



Red Hat Gluster Storage 3.5

3.5 リリースノート

Red Hat Gluster Storage 3.5 リリースノート

Red Hat Gluster Storage 3.5 リリースノート

Red Hat Gluster Storage 3.5 リリースノート

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2023 | You need to change the HOLDER entity in the en-US/3.5_Release_Notes.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本リリースノートでは、Red Hat Gluster Storage 3.5 で実装された改良点と追加機能の概要を説明します。

目次

第1章 はじめに	3
第2章 本リリースで変更された内容	4
2.1. 本リリースにおける最新情報	4
2.2. サポートの制限と廃止	6
2.3. 非推奨の機能	6
第3章 主なバグ修正	8
第4章 既知の問題	13
4.1. RED HAT GLUSTER STORAGE	13
4.2. RED HAT GLUSTER STORAGE と RED HAT ENTERPRISE VIRTUALIZATION の統合	28
第5章 テクノロジーレビュー	29
5.1. SMB マルチチャネル	29
付録A 改訂履歴	30

第1章 はじめに

Red Hat Gluster Storage は、エンタープライズ向けに柔軟かつアジャイルな非構造化データストレージを提供するソフトウェアで、スケールアウトした唯一のストレージソリューションです。Red Hat Gluster Storage は、データストレージとインフラストラクチャーを統合し、パフォーマンスを向上して、可用性および管理性を改善し、組織の課題やニーズに対応する機会を新たに提供しています。

GlusterFS は、Red Hat Gluster Storage の主要なビルディングブロックで、スタック可能なユーザー空間設計をベースとしており、さまざまなワークロードのために例外のパフォーマンスを提供できます。GlusterFS は、複数のネットワークインターフェースを通じてさまざまなストレージサーバーを集約し、これらをつなぎ合わせて単一の大きな並列ネットワークファイルシステムを形成します。POSIX と互換性のある GlusterFS サーバーは、XFS ファイルシステム形式で、ディスクにデータを保存します。これらのサーバーには、Network File System (NFS) および Server Message Block SMB (CIFS としても知られている) などの業界標準アクセスプロトコルを使用してアクセスできます。

オンプレミスの Red Hat Gluster Storage Server は、プライベートクラウドまたはデータセンターのデプロイメントで使用できます。Red Hat Gluster Storage は、コモディティサーバーおよびストレージハードウェアにインストールできるため、強力かつ拡張性の高い、高可用性 NAS 環境になります。さらに、Red Hat Gluster Storage を Amazon Web Services (AWS)、Microsoft Azure、または Google Cloud と共にパブリッククラウドにデプロイすることもできます。これは、プライベートクラウドまたはデータセンターで可能なすべての機能をパブリッククラウドに提供します。そのためには、クラウドに非常にスケラブルで高可用性の NAS を提供します。

Red Hat Gluster Storage Server for On-premises

Red Hat Gluster Storage Server for On-premises を使用すると、企業は製品サーバーおよびストレージハードウェアを使用して、仮想の、スケラブルな、集中管理プールとして物理ストレージを扱うことができます。

Red Hat Gluster Storage Server for Public Cloud

Red Hat Gluster Storage Server for Public Cloud は、AWS、Microsoft Azure、および Google Cloud でスケラブルな NAS をデプロイするための GlusterFS をパッケージ化しています。この強力なストレージサーバーは、これらのパブリッククラウドプロバイダーのユーザーに、可用性が高く、スケラブルで仮想化された、集中管理プールを提供します。

第2章 本リリースで変更された内容

2.1. 本リリースにおける最新情報

本セクションでは、Red Hat Gluster Storage 3.5 リリースの主要機能および機能拡張を説明します。

- Red Hat Gluster Storage がアップストリーム glusterfs バージョン 6 に更新されました。
- Samba がアップストリームバージョン 4.9.8 に更新されました。
- NFS-Ganesha がアップストリームバージョン 2.7.3 にアップデートされました。
- NFSのバージョン4.1に対応しました。
- ディレクトリのコンテンツが設定可能なチャンクで読み込まれるようになったため、非常に大きなディレクトリリストでも、NFS-Ganeshaクライアントに提供する前にディレクトリ全体が読み込まれるのを待つ必要がなくなり、より高速に提供されるようになりました。
- Red Hat Gluster Storage では、レプリケートされたサブボリューム間の拡張属性として独自の `ctime` 属性のオプションが提供されるようになり、自己修復が発生した後など、ファイルシステムベースの `ctime` を使用した場合に発生していたレプリケートされたブリックと配布されたブリック間の一貫性の問題が回避されるようになりました。
- 異なるサブボリューム内のブリックを異なるサイズにすることができ、Glusterのアルゴリズムはファイルの配置範囲を決定する際にこれを考慮するようになりました。利用可能なスペースのアルゴリズムが更新され、異なるブリックサイズに対してより良く機能するようになりました。同じレプリカセットや分散セットに属するブリックは、これまでどおり同じサイズでなければなりません。
- ブリックのデフォルトの最大ポート番号は、65535ではなく60999になりました。
- 管理者は、禁止されたノードの証明書をCertificate Revocation Listファイルに追加し、そのファイルのパスを新しい`ssl.crl-path`ボリュームオプションに指定することで、証明書が失効したノードがクラスタにアクセスしないようにできるようになりました。
- Gluster-ansible が Red Hat Gluster Storage の IPv6 ネットワーキングを設定できるようになりました。
- `op-version`が70000以上のクラスタ上の新しいボリュームに対して、`storage.fips-mode-rchecksum`ボリュームオプションがデフォルトで有効になりました。
- Geo レプリケーションの設定は、時間がかかり、エラーが発生しやすいプロセスでした。Geo レプリケーションのサポートが`gdeploy`によって提供されるようになり、設定の自動化とこのプロセスにおけるエラーの削減が可能になりました。
- 新しい API である `glfs_set_statedump_path` により、ユーザーは `statedump` 出力を保存するディレクトリを設定することができます。例えば、以下のコールでは、`/tmp`ディレクトリを `statump`パスとして設定します。

```
glfs_set_statedump_path(fs2, "/tmp");
```

- `umask`のオーバーライドを可能にする新しい設定オプションが追加されました。**`storage.create-directory-mask`**および**`storage.create-mask`**オプションは、ディレクトリおよびその他のファイルのファイルモードをそれぞれ指定されたマスクに制限します。**`storage.force-directory-mode`**および**`storage.force-create-mode`**オプションは、ディレク

トリおよびその他のファイルに対して、それぞれ指定されたモードビットの存在を強制します。なお、これらのモード制約は、与えられたオプション値が有効である限り維持され、ファイル作成時にのみ適用されるわけではありません。

- NFS-Ganeshaは、アクティブ-アクティブの高可用性構成を維持するために必要なアップコール通知を、非同期コールバックで受け取るようになりました。これにより、継続的なポーリングが不要になり、CPUとメモリの使用量が削減されます。
- NFS-Ganeshaサーバーからアクセス制御リストやその他のエクスポート情報を取得するためのdbusコマンドが新たに追加されました。
- storage.reserveオプションは、利用可能なスペースをパーセンテージだけでなく、サイズで確保できるようになりました。
- 非同期のI/O操作は、正常終了の通知を受ける時点で、ワークフローのボトルネックにより、遅延していました。ボトルネックが解消され、非同期のI/O操作のパフォーマンスが向上しました。
- Red Hat Gluster Storage 3.4 と比較して、パフォーマンスが向上しました。

NFS-Ganesha v4.1のマウント

- レプリカ3およびarbiterボリューム上の小さなファイルに対するls -l/stat/chmodの動作が6-10倍向上しました。
- メタデータを多用する操作が改善され、分散したボリュームの小さなファイルのパフォーマンスを向上しました。
 - chmod - 106%
 - creates - 27%
 - stat - 808%
 - reads - 130%
 - appends - 52%
 - rename - 36%
 - delete-renamed - 70%
 - mkdir - 52%
 - rmdir - 58%
- 分散されたボリュームでの大きなファイルの連続した書き込みパフォーマンスが47%向上しました。
- レプリカ3ボリュームでの大きなファイルの連続した読み取り/書き込みパフォーマンスが向上しました（それぞれ20%と60%）。

Gluster-FUSEマウント

- arbiterボリュームにでの大きなファイルの連続した読み取りパフォーマンスが20%向上しました。

- 分散されたボリュームでの小さなファイルのmkdir/rmdir/rename操作が向上しました（それぞれ24/25/10%）。

2.2. サポートの制限と廃止

このセクションでは、Red Hat Gluster Storage 3.5 でサポートされなくなったテクノロジー、またはサポートが制限される機能について説明します。

gluster-swift (オブジェクトストア)

Red Hat OpenStack Platform のオブジェクトストアとして Red Hat Gluster Storage を使用することはサポートされなくなりました。

gstatus

Red Hat Gluster Storage 3.5 は gstatus をサポートしていません。

スタンドアロンの Red Hat Gluster Storage に対する Heketi

Heketiは、OpenShift Container Storageとの組み合わせでのみサポートされています。Red Hat Gluster Storageでの一般的な使用には対応していません。

Nagios Monitoring

Red Hat Gluster Storage 3.5 は Nagios による監視をサポートしていません。Red Hat Gluster Storage Web 管理が、Red Hat Storage Gluster クラスター用に推奨されるモニタリングツールになりました。

Red Hat Atomic Host

Red Hat Atomic Hostは、Red Hat Gluster Storage 3.5での使用はサポートされていません。

Red Hat Gluster Storage Console

Red Hat Gluster Storage Consoleは、Red Hat Gluster Storage 3.5での使用はサポートされていません。Red Hat Gluster Storage Web 管理が、Red Hat Storage Gluster クラスター用に推奨されるモニタリングツールになりました。

VDO (Virtual Disk Optimizer)

VDO (Virtual Disk Optimizer) ボリュームは、Red Hat Hyperconverged Infrastructure for Virtualization 2.0 デプロイメントの一部としてのみサポートされます。VDO は、他の Red Hat Gluster Storage デプロイメントではサポートされません。

2.3. 非推奨の機能



注記

本リリースは、Red Hat Enterprise Linux 6 をサポートする最後の Red Hat Gluster Storage リリースです。お客様には[Red Hat Enterprise Linux 7](#) へのアップグレードをお勧めします。

以下の機能は、Red Hat Gluster Storage 3.5 では非推奨となり、後続のリリースでは非推奨とみなされます。機能の削除タイムフレームについての詳細は、個別の項目を確認します。

gluster-NFS

Gluster-NFS は、Red Hat Gluster Storage 3.5 では非推奨です。Red Hat では Gluster-NFS の使用を推奨しておらず、Red Hat Gluster Storage 3.5以降での新規デプロイメントでのその使用をサポートしません。Red Hat Gluster Storage 3.5 にアップグレードした既存のデプロイメントは引き続きサポートされますが、ユーザーには NFS-Ganesha への移行を推奨します。この NFS-Ganesha は、「[本リリースにおける最新情報](#)」で詳述されている強化された機能、追加のセキュリティ機能、およびパフォーマンスの改善を提供します。

Remote Direct Memory Access (RDMA)

RDMA をトランスポートプロトコルとして使用することは、Red Hat Gluster Storage 3.5 で非推奨となっています。Red Hat ではその使用を推奨しておらず、Red Hat Gluster Storage 3.5での新規デプロイメントをサポートしません。Red Hat Gluster Storage 3.5 にアップグレードした既存のデプロイメントは引き続きサポートされます。

階層化

階層化は、Red Hat Gluster Storage 3.5 では非推奨です。Red Hat ではその使用を推奨しておらず、Red Hat Gluster Storage 3.5での新規デプロイメントでの階層化をサポートしません。Red Hat Gluster Storage 3.5 にアップグレードした既存のデプロイメントは引き続きサポートされます。

RHGSサーバー用のRed Hat Enterprise Linux 6 (RHEL 6) ベースのISOイメージ

Red Hat Gluster Storage 3.5 のリリースから、Red Hat Gluster Storage サーバー用の Red Hat Enterprise Linux 6 ベースの ISO イメージが含まれなくなりました。

Parallel NFS (pNFS)

Red Hat Gluster Storage 3.4 の時点で、Parallel NFS はサポートされておらず、テクノロジープレビューとして利用できなくなりました。安定性に影響を及ぼす、この機能の長期的な問題は、アップストリームで未解決のままです。この機能の使用に関する情報は、Red Hat Gluster Storage 3.4 から削除されましたが、テクノロジープレビューとして Parallel NFS を提供したリリースのドキュメントで引き続き参照できます。

Parallel NFS (pNFS)

Red Hat Gluster Storage 3.4 の時点で、Parallel NFS はサポートされておらず、テクノロジープレビューとして利用できなくなりました。安定性に影響を及ぼす、この機能の長期的な問題は、アップストリームで未解決のままです。この機能の使用に関する情報は、Red Hat Gluster Storage 3.4 から削除されましたが、テクノロジープレビューとして Parallel NFS を提供したリリースのドキュメントで引き続き参照できます。

第3章 主なバグ修正

本章では、ユーザーに大きな影響を及ぼす Red Hat Gluster Storage の今回のリリースで修正されたバグについて説明します。



注記

ハイパーリンクの付いていない Bugzilla ID は、機密データが割り当てられている可能性のあるプライベートバグです。

セキュリティの修正

CVE-2019-10197 (中程度)

パラメーターとパーミッションの組み合わせにより、ユーザーが共有パスの定義から逃れることができる可能性があります。

一般的な修正

BZ#1578703

以前は、**gluster volume status <volname> inode**を実行するとinodeテーブル全体が出力され、タイムアウトしてパフォーマンス問題が発生することがありました。このコマンドの出力はより合理的になり、元の情報はstetumpを実行することで得られるようになりました。

BZ#1734423、BZ#1736830、BZ#1737674

これまでは、動的に割り当てられたメモリが正しく解放されないため、glusterクライアントでのメモリ消費量の増加やメモリ不足の管理が行われていました。メモリが正しく解放されるようになり、メモリオーバーランが発生しないようになりました。

BZ#1676468

以前は、glusterfsはカーネルの自動無効化を有効にしており、ctimeが変更されるとページキャッシュを無効にしていました。つまり、ctimeの変更前、変更中、変更後に書き込みが発生するたびに、ページキャッシュがページされ、その後の書き込みのパフォーマンスはキャッシュの恩恵を受けないことになりました。

パフォーマンスを向上させるための2つの新しいオプションが用意されました。

マウントオプションの**auto-invalidation[=on|off]**がデフォルトで有効になりました。これは、カーネルが属性、dentry、およびページキャッシュを自動的に無効にするかどうかを指定するものです。書き込み後にページキャッシュを保持するには、これを「off」に設定します。ただし、ファイルが2つの異なるマウントポイントから同時にアクセスできない場合に限り、

ボリュームオプションの**performance.global-cache-invalidation=[on|off]**は、**performance.cache-invalidation**の値を上書きします。このオプションは、デフォルトでは無効ですが、有効にすると、統計の変更が検出されたときに、glusterに関連するすべての読み取りキャッシュをページします。ファイルが異なるマウントポイントからアクセス可能で、これらのマウントポイント間のキャッシュが一貫していることが要求される場合に限り、このオプションをオンにしてください。

両方のオプションをオフにすると、書き込まれたデータはページキャッシュに保持され、同一領域でのオーバーラップ読み取りのパフォーマンスが向上します。

BZ#1726991

get-status操作は開始および停止の状態しか追跡しないため、ブリックの状態が開始または停止の状態にあるときに、ブリックの状態が開始と表示されていました。**get-status**操作は、より正確に状態を報告するようになりました。

BZ#1720192

glusterボリュームに**bind-address**が指定されている場合、リバランスソケットファイルの名前が許容される文字数よりも大きくなり、リバランスが開始されないという問題がありました。ボリューム名とUUIDに基づいてハッシュが生成されるようになり、この問題が回避されました。

BZ#1670415、BZ#1686255

すべてのボリュームの状態を表示するときに発生していた小さなメモリーリークが修正されました。

BZ#1652461

3ノードクラスターで1500以上のボリュームを設定している場合に、ノードやglusterdサービスが利用できなくなると、再接続時にハンドシェイクプロセスがタイムアウトする前に収集するボリューム情報が多すぎるという問題がありました。この問題は、ボリューム情報の収集プロセスにいくつかの最適化を加えることで解決されます。

BZ#1058032

これまでは、仮想マシンの移行中に、マシンイメージが共有ファイルシステム上にあることが検出されると、libvirtがマシンイメージの所有権を変更していました。そのため、仮想マシンがイメージにアクセスできなくなっていました。この問題はもう再現しません。

BZ#1685246

removeexattr システムコールがブリックプロセスに渡されなかったため、アクセス制御リスト設定が Red Hat Gluster Storage ボリュームから削除されませんでした。この問題は修正され、期待通りに属性が削除されるようになりました。

分散されたボリュームの修正

BZ#1732774

不良ブリック上のファイルに対する書き込み要求が実行されている間に、そのファイルがヒーリングされていた場合、書き込み操作中に発生する読み取りは、不良ブリックからファイルを読み取ることができます。これにより、正常なブリックのデータが破損してしまう可能性があります。この問題を回避するために、すべての読み取りは良好なブリックからのみ行われるようになりました。

BZ#1706549

ブリックがダウンしても、**O_TRUNC**フラグを使用してファイルを変更することができます。ブリック機能が復活すると、ファイル記述子を使用してファイルを変更した操作は、open-fdヒールを開始します。これまでは、**O_TRUNC**を使用してオープンされたファイルに対してopen-fd healを実行すると、ファイルに対して切り捨て操作が行われていました。切り捨て操作は通常、すでにロックを取得している操作の一部として行われるため、明示的なロックを取得しませんでした。この場合、NULLロック構造が発生し、最終的にNULLロック構造が参照解除されたときにクラッシュしました。open-fdヒール時に**O_TRUNC**フラグが無視され、ファイルのデータヒール時に切り捨て操作が行われるようになり、この問題が回避されるようになりました。

BZ#1745107

これまでは、ファイルのサイズやバージョンの更新に失敗しても、ファイル記述子が不良としてマークされることはありませんでした。そのため、必ずしもそうではないのに、ブリックが正常だ

と思われたり、ファイルに誤ったデータが表示されたりすることがありました。この更新により、更新に失敗した後のファイル同期の変更やフラッシュの失敗により、ファイル記述子が不良とマークされるようになりました。

分散ボリュームの修正

BZ#1672869

これまでは、**parallel-readdir**を有効にすると、古くなったlinktoファイルがデータファイルと誤って解釈され、削除できませんでした。古くなったlinktoファイルが正しく認識されるようになりました。

イベントに関する修正

BZ#1732443

これまでは、イベントソケットの初期化時にネットワークファミリーが正しく設定されませんでした。この結果、無効な引数エラーが発生し、イベントがコンシューマーに送信されませんでした。ネットワークファミリーが正しく設定され、イベントが期待通りに動作するようになりました。

gdeployによる自動化の修正

BZ#1759810

gdeployによるSambaのセットアップ時に、設定オプション**group=samba**と**user.cifs=enable**がボリュームに設定されるようになり、セットアップが成功するようになりました。

BZ#1712904

これまでは、gdeployを使用してsambaを設定すると、クラスタ内のすべてのノードでsambaユーザが作成されませんでした。これにより、CTDBのフェイルオーバー時に、必要なユーザーが存在しないという問題が発生しました。gdeployは、すべてのノードでsambaユーザーを作成するようになり、この問題を回避しました。

Geo レプリケーションに関する修正

BZ#1708116

ジオレプリケーションで、リンクが解除されてマスターに存在しなくなった大量のファイルの同期を試みると、デッドロックが発生してtarsshプロセスがハングアップしてしまうという問題がありました。tarが完了する前に、tarプロセスのstderrバッファがいっぱいになると、ハングアップしました。ワーカーはstderrを読み取る前にtarが完了することを期待していましたが、tarは読み取られてバッファが解放されるまで完了できませんでした。ワーカーは、tarプロセスが作成されるとすぐにstderr出力の読み取りを開始するようになり、この問題が回避されました。

BZ#1708121

Geo レプリケーションでは、多数の異なるファイルが作成され、同じ宛先パスにリネームされた場合に、追加のファイルを作成するのではなく、正しく同期するようになりました。

BZ#1712591

非rootのGeo レプリケーションセッションでは、glusterのバイナリパスがPATH変数に追加されないため、glusterコマンドがセッションで利用できないという問題がありました。既存の**gluster-command-dir**および**gluster-command-slave-dir**オプションを使用して、セッションがglusterコマ

ンドにアクセスできるようにすることができます。

BZ#1670429

シンボリックリンクの名前が同期の間に複数回変更されても、Geo レプリケーションが成功するようになりました。

NFS-Ganeshaの修正

BZ#1728588

NFSクライアントとの接続を再確立しようとしたときに、サーバーが既存の状態を時間内にクリーンアップしないという競合状態がありました。これにより、新しい接続が期限切れと誤って認識され、マウントポイントにアクセスできなくなるという問題がありました。新しい接続を受け入れる前に状態がクリーンアップされるようになったため、この問題は発生しなくなりました。

BZ#1751210

NFS-Ganeshaでは、Glusterストレージに対するすべての操作にクライアントの認証情報を使用していました。非rootユーザーが読み取り専用のファイルを操作した場合に、「permission denied」というエラーが発生することがありました。ルート権限が必要に応じて使用されるようになり、非rootユーザーが0444モードを使用してファイルを作成および書き込むことができるようになりました。

レプリケーションの修正

BZ#1688395

書き込みトランザクション中にeager-lockのロック取得が失敗すると、前のロックが保持され、後続の書き込みがすべてブロックされ、ハングアップすることがありました。この状況が正しく処理されるようになり、関連する問題の診断に役立つ、より具体的なログメッセージが追加されました。

BZ#1642425

Gluster NFSボリュームのボリューム設定ファイルで**cluster.quorum-count**ボリュームオプションが更新されないという問題がありました。これは、読み取ったファイルの最後の部分がバッファサイズよりも小さい場合、バッファから書き込まれるデータが新しいデータと古いデータの組み合わせになったためです。この問題は修正され、Gluster NFSクライアントは、**cluster.quorum-type**が**fixed**に設定されている場合、**cluster.quorum-count**を尊重するようになりました。

Shardingの修正

BZ#1568758

多数のシャードを持つファイルを削除するときに、タイムアウトしていました。すべてのシャードで並行してアンリンク操作が行われ、**.shard**ディレクトリで競合が発生したためです。タイムアウトにより削除が失敗し、**.shard**ディレクトリに古いシャードが残るという問題がありました。**.shard**ディレクトリの競合を抑制し、タイムアウトを防ぐために、シャードの削除は、一度に1バッチのシャードを削除するバックグラウンドプロセスになりました。シャードの削除バッチのサイズは、**features.shard-deletion-rate**オプションで制御されますが、デフォルトでは**100**に設定されています。

Web Administrationに関する修正

BZ#1645428

以前に出荷されたバージョンのpython2-pyasn1パッケージを使用すると、IPAクライアントのインストールに失敗するという問題がありました。python2-pyasn1の代わりにpysnmpが使用されるように、このパッケージはtendr1-notifierとtendr1-commonsのアップデートで置き換えられ、インストールは期待通りに動作します。

Red Hat Gluster Storage Web Administration 3.5 にアップグレードする前に、以下のコマンドを実行してpython2-pyasn1とpysnmpパッケージ（依存関係は除く）を削除してください。

```
# rpm -e --nodeps $(rpm -qa 'python2-pyasn1')
# rpm -e --nodeps $(rpm -qa 'pysnmp')
```

BZ#1647322

以前は、tendr1は/var/lib/carbon/whisper/tendr1ディレクトリにオーナーを設定しませんでした。このディレクトリの所有者がcarbonユーザーでない場合、carbon-cacheはこの場所にwhisperファイルを作成できませんでした。whisperファイルを確実に作成できるように、Tendr1はディレクトリの所有者をcarbonユーザーにするようになりました。

BZ#1688630

これまでは、tendr1-monitoring-integrationが実行されていないために発生したエラーは、一般的なエラーメッセージで報告されていました。この状況で、tendr1-monitoring-integrationのステータスに関するより具体的なエラーメッセージが記録されるようになりました。

BZ#1645221

これまでは、Red Hat Gluster Storage Web Administrationでは、Web Administrationによるノードの管理を停止する前に、すべてのノードがオンラインになっている必要がありました。クラスタ内の1つまたは複数のノードがオンラインでない場合でも、ノードを管理対象から外すことができるようになりました。

BZ#1666386

Red Hat Gluster Storage Web Administrationでは、正常に動作しているヒールプロセスの一部であっても、スプリットブレイン関連のイベントをすべて受信し、ユーザーインターフェースにエラーとして表示していました。イベントがクライアントの識別子に基づいてフィルタリングされるようになり、ユーザーインターフェースから不要なエラーや誤ったエラーを取り除くことができるようになりました。

BZ#1687333

これまでは、クラスタ内のすべてのノードがオフラインになったときに、Web Administrationインターフェイスでオフラインのノード数が正しく報告されませんでした。ノードの状態が正しく追跡され、報告されるようになりました。

BZ#1686888

node-agentサービスは、インポートと削除（管理停止）の操作を行います。node-agentサービスが実行されていない場合、これらの操作は一般的なログメッセージと共にタイムアウトしました。この問題が発生した場合、より明確に記録されるようになりました。

BZ#1702412

これまでは、Ansible 2.8との互換性が正しく機能しませんでした。Red Hat Storage Web Administration が Ansible 2.8 と互換性を持つようになりました。

第4章 既知の問題

本章では、リリース時の既知の問題を紹介します。



注記

ハイパーリンクの付いていない Bugzilla ID は、機密データが割り当てられている可能性のあるプライベートバグです。

4.1. RED HAT GLUSTER STORAGE

glusterd に関連する問題

BZ#1567616

glusterd のいずれかが停止したときに **enable-shared-storage** オプションが無効になっていると、共有ストレージ操作が正常に行われます。ただし、**enable-shared-storage** 操作の有効化および無効化の要求は失敗します。

回避策: 以下のコマンドを実行し、この動作を解消します。

```
# gluster v delete gluster_shared_storage
```

```
# gluster v set all cluster.enable-shared-storage enable
```

BZ#1400092

I/O の実行中にレプリカ数を増やすために `add-brick` を実行すると、データが失われる可能性があります。

回避策: レプリカ数の増加がオフラインになることを確認します (例: クライアントがボリュームにアクセスすることなく)。

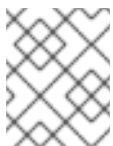
BZ#1417097

glusterd は、設定が遅い場合に初期化に時間がかかります。その結果、`/etc/fstab` エントリーがマウントされる際に、ノード上の glusterd がそのマウントに対応する準備ができず、glusterd マウントは失敗します。このため、ノードの再起動後に共有ストレージがマウントされない可能性があります。

回避策: ノードの再起動後に共有ストレージがマウントされていない場合は、glusterd が稼働中かどうかを確認し、共有ストレージボリュームを手動でマウントします。

BZ#1394138

`umount` を実行せずに NFS-Ganesha HA クラスターからノードが削除されてから、そのノードのピアダタッチが実行されると、HA-Cluster のノードを削除しても、そのボリュームは `/var/run/gluster/shared_storage/` の場所で引き続きアクセスできます。



注記

3.5 Batch Update 3 のリリースにより、共有ストレージのマウントポイントが `/var/run/gluster/` から `/run/gluster/` に変更されました。

回避策: ピアをクラスターからデタッチした後に、そのピアの共有ストレージを手動でアンマウントします。

gdeploy に関連する問題

BZ#1408926

現在、gdeploy 設定ファイルでは、**ssl_enable** オプションは **volume** セクションの一部です。1つのストレージプールの単一の gdeploy 設定ファイル内で複数のボリュームセクションが使用され、**ssl_enable** がすべてのボリュームセクションに設定されていると、SSL 操作は複数回実行されます。これにより、古いボリュームのマウントに失敗します。そのため、gdeploy 設定ファイルに1行で SSL を設定することはできません。

回避策: 1つのストレージプールの1つの gdeploy 設定ファイルに複数のボリュームセクションがある場合は、変数 **enable_ssl** を1つのボリュームセクションのみに設定し、キーを以下のように設定します。**'client.ssl'**、値: 'on'、**'server.ssl'**、値: 'on'、**'auth.ssl-allow'**、値: *comma separated SSL hostnames*

Arbitrated ボリュームに関連する問題

BZ#1387494

現在、arbiter ボリュームのデータブリックが完全に消費されると、**ENOSPC** エラーで失敗せずに、新しいデータエントリーの作成がさらに rbiter ブリックで作成される可能性があります。ただし、クライアントは、マウントポイントで作成失敗エラーを正しく受信します。そのため、arbiter ブリックにはさらにエントリーが含まれる可能性があります。クライアントから **rm -rf** コマンドを実行すると、削除するファイルの一覧を取得するために databricks のいずれかに対して **readdir** 操作が実行されます。したがって、すべてのブリックでこれらのエントリーのみが削除されます。親ディレクトリーで **rmdir** コマンドを実行すると、データブリックでは成功しますが、その中にファイルがいくつかあるため、**ENOTEMPTY** エラーとともに arbiter で失敗します。

回避策: arbiter ブリックにディレクトリーが依然として含まれていて、マウントからの削除でエラーが発生しない場合は、ディレクトリーとその関連 GFID シンボリックリンクを手動で削除する必要があります。削除するディレクトリーにファイルが含まれている場合は、これらのファイルと関連する GFID ハードリンクを手動で削除する必要があります。

BZ#1388074

rm -rf コマンドの実行中にレプリカまたは arbiter サブボリュームがオフラインになるか、クライアントから切断された場合は、ブリックがオンラインに戻され、セルフ修復が完了したときに、ディレクトリーが再表示されることがあります。ユーザーがマウントから同じ名前のディレクトリーを作成しようとする、この既存のディレクトリーをボリュームの他の DHT サブボリュームに修復する可能性があります。

回避策: マウントから削除が完了しませんでした、ブリックにディレクトリーが含まれている場合は、ディレクトリーとその関連 GFID symlink を手動で削除する必要があります。削除するディレクトリーにファイルが含まれている場合は、これらのファイルと関連する GFID ハードリンクを手動で削除する必要があります。

BZ#1361518

すべてのブリックでファイルを作成していても、arbiter ブリック上でのみ成功する場合、ファイルを使用するアプリケーションは失敗します。ただし、自己修復中に、このファイルは、データソースとマークされた Arbiter ブリックを使用して、データブリックに作成されます。データ自己修復は arbiter ブリックから実行すべきではないため、**gluster volume heal volname info** コマンドの出力にはエントリーが無限に一覧表示されます。

回避策: `gluster volume heal volname info` コマンドの出力に、ファイルの保留中のペナルダーが無期限に表示される場合には、以下の手順を実行して問題が永続化されているかどうかを確認します。

1. `getfattr` コマンドを使用して、以下を確認します。

- `trusted.afr.volname-client* xattrs` はデータブリック上でゼロである場合
- `trusted.afr.volname-client* xattrs` がデータの部分のみの `arbiter brick` でゼロ以外の場合。データ部分は最初の 4 バイトです。

以下に例を示します。

```
# getfattr -d -m . -e hex /bricks/arbiterbrick/file | grep trusted.afr.volname*
getfattr: Removing leading '/' from absolute path names
trusted.afr.volname-client-0=0x00000054000000000000000000
trusted.afr.volname-client-1=0x00000054000000000000000000
```

2. コマンド出力が上記の状態と一致する場合は、以下のコマンドを使用して `xattr` を削除します。

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

分散したボリュームに関連する問題

BZ#1766640

インサービスアップグレードの際に、古いバージョンのクライアントのI/Oが正しく動作するように、特別な処理が必要な場合があります。Red Hat Gluster Storage 3.3.1のクライアントがバージョン3.5にアップグレードされる際に、分散したボリュームを持つサーバーはこの処理を行いません。

回避策:分散ボリュームを使用しており、クライアントがRed Hat Gluster Storage 3.3.1にある場合、サーバーとクライアントをバージョン3.5に移行する際にオフラインアップグレードを実行します。

BZ#1735666

一部のケースでは、カーネルのread-aheadとglusterのread-aheadの両方のメカニズムにより、連続した読み取り操作時に読み取られるデータ量が大幅に増加します。このような場合、特に分散したボリュームでは、連続した読み取りのパフォーマンスが著しく低下します。

この問題が発生した場合は、以下のコマンドを使用してglusterのread-aheadおよびio-cacheの動作を手動で無効にすることで、問題を回避できます。

```
# gluster volume set <volname> read-ahead off
# gluster volume set <volname> io-cache off
```

DHT (Distribute) Translator に関連する問題

BZ#1136718

自動ファイルレプリケーション (AFR) の自己修復ファイルには、AFR 自己修復ファイルを含むブ

リックが修復操作でオフラインになると、部分的に修復されます。リックがオンラインに戻る前に部分的に修復されたファイルに移行すると、移行したファイルには正しくないデータが存在し、元のファイルが削除されます。

BZ#1760713

cluster.lookup-optimize オプションが有効な場合に、リバランス後に名前の変更操作を実行すると、クライアントから一部のファイルにアクセスできなくなるという問題があります。

回避策: ボリュームの **cluster.lookup-optimize** を無効にします。

```
# gluster volume set VOLNAME cluster.lookup-optimize off
```

レプリケーションに関連する問題 (AFR)

BZ#1426128

複製ボリュームでは、ファイルの作成時に gluster ボリュームスナップショットが作成されると、ファイルはレプリカの1つのブリックに存在しますが、スナップショットを作成したボリュームにその他のブリックが存在しない可能性があります。このため、このスナップショットを復元し、マウントからのディレクトリーで **rm -rf dir** コマンドを実行すると、**ENOTEMPTY** エラーで失敗する可能性があります。

回避策: **rm -rf dir** コマンドの実行中に **ENOTEMPTY** エラーが発生しても、ディレクトリーの **ls** コマンドの出力にエントリーが含まれない場合は、レプリカのバックエンドブリックを確認して、ファイルが一部のブリックに存在するかどうかを確認します。マウントからファイル名を指定して **stat** コマンドを実行し、レプリカのすべてのブリックに修復されるようにします。ファイルを完全に修復したら、**rm -rf dir** コマンドの実行に成功します。

BZ#1721355

ボリュームへの書き込み、および **gluster heal info** コマンドの情報の読み取りには、いずれもブロッキングロックが必要です。これは、ボリュームの書き込み負荷が高いときに、**gluster heal info** を実行すると、既存のロックによってブロックされることを意味します。

回避策: クライアントからの書き込み操作の数を減らし、ロックの解放と **gluster heal info** 操作の完了を支援します。

Gluster NFSに関する問題

古い NFS over UDP の問題

- **nfs.mount-udp** オプションを有効にする場合は、Linux でサブディレクトリー (**nfs.export-dir** オプションを使用してエクスポート) をマウントしている間に、**-o proto=tcp** オプションを使用してマウントする必要があります。UDP は、Gluster-NFS サーバー上のサブディレクトリーマウントには対応していません。
- **nfs.mount-udp** オプションはデフォルトで無効になっています。NFS を使用して Red Hat Gluster Storage ボリュームにマウントする場合は、Solaris で **posix-locks** を使用するよう有効にする必要があります。

古いロックの問題

- **fcntl** ロック (NFS Lock Manager) は IPv6 では機能しません。

- NFS クライアントが NAT (ネットワークアドレス変換) ルーターまたはファイアウォールの背後にある場合、ロックの動作は予測できません。現在の NLM の実装では、クライアントの IP アドレスのネットワークアドレス移行が発生しないことを想定します。
- NFS Lock Manager が適切に機能するには、すべてのサーバーとクライアントに解決可能なホスト名があるようにする必要があります。つまり、サーバーはクライアント名を解決でき、クライアントはサーバーのホスト名を解決できる必要があります。

BZ#1413910

Red Hat Gluster Storage 3.2 以降では、すべてのボリュームで、**nfs.disable** オプションが明示的に **on** または **off** のいずれかに設定されます。作成される新規ボリュームのデフォルト値は **on** であるため、これらのボリュームは経由でエクスポートされません。Gluster NFS。バージョン 3.1.x 以前から作成されたスナップショットには、このボリュームオプションがありません。

回避策: ボリュームで以下のコマンドを実行します。

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

復元されたボリュームは、経由でエクスポートされません。Gluster NFS。

階層に関連する問題

BZ#1334262

gluster volume tier attach コマンドがタイムアウトすると、いずれかの状況になります。ボリュームが階層化ボリュームにならないか、階層デーモンが起動しません。

回避策: タイムアウトが確認されたら、以下を実行します。

1. ボリュームが階層化されたボリュームかどうかを確認します。
 - インストールされていない場合は、**gluster volume tier attach** コマンドを再実行します。
 - 存在する場合は次のステップに進みます。
2. 各サーバーに階層デーモンが作成されているかどうかを確認します。
 - 階層デーモンが作成されていない場合は、以下のコマンドを実行します。

```
# gluster volume tier volname start
```

BZ#1303298

階層化されたボリュームのスナップショットにエントリーを一覧表示すると、一部のファイルに対する誤ったパーミッションが表示されます。これは、ユーザーサービス可能なスナップショット (USS) が実際のデータファイルではなく、コールド層内のファイルへのリンクの **stat** 情報を返すためです。これらのファイルは **-----T** パーミッションがあるように見えます。

回避策: FUSE クライアントは、以下のオプションのいずれかを適用することで、この問題を回避できます。

- **use-readdir=no** これは推奨されるオプションです。
- **attribute-timeout=0**

- **entry-timeout=0**

NFS クライアントは、**noac** オプションを適用して問題を回避できます。

BZ#1303045

NFS マウントで I/O 操作が進行中に層がアタッチされると、通常は 3 分から 5 分間、I/O が一時的に一時停止します。

回避策: I/O が 5 分以内に再開されない場合は、**gluster volume start volname force** コマンドを使用して中断せずに I/O を再開してください。

BZ#1273741

ハードリンクのあるファイルは、階層化ボリュームではプロモートされず、降格されません。

この問題に対する既知の回避策はありません。

BZ#1305490

階層移行とハードリンク作成間の競合状態により、**File exists** エラーでハードリンク操作が失敗し、クライアント上の **Stale file handle** メッセージを記録します。これは機能には影響しません。また、ファイルアクセスは期待どおりに機能します。

コールド層でハードリンクが作成された後でも、ホットレイヤーのデータにハードリンクが作成される前に、ファイルがコールド層に移行すると、この競合が発生します。このような場合、ホット層でハードリンクの作成を試みると失敗します。ただし、移行はコールド層のハードリンクをデータファイルに変換し、コールド層にすでにリンクがすでに存在するため、リンクが存在し、予想通りに機能します。

BZ#1277112

ホット層ストレージが満杯になると、コールド層ストレージに書き込みをリダイレクトしたり、データをフラッシュしたりせずに、ファイルの作成や既存のファイルへの新規書き込みなどの書き込み操作が、**No space left on device** で失敗します。

回避策: ホットレイヤーが完全に満杯になっていない場合は、書き込み操作を続行する前に次の CTR プロモート/降格サイクルを待つことで、この問題を回避できます。

ホット層が完全に一杯になると、管理者はホット層から安全な場所にファイルをコピーして、ホット層から元のファイルを削除して、ホット層の空き領域が解放されるまで待つことができます。

BZ#1278391

移行をトリガーする拡張属性を設定する領域がないため、ホット層からの移行は、ホット層がフル状態になると失敗します。

BZ#1283507

破損したファイルをプロモート用に特定して、ホット階層ストレージにプロモートできます。

まれに、BitRot スクラビターにより破損が見逃される可能性があります。これは 2 つの方法で発生する可能性があります。

1. チェックサムが作成される前にファイルが破損しているため、チェックサムが破損したファイルと一致して、BitRot スクラビターはファイルを破損と識別しません。

2. 正常なファイル用にチェックサムが作成され、ファイルが破損します。また、昇格用に識別されて、新しい(破損した)チェックサムが作成されるホット層に昇格される前に、破損したファイルがチェックサムに対して比較されません。

階層が使用中の場合、これらの識別されていない破損したファイルはヒートでき、ホット層にプロモートするために選択できます。破損したファイルがホット層に移行され、ホット層が複製されていない場合は、破損したファイルにアクセスしたり、コールド層に移行したりすることはできません。

BZ#1306917

User Serviceable Snapshot を有効にすると、階層の割り当ては成功しますが、attach tier 操作の実行中に進行中の I/O 操作が古いファイル処理エラーで失敗する可能性があります。

回避策: **attach tier** を実行する前に、ユーザーサービス可能なスナップショットを無効にします。**attach tier** に成功すると、User Serviceable Snapshots が有効になります。

スナップショットに関連する問題

BZ#1403169

スナップショットの取得中に NFS-ganesha が有効化され、そのスナップショットの復元中にそのスナップショットが無効化されるか、共有ストレージがダウンしたときに、スナップショットの復元に失敗します。

BZ#1403195

ブリックが起動してもすべてのトランスレーターが初期化されていないと、スナップショットの作成に失敗する可能性があります。

BZ#1169790

ボリュームが停止し、**.snaps** ディレクトリーにアクセスしようとする、**.snaps** ディレクトリーのカーネル仮想ファイルシステム (VFS) キャッシュに負のキャッシュエントリーが作成されます。ボリュームをオンラインに戻した後、負のキャッシュエントリーが原因で ENOENT エラーで **.snaps** ディレクトリーへのアクセスに失敗します。

回避策: 以下のコマンドを実行して、カーネルの VFS キャッシュを削除します。

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

これにより、一時的なパフォーマンスが低下する可能性があることに注意してください。

BZ#1174618

User Serviceable Snapshot 機能が有効で、ディレクトリーに既存の **.snaps** フォルダーがある場合、そのフォルダーにアクセスすると、予期しない動作が発生する可能性があります。

回避策: 既存の **.snaps** フォルダーの名前を別の名前に変更します。

BZ#1394229

ボリュームセット操作などのクライアントグラフの変更、スナップショットの復元など、最終的にボリュームをマウントするクライアントプロセスのメモリーシナリオが不足します。

BZ#1129675

glusterd が利用できない場合や、ノードが利用できない場合に、クラスターノードでスナップショットの復元を実行すると、以下のエラーが発生します。

- **gluster volume heal vol-name info** コマンドを実行すると、エラーメッセージ **Transport endpoint not connected** が表示されます。
- クライアントが **glusterd** サービスへの接続を試みると、エラーが発生します。

回避策: 全ノードおよび対応する **glusterd** サービスが実行中の場合にのみ、スナップショットを復元します。以下のコマンドを実行して **glusterd** を起動します。

```
# service glusterd start
```

BZ#1118780

ディレクトリーの名前変更中に作成されたスナップショットの復元では (ディレクトリーはハッシュ化サブボリュームで名前変更されていますが、すべてのサブボリュームに変更されています)、古いディレクトリーと新しいディレクトリーがともに存在し、同じ GFID を持ちます。これにより、不整合が生じ、それらのディレクトリー内のファイルへのアクセスが問題が発生する可能性があります。

DHT では、ディレクトリーの名前を変更 (ソース、宛先) が最初にハッシュ化されたサブボリュームで実行し、成功すると残りのサブボリュームで行われます。この時点で、ソースディレクトリーと宛先ディレクトリーの両方が同じ GFID のボリュームに存在します。宛先は、ハッシュ化されたサブボリュームの宛先にあり、ソースは残りの部分にあります。現時点では、(ソースまたは宛先のいずれかでの) 並行ロックアップにより、ディレクトリーがまだ存在していないサブボリュームに、ディレクトリーが作成される可能性があります (ハッシュにはソースディレクトリーエントリー、残りのサブディレクトリーには宛先ディレクトリーエントリー)。したがって、同じ GFID を持つディレクトリーエントリーは source と destination の 2 つになります。

BZ#1236149

ノード/ブリックがダウンすると、force オプションを指定しても **snapshot create** コマンドが失敗します。これは想定される動作です。

BZ#1240227

LVM における LUKS 暗号化には現在対応していません。

BZ#1246183

User Serviceable Snapshots は、Erasure Coded (EC) ボリュームではサポートされていません。

Geo レプリケーションに関連する問題

BZ#1393362

gluster ボリュームのリバランスが進行中にジオメトリーセッションが作成されると、geo レプリケーションによってスレーブボリュームに同期している一部のファイル/ディレクトリーが欠落する可能性があります。これは、リバランスによるファイルの内部移動が原因で生じます。

回避策: マスターボリュームのリバランスが進行中であれば、geo レプリケーションのセッションを作成しないでください。

BZ#1344861

マスタークラスターで1つ以上のノードがダウンすると、geo レプリケーション設定が変更されません。このため、ダウンしたノードには、ノードの起動時に古い設定が適用されます。

回避策: 全ノードが起動すると、Geo レプリケーションの設定コマンドを再度実行します。これにより、プライマリークラスターのすべてのノードには同じ Geo レプリケーション設定オプションが含まれます。

BZ#1293634

プライマリーボリュームが階層化されると、geo レプリケーションが適用されたストレージのパフォーマンスが向上します。これにより、階層化されたボリュームで geo レプリケーションのパフォーマンスが低下します。

BZ#1302320

ファイルの昇格中に、リバランス操作はスティッキービットと suid/sgid ビットを設定します。通常、移行が完了するとこれらのビットが削除されます。移行が完了する前にファイルで `readdirp` が呼び出されると、これらのビットは削除されず、クライアントで適用されません。

ビットの適用中に `rsync` が発生すると、ビットは宛先と同期され、宛先でアクセスに障害が発生しないように、ビットがファイルに適用されます。これは、geo レプリケーションが適用された設定でも発生する可能性があります。リバランスプロセスが継続的であるため、階層化の可能性が高まります。

BZ#1102524

Geo レプリケーションワーカーは障害状態になり、再開時に再起動します。これは、再起動時には通常通りに機能しますが、再開まで同期するのに時間がかかります。

BZ#1238699

Changelog History API は、ブリックパスのセッションが同じままになることを想定します。ただし、スナップショットの復元時に、ブリックパスが変更されます。これにより、履歴 API が失敗し、geo-rep が **Faulty** に変わります。

回避策:

1. スナップショットの復元後に、プライマリーボリュームおよびセカンダリーボリュームが停止していることを確認します。
2. **htime** ディレクトリー (プライマリーボリューム) のバックアップを作成します。

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



注記

拡張属性を保持するには、**-a** オプションの指定が重要です。

以下に例を示します。

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

- 以下のコマンドを実行して、htime ファイルの **OLD** パスを新しいブリックパスに置き換えます。ここでの `OLD_BRICK_PATH` は現在のボリュームのブリックパスで、`NEW_BRICK_PATH` はスナップショットの復元後のブリックパスになります。

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

以下に例を示します。

```
# 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/a4e2c4647cf642f68d0f8259b43494c0/brick0/b0|g'
```

- 復元されたボリュームで、プライマリボリュームおよびセカンダリボリュームおよび Geo レプリケーションセッションを開始します。ステータスは **Active** に更新する必要があります。

自己修復に関連する問題

BZ#1240658

バックエンドのレプリカペアのブリックからファイルが誤って削除され、**gluster volume heal VOLNAME full** を実行すると、ファイルが修復されない可能性があります。

回避策: クライアント (マウント) からファイルでルックアップを実行します。これにより、修復がトリガーされます。

BZ#1173519

既存のファイルに書き込みを行い `_AVAILABLE_BRICK_SPACE_` をチェックすると、書き込みは I/O エラーで失敗します。

回避策: `cluster.min-free-disk` オプションを使用します。 n GB までのファイルを頻繁に書き込む場合は、`min-free-disk` を n より大きい m GB 値に設定できます。

たとえば、ファイルサイズが 5 GB で、これが書き込むファイルサイズの上限である場合は、`min-free-disk` を 8 GB に設定することを検討してください。これにより、ファイルが利用可能な十分な容量を持つブリックに書き込まれます (存在する場合)。

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

replace-brick 操作に関連する問題

- gluster volume replace-brick VOLNAME Brick New-Brick commit force** コマンドを実行すると、その特定ボリュームのファイルシステム操作 (移動中のボリューム) は失敗します。
- replace-brick 操作後、NFS マウントと FUSE マウントで stat 情報が異なります。これは、**replace-brick** 操作の実行時に内部のタイムスタンプが変更されたために発生します。

NFS-Ganesha に関連する問題

再起動後にロック解除ができないことがあります。

NFS サーバーを再起動すると、`grace-period` 機能内でロック解除が失敗する可能性があり、ロックが以前に保持されない可能性があります。

BZ#1570084

`nfs-ganesha` の起動を開始する前にボリュームがエクスポートされる場合は、ボリュームのエクスポートに使用される `dbus` コマンドは失敗します。

回避策: `nfs-ganesha` プロセスを再起動してボリュームをエクスポートします。

BZ#1535849

NFS-Ganesha の場合、キャッシュエントリ用に作成されたメモリーは解放されずに再利用されます。たとえば、「foo」ファイルがあり、「foo」の異なるクライアントキャッシュエントリから削除しても、そのファイルは存在します。その結果、NFS-Ganesha が使用するメモリーはキャッシュが満杯になるまで増大します。

BZ#1461114

ノードを既存の Ganesha クラスターに追加する際に、以下のエラーメッセージが断続的に表示されます。

```
Error: Some nodes had a newer tokens than the local node. Local node's tokens were updated.
Please repeat the authentication if needed
```

```
Error: Unable to communicate with pcsd
```

回避策: 既知の機能への影響がないため、これらのメッセージは無視しても問題ありません。

BZ#1402308

`ganesha` クラスターの設定後に `ifdown` が実行された場合、Corosync サービスがクラッシュします。これは HA 機能に影響を与える可能性があります。

BZ#1416371

NFS-ganesha サーバーでエクスポートされるボリュームで **gluster volume stop** 操作が失敗した場合、コマンドが失敗するにもかかわらず、ボリュームがいくつかのノードでエクスポートされない可能性が高くなります。これにより、NFS-ganesha クラスター全体で一貫性のない状態になります。

回避策: クラスターを通常の状態に復元するには、以下を実行します。

- ボリュームがエクスポートされないノードを特定します。
- 以下の `dbus` コマンドを使用して、ボリュームを手動で再エクスポートします。

```
# 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>)""
```

**注記**

3.5 Batch Update 3 のリリースにより、共有ストレージのマウントポイントが `/var/run/gluster/` から `/run/gluster/` に変更されました。

BZ#1398280

PCS リソースのいずれかが失敗した状態にある場合、ティアダウン完了には多くの時間がかかります。このため、**gluster nfs-ganesha disable** コマンドはタイムアウトします。

回避策: **gluster nfs-ganesha disable** がタイムアウトが発生した場合には、**pcs status** を実行し、リソースが失敗した状態にあるかどうかを確認します。次に、以下のコマンドを使用して、そのリソースのクリーンアップを実行します。

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

gluster nfs-ganesha disable コマンドを再実行してください。

BZ#1328581

ファイルを削除すると、nfs-ganesha プロセスは、削除されたエントリでロックアップを実行して、リンクが存在する場合は属性を更新します。このため、ファイルが削除されると ENOENT で検索に失敗し、**gfapi.log** に誤解を招くログメッセージが発生します。

これは想定される動作であり、機能の問題はありません。このような場合、ログメッセージは無視する必要があります。

BZ#1259402

vdsmd サービスと abrt サービスが相互にインストールされると、vdsmd は **/proc/sys/kernel/core_pattern** ファイルの abrt コアダンプ設定を上書きします。これにより、NFS-Ganesha がコアダンプを生成するのを防ぎます。

回避策: **/etc/vdsm/vdsm.conf** ファイルで **core_dump_enable** を **false** に設定してコアダンプを無効にし、**abrt-ccpp** サービスを再起動します。

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

BZ#1257548

IP フェイルオーバーをトリガーする **nfs-ganesha** サービスモニタースクリプトは、10 秒ごとに定期的に実行されます。デフォルトでは、glusterFS サーバーの ping-timeout (到達できないクライアントのロックがフラッシュされる後) は 42 秒です。IP フェイルオーバー後に、一部のロックは glusterFS サーバードプロセスでクリーンアップされません。したがって、NFS クライアントによるロック状態の回収に失敗します。

回避策: **nfs-ganesha** サービスモニター期間 (デフォルトは 10s) を、Gluster サーバーの ping-timeout (デフォルトの 42) 期間に設定することを推奨します。

したがって、以下のコマンドを使用してネットワークの ping-timeout を減らす必要があります。

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

以下のコマンドを使用して、nfs-service モニターの間隔を増やします。

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

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

BZ#1470025

PCS クラスター IP リソースは、NFS-Ganesha HA クラスターで VIP のフェイルオーバー/フェイルバック中に FAILED 状態になる場合があります。その結果、VIP にアクセスできなくなると、マウントの失敗やシステムのフリーズが生じます。

回避策: 以下のコマンドを使用して、障害が発生したリソースを消去します。

```
# pcs resource cleanup resource-id
```

BZ#1474716

再起動後に、systemd は NFS-Ganesha が STARTED 状態にあると解釈する場合があります (実行されていない場合)。

回避策: NFS-Ganesha プロセスを手動で起動します。

BZ#1473280

gluster nfs-ganesha disable コマンドを実行すると、NFS-Ganesha サービスが停止します。エクスポート前のエントリーの場合には、NFS-Ganesha は FAILED 状態になります。

回避策: 失敗後に NFS-Ganesha プロセスを再起動して、以下のコマンドを再度実行します。

```
# gluster nfs-ganesha disable
```

Red Hat Gluster Storage ボリュームに関連する問題

BZ#1286050

ボリュームでは、読み取り操作および書き込み操作が進行中で、同時にリバランス操作が実行され、そのボリュームで remove-brick 操作が実行されると、**rm -rf** コマンドはいくつかのファイルで失敗します。

BZ#1224153

ブリックプロセスがなくなると、BitD は対応するブリックとの通信に使用されるソケットから読み込みを試みます。失敗すると、BitD はログファイルに障害を記録します。これにより、多くのメッセージがログファイルに置かれ、ソケットからの読み取りに失敗し、ログファイルのサイズが増えます。

BZ#1227672

指定のオブジェクトがクリーンな状態か破損しているかを確認するには、ファイルシステム (オブジェクト) の正常なスクラブが必要になります。ファイルが破損し、ファイルシステムでスクラブが実行されていない場合は、I/O の実行時に適切なコピーを保持するブリックがオフラインであった場合に、破損したオブジェクトが複製される可能性があります。

回避策: 修復中に破損をオンデマンドでオブジェクトが確認する必要があります。

Samba に関連する問題

BZ#1329718

スナップショットボリュームは読み取り専用です。すべてのスナップショットは、.snaps ディレクトリー内のディレクトリーとして利用できます。スナップショットは読み取り専用ですが、スナップショットのディレクトリー属性は、スナップショットボリュームのディレクトリー属性と同じで、読み取り/書き込みが可能です。これにより、Windows はスナップショットディレクトリーが読

み取り/書き込みと仮定されるため、混乱を生じさせる可能性があります。ファイルプロパティーの **Restore previous version** オプションでは、**open** オプションを指定できます。対応するスナップショットからファイルを開きます。ファイルを開いても一時ファイル (Microsoft Word ファイルなど) も作成すると、オープンに失敗します。これは、スナップショットボリュームが読み取り専用であるため、一時ファイルの作成に失敗するためです。

回避策: このようなファイルを、直接開くのではなく、別の場所にコピーします。

BZ#1322672

vdsmd と abrt の ccpp アドオンが相互にインストールされると、vdsmd は `/proc/sys/kernel/core_pattern` で abrt のコアダンプ設定を上書きします。これにより、vdsmd が設定した新しいコアダンプの場所での SELinux 検索拒否により、Samba がコアダンプを生成できなくなります。

回避策: 以下の手順を実行してください。

1. `/etc/vdsm/vdsm.conf` でコアダンプを無効にします。

```
core_dump_enable = false
```

2. abrt-ccpp サービスおよび smb サービスを再起動します。

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

BZ#1282452

ctdb2.5-debuginfo パッケージは現在 samba-debuginfo パッケージと競合するため、ctdb2.5-debuginfo がインストールされていると、ctdb バージョン 4 へのアップグレードは失敗します。

回避策: ctdb バージョン 4 にアップグレードする前に、ctdb2.5-debuginfo パッケージを手動で削除します。必要に応じて、アップグレード後に samba-debuginfo をインストールします。

SELinux に関連する問題

BZ#1256635

Red Hat Gluster Storage は現在、SELinux ラベル付きのマウントをサポートしていません。

FUSE マウントでは、現在 SELinux はサブタイプでファイルシステムを区別できないため、異なる FUSE ファイルシステム (BZ#1291606) を区別することはできません。つまり、Red Hat Gluster Storage のクライアント固有のポリシーは定義できず、SELinux は Red Hat Gluster Storage が追跡するファイルのクライアント側の拡張属性を安全に変換することができません。

BZ#1269584 の一部として、NFS-Ganesha マウントの回避策が進行中です。BZ#1269584 が完了すると、SELinux ラベル付きサポートを含む、NFS バージョン 4.2 の Red Hat Gluster Storage のサポートが有効になります。

BZ#1291194, BZ#1292783

現在の SELinux ポリシーは、ctdb の 49.winbind イベントスクリプトが smbcontrol の実行を阻止します。パブリック IP アドレスがノードから移動されると、winbind で整合性のない状態が発生することがあります。これは、winbind が、その IP アドレスを介して接続を破棄することができないためです。

シャード化に関連する問題

BZ#1332861

シャード化は、基礎となるファイルシステムからの書き込みとして毎回ブロック数の差異に依存し、その数をシャード化されたファイルの既存のブロック数に追加します。ただし、ブロックのXFSの予測事前割り当てにより、予測事前割り当てで xfs が提示した書き込み後のシャードのブロック数が、実際に書き込まれたブロック数よりも大きくなる可能性があるため、この計算は適切ではありません。

このため、シャード化されたファイルの block-count は、ディスク上で消費されるブロックの数よりも大きくなるように示される可能性があります。その結果、**du -sh** などのコマンドが、ファイルが使用する物理ブロックの数よりも高いサイズを報告する可能性があります。

一般的な問題

GFID 不一致によりエラーが発生する

ファイルおよびディレクトリーが異なるバックエンドに GFID と異なる場合は、glusterFS クライアントがハングしたり、エラーを表示する可能性があります。この問題の詳細は、Red Hat サポートにお問い合わせください。

BZ#1236025

スナップショットの復元におけるファイルおよびディレクトリーのタイムスタンプが変更され、適切な変更ログの読み取りに失敗します。**glusterfind pre** は、**historical changelogs not available** というエラーで失敗します。スナップショットの復元後に既存の glusterfind セッションが機能しなくなります。

回避策: 既存の glusterfind セッションから必要な情報を収集し、セッションを削除し、スナップショットの復元を実行してから glusterfind セッションを新たに作成します。

BZ#1573083

storage.reserve の制限に到達し、ディレクトリーが作成されると、ディレクトリーの作成が ENOSPC エラーで失敗し、ディレクトリーの検索で ESTALE エラーが発生します。したがって、ファイル操作が完了しません。

回避策: 回避策はありません。

BZ#1260119

glusterfind コマンドは、クラスターのいずれかのノードから実行する必要があります。クラスターのすべてのノードが、コマンド開始したノードの **known_hosts** 一覧に追加されていない場合は、**glusterfind create** コマンドがハングします。

回避策: ローカルノードを含め、ピアのすべてのホストを **known_hosts** に追加します。

BZ#1127178

レプリカブリックがオンラインに戻る場合は、自己修復が保留している可能性があります。ブリックで **rm -rf** コマンドを実行すると、**Directory not empty** というエラーメッセージで失敗します。

回避策: 保留中の自己修復がないときに操作を再試行します。

BZ#1460629

rm -rf コマンドが、シンクブリックからのファイルパーズに関連する保留中の自己修復エントリーを持つ親ディレクトリーで実行される場合、ディレクトリーおよびファイルはシンクのブリックか

ら削除されない可能性があります。**rm -rf** の **readdir** はソースブリックから提供されるため、エントリー修復保留中のファイルはシンクブリックから削除されません。このようなファイルで修復保留中のデータまたはメタデータは、問題が解決されるまで **heal-info** コマンドの出力に表示されます。

回避策: ファイルおよび親ディレクトリーがその他のブリックに存在しない場合は、シンクブリックから削除します。これにより、それらが次の 'heal-info' 出力に一覧表示されなくなります。

BZ#1462079

クライアントのホスト値が正しくないなどの理由で失敗した場合でも、Statedumpは成功を報告します。

```
# gluster volume statedump volume client not-a-host:port
```

回避策: *host*の値が正しいか確認してください。

Red Hat Gluster Storage AMI に関する問題

BZ#1267209

redhat-storage-server パッケージは、Red Hat Enterprise Linux 7 AMI イメージの Red Hat Gluster Storage Server 3 にデフォルトでインストールされません。

回避策: yum を使用してこのパッケージを手動でインストールすることを強く推奨します。

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

redhat-storage-server パッケージは主に **/etc/redhat-storage-release** ファイルを提供し、ストレージノードの環境を設定します。パッケージは主に **/etc/redhat-storage-release** ファイルを提供し、ストレージノードの環境を設定します。

Red Hat Gluster Storage Web 管理に関連する問題

BZ#1622461

etcd を停止したり、Web Administration サーバーノード自体をシャットダウンしたりすることで発生する可能性のある中央ストア (etcd) が停止すると、すべての Web Administration サービスは etcd の到達可能性に関する例外を報告します。その結果、etcd に到達できなくなるため、Web 管理サービスがクラッシュします。

回避策: etcd が返されたら、Web Administration サービスを再起動します。

4.2. RED HAT GLUSTER STORAGE と RED HAT ENTERPRISE VIRTUALIZATION の統合

コンテキストに関係なく表示されるデータセンター内の全イメージ

Red Hat Gluster Storage サーバーノードと Red Hat Enterprise Virtualization Hypervisor が同じデータセンターにある場合は、仮想マシンの作成時やストレージドメインを追加する際に、両タイプのサーバーが選択できるように一覧表示されます。Red Hat は、Red Hat Gluster Storage サーバーノード用に別のデータセンターを作成することを推奨します。

第5章 テクノロジープレビュー

本章では、本リリースで利用可能なテクノロジープレビュー機能の一覧を紹介します。

現在、テクノロジープレビュー機能は Red Hat Gluster Storage サブスクリプションサービスではサポートされていませんが、機能的に完全ではないことがあるため、通常は実稼働環境には適していません。ただし、この機能はお客様の利便性のために含まれており、機能により多くの公開を提供します。

テクノロジープレビューの機能は、本番環境以外の環境で役に立ちます。また、完全にサポートされる前に、テクノロジープレビュー機能に関するフィードバックおよび機能についてのご提案をお寄せください。重大度の高いセキュリティー問題に対するエラータが提供されます。

テクノロジープレビュー機能の開発中に、追加コンポーネントがテスト用に一般利用可能になる場合があります。Red Hat では、今後のリリースでテクノロジープレビュー機能が完全にサポートされる予定です。



注記

Red Hat Enterprise Linux 6.10 および 7.7 のテクノロジープレビュー機能はすべて、Red Hat Gluster Storage 3.5 のテクノロジープレビュー機能として考慮されています。Red Hat Enterprise Linux 6.10 で利用可能なテクノロジープレビュー機能の詳細は、『『Red Hat Enterprise Linux 6.10 テクニカルノート』』の「[テクノロジープレビュー機能](#)」の章を参照してください。Red Hat Enterprise Linux 7.7 の場合は、「[テクノロジープレビュー機能](#)」を参照してください。

5.1. SMB マルチチャネル

マルチチャネルは、クライアントが複数のトランスポート接続を認証された SMB セッションにバインドできるようにする SMB バージョン 3 機能です。これにより、Windows 8 以降および Windows Server 2012 以降を実行しているクライアントのフォールトトレランスおよびスループットが向上します。

詳細は、「[SMB3 Multi-Channel with Samba on Red Hat Gluster Storage \(テクノロジープレビュー\)](#)」を参照してください。

付録A 改訂履歴

改訂 3.5-0

Wed Oct 30 2019

Red Hat Gluster Storage
Documentation Team

Red Hat Gluster Storage 3.5 のリリースノートの作成および反映