to internal time stamp changes when the **replace-brick** operation is performed.

Issues related to Quota

BZ#1418227

If a directory was removed before removing the quota limits previously set on them, then a stale gfid entry corresponding to that directory remains in the quota configuration file. In the case that the last gfid entry in the quota configuration file happens to be stale, the quota list would end up showing blank output.

Workaround: The limits can be examined on the individual directories by giving the path in quota list command.

To resolve the list issue:

- Take a backup of quota.conf for safety (/var/lib/glusterd/vols/<volname>/quota.conf)
- Remove the last gfid entry which is either 16 bytes or 17 bytes based on quota.conf version.
  1. Check the quota.conf version by performing a **cat** on the file.
  2. If the conf version is 1.2, then remove the last 17 bytes. Otherwise remove the last 16 bytes from the conf file.

3. Perform the quota list operation and check if all the limits are listed now.

- Delete/Restore the backup file based on whether the above step worked/failed.

Issues related to NFS

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

Issues related to NFS-Ganesha

BZ#1451981

As of Red Hat Gluster Storage 3.2, the NFS Ganesha configuration files **`ganesha.conf`** and **`ganesha-ha.conf`** are stored in shared storage (**`/var/run/gluster/shared_storage`**). However, it is not possible to ensure that this shared storage is mounted before NFS Ganesha is started. This means that when shared storage is not yet available, NFS Ganesha fails to start. This is corrected in Red Hat Gluster Storage 3.3 but cannot be corrected in Red Hat Gluster Storage 3.2.

Workaround: Ensure that NFS Ganesha starts after shared storage is mounted. You can do this by preventing the `nfs-ganesha` service from starting at boot time, and starting the service manually after you have verified that the shared storage is mounted.

To disable the service from starting automatically at boot time, run the following command:

```
# systemctl disable nfs-ganesha
```

To verify that shared storage is mounted, run the following command:

```
# df -h | grep -i shared
server1:/gluster_shared_storage  14G  2.4G  12G  17%
    /run/gluster/shared_storage
```

To start the service manually, run the following command:

```
# systemctl start nfs-ganesha
```

BZ#1425504

The logrorate system is missing options in the configuration file which will enable the deletion of **`ganesha.log`** and **`ganesha-gfapi.log`**. The absence of configuration options results in the log files being rotated but never being deleted or removed, resulting in the consumption of a lot of space.

Workaround: Manually delete the log files to recover the space.

BZ#1425753

When there are multiple paths with the same parent volume exported via NFS-ganesha server, the handles maintained by the server of the files/directories common to those paths may get merged. Due to this, unexporting one of those shares may result in segmentation fault of the server when accessed via another share mount.

Workaround: Unexport such shares in the reverse order of how they were exported. For eg., if the shares are exported in the order as mentioned below:

```
/testvol
/testvol/a
/testvol/a/b
```

Then unexport those paths in the reverse order i.e,

```
/testvol/a/b
/testvol/a
/testvol
```

The handles merged by the server shall not get freed as long as all the shares accessing them do not get unexported, thus avoiding the crash.

BZ#1426523

The ganesha.conf file is not cleaned completely during nfs-ganesha disable leading to several stale export entries in the ganesha.conf file. Due to this, enabling nfs-ganesha after this fails to bring up the ganesha process.

Workaround: Remove the stale entries manually.

BZ#1403654

In a nfs-ganesha cluster, when multiple nodes shutdown/reboot, pacemaker resources may enter FAILED/STOPPED state. This may then affect IP failover/failback behaviour.

Workaround: Execute the following command for each such resource which went into FAILED/STOPPED state to restore them back to normal state.

```
# pcs resource cleanup <resource-id>
```

BZ#1398843

When a parallel `rm -rf` from multiple nfs clients which has large number of directory hierarchy and files in it is performed, due to client side caching, deletion of certain files results in ESTALE, and the parent directory will not be removed with ENOEMPTY.

Workaround: Perform `rm -rf *` again on the mount point.

BZ#1402308

The Corosync service will crash, if ifdown is performed after setting up the ganesha cluster. This may impact the HA functionality.

BZ#1330218

If a volume is being accessed by heterogeneous clients (i.e, both NFSv3 and NFSv4 clients), it is observed that NFSv4 clients take longer time to recover post virtual-IP failover due to a node shutdown.

Workaround: Use different VIPs for different access protocol (i.e, NFSv3 or NFSv4) access.

BZ#1416371

If **`gluster volume stop`** operation on a volume exported via NFS-ganesha server fails, there is a probability that the volume will get unexported on few nodes, inspite of the command failure. This will lead to inconsistent state across the NFS-ganesha cluster.

Workaround: To restore the cluster back to normal state, perform the following

- Identify the nodes where the volume got unexported
- Re-export the volume manually using the following dbus command:

```
# dbus-send --print-reply --system --dest=org.ganesha.nfsd
/org/ganesha/nfsd/ExportMgr org.ganesha.nfsd.exportmgr.AddExport
string:/var/run/gluster/shared_storage/nfs-ganesha/exports/export.
<volname>.conf string:""EXPORT(Path=/<volname>)"""
```

BZ#1381416

When a READDIR is issued on directory which is mutating, the cookie sent as part of request could be of the file already deleted. In such cases, server returns **BAD_COOKIE** error. Due to this, some applications (like bonnie test-suite) which do not handle such errors may error out.

This is an expected behaviour of NFS server and the applications has to be fixed to fix such errors.

BZ#1398280

If any of the PCS resources are in the failed state, then the teardown requires a lot of time to complete. Due to this, the command **`gluster nfs-ganesha disable`** will timeout.

Workaround: If **`gluster nfs-ganesha disable`** is errored with a timeout, then perform the pcs status and check whether any resource is in failed state. Then perform the cleanup for that resource using following command:

```
# pcs resource --cleanup <resource id>
```

Re-execute the **`gluster nfs-ganesha disable`** command.

BZ#1328581

After removing a file, the nfs-ganesha process does a lookup on the removed entry to update the attributes in case of any links present. Due to this, as the file is deleted, lookup will fail with ENOENT resulting in a misleading log message in gfapi.log.

This is an expected behaviour and there is no functionality issue here. The log message needs to be ignored in such cases.

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-timout (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#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#1235597

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

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#1286050

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#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#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#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
```

Issues related to Samba

BZ#1419633

CTDB fails to start on those setups where the real time schedulers have been disabled. One such example is where vdsm is installed.

Workaround: Enable real time schedulers by **echo 950000 > /sys/fs/cgroup/cpu,cpuacct/system.slice/cpu.rt_runtime_us** and then restart the ctdb service. For more information, refer the cgroup section of Red Hat Enterprise Linux administration guide, https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html-single/System_Administrators_Guide/index.html

BZ#1379444

Sharing of subdirectories of Gluster volume does not work if shadow_copy2 vfs module is also used. This is because shadow_copy2 checks on local filesystem for path being shared and Gluster volumes are remote filesystems accessed using libgfapi.

Workaround: Add **shadow:mountpoint = /** in share section of **smb.conf** to bypass this check.

BZ#1329718

Snapshot volumes are read-only. All snapshots are made available as directories inside .snaps. Even though snapshots are read-only the directory attribute of snapshots is same as the directory attribute of root of snapshot volume, which can be read-write. This can lead to confusion, because Windows will assume that the snapshots directory is read-write. **Restore previous version** option in file properties gives **open** option. It will open the file from the corresponding snapshot. If opening of the file also create temp files (e.g. Microsoft Word files), the open will fail. This is because temp file creation will fail because snapshot volume is read-only.

Workaround: Copy such files to a different location instead of directly opening them.

BZ#1322672

When vdsm and abrt's ccpp addon are installed alongside each other, vdsmd overwrites abrt's core dump configuration in /proc/sys/kernel/core_pattern. This prevents Samba from generating core dumps due to SELinux search denial on new coredump location set by vdsmd.

Workaround: To workaround this issue, execute the following steps:

1. Disable core dumps in /etc/vdsm/vdsm.conf:

```
core_dump_enable = false
```

2. Restart the abrt-ccpp and smb services:

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

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 smb.conf file. Disabling asynchronus I/O may have performance impact 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#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#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.

Issues related to Sharding

BZ#1332861

Sharding relies on block count difference before and after every write as gotten from the underlying file system and adds that to the existing block count of a sharded file. But XFS' speculative preallocation of blocks causes this accounting to go bad as it may so happen that with speculative preallocation the block count of the shards after the write projected by xfs could be greater than the number of blocks actually written to.

Due to this, the block-count of a sharded file might sometimes be projected to be higher than the actual number of blocks consumed on disk. As a result, commands like **du -sh** might report higher size than the actual number of physical blocks used by the file.

General issues

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#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#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#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.

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.

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

4.2. RED HAT GLUSTER STORAGE CONSOLE

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#838329

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

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#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#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#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#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#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#1167425

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#1134319

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

4.3. RED HAT GLUSTER STORAGE AND RED HAT ENTERPRISE VIRTUALIZATION INTEGRATION

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.

CHAPTER 5. 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 3.2. 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

5.1. 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 Stopping a **`remove-brick`** Operation in *Red Hat Gluster Storage 3.2 Administration Guide*.

5.2. SMB MULTI-CHANNEL

Multi-Channel is an SMB3 protocol feature that allows the client to bind multiple transport connections into one authenticated SMB session. This allows for increased fault tolerance and throughput on Windows 8 and newer and Windows Server 2012 and newer.

For more information, see SMB3 Multi-Channel with Samba on Red Hat Gluster Storage (Technology Preview).

5.3. 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.

5.4. 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 *pNFS* under pNFS in *Red Hat Gluster Storage 3.2 Administration Guide*

APPENDIX A. REVISION HISTORY

Revision 3.2-5 **Wed Mar 22 2017** **Bhavana. Mohan.**

Version for the Red Hat Gluster Storage 3.2 release.

Revision 3.2-4 **Tue Mar 21 2017** **Bhavana. Mohan.**

Included the Notable Bug Fixes chapter and incorporated review comments.

Revision 3.2-3 **Mon Mar 20 2017** **Bhavana. Mohan.**

Included all the major features in the What is new in this release chapter and added known issues pertaining to the Red Hat Gluster Storage 3.2 release

Revision 3.2-2 **Mon Mar 20 2017** **Bhavana. Mohan.**

Removed the technology preview features based on bug 1427007 and cleaned up the known issues chapter.

Revision 3.2-1 **Thu Mar 16 2017** **Bhavana. Mohan.**

Initial creation by publican