



Red Hat OpenStack Platform 16.1

ストレージガイド

OpenStack での永続ストレージの理解、使用、管理

Red Hat OpenStack Platform 16.1 ストレージガイド

OpenStack での永続ストレージの理解、使用、管理

OpenStack Team
rhos-docs@redhat.com

法律上の通知

Copyright © 2023 Red Hat, Inc.

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

本ガイドでは、Red Hat OpenStack Platform 環境における永続ストレージの使用/管理手順について詳しく説明します。また、各永続ストレージの種別に対応する OpenStack サービスの設定および管理の手順も記載しています。

目次

多様性を受け入れるオープンソースの強化	3
RED HAT ドキュメントへのフィードバック (英語のみ)	4
第1章 RED HAT OPENSTACK PLATFORM での永続ストレージの概要	5
1.1. スケーラビリティおよびバックエンド	6
1.2. アクセシビリティと管理	7
1.3. セキュリティ	7
1.4. 冗長性および災害復旧	7
第2章 BLOCK STORAGE とボリューム	9
2.1. バックエンド	9
2.2. BLOCK STORAGE サービスの管理	9
2.3. ボリュームの基本的な使用方法と設定	28
2.4. ボリュームの高度な設定	45
2.5. マルチパス設定	49
第3章 OBJECT STORAGE サービス	57
3.1. オブジェクトストレージリング	57
3.2. OBJECT STORAGE サービスの管理	59
3.3. RED HAT ENTERPRISE LINUX 用ストレージノードのインストールおよび設定	63
3.4. 基本的なコンテナ管理	67
第4章 SHARED FILE SYSTEMS サービス	70
4.1. SHARED FILE SYSTEMS サービス (MANILA) バックエンド	71
4.2. ファイル共有へのネットワーク接続の確保	76
4.3. ファイル共有に対するアクセス権限の付与	78
4.4. コンピュートインスタンスへのファイル共有のマウント	81
4.5. ファイル共有の削除	82
4.6. SHARED FILE SYSTEMS サービスのデフォルトクォータの変更	83
4.7. エラーのトラブルシューティング	83

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) を参照してください。

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

ドキュメントへのダイレクトフィードバック (DDF) 機能の使用 (英語版のみ)

特定の文章、段落、またはコードブロックに対して直接コメントを送付するには、DDF の **Add Feedback** 機能を使用してください。なお、この機能は英語版のドキュメントでのみご利用いただけます。

1. **Multi-page HTML** 形式でドキュメントを表示します。
2. ドキュメントの右上隅に **Feedback** ボタンが表示されていることを確認してください。
3. コメントするテキスト部分をハイライト表示します。
4. **Add Feedback** をクリックします。
5. **Add Feedback** フィールドにコメントを入力します。
6. オプション: ドキュメントチームが問題の詳細を確認する際に使用できるメールアドレスを記入してください。
7. **Submit** をクリックします。

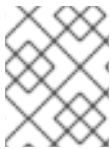
第1章 RED HAT OPENSTACK PLATFORM での永続ストレージの概要

本ガイドは、永続ストレージの作成と管理の手順を説明します。Red Hat OpenStack Platform (RHOSP) では、永続ストレージは主に 3 つのサービスで提供されます。

- Block Storage (**openstack-cinder**)
- Object Storage (**openstack-swift**)
- Shared File Systems ストレージ (**openstack-manila**)

これらのサービスは、異なる種別の永続ストレージを提供します。それぞれのストレージは、異なるユースケースで独自の利点があります。本ガイドでは、一般的なエンタープライズストレージ要件に対する各ストレージの適合性について説明します。

RHOSP Dashboard またはコマンドラインクライアントのどちらかを使用して、クラウドストレージの管理を行うことができます。どちらの方法を使用しても、ほとんどの操作を実施することができます。ただし、一部のより高度な操作は、コマンドラインでのみ実施することができます。本ガイドでは、可能な場合には Dashboard を使用する手順を記載しています。



注記

Red Hat OpenStack Platform の全ドキュメントスイートは [Red Hat OpenStack Platform Documentation](#) で参照してください。



重要

本ガイドでは、**crudini** を使用してカスタムのサービス設定を適用する方法について説明します。そのため、**crudini** パッケージを最初にインストールする必要があります。

```
# dnf install crudini -y
```

RHOSP は、一時ストレージと永続ストレージの 2 種類を認識します。一時ストレージは、特定のコンピュートインスタンスにのみ関連付けられるストレージです。インスタンスが終了されると、一時ストレージも終了します。この種別のストレージは、インスタンスのオペレーティングシステムの保存など、ランタイム時の基本的要件に対応する際に役立ちます。

永続ストレージは、実行中のインスタンスからは独立して存続(永続)するように設計されています。このストレージは、別のインスタンスまたは有効期間を超えた特定のインスタンスが再利用する必要のあるデータに使用されます。RHOSP は以下の種別の永続ストレージを使用します。

ボリューム

OpenStack Block Storage サービス (**openstack-cinder**) により、ユーザーは **ボリューム** を使用してブロックストレージデバイスにアクセスすることができます。一時ストレージを汎用の永続ストレージで拡張するために、インスタンスにボリュームを接続することができます。ボリュームは、任意でインスタンスからデタッチすることも、再度接続することもできます。接続したインスタンス経由でないと、ボリュームにはアクセスできません。

一時ストレージを使用しないようにインスタンスを設定することもできます。一時ストレージを使用する代わりに、Block Storage サービスがイメージをボリュームに書き込むように設定できます。その後、インスタンスのブート可能なルートボリュームとしてボリュームを使用することができます。

ボリュームには、バックアップやスナップショットを使用することで冗長性と災害復旧機能も備わっています。さらに、ボリュームを暗号化できるため、セキュリティが強化されます。

コンテナ

OpenStack Object Storage サービス (openstack-swift) は、メディアファイル、大容量のデータセット、ディスクイメージなど、静的データやバイナリーオブジェクトを保存するために使用する完全に分散されたストレージソリューションを提供します。Object Storage サービスは、コンテナを使用してこれらのオブジェクトを整理します。

ボリュームのコンテンツにはインスタンス経由でしかアクセスできませんが、コンテナの中のオブジェクトには Object Storage REST API 経由でアクセスすることができます。そのため、クラウド内にあるほぼすべてのサービスが、Object Storage サービスをリポジトリとして使用することができます。

ファイル共有

OpenStack Shared File Systems サービス (openstack-manila) は、リモートにある共有可能なファイルシステムまたは **ファイル共有** を簡単にプロビジョニングする手段を提供します。ファイル共有により、クラウド内のテナントはストレージをオープンに共有できます。また、ファイル共有は、複数のインスタンスが同時に消費することが可能です。

各ストレージの種別は、特定のストレージ要件に対応するために設計されています。コンテナは、幅広いアクセスに対応できるように設計されているため、全ストレージ種別において最高レベルのスループット、アクセス、フォールトトレランスが備えられています。コンテナは主にサービスへの使用を対象としています。

一方で、ボリュームは主にインスタンスの消費に使用されます。ボリュームは、コンテナと同じレベルのアクセスやパフォーマンスには対応しにくくなっていますが、コンテナに比べ、機能セットが幅広く、ネイティブのセキュリティ機能も多くなっています。この点では、ファイル共有はボリュームとよく似ていますが、複数のインスタンスにより消費可能である点が異なります。

以下のセクションでは、具体的なストレージ基準との関連において、各ストレージ種別のアーキテクチャーおよび機能セットについて考察します。

1.1. スケーラビリティおよびバックエンド

一般的に、クラスターストレージソリューションは、バックエンドのスケラビリティが高くなっています。Red Hat Ceph を Block Storage (cinder) のバックエンドとして使用する場合は、Ceph Object Storage Daemon (OSD) ノードをさらに追加することで、ストレージの容量および冗長性をスケラリングできます。Block Storage, Object Storage (swift) および Shared File Systems Storage (manila) サービスは、バックエンドとして Red Hat Ceph Storage をサポートします。

Block Storage サービスは、個別のバックエンドとして複数のストレージソリューションを使用することができます。バックエンドレベルでは、バックエンドを追加してサービスを再起動することで、容量をスケラリングすることができます。Block Storage サービスには、多くのサポート対象バックエンドソリューションリストも含まれており、その一部には追加のスケラビリティ機能が備えられています。

デフォルトでは、Object Storage サービスは設定済みのストレージノードを使用しており、空き容量がある分だけ使用することができます。Object Storage サービスは、XFS および ext4 ファイルシステムをサポートし、いずれのサービスもスケラリングして、下層にあるブロックストレージで容量を利用可能な分だけ消費することができます。また、ストレージデバイスをストレージノードに追加して、容量をスケラリングすることも可能です。

Shared File Systems サービスは、1つ以上のサードパーティーのバックエンドのストレージシステムが管理する指定されたストレージプールからファイル共有をプロビジョニングします。この共有ストレージは、サービスで利用可能なストレージプールのサイズまたは数を増やすか、サードパーティーのバツ

クエンドのストレージシステムをデプロイメントに追加してスケーリングできます。

1.2. アクセシビリティと管理

ボリュームは、インスタンスによってのみ消費され、1回に1つのインスタンスにしか接続できず、またそのインスタンス内にしかマウントできません。ボリュームのスナップショットを作成して、クローンを作成する際や以前の状態にボリュームをリストアする際に使用することができます (「冗長性および災害復旧」を参照)。Block Storage サービスでは、新規ボリュームを作成する際にこれらの設定を簡単に呼び出すことができるようにボリュームの各種設定 (例: サイズおよびバックエンド) をまとめた **ボリューム種別** を作成することも可能です。これらの種別はさらに QoS スペックに関連付けて、ユーザー向けに異なるストレージ階層を作成することができます。

ファイル共有は、ボリュームと同様にインスタンスにより消費されます。しかし、ファイル共有の場合はインスタンス内に直接マウントすることができるので、ダッシュボードまたは CLI 経由で接続する必要がありません。ファイル共有は、同時に複数のインスタンスによりマウントすることができます。Shared File Systems サービスは、ファイル共有のスナップショットやクローン作成もサポートしており、(ボリューム種別と同様に) 設定をまとめた **共有種別** を作成することも可能です。

コンテナ内のオブジェクトは、API 経由でアクセスすることができ、クラウド内のインスタンスやサービスからアクセスすることができます。したがって、サービスのオブジェクトリポジトリとして理想的です。たとえば、Image サービス (`openstack-glance`) は Object Storage サービスで管理するコンテナにイメージを保存することができます。

1.3. セキュリティー

Block Storage サービス (`cinder`) は、ボリュームの暗号化を使用して基本的なデータセキュリティーを確保します。これにより、静的キーでボリューム種別を暗号化するように設定できます。このキーは設定したボリュームの種別から作成するボリュームすべてを暗号化する際に使用されます。詳細は、「[ボリュームの暗号化設定](#)」を参照してください。

オブジェクトとコンテナのセキュリティーは、サービスおよびノードレベルで設定されます。Object Storage サービス (`swift`) は、コンテナおよびオブジェクトに対するネイティブの暗号化を提供しません。Object Storage サービスによりクラウド内のアクセス性の優先順位が付けられるため、オブジェクトデータの保護はクラウドのネットワークセキュリティーにのみ依存します。

Shared File Systems サービス (`manila`) では、インスタンスの IP アドレス、ユーザーもしくはグループ、または TLS 証明書別にアクセス制限することでファイル共有のセキュリティーを確保することができます。さらに、一部の Shared File Systems サービスのデプロイメントは、別の共有用サーバーが備えられているため、ファイル共有ネットワークとファイル共有間の関係を管理することができます。共有用サーバーによっては追加のネットワークセキュリティーをサポートする、または必要とする場合があります。たとえば、CIFS ファイル共有サーバーでは LDAP、Active Directory または Kerberos 認証サービスのデプロイメントが必要です。

イメージの署名および検証ならびにメタデータ定義 (`metadef`) API の制限など、Image サービス (`glance`) のセキュリティーを保護する方法についての詳細は、[Creating and Managing Images of the Image service](#) を参照してください。

1.4. 冗長性および災害復旧

Block Storage サービスには、ボリュームのバックアップとリストア機能があり、ユーザーストレージの基本的な災害復旧を行います。バックアップで、ボリュームのコンテンツを保護することができます。それに加え、サービスはクローン作成以外にスナップショットもサポートしており、以前の状態にボリュームをリストアする際に役立ちます。

マルチバックエンドの環境では、バックエンド間でボリュームを移行することも可能です。この機能

は、メンテナンスでバックエンドをオフラインにする必要がある場合に役立ちます。バックアップは通常、データが保護できるように、ソースのボリュームとは別のストレージバックエンドに保存されます。ただし、スナップショットはソースのボリュームに依存するため、この方法を用いることはできません。

Block Storage サービスは、**整合性グループ**の作成もサポートしており、ボリュームをグループ化して同時にスナップショットの作成ができます。これにより、複数のボリューム間のデータの整合性レベルを向上することができます。詳しくは、「[整合性グループの設定と使用](#)」を参照してください。

Object Storage サービスには、ビルトインのバックアップ機能はありません。したがって、すべてのバックアップはファイルシステムまたはノードレベルで行う必要があります。ただし、このサービスにはより強力な冗長機能とフォールトトレランスが備えられており、Object Storage サービスの最も基本的なデプロイメントでさえ、複数回オブジェクトを複製します。**dm-multipath**などのフェイルオーバー機能を使用して、冗長性を強化することができます。

Shared File Systems サービスには、ファイル共有向けのバックアップ機能は組み込まれていませんが、スナップショットを作成してクローンを作成したり、リストアしたりすることができます。

第2章 BLOCK STORAGE とボリューム

Block Storage サービス (`openstack-cinder`) は、全ボリュームの管理タスク、セキュリティ、スケジューリング、全体を管理します。コンピュートインスタンス用の永続ストレージとしては、ボリュームが主に使用されます。

ボリュームのバックアップについての詳しい情報は、[Block Storage Backup Guide](#) を参照してください。

2.1. バックエンド

Red Hat OpenStack Platform は、OpenStack Platform director を使用してデプロイされます。director を使用することで、Block Storage サービス (拡張でバックエンドも含む) など、各サービスが正しく設定されるようにします。director には、複数のバックエンド設定が統合されています。

Red Hat OpenStack Platform は、Block Storage バックエンドとして [Red Hat Ceph](#) および NFS をサポートします。デフォルトでは、Block Storage サービスは、ボリュームのリポジトリとして LVM バックエンドを使用します。このバックエンドはテスト環境に適しますが、LVM は実稼働環境ではサポートされません。

OpenStack で Ceph をデプロイメントする方法は、[Deploying an Overcloud with Containerized Red Hat Ceph](#) を参照してください。

オーバークラウドで NFS ストレージを設定する方法は、[Advanced Overcloud Customization Guide](#) の [Configuring NFS Storage](#) を参照してください。

2.1.1. サードパーティーのストレージプロバイダー

Block Storage サービスをサポート対象のサードパーティー製ストレージアプライアンスを使用するように設定することも可能です。さまざまなバックエンドソリューションを簡単にデプロイできるように、director にはそれに必要なコンポーネントが含まれています。

サポート対象のバックエンドアプライアンスおよびドライバーの完全なリストは、[Component, Plug-In, and Driver Support in Red Hat OpenStack Platform](#) を参照してください。すべてのサードパーティーのバックエンドアプライアンスおよびドライバーには、追加のデプロイメントガイドがあります。適切なデプロイメントガイドを確認し、バックエンドアプライアンスまたはドライバーにプラグインが必要かどうかを判断します。サードパーティーのストレージアプライアンスプラグインのデプロイに関する詳細は、[Advanced Overcloud Customization](#) ガイドの [Deploying a vendor plugin](#) を参照してください。

2.2. BLOCK STORAGE サービスの管理

以下の手順で、Block Storage サービスをニーズに合わせて設定する方法を説明します。以下の手順はすべて管理者権限が必要です。



重要

Block Storage サービス (`cinder`) およびファイバーチャネル (FC) バックエンドを使用するすべてのデプロイメントにおいて、すべてのコントローラーノードおよびコンピューターノードにホストバスアダプター (HBA) をインストールする必要があります。

2.2.1. 高可用性のためのアクティブ/アクティブデプロイメント

重要

アクティブ/アクティブモードは、エッジサイトの分散コンピュートノード (DCN) アーキテクチャーでのみサポートされます。

アクティブ/パッシブモードでは、Block Storage サービスがハイパーコンバージドデプロイメントで失敗した場合、ノードのフェンシングは望ましくありません。これは、ノードのフェンシングがトリガーとなり、不要なストレージのリバランスが行われる場合があるためです。Pacemaker は制御サイトに存在しますが、エッジサイトは Pacemaker をデプロイしません。その代わりに、エッジサイトは、高可用性ハイパーコンバージドデプロイメントをサポートするために、アクティブ/アクティブ設定の Block Storage サービスをデプロイします。

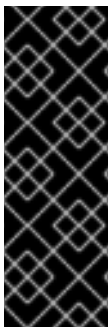
アクティブ/アクティブのデプロイメントでは、利用可能なすべてのノードに負荷をバランスさせることにより、スケーリング機能およびパフォーマンスが向上し、応答時間が短縮されます。アクティブ/アクティブ設定の Block Storage サービスをデプロイすると高可用性環境が構築され、ネットワークの部分的な障害や単一または複数ノードでのハードウェア異常が発生している間に管理レイヤーが維持されます。アクティブ/アクティブのデプロイメントにより、クラスターはノードに障害が発生している間 Block Storage サービスの提供を続けることができます。

ただし、アクティブ/アクティブのデプロイメントでは、ワークフローが自動的に再開することはありません。サービスが停止すると、障害が発生したノードで実行中の個々の操作も、障害が発生している間は実施に失敗します。この場合は、サービスが停止していることを確認し、インフライトの操作に提供されていたリソースのクリーンアップを開始します。

2.2.1.1. 高可用性のためのアクティブ/アクティブ設定の有効化

cinder-volume-active-active.yaml ファイルを使用すると、アクティブ/アクティブ設定で Block Storage サービスをデプロイすることができます。このファイルにより、director は Pacemaker 以外の cinder-volume heat テンプレートをを使用し、**etcd** サービスを分散ロックマネージャー (DLM) としてデプロイメントに追加するようになります。

cinder-volume-active-active.yaml ファイルの **CinderVolumeCluster** パラメーターに値を割り当てて、アクティブ/アクティブ設定のクラスターの名前も定義します。**CinderVolumeCluster** は、Block Storage のグローバルパラメーターです。したがって、同じデプロイメントにクラスター化した (アクティブ/アクティブ) バックエンドとクラスター化していないバックエンドを含めることはできません。



重要

現在、アクティブ/アクティブ設定の Block Storage は、Ceph RADOS Block Device (RBD) バックエンドでのみ機能します。複数のバックエンドを使用する場合、すべてのバックエンドがアクティブ/アクティブ設定をサポートする必要があります。アクティブ/アクティブ設定をサポートしないバックエンドがデプロイメントに含まれている場合には、そのバックエンドをストレージに使用することはできません。アクティブ/アクティブのデプロイメントでは、アクティブ/アクティブ設定をサポートしないバックエンドにデータを保存した場合、データ損失の危険性があります。

手順

アクティブ/アクティブ設定の Block Storage サービスのボリュームを有効にするには、オーバークラウドのデプロイメントに以下の環境ファイルを追加します。

```
-e /usr/share/openstack-tripleo-heat-templates/environments/cinder-volume-active-active.yaml
```

2.2.1.2. アクティブ/アクティブ設定のメンテナンスコマンド

API バージョン 3.17 以降を使用している場合、アクティブ/アクティブ設定のデプロイ後に、さまざまなコマンドを使用して環境と対話することができます。

ユーザーのアクション	コマンド
<p>クラスター名、ホスト、ゾーン、ステータス、状態、無効にした理由、およびバックエンドの状態などの情報を含む、サービスのリストを表示する。</p> <p>注記: Ceph バックエンド用に director によりデプロイされた場合、デフォルトのクラスター名は tripleo@tripleo_ceph になります。</p>	<p>cinder service-list</p>
<p>個別サービスではなく、クラスター全体についての詳細および概要情報を表示する。</p>	<p>cinder cluster-list</p> <p>注記: このコマンドには、cinder API のマイクロバージョン 3.7 以降が必要です。</p>
<p>特定クラスターについての詳細情報を表示する。</p>	<p>cinder cluster-show <cluster_name></p> <p>注記: このコマンドには、cinder API のマイクロバージョン 3.7 以降が必要です。</p>
<p>無効にしたサービスを有効にする。</p>	<p>cinder cluster-enable <cluster_name></p> <p>注記: このコマンドには、cinder API のマイクロバージョン 3.7 以降が必要です。</p>
<p>クラスター化したサービスを無効にする。</p>	<p>cinder cluster-disable <cluster_name></p> <p>注記: このコマンドには、cinder API のマイクロバージョン 3.7 以降が必要です。</p>

2.2.1.3. ボリュームの管理と管理解除

管理解除および管理のメカニズムにより、バージョン X を使用するあるサービスからバージョン X+1 を使用する別のサービスに、ボリュームを容易に移行することができます。このプロセス中、どちらのサービスも稼働を続けます。

API バージョン 3.17 以降では、Block Storage クラスター内の管理に使用できるボリュームおよびスナップショットのリストを確認することができます。これらのリストを表示するには、**cinder manageable-list** または **cinder snapshot-manageable-list** で **--cluster** 引数を使用します。

API バージョン 3.16 以降では、**cinder manage** コマンドにもオプションの **--cluster** 引数を指定できるので、以前に管理解除したボリュームを Block Storage クラスターに追加することができます。

2.2.1.4. クラスター化したサービスでのボリュームの移行

API バージョン 3.16 以降を使用すると、**cinder migrate** および **cinder-manage** コマンドに **--cluster** 引数を指定して、アクティブ/アクティブデプロイメントのデプロイ先を定義することができます。

Block Storage クラスター化サービスのボリュームを移行する場合、オプションの **--cluster** 引数を渡し、位置に関する **host** 引数を省略します。これは、引数が互いに排他的であるためです。

2.2.1.5. サーバーメンテナンスの開始

すべての Block Storage ボリュームサービスは、起動時に独自のメンテナンスを実行します。複数のボリュームサービスがクラスターにグループ化されている環境では、現在実行されていないサービスをクリーンアップすることができます。

コマンド **work-cleanup** により、サーバーのクリーンアップがトリガーされます。このコマンドは以下の出力を返します。

- コマンドでクリーンアップすることのできるサービスのリスト
- 現在クラスター内で実行されていないため、コマンドでクリーンアップすることのできないサービスのリスト



注記

work-cleanup コマンドは、API バージョン 3.24 以降を実行しているサーバーでのみ機能します。

手順

1. 以下のコマンドを実行して、クラスターのすべてのサービスが実行中かどうかを確認します。

```
$ cinder cluster-list --detailed
```

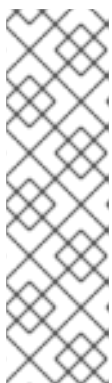
あるいは、**cluster show** コマンドを実行します。

2. いずれかのサービスが実行されていない場合は、以下のコマンドを実行してそのサービスを特定します。

```
$ cinder service-list
```

3. 以下のコマンドを実行してサーバーのクリーンアップを開始します。

```
$ cinder work-cleanup [--cluster <cluster-name>] [--host <hostname>] [--binary <binary>] [--is-up <True|true|False|false>] [--disabled <True|true|False|false>] [--resource-id <resource-id>] [--resource-type <Volume|Snapshot>]
```



注記

--cluster、**--host**、**--binary** 等のフィルターで、コマンドでクリーンアップする対象を定義します。特定のリソースを含め、クラスター名、ホスト名、サービスの種別、およびリソースの種別で絞り込むことができます。フィルターを適用しない場合、コマンドはクリーンアップ可能なすべてを対象に処理を試みます。

クラスター名で絞り込む例を以下に示します。

```
$ cinder work-cleanup --cluster tripleo@tripleo_ceph
```

2.2.2. グループボリューム設定とボリューム種別

Red Hat OpenStack Platform では、ボリューム種別を作成することができ、関連する設定をボリューム種別に適用することができます。ボリュームの作成時に設定を適用することができます。[ボリュームの作成](#) を参照してください。ボリュームの作成後に設定を適用することもできます。[ボリューム種別の](#)

[変更](#) を参照してください。ボリューム種別に適用することができる関連設定の一部を、リストにして以下に示します。

- ボリュームの暗号化。詳細は、[Dashboard を使用したボリューム種別の暗号化設定](#) を参照してください。
- ボリュームが使用するバックエンド。詳細は、[ボリュームを作成するバックエンドの指定](#) および [ボリュームの移行](#) を参照してください。
- Quality-of-Service (QoS) スペック

設定は、追加スペックと呼ばれるキーと値のペアを使用してボリューム種別に関連付けられます。ボリュームの作成時にボリューム種別を指定する際には、Block Storage のスケジューラーがこれらのキーと値のペアを設定として適用します。また、複数のキーと値のペアを同じボリューム種別に関連付けることができます。

ボリューム種別は、異なるユーザーにストレージ階層を使用できるようにする機能をします。特定のパフォーマンス、耐障害性、およびその他の設定をキーと値のペアとしてボリューム種別に関連付けることにより、階層固有の設定を異なるボリューム種別にマッピングすることができます。マッピングされた階層固有の設定は、ボリュームの作成時に対応するボリューム種別を指定することによって適用が可能です。

2.2.2.1. ホストドライバーの機能のリスト表示



注記

利用可能な追加スペックやサポートされている追加スペックは、バックエンドのドライバーにより異なります。有効な追加スペックのリストについては、ボリュームドライバーのマニュアルを参照してください。

または、Block Storage ホストに直接クエリーを送信して、そのドライバーがサポートしている、明確に定義された標準の追加スペックを確認することができます。まず、コマンドラインから Block Storage サービスをホストするノードにログインします。次に、以下のコマンドを実行します。

```
# cinder service-list
```

このコマンドは、各 Block Storage サービス (**cinder-backup**、**cinder-scheduler**、および **cinder-volume**) のホストが含まれるリストを返します。以下に例を示します。

```
+-----+-----+-----+-----+
| Binary | Host      | Zone | Status ...
+-----+-----+-----+-----+
| cinder-backup | localhost.localdomain | nova | enabled ...
| cinder-scheduler | localhost.localdomain | nova | enabled ...
| cinder-volume | *localhost.localdomain@lvm* | nova | enabled ...
+-----+-----+-----+-----+
```

Block Storage サービスのドライバーの機能を表示するには (これによりサポートされる追加スペックを判断)、以下のコマンドを実行します。

```
# cinder get-capabilities _VOLSVCHOST_
```

VOLSVCHOST は **cinder-volume** のホストの完全な名前に置き換えます。以下に例を示します。

```
# cinder get-capabilities localhost.localdomain@lvm
+-----+
| Volume stats | Value |
+-----+
| description | None |
| display_name | None |
| driver_version | 3.0.0 |
| namespace | OS::Storage::Capabilities::localhost.loc...
| pool_name | None |
| storage_protocol | iSCSI |
| vendor_name | Open Source |
| visibility | None |
| volume_backend_name | lvm |
+-----+
+-----+
| Backend properties | Value |
+-----+
| compression | {u'type': u'boolean', u'description'...
| qos | {u'type': u'boolean', u'des ...
| replication | {u'type': u'boolean', u'description'...
| thin_provisioning | {u'type': u'boolean', u'description': u'S...
+-----+
```

Backend properties のコラムには設定可能な追加スペックキーのリストが、**Value** のコラムには、Backend properties に対する有効な値が表示されます。

2.2.2.2. ボリューム種別の作成と設定

1. Dashboard に管理ユーザーとしてログインして **管理 > ボリューム > ボリューム種別** を選択します。
2. **ボリューム種別の作成** をクリックします。
3. **Name** フィールドにボリューム種別の名前を入力します。
4. **Create Volume Type** をクリックします。 **Volume Types** の表に新しい種別が表示されます。
5. ボリューム種別の **View Extra Specs** のアクションを選択します。
6. **Create** をクリックして **Key** と **Value** を指定します。キーと値のペアは有効である必要があります。有効でない場合には、ボリュームの作成時にそのボリューム種別を指定するとエラーが発生してしまいます。
7. **Create** をクリックします。関連付けられた設定 (キー/値のペア) が **追加スペック** の表に表示されます。

デフォルトでは、OpenStack の全テナントがすべてのボリューム種別にアクセス可能です。アクセスが制限されたボリューム種別を作成する必要がある場合は、CLI から作成する必要があります。その方法は、「[プライベートのボリューム種別の作成と設定](#)」を参照してください。



注記

QoS スペックをボリューム種別に関連付けることも可能です。詳細は、「[QoS スペックとボリューム種別に関連付け](#)」を参照してください。

2.2.2.3. ボリューム種別の編集

1. Dashboard に管理ユーザーとしてログインして **管理 > ボリューム > ボリューム種別** を選択します。
2. **ボリューム種別** の表で、ボリューム種別の **追加スペックの表示** のアクションを選択します。
3. このページの **追加スペック** の表では、以下のような操作を行うことができます。
 - ボリューム種別への新規設定の追加。そのためには、**作成** をクリックして、ボリューム種別に関連付ける新規設定のキー/値ペアを指定します。
 - ボリューム種別に関連付けられている既存の設定の編集。そのためには、設定の **編集** アクションを選択します。
 - ボリューム種別に関連付けられている既存の設定の削除。そのためには、追加スペックのチェックボックスを選択して、表示中のダイアログ画面と次の画面で **追加スペックの削除** をクリックします。

2.2.2.4. ボリューム種別の削除

ボリューム種別を削除するには、**ボリューム種別** の表でそのボリューム種別のチェックボックスを選択して **ボリューム種別の削除** をクリックします。

2.2.2.5. プライベートのボリューム種別の作成と設定

デフォルトでは、全テナントですべてのボリューム種別を使用することができます。アクセスが制限されたボリューム種別を作成するには、ボリューム種別を **プライベート** に指定します。そのためには、ボリューム種別の **is-public** フラグを **false** に設定します。

プライベートのボリューム種別は、特定の属性を持つボリュームへのアクセスを制限するのに役立ちます。これは通常は、特定のテナントのみが使用可能とする必要のある設定です。たとえば、テスト中の新規バックエンドや超ハイパフォーマンスの設定などが例としてあげられます。

プライベートのボリューム種別を作成するには、以下のコマンドを実行します。

```
$ cinder type-create --is-public false <TYPE-NAME>
```

デフォルトでは、プライベートのボリューム種別には作成者のみがアクセスできます。ただし、管理ユーザーは、以下のコマンドを使用してプライベートボリューム種別を特定/表示することができます。

```
$ cinder type-list --all
```

このコマンドは、パブリックとプライベートの両方のボリューム種別をリスト表示します。リストには、各ボリューム種別の名前と ID も表示されます。ボリューム種別にアクセスするには、そのボリューム種別の ID が必要となります。

プライベートのボリューム種別へのアクセスは、テナントレベルで許可されます。テナントがプライベートのボリューム種別にアクセスできるようにするには、以下のコマンドを実行します。

```
$ cinder type-access-add --volume-type <TYPE-ID> --project-id <TENANT-ID>
```

プライベートのボリューム種別にアクセス可能なテナントを確認するには、以下のコマンドを実行します。

```
$ cinder type-access-list --volume-type <TYPE-ID>
```

プライベートのボリューム種別のアクセスリストからテナントを削除するには、以下のコマンドを実行します。

```
$ cinder type-access-remove --volume-type <TYPE-ID> --project-id <TENANT-ID>
```



注記

プライベートのボリューム種別へのアクセスは、デフォルトでは管理権限のあるユーザーのみが作成、表示、設定することが可能です。

2.2.3. Block Storage サービスの内部テナントの作成および設定

Block Storage 機能の一部 (例: Image-Volume キャッシュ) では、内部テナントの設定を必要とします。Block Storage サービスは、このテナント (またはプロジェクト) を使用して、通常ユーザーに公開する必要のないブロックストレージアイテムを管理します。このようなアイテムの例として、ボリュームの頻繁なクローン作成のためにキャッシュされたイメージや、移行中のボリュームの一時コピーなどが挙げられます。

内部プロジェクトを設定するには、まず `cinder-internal` という名前の一般プロジェクトとユーザーを作成します。そのためには、コントローラーノードにログインして以下のコマンドを実行します。

```
# openstack project create --enable --description "Block Storage Internal Tenant" cinder-internal
+-----+-----+
| Property | Value |
+-----+-----+
| description | Block Storage Internal Tenant |
| enabled | True |
| id | cb91e1fe446a45628bb2b139d7dccaef |
| name | cinder-internal |
+-----+-----+
# openstack user create --project cinder-internal cinder-internal
+-----+-----+
| Property | Value |
+-----+-----+
| email | None |
| enabled | True |
| id | 84e9672c64f041d6bfa7a930f558d946 |
| name | cinder-internal |
| project_id | cb91e1fe446a45628bb2b139d7dccaef |
| username | cinder-internal |
+-----+-----+
```

新たな設定オプションを追加する手順により、内部テナントが作成されます。詳細は、「[Image-Volume キャッシュの設定および有効化](#)」を参照してください。

2.2.4. Image-Volume キャッシュの設定および有効化

Block Storage サービスには、任意の `Image-Volume キャッシュ` が含まれており、イメージからボリュームを作成する際にこのキャッシュを使用できます。このキャッシュは、頻繁に使用するイメージからボリュームを作成する際の時間を短縮するように設計されています。イメージからボリュームを作成する方法は、「[ボリュームの作成](#)」を参照してください。

Image-Volume のキャッシュを有効化すると、ボリュームの初回作成時にベースとなったイメージのコピーが保存されます。この保存されたイメージは、Block Storage バックエンドのローカルにキャッシュされ、次回このイメージを使用してボリュームを作成する際のパフォーマンス向上に役立ちます。Image-Volume キャッシュは、サイズ (GB)、イメージ数、または両方を指定して上限を設定することができます。

Image-Volume キャッシュは、複数のバックエンドでサポートされます。サードパーティーのバックエンドを使用する場合は、Image-Volume キャッシュサポートに関する情報については、サードパーティーのドキュメントを参照してください。



注記

Image-Volume キャッシュでは、**内部テナント** を Block Storage サービスに設定する必要があります。手順は、「[Block Storage サービスの内部テナントの作成および設定](#)」を参照してください。

バックエンド (**BACKEND**) で Image-Volume キャッシュを有効にして設定するには、アンダークラウドの環境ファイルの **ExtraConfig** セクションに値を追加します。以下に例を示します。

```
parameter_defaults:
  ExtraConfig:
    cinder::config::cinder_config:
      DEFAULT/cinder_internal_tenant_project_id:
        value: TENANTID
      DEFAULT/cinder_internal_tenant_user_id:
        value: USERID
      BACKEND/image_volume_cache_enabled: ❶
        value: True
      BACKEND/image_volume_cache_max_size_gb:
        value: MAXSIZE ❷
      BACKEND/image_volume_cache_max_count:
        value: MAXNUMBER ❸
```

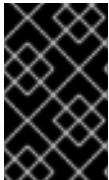
- ❶ **BACKEND** は、ターゲットのバックエンドの名前に置き換えてください (具体的には、その **volume_backend_name** の値)。
- ❷ デフォルトでは、Image-Volume キャッシュサイズはバックエンドによってのみ制限されません。 **MAXSIZE** を GB 単位の数値に変更します。
- ❸ **MAXNUMBER** を使用して、イメージの最大値を設定することもできます。

Block Storage サービスのデータベースは、タイムスタンプを使用して、キャッシュされた各イメージの最終使用日時をトラッキングします。 **MAXSIZE** と **MAXNUMBER** のいずれか一方または両方が設定されている場合は、Block Storage サービスは必要に応じてキャッシュされたイメージを削除し、新たにイメージをキャッシュするためのスペースを解放します。Image-Volume キャッシュが上限に達すると、最も古いタイムスタンプが付いたキャッシュイメージが最初に削除されます。

/home/stack/templates/ に環境ファイルを作成したら、stack ユーザーとしてログインして、以下のコマンドを実行して設定をデプロイします。

```
$ openstack overcloud deploy --templates \
-e /home/stack/templates/<ENV_FILE>.yaml
```

ここで、 **ENV_FILE.yaml** は、前のステップで **ExtraConfig** 設定を追加したファイルの名前です。



重要

オーバークラウドの作成時に追加の環境ファイルを渡した場合は、予定外の変更がオーバークラウドに加えられないように、ここで **-e** オプションを使用して環境ファイルを再度渡します。

openstack overcloud deploy コマンドについての詳しい情報は、**director** のインストールと使用方法の [デプロイメントコマンド](#) を参照してください。

2.2.5. Quality-of-Service スペックの使用

複数のパフォーマンス設定を単一の Quality-of-Service の仕様 (QoS スペック) にマッピングすることができます。これにより、ユーザータイプ別のパフォーマンス階層を提供することができます。

パフォーマンス設定はキーと値のペアとして QoS スペックにマッピングされます。これは、ボリュームの設定がボリューム種別に関連付けられる方法と似ています。ただし、QoS スペックの場合は以下の面でボリューム種別の場合とは異なります。

- QoS スペックは、ディスクの読み取り/書き込み操作を制限するなどのパフォーマンス設定を適用するのに使用されます。利用可能かつサポートされているパフォーマンス設定はストレージドライバーによって異なります。
バックエンドがサポートしている QoS スペックを確認するには、そのバックエンドデバイスのボリュームドライバーのマニュアルを参照してください。
- QoS スペックとは異なり、ボリューム種別はボリュームに直接適用されます。QoS スペックは、ボリューム種別に関連付けられます。また、ボリュームの作成時にボリューム種別を指定すると、そのボリューム種別に関連付けられた QoS スペックにマッピングされたパフォーマンス設定も適用されます。

2.2.5.1. ボリュームの基本 Quality-of-Service

ボリュームの基本 QoS 値を使用して、ボリュームごとにボリュームのパフォーマンス制限を定義することができます。Block Storage サービスでは、以下のオプションがサポートされます。

- **read_iops_sec**
- **write_iops_sec**
- **total_iops_sec**
- **read_bytes_sec**
- **write_bytes_sec**
- **total_bytes_sec**
- **read_iops_sec_max**
- **write_iops_sec_max**
- **total_iops_sec_max**
- **read_bytes_sec_max**
- **write_bytes_sec_max**

- `total_bytes_sec_max`
- `size_iops_sec`

2.2.5.2. QoS スペックの作成と設定

管理者は、QoS スペックの表で QoS スペックの作成および設定を行うことができます。同じ QoS スペックには、複数のキー/値のペアを関連付けることができます。

1. Dashboard に管理ユーザーとしてログインして **管理 > ボリューム > ボリューム種別** を選択します。
2. **QoS スペック** の表で **QoS スペックの作成** をクリックします。
3. **QoS スペック** の名前を入力します。
4. **使用者** フィールドで、QoS ポリシーを適用する対象を指定します。

表2.1 使用者の種別

型	説明
back-end	QoS ポリシーが Block Storage バックエンドに適用されます。
front-end	QoS ポリシーが Compute に適用されます。
both	QoS ポリシーが Block Storage と Compute の両方に適用されます。

5. **Create** をクリックします。新規 QoS スペックが **QoS スペック** の表に表示されるはずですが。
6. **QoS スペック** の表で、新規スペックの **スペックの管理** アクションを選択します。
7. **作成** をクリックして **キー** と **値** を指定します。キーと値のペアは有効である必要があります。有効でない場合には、ボリュームの作成時に、この QoS スペックに関連付けられたボリューム種別を指定するとエラーが発生してしまいます。たとえば、読み取りの IOPS 上限を **500** に設定するには、以下のキー/値のペアを使用します。

```
read_iops_sec=500
```

8. **Create** をクリックします。関連付けられた設定 (キー/値のペア) が **キーと値のペア** の表に表示されます。

2.2.5.3. 容量ベースの QoS 上限の設定

ボリュームの種別を使用して、容量ベースの Quality-of-Service (QoS) 上限をボリュームに実装することができます。これにより、プロビジョニングされるボリュームのサイズに基づいて、確定的な IOPS スループットを設定することができます。このように設定すると、ストレージリソースがユーザーにプロビジョニングされる方法が簡素化され、プロビジョニングされるボリュームのサイズに基づいて、事前に決定された (最終的には高度に予測可能な) スループット速度が提供されます。

特に、Block Storage サービスでは、実際にプロビジョニングされるサイズに基づいてボリュームに割り当てる IOPS を設定することができます。このスループットは、以下の QoS キーを使用して、1GB あたりの IOPS で設定されます。

```
read_iops_sec_per_gb
write_iops_sec_per_gb
total_iops_sec_per_gb
```

これらのキーにより、プロビジョニングされるボリュームのサイズに応じてスケーリングするための読み取り、書き込み、IOPS 合計を設定することができます。たとえば、ボリューム種別に **read_iops_sec_per_gb=500** を指定した場合には、プロビジョニングされる 3 GB のボリュームには、読み取り IOPS が 1500 に自動設定されます。

容量ベースの QoS 上限はボリューム種別ごとに設定され、通常の QoS スペックと同様に設定されます。また、これらの上限は配下の Block Storage サービスにより直接サポートされており、特定のドライバーに依存しません。

ボリューム種別に関する詳しい情報は、「[グループボリューム設定とボリューム種別](#)」および「[ボリューム種別の作成と設定](#)」を参照してください。QoS スペックの設定方法については、「[Quality-of-Service スペックの使用](#)」を参照してください。



警告

容量ベースの QoS 上限を使用して、接続済みのボリュームにボリューム種別を適用した場合 (またはボリューム種別を変更した場合) には、上限は適用されません。この上限は、そのボリュームをインスタンスから接続解除した後のみ適用されます。

ボリューム種別の変更に関する情報は、「[ボリューム種別の変更](#)」を参照してください。

2.2.5.4. QoS スペックとボリューム種別の関連付け

管理者は、**ボリューム種別** の表で QoS スペックを既存のボリューム種別に関連付ける事ができます。

1. Dashboard に管理者としてログインして **管理 > ボリューム > ボリューム種別** を選択します。
2. **ボリューム種別** の表で、その種別の **QoS スペックの関連付けの管理** のアクションを選択します。
3. **関連付ける QoS スペック** のリストから QoS スペックを選択します。
4. **割り当て** をクリックします。選択した QoS スペックが、編集したボリューム種別の **QoS スペックの関連付け** のコラムに表示されるようになります。

2.2.5.5. ボリューム種別からの QoS スペックの関連付け解除

1. Dashboard に管理者としてログインして **管理 > ボリューム > ボリューム種別** を選択します。
2. **ボリューム種別** の表で、その種別の **QoS スペックの関連付けの管理** のアクションを選択します。
3. **関連付ける QoS スペック** のリストから **なし** を選択します。

4. **割り当て** をクリックします。選択した QoS スペックは、編集したボリューム種別の **QoS スペックの関連付け** のコラムに表示されなくなります。

2.2.6. ボリュームの暗号化設定

ボリュームの暗号化は、ボリュームのバックエンドのセキュリティが侵害されたり、盗難されたりした場合に、基本的なデータ保護を提供します。Compute および Block Storage サービスを両方統合して、インスタンスがアクセスを読み込み、暗号化されたボリュームを使用できるようにします。ボリュームの暗号化を使用するには、Key Manager サービス (barbican) をデプロイする必要があります。



重要

- ボリュームの暗号化は、ファイルベースのボリューム (例: NFS) ではサポートされていません。
- 暗号化されていないボリュームを同じサイズの暗号化されたボリュームに種別変更する操作はサポートされません。暗号化したボリュームには、暗号化データを格納するための追加領域が必要なためです。暗号化されていないボリュームの暗号化に関する詳細は、[暗号化されていないボリュームの暗号化](#) を参照してください。

ボリュームの暗号化は、ボリューム種別を使用して適用されます。暗号化されたボリューム種別に関する情報は、[「Dashboard を使用したボリューム種別の暗号化設定」](#) を参照してください。

2.2.6.1. Dashboard を使用したボリューム種別の暗号化設定

暗号化されたボリュームを作成するには、まず **暗号化されたボリューム種別** が必要です。ボリューム種別を暗号化するには、使用すべきプロバイダークラス、暗号、キーサイズを設定する必要があります。

1. Dashboard に管理ユーザーとしてログインして **管理 > ボリューム > ボリューム種別** を選択します。
2. 暗号化するボリューム種別の **アクション** コラムで **暗号化設定の作成** を選択して、**ボリューム種別の暗号化設定の作成** ウィザードを開きます。
3. このウィザードで、ボリューム種別の暗号化の **プロバイダー**、**制御場所**、**暗号**、および **キーサイズ** を設定します。**説明** のコラムで各設定について説明されています。



重要

プロバイダー、**暗号**、および **キーサイズ** のオプションとしてサポートされるは、以下に示す値だけです。

- a. **プロバイダー** に **luks** と入力します。
 - b. **暗号** に **aes-xts-plain64** と入力します。
 - c. **キーサイズ** に **256** と入力します。
4. **ボリューム種別の暗号化設定の作成** をクリックします。

ボリューム種別の暗号化設定が完了したら、その設定を使用して、暗号化されたボリュームを自動的に作成することができます。ボリューム種別の作成に関する詳しい情報は、[「ボリューム種別の作成と設定」](#) を参照してください。具体的には、**ボリュームの作成** ウィンドウの種別のドロップダウンから暗

号化されたボリューム種別を選択します (「[ボリュームの基本的な使用方法と設定](#)」を参照)。

CLI を使用して暗号化されたボリューム種別を設定するには、「[CLI を使用したボリューム種別の暗号化設定](#)」を参照してください。

暗号化されたボリューム種別の暗号化設定を再設定することも可能です。

1. ボリューム種別の **アクション** コラムから **暗号化設定の更新** を選択して、**ボリューム種別の暗号化設定の更新** ウィザードを開きます。
2. ボリュームが暗号化されているかどうかを判断するには、**プロジェクト > コンピュート > ボリューム** にある **ボリューム** テーブルの **暗号化** コラムを確認します。
3. ボリュームが暗号化されている場合には、暗号化のコラムの **はい** をクリックすると暗号化設定が表示されます。

2.2.6.2. CLI を使用したボリューム種別の暗号化設定

Block Storage ボリュームの暗号化を設定するには、以下の手順を実施します。

1. ボリューム種別を作成します。

```
$ cinder type-create encrypt-type
```

2. 暗号、キーサイズ、制御場所、およびプロバイダー設定を定義します。

```
$ cinder encryption-type-create --cipher aes-xts-plain64 --key-size 256 --control-location front-end encrypt-type luks
```

3. 暗号化されたボリュームを作成します。

```
$ cinder --debug create 1 --volume-type encrypt-type --name DemoEncVol
```

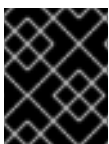
2.2.6.3. ボリュームイメージ暗号化キーの自動削除

Block Storage サービス (cinder) が暗号化されたボリュームを Image サービス (glance) にアップロードする際に、Key Management サービス (barbican) に暗号鍵を作成します。これにより、暗号鍵と保存されるイメージに 1対1 の関係が形成されます。

暗号鍵を削除することで、Key Management サービスがリソースを無制限に消費するのを防ぐことができます。Block Storage サービス、Key Management サービス、および Image サービスは、暗号化されたボリュームの鍵を自動的に管理します。これには、鍵の削除が含まれます。

Block Storage サービスは、自動的に 2つの属性をボリュームイメージに追加します。

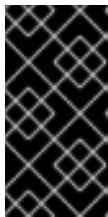
- **cinder_encryption_key_id**: Key Management サービスが特定のイメージ用に保存する暗号鍵の識別子
- **cinder_encryption_key_deletion_policy**: Image サービスはこのポリシーにしたがって、このイメージに関連付けられた鍵を削除するかどうかを Key Management サービスに指示します。



重要

これらの属性の値は、自動的に割り当てられます。意図しないデータ損失を避けるため、これらの値を調整しないでください。

ボリュームイメージを作成すると、Block Storage サービスは `cinder_encryption_key_deletion_policy` 属性を `on_image_deletion` に設定します。`cinder_encryption_key_deletion_policy` が `on_image_deletion` に設定されている場合、ボリュームイメージを削除すると、Image サービスは対応する暗号鍵を削除します。



重要

Red Hat では、`cinder_encryption_key_id` または `cinder_encryption_key_deletion_policy` 属性を手動で操作することを推奨しません。`cinder_encryption_key_id` の値で識別される暗号鍵を他の目的で使用すると、データが失われる危険性があります。

詳しい情報は、[Manage Secret with the OpenStack Key Manager](#) を参照してください。

2.2.7. ボリュームを複数のバックエンドに割り当てる方法の設定

Block Storage サービスが複数のバックエンドを使用するように設定されている場合には、設定済みのボリューム種別を使用して、ボリュームの作成先を指定することができます。詳しくは、「[ボリュームを作成するバックエンドの指定](#)」を参照してください。

ボリュームの作成時にバックエンドを指定していない場合には、Block Storage サービスにより、自動的に選択されます。Block Storage は、最初に定義したバックエンドをデフォルトとして設定します。このバックエンドは、容量がなくなるまで使用されます。容量がなくなった時点で、Block Storage は 2 番目に定義されたバックエンドをデフォルトに設定し、その容量がなくなるとさらに次のバックエンドがデフォルトに設定されるようになっていきます。

このメカニズムが必要な条件を満たさない場合には、フィルタースケジューラーを使用して、Block Storage がバックエンドを選択する方法を制御することができます。このスケジューラーは、以下の例に示したような異なるフィルターを使用して適切なバックエンドをトリアーゼすることができます。

AvailabilityZoneFilter

要求されたボリュームのアベイラビリティゾーン要件を満たさないバックエンドを除外します。

CapacityFilter

ボリュームを収容するのに十分な容量のあるバックエンドのみを選択します。

CapabilitiesFilter

ボリュームで指定した設定に対応可能なバックエンドのみを選択します。

InstanceLocality

クラスターが、同じノードに対してローカルのボリュームを使用するように設定します。

フィルタースケジューラーを設定するには、以下の内容を記載した環境ファイルをデプロイメントに追加します。

```
parameter_defaults:
  ControllerExtraConfig: # 1
    cinder::config::cinder_config:
      DEFAULT/scheduler_default_filters:
        value: 'AvailabilityZoneFilter,CapacityFilter,CapabilitiesFilter,InstanceLocality'
```

1 ControllerExtraConfig: フックとそのネストされているセクションを、既存の環境ファイルの `parameter_defaults:` セクションに追加することもできます。

2.2.8. アベイラビリティゾーンのデプロイ

アベイラビリティゾーンは、クラウドインスタンスおよびサービスをグループ化するためのプロバイダー固有の方法です。director は **CinderXXXAvailabilityZone** パラメーターを使用して、Block Storage ポリユームのバックエンドごとに異なるアベイラビリティゾーンを設定します (**XXX** は特定のバックエンドに対応する値です)。

手順

Block Storage ポリユームのバックエンドごとに異なるアベイラビリティゾーンをデプロイするには、以下の手順を実施します。

1. 以下のパラメーターを環境ファイルに追加して、2つのアベイラビリティゾーンを作成します。

```
parameter_defaults:
  CinderXXXAvailabilityZone: zone1
  CinderYYYAvailabilityZone: zone2
```

以下に示す例のように、**XXX** および **YYY** を、サポートされるバックエンドの値に置き換えます。

```
CinderISCSIAvailabilityZone
CinderNfsAvailabilityZone
CinderRbdAvailabilityZone
```



注記

/usr/share/openstack-tripleo-heat-templates/deployment/cinder/ ディレクトリでバックエンドに関連付けられた heat テンプレートを探し、正しいバックエンドの値を確認してください。

2つのバックエンドをデプロイする例を以下に示します。ここでは、**rbd** がゾーン1で **iSCSI** がゾーン2です。

```
parameter_defaults:
  CinderRbdAvailabilityZone: zone1
  CinderISCSIAvailabilityZone: zone2
```

2. 更新された環境ファイルを追加して、オーバークラウドをデプロイします。

2.2.9. 整合性グループの設定と使用

Block Storage (cinder) サービスを使用して、整合性グループを設定して複数のポリユームを単一のエンティティとしてグループ化することができます。つまり、複数のポリユームに対して個別に操作を実行するのではなく、同時に複数のポリユームに対して操作を実行することができます。整合性グループを使用して、複数のポリユームのスナップショットを同時に作成することができます。また、これらのポリユームを同時にリストアまたはクローン作成することも可能です。

1つのポリユームを複数の整合性グループのメンバーにすることができます。ただし、ポリユームを整合性グループに追加した後に、そのポリユームを削除、種別変更、移行することはできません。

2.2.9.1. 整合性グループの設定

デフォルトでは、整合性グループの API は Block Storage のセキュリティーポリシーにより無効にされています。この機能を使用するには、ここで有効にする必要があります。Block Storage API サービスをホストするノードの `/etc/cinder/policy.json` ファイルの関連する整合性グループエントリー `openstack-cinder-api` にデフォルト設定がリストされています。

```
"consistencygroup:create" : "group:nobody",
"consistencygroup:delete": "group:nobody",
"consistencygroup:update": "group:nobody",
"consistencygroup:get": "group:nobody",
"consistencygroup:get_all": "group:nobody",
"consistencygroup:create_cgsnapshot" : "group:nobody",
"consistencygroup:delete_cgsnapshot": "group:nobody",
"consistencygroup:get_cgsnapshot": "group:nobody",
"consistencygroup:get_all_cgsnapshots": "group:nobody",
```

環境ファイルでこれらの設定を変更してから、`openstack overcloud deploy` コマンドを使用してオーバークラウドにデプロイする必要があります。JSON ファイルを直接編集しないでください。次回オーバークラウドがデプロイされる際に変更が上書きされてしまうためです。

手順

1. 環境ファイルを編集し、`parameter_defaults` セクションに新しいエントリーを追加します。これにより、`openstack overcloud deploy` コマンドを使用して環境が再デプロイされるたびに、エントリーがコンテナで更新され保持されるようになります。
2. `CinderApiPolicies` を使用して環境ファイルに新規セクションを追加し、整合性グループの設定を定義します。JSON ファイルのデフォルト設定を持つ同等の `parameter_defaults` セクションは、次のように表示されます。

```
parameter_defaults:
  CinderApiPolicies: { \
    cinder-consistencygroup_create: { key: 'consistencygroup:create', value: 'group:nobody' }, \
    \
    cinder-consistencygroup_delete: { key: 'consistencygroup:delete', value: 'group:nobody' }, \
    \
    cinder-consistencygroup_update: { key: 'consistencygroup:update', value: 'group:nobody' \
  }, \
    cinder-consistencygroup_get: { key: 'consistencygroup:get', value: 'group:nobody' }, \
    cinder-consistencygroup_get_all: { key: 'consistencygroup:get_all', value: 'group:nobody' }, \
    \
    cinder-consistencygroup_create_cgsnapshot: { key: \
'consistencygroup:create_cgsnapshot', value: 'group:nobody' }, \
    cinder-consistencygroup_delete_cgsnapshot: { key: \
'consistencygroup:delete_cgsnapshot', value: 'group:nobody' }, \
    cinder-consistencygroup_get_cgsnapshot: { key: 'consistencygroup:get_cgsnapshot', \
value: 'group:nobody' }, \
    cinder-consistencygroup_get_all_cgsnapshots: { key: \
'consistencygroup:get_all_cgsnapshots', value: 'group:nobody' }, \
  }
```

3. 値 `'group:nobody'` は、この機能を使用できるグループがないことを決定するため、効果的に無効になります。これを有効にするには、グループを別の値に変更します。

4. セキュリティーを強化するためには、整合性グループの API とボリューム種別管理の API の両方に、同じアクセス権限を設定します。デフォルトでは、ボリューム種別管理の API は (同じ `/etc/cinder/policy.json_file`) で `"rule:admin_or_owner"` に設定されています。

```
"volume_extension:types_manage": "rule:admin_or_owner",
```

5. 整合性グループの機能をすべてのユーザーが利用できるようにするには、API ポリシーのエントリーを設定して、ユーザーが専用の整合性グループを作成、使用、および管理できるようにします。そのためには、`rule:admin_or_owner` を使用します。

```
CinderApiPolicies: { \
  cinder-consistencygroup_create: { key: 'consistencygroup:create', value:
'rule:admin_or_owner' }, \
  cinder-consistencygroup_delete: { key: 'consistencygroup:delete', value:
'rule:admin_or_owner' }, \
  cinder-consistencygroup_update: { key: 'consistencygroup:update', value:
'rule:admin_or_owner' }, \
  cinder-consistencygroup_get: { key: 'consistencygroup:get', value: 'rule:admin_or_owner'
}, \
  cinder-consistencygroup_get_all: { key: 'consistencygroup:get_all', value:
'rule:admin_or_owner' }, \
  cinder-consistencygroup_create_cgnsnapshot: { key:
'consistencygroup:create_cgnsnapshot', value: 'rule:admin_or_owner' }, \
  cinder-consistencygroup_delete_cgnsnapshot: { key:
'consistencygroup:delete_cgnsnapshot', value: 'rule:admin_or_owner' }, \
  cinder-consistencygroup_get_cgnsnapshot: { key: 'consistencygroup:get_cgnsnapshot',
value: 'rule:admin_or_owner' }, \
  cinder-consistencygroup_get_all_cgnsnapshots: { key:
'consistencygroup:get_all_cgnsnapshots', value: 'rule:admin_or_owner' }, \
}
```

6. `/home/stack/templates/` に環境ファイルを作成したら、スタックユーザーとしてログインし、設定をデプロイします。

```
$ openstack overcloud deploy --templates \
-e /home/stack/templates/<ENV_FILE>.yaml
```

`<ENV_FILE.yaml>` を、追加した **ExtraConfig** 設定のファイル名に置き換えます。



重要

オーバークラウドの作成時に追加の環境ファイルを渡した場合には、予定外の変更がオーバークラウドに加えられないように、ここで `-e` オプションを使用して環境ファイルを再度渡します。

`openstack overcloud deploy` コマンドの詳細については、[Director インストールおよび使用ガイドの CLI ツールを使用したオーバークラウドの作成](#) を参照してください。

2.2.9.2. 整合性グループの作成

整合性グループの API を有効にしたら、整合性グループの作成を開始することができます。

手順

1. Dashboard で管理者ユーザーとして **プロジェクト > コンピュート > ボリューム > ボリュームの整合性グループ** を選択します。
2. **整合性グループの作成** をクリックします。
3. ウィザードの **整合性グループの情報** タブで、整合性グループの名前と説明を入力します。次に **アベイラビリティゾーン** を指定します。
4. 整合性グループにボリューム種別を追加することもできます。整合性グループにボリュームを作成する際には、Block Storage サービスにより、これらのボリューム種別から互換性のある設定が適用されます。ボリューム種別を追加するには、**利用可能な全ボリューム種別** リストから追加するボリューム種別の **+** ボタンをクリックします。
5. **整合性グループの作成** をクリックします。次回、作成した整合性グループが **ボリュームの整合性グループ** テーブルに表示されます。

2.2.9.3. 整合性グループの管理

手順

1. (オプション) **アクション** コラムから **整合性グループの編集** を選択して、整合性グループの名前または説明を変更することができます。
2. 整合性グループから直接にボリュームを追加または削除するには、Dashboard で管理者ユーザーとして **プロジェクト > コンピュート > ボリューム > ボリュームの整合性グループ** を選択します。
3. 設定する整合性グループを特定します。その整合性グループの **アクション** コラムで、**ボリュームの管理** を選択します。これにより、**整合性グループボリュームの追加/削除** ウィザードが起動します。
 - a. 整合性グループにボリュームを追加するには、**利用可能な全ボリューム** リストから追加するボリュームの **+** ボタンをクリックします。
 - b. 整合性グループからボリュームを削除するには、**選択済みのボリューム** リストから削除するボリュームの **-** ボタンをクリックします。
4. **整合性グループの編集** をクリックします。

2.2.9.4. 整合性グループのスナップショットの作成および管理

整合性グループにボリュームを追加したら、そこからスナップショットを作成することができます。

手順

1. **openstack-cinder-api** をホストするノードのコマンドラインから **admin** ユーザーとしてログインし、次のように入力します。

```
# export OS_VOLUME_API_VERSION=2
```

これにより、**openstack-cinder-api** のバージョン **2** を使用するようにクライアントが設定されます。

2. 利用可能な整合性グループおよびその ID をすべて表示します。

```
# cinder consisgroup-list
```

3. 整合性グループを使用してスナップショットを作成します。

```
# cinder cgsnapshot-create --name <CGSNAPNAME> --description "<DESCRIPTION>"
<CGNAMEID>
```

以下を置き換えます。

- **<CGSNAPNAME>** とスナップショットの名前 (オプション)。
- スナップショットの説明を含む **<DESCRIPTION>** (オプション)。
- **<CGNAMEID>** と整合性グループの名前または ID。

4. 利用可能な整合性グループのスナップショットの全リストを表示します。

```
# cinder cgsnapshot-list
```

2.2.9.5. 整合性グループのクローン作成

整合性グループを使用して、事前に設定されたボリューム群を一括で同時に作成することもできます。この操作は、既存の整合性グループをクローンするか、整合性グループのスナップショットをリストアすることによって実行できます。いずれのプロセスも同じコマンドを使用します。

手順

1. 既存の整合性グループのクローンを作成するには、以下のコマンドを実行します。

```
# cinder consisgroup-create-from-src --source-cg <CGNAMEID> --name <CGNAME> --
description "<DESCRIPTION>"
```

以下を置き換えます。

- **<CGNAMEID>** は、複製する整合性グループの名前または ID です。
- **<CGNAME>** は、整合性グループの名前です (オプション)。
- **<DESCRIPTION>** は、整合性グループの説明です (オプション)。

2. 整合性グループのスナップショットから整合性グループを作成するには、以下のコマンドを実行します。

```
# cinder consisgroup-create-from-src --cgsnapshot <CGSNAPNAME> --name <CGNAME> -
-description "<DESCRIPTION>"
```

<CGSNAPNAME> は、整合性グループの作成に使用するスナップショットの名前または ID に置き換えてください。

2.3. ボリュームの基本的な使用方法と設定

以下の手順で、基本的なエンドユーザー向けのボリューム管理方法について説明します。これらの手順には管理者の権限は必要ありません。

**重要**

Block Storage サービス (cinder) およびファイバーチャネル (FC) バックエンドを使用するすべてのデプロイメントにおいて、すべてのコントローラーノードおよびコンピュータノードにホストバスアダプター (HBA) をインストールする必要があります。

2.3.1. ボリュームの作成

**重要**

デフォルトでは、1つのプロジェクトで作成可能な最大のボリューム数は10です。

手順

1. Dashboard で **プロジェクト > コンピュート > ボリューム** を選択します。
2. **ボリュームの作成** をクリックして、以下のフィールドを編集します。

フィールド	説明
ボリューム名	ボリュームの名前
説明	ボリュームの簡単な説明 (オプション)
型	<p>オプションのボリューム種別 (「グループボリューム設定とボリューム種別」を参照)</p> <p>複数の Block Storage バックエンドがある場合には、このフィールドを使用して特定のバックエンドを選択します。「ボリュームを作成するバックエンドの指定」を参照してください。</p>
容量 (GB)	<p>ボリュームの容量 (ギガバイト単位)暗号化されていないイメージから暗号化されたボリュームを作成する場合は、暗号化データがボリュームデータを切り捨てないように、ボリュームのサイズがイメージサイズより1GB 以上大きいようにする必要があります。</p>
アベイラビリティゾーン	<p>アベイラビリティゾーン (論理サーバーグループ) は、ホストアグリゲートと併せて、OpenStack 内のリソースを分離する一般的な方法です。アベイラビリティゾーンは、インストール中に定義されます。アベイラビリティゾーンとホストアグリゲートについてのさらに詳しい説明は、ホストアグリゲートの作成と管理 を参照してください。</p>

3. **ボリュームソース** を指定します。

ソース	説明
ソースの指定なし (空のボリューム)	<p>ボリュームは空で、ファイルシステムやパーティションテーブルは含まれません。</p>

ソース	説明
スナップショット	既存のスナップショットをボリュームソースとして使用します。このオプションを選択すると、 スナップショットをソースとして使用する のリストが新たに表示され、スナップショットを選択できるようになります。暗号化されたボリュームのスナップショットから新規ボリュームを作成する場合は、新規ボリュームが古いボリュームより1GB以上大きいようにする必要があります。ボリュームのスナップショットについての詳しい情報は、「 ボリュームスナップショットの作成、使用、削除 」を参照してください。
Image	既存のイメージをボリュームソースとして使用します。このオプションを選択すると、 スナップショットをソースとして使用する のリストが新たに表示され、イメージを選択できるようになります。
ボリューム	既存のボリュームをボリュームソースとして使用します。このオプションを選択すると、 スナップショットをソースとして使用する のリストが新たに表示され、ボリュームを選択できるようになります。

4. **ボリュームの作成** をクリックします。ボリュームが作成されると、**ボリューム** の表に名前が表示されます。

後ほどボリュームの種別を変更することも可能です。詳細は、「[ボリューム種別の変更](#)」を参照してください。

2.3.2. ボリュームを作成するバックエンドの指定

複数の Block Storage (cinder) バックエンドが設定された場合には、必ず、バックエンドごとにボリューム種別も作成する必要があります。その種別を使用して、作成したボリュームに、どのバックエンドを使用するかを指定することができます。ボリューム種別の詳しい情報は、「[グループボリューム設定とボリューム種別](#)」を参照してください。

ボリュームの作成時にバックエンドを指定するには、種別のリストから適切なボリューム種別を選択します（「[ボリュームの作成](#)」を参照）。

ボリュームの作成時にバックエンドを指定しない場合には、Block Storage サービスにより自動的に選択されます。デフォルトでは、このサービスは、最も空き容量の多いバックエンドを選択します。また、Block Storage サービスが利用可能な全バックエンドから無作為に選択するように設定することも可能です。詳細は、「[ボリュームを複数のバックエンドに割り当てる方法の設定](#)」を参照してください。

2.3.3. ボリュームの名前と説明の編集

1. Dashboard で **プロジェクト > コンピュート > ボリューム** を選択します。

2. 対象のボリュームの **ボリュームの編集** ボタンをクリックします。
3. 必要に応じて、ボリュームの名前または説明を編集します。
4. **ボリュームの編集** をクリックして、変更を保存します。



注記

暗号化ボリュームを作成するには、最初にボリュームの暗号化専用設定されたボリューム種別を使用する必要があります。また、Compute サービスと Block Storage サービスの両方で、同じ静的キーを使用するように設定しておく必要があります。ボリュームの暗号化に必要な設定の方法についての説明は、「[ボリュームの暗号化設定](#)」を参照してください。

2.3.4. ボリュームのサイズ変更 (拡張)



注記

使用中のボリュームのサイズを変更する機能はサポートされていますが、ドライバーに依存します。RBD がサポートされています。使用中のマルチ接続ボリュームを拡張することはできません。この機能のサポートについての詳細は、Red Hat のサポートにお問い合わせください。

1. ボリュームをリスト表示し、拡張するボリュームの ID を取得します。

```
$ cinder list
```

2. ボリュームのサイズを変更するには、次のコマンドを実行して正しい API マイクロバージョンを指定し、ボリューム ID と新しいサイズ (古いサイズより大きい値) をパラメーターとして渡します。

```
$ OS_VOLUME_API_VERSION=<API microversion>
$ cinder extend <volume ID> <size>
```

<API_microversion>、<volume_ID>、および <size> を適切な値に置き換えます。次に例を示します。

```
$ OS_VOLUME_API_VERSION=3.42
$ cinder extend 573e024d-5235-49ce-8332-be1576d323f8 10
```

2.3.5. ボリュームの削除

1. Dashboard で **プロジェクト > コンピュート > ボリューム** を選択します。
2. **ボリューム** の表で、削除するボリュームを選択します。
3. **ボリュームの削除** をクリックします。



注記

スナップショットが存在する場合には、ボリュームは削除できません。スナップショットの削除手順については、「[ボリュームスナップショットの作成、使用、削除](#)」を参照してください。

2.3.6. インスタンスへのボリュームの接続と切断

インスタンスでは永続ストレージにボリュームを使用することができます。ボリュームは、1度に1つのインスタンスにしか接続できません。詳細については、[インスタンスの作成と管理ガイドのインスタンスへのボリュームのタッチ](#)を参照してください。

2.3.6.1. インスタンスへのボリュームの接続

1. Dashboard で **プロジェクト > コンピュート > ボリューム** を選択します。
2. **接続の編集** アクションを選択します。ボリュームがインスタンスに接続されていない場合には、**インスタンスへの接続** のドロップダウンリストが表示されます。
3. **インスタンスへの接続** のリストから、ボリュームの接続先となるインスタンスを選択します。
4. **ボリュームの接続** をクリックします。

2.3.6.2. インスタンスからのボリュームの切断

1. Dashboard で **プロジェクト > コンピュート > ボリューム** を選択します。
2. 対象のボリュームの **接続の管理** アクションを選択します。ボリュームがインスタンスに接続されている場合には、そのインスタンスの名前が **接続状況** の表に表示されます。
3. 表示中のダイアログ画面と次の画面で **ボリュームの切断** をクリックします。

2.3.7. 複数インスタンスへのボリュームの接続

ボリュームのマルチ接続により、複数のインスタンスが Block Storage ボリュームに同時に読み取り/書き込みアクセスを行うことができます。Ceph RBD ドライバーがサポートされます。



警告

複数インスタンスからの書き込み操作を管理するには、マルチ接続またはクラスター対応のファイルシステムを使用する必要があります。それ以外の設定では、データの破損が生じます。

マルチ接続ボリュームの制限

- Block Storage (cinder) のバックエンドは、マルチ接続ボリュームをサポートしている必要があります。サポートされるバックエンドについての情報は、Red Hat のサポートにお問い合わせください。
- Block Storage (cinder) のドライバーは、マルチ接続ボリュームをサポートしている必要があります。ベンダープラグインでマルチ接続がサポートされることを確認するには、Red Hat のサポートにお問い合わせください。ベンダープラグインの認定についての詳しい情報は、以下のアティクルおよび Web サイトを参照してください。
 - [Component, Plug-In, and Driver Support in Red Hat OpenStack Platform](#)
 - [Red Hat Ecosystem Catalog](#)

- 読み取り専用のマルチ接続ボリュームはサポートされていません。
- マルチ接続ボリュームのライブマイグレーションは利用できません。
- マルチ接続ボリュームの暗号化はサポートされていません。
- マルチ接続ボリュームは、ベアメタルプロビジョニングサービス (ironic) virt ドライバーではサポートされていません。マルチ接続ボリュームは、libvirt virt ドライバーによってのみサポートされます。
- アタッチされたボリュームをマルチ接続タイプから非マルチ接続タイプに再入力したり、非マルチ接続タイプをマルチ接続タイプに再入力したりすることはできません。
- アタッチされたボリュームの移行中に、複数の読み取り/書き込みアタッチメントを持つマルチ接続ボリュームをソースボリュームまたは宛先ボリュームとして使用することはできません。
- 見送られていたオフロードされたインスタンスにマルチ接続ボリュームをアタッチすることはできません。

2.3.7.1. マルチ接続ボリューム種別の作成

複数のインスタンスにボリュームを接続するには、ボリュームの追加スペックで **multiattach** フラグを **<is> True** に設定します。マルチ接続のボリューム種別を作成すると、ボリュームはフラグを継承し、マルチ接続のボリュームになります。



注記

デフォルトでは、新規ボリューム種別の作成は管理者だけができる操作です。

手順

1. 以下のコマンドを実行し、マルチ接続のボリューム種別を作成します。

```
$ cinder type-create multiattach
$ cinder type-key multiattach set multiattach="<is> True"
```



注記

この手順では、マルチ接続をサポートする任意のバックエンドにボリュームを作成します。したがって、マルチ接続をサポートするバックエンドが2つある場合、スケジューラーは、ボリューム作成時に利用可能な領域に基づいて使用するバックエンドを決定します。

2. バックエンドを指定するには、以下のコマンドを実行します。

```
$ cinder type-key multiattach set volume_backend_name=<backend_name>
```

2.3.7.2. ボリューム種別の変更

ボリュームをマルチ接続可能な種別に変更することや、マルチ接続可能なボリュームを複数インスタンスに接続できない種別に変更することが可能です。ただし、ボリューム種別を変更することができるのは、ボリュームが使用中ではなくそのステータスが **available** の場合に限られます。

マルチ接続のボリュームを接続する場合、一部のハイパーバイザーでは、キャッシュを無効にする場合

など、特別な考慮が必要になります。現在、接続済みのボリュームを接続したまま安全に更新する方法はありません。複数のインスタンスに接続されているボリュームの種別変更を試みると、変更に失敗します。

2.3.7.3. マルチ接続ボリュームの作成

マルチ接続のボリューム種別を作成したら、マルチ接続のボリュームを作成します。

手順

1. 以下のコマンドを実行し、マルチ接続のボリュームを作成します。

```
$ cinder create <volume_size> --name <volume_name> --volume-type multiattach
```

2. 以下のコマンドを実行して、ボリュームがマルチ接続に対応していることを確認します。ボリュームがマルチ接続に対応している場合、**multiattach** フィールドは **True** と表示されます。

```
$ cinder show <vol_id> | grep multiattach  
  
| multiattach | True |
```

これで、複数のインスタンスにボリュームを接続できるようになりました。インスタンスにボリュームをアタッチする方法についての詳細は、[インスタンスへのボリュームの接続と切断](#)を参照してください。

2.3.7.4. サポート対象のバックエンド

Block Storage のバックエンドは、マルチ接続をサポートしている必要があります。サポートされるバックエンドについての情報は、Red Hat のサポートにお問い合わせください。

2.3.8. 読み取り専用ボリューム

データが誤って上書きまたは削除されないように、ボリュームを読み取り専用に指定することができます。これを行うには、次のコマンドを使用して、ボリュームを読み取り専用に設定します。

```
# cinder readonly-mode-update <VOLUME-ID> true
```

読み取り専用ボリュームを読み取り/書き込み可能に戻すには、以下のコマンドを実行します。

```
# cinder readonly-mode-update <VOLUME-ID> false
```

2.3.9. ボリュームの所有者の変更

ボリュームの所有者を変更するには、ボリューム転送を実行する必要があります。ボリュームの転送はボリュームの所有者によって開始され、所有権の変更は、転送が新しい所有者によって受け入れられた後に完了します。

2.3.9.1. コマンドラインを使用したボリュームの譲渡

1. コマンドラインから、ボリュームの現在の所有者としてログインします。
2. 利用可能なボリュームをリスト表示します。

■

```
# cinder list
```

- 以下のコマンドを実行して、ボリュームの譲渡を開始します。

```
# cinder transfer-create VOLUME
```

VOLUME は譲渡するボリュームの名前または **ID** に置き換えます。以下に例を示します。

```
+-----+-----+
| Property |          Value          |
+-----+-----+
| auth_key | f03bf51ce7ead189      |
| created_at | 2014-12-08T03:46:31.884066 |
| id | 3f5dc551-c675-4205-a13a-d30f88527490 |
| name | None                  |
| volume_id | bcf7d015-4843-464c-880d-7376851ca728 |
+-----+-----+
```

cinder transfer-create コマンドはボリュームの所有権を消去し、譲渡用の **id** と **auth_key** を作成します。この値は別のユーザーに渡すことができます。受け取ったユーザーは、その値を使用して譲渡を承認し、ボリュームの新しい所有者となります。

- 新規ユーザーがボリュームの所有権を宣言できる状態となりました。所有権を宣言するには、ユーザーは最初にコマンドラインからログインして以下のコマンドを実行する必要があります。

```
# cinder transfer-accept TRANSFERID TRANSFERKEY
```

TRANSFERID と **TRANSFERKEY** はそれぞれ、**cinder transfer-create** コマンドで返された **id** と **auth_key** の値に置き換えます。以下に例を示します。

```
# cinder transfer-accept 3f5dc551-c675-4205-a13a-d30f88527490 f03bf51ce7ead189
```

注記

利用可能なボリュームの譲渡をすべて表示するには、以下のコマンドを実行します。

```
# cinder transfer-list
```



2.3.9.2. ダッシュボードを使用してボリュームを転送する

Dashboard を使用したボリューム譲渡の作成

- Dashboard にボリュームの所有者としてログインして **プロジェクト > ボリューム** を選択します。
- 譲渡するボリュームの **アクション** のコラムで、**譲渡の作成** を選択します。
- ボリュームの譲渡の作成** ダイアログボックスで、譲渡名を入力して **ボリュームの譲渡の作成** をクリックします。
ボリュームの譲渡が作成され、**ボリュームの譲渡** の画面で **譲渡 ID** と **認証キー** を取得して譲渡先のプロジェクトに送信することができます。

譲渡認証情報のダウンロード ボタンをクリックして **transfer name**、**transfer ID**、**authorization key** が記載されている **.txt** ファイルをダウンロードします。



注記

認証キーは **ボリュームの譲渡** の画面にしか表示されません。この認証キーをなくした場合には、譲渡をキャンセルし、別の譲渡を作成して新たな認証キーを生成する必要があります。

4. **ボリュームの譲渡** の画面を閉じて、ボリュームのリストに戻ります。
譲渡先のプロジェクトが譲渡を受理するまで、ボリュームのステータスは **awaiting-transfer** と表示されます。

Dashboard を使用したボリューム譲渡の受理

1. Dashboard にボリュームの譲渡先としてログインして **プロジェクト > ボリューム** を選択します。
2. **譲渡の受理** をクリックします。
3. **ボリュームの譲渡の受理** のダイアログボックスで、ボリュームの所有者から受け取った **譲渡 ID** と **認証キー** を入力して、**ボリュームの譲渡の受理** をクリックします。
譲渡先のプロジェクトのボリュームリストに、そのボリュームが表示されるようになります。

2.3.10. ボリュームスナップショットの作成、使用、削除

ボリュームのスナップショットを作成することによって、ある特定の時点のボリュームの状態を保持することができます。そのスナップショットを使用して、新規ボリュームをクローン作成することが可能です。



注記

ボリュームのバックアップはスナップショットとは異なります。バックアップはボリューム内のデータを保持するのに対して、スナップショットはある特定の時点におけるボリュームの状態を保持します。スナップショットが存在している場合にはボリュームを削除することはできません。ボリュームのバックアップはデータ損失を防ぎます。一方、スナップショットはクローン作成を円滑化します。

このため、スナップショットのバックエンドは、クローン作成中のレイテンシーを最小限に抑えるように、通常ボリュームのバックエンドと同じ場所に配置されます。一方、バックアップのリポジトリは通常、一般的なエンタープライズデプロイメント内の別の場所に配置されます (例: 異なるノード、物理ストレージ、あるいは別の地理的ロケーションの場合もあり)。これは、ボリュームのバックエンドが一切ダメージを受けないように保護することを目的とします。

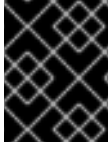
ボリュームのバックアップについての詳しい情報は、[Block Storage Backup Guide](#) を参照してください。

ボリュームのスナップショットを作成するには、以下の手順を実施します。

1. Dashboard で **プロジェクト > コンピュート > ボリューム** を選択します。
2. ターゲットボリュームの**スナップショットの作成アクション**を選択します。

- 作成するスナップショットの **スナップショット名** を指定して **ボリュームのスナップショットの作成** をクリックします。ボリュームのスナップショット タブに全スナップショットが表示されます。

ボリュームのスナップショット の表にスナップショットが表示されたら、そのスナップショットから新規ボリュームをクローン作成することができます。スナップショットの **ボリューム作成** アクションを選択します。ボリュームの作成に関する詳しい説明は、「[ボリュームの作成](#)」を参照してください。



重要

暗号化されたボリュームのスナップショットから新規ボリュームを作成する場合は、新規ボリュームが古いボリュームより1GB以上大きいようにします。

スナップショットを削除するには、**ボリュームスナップショットの削除** アクションを選択します。

OpenStack デプロイメントで Red Hat Ceph バックエンドを使用している場合には、「[Red Hat Ceph Storage バックエンドにおけるスナップショットの保護と保護解除](#)」でスナップショットのセキュリティーとトラブルシューティングについての詳しい情報を参照してください。



注記

スナップショットから作成される Block Storage サービス (cinder) の RADOS ブロックデバイス (RBD) ボリュームの場合は、**CinderRbdFlattenVolumeFromSnapshot** heat パラメーターを使用してフラット化し、スナップショットの依存関係を削除することができます。**CinderRbdFlattenVolumeFromSnapshot** を **true** に設定すると、Block Storage サービスは RBD ボリュームをフラット化してスナップショットの依存関係を削除すると共に、それ以降のスナップショットもすべてフラット化します。デフォルト値は **false** で、cinder RBD ドライバーのデフォルト値も **false** です。

スナップショットをフラット化すると、親との潜在的なブロック共有が削除され、バックエンドでのスナップショットサイズが大きくなり、スナップショット作成の時間が長くなることに注意してください。

2.3.10.1. Red Hat Ceph Storage バックエンドにおけるスナップショットの保護と保護解除

Red Hat Ceph Storage を OpenStack デプロイメントのバックエンドとして使用する場合には、そのバックエンドでスナップショットの **保護** を設定することができます。OpenStack で (Dashboard または **cinder snapshot-delete** コマンドを使用して) 保護されているスナップショットの削除を試みると、操作は失敗します。

このようなエラーが発生した場合には、まず Red Hat Ceph バックエンドでスナップショットを **保護解除** に設定します。それ以降は、通常どおりに OpenStack でスナップショットを削除することができるはずです。

詳細については、[Red Hat Ceph ストレージブロックデバイスガイド](#)の次のリンクを参照してください。

- [ブロックデバイススナップショットの保護](#)
- [ブロックデバイススナップショットの保護解除](#)

2.3.11. スナップショットを使用したボリュームの最後の状態への復元

ボリュームの最新スナップショットの状態に復元することができます。つまり、インプレースでボリュームデータをその最新スナップショットの状態に戻すことができます。

警告

ボリュームの最新スナップショットの状態に復元する機能はサポート対象ですが、ドライバーに依存します。この機能を正しく実装するには、ドライバー側の支援が必要です。この機能に対するサポートの詳細は、ドライバーのベンダーにお問い合わせください。

制限

- マルチ接続のボリュームの場合、スナップショットの状態に戻す機能の使用には制限が適用される可能性があります。この機能を使用する前に、このような制限が適用されるかどうかを確認してください。
- スナップショットの作成後にサイズを変更 (拡張) したボリュームを元に戻すことはできません。
- 接続済みまたは使用中のボリュームに対して、スナップショットの状態に戻す機能を使用することはできません。

前提条件

- Block Storage (cinder) API マイクロバージョン 3.40 またはそれ以降
- ボリュームのスナップショットを少なくとも1つ作成していること。

手順

1. アンダークラウドに **stack** ユーザーとしてログインします。
2. source コマンドで **overcloudrc** ファイルを読み込みます。

```
[stack@undercloud ~] $ source overcloudrc
```

3. ボリュームを切断します。

```
$ nova volume-detach <instance_id> <vol_id>
```

<instance_id> および <vol_id> を、元に戻すインスタンスおよびボリュームの ID に置き換えてください。

4. 元に戻すスナップショットの ID または名前を探します。元に戻すことができるのは最新のスナップショットだけです。

```
$ cinder snapshot-list
```

5. スナップショットの状態に戻します。

```
$ cinder --os-volume-api-version=3.40 revert-to-snapshot <snapshot_id or snapshot_name>
```

<snapshot_id or snapshot_name> をスナップショットの ID または名前に置き換えてください。

6. オプション: **cinder snapshot-list** コマンドを使用して、元に戻しているボリュームが reverting の状態にあることを確認することができます。

```
$ cinder snapshot-list
```

7. ボリュームを再接続します。

```
$ nova volume-attach <instance_id> <vol_id>
```

<instance_id> および <vol_id> を、元に戻したインスタンスおよびボリュームの ID に置き換えてください。

2.3.11.1. 正常に復元していることの確認

手順

- 手順が正常に行われたことを確認するには、**cinder list** コマンドを使用して、元に戻したボリュームが `available` の状態にあることを検証します。

```
$ cinder list
```

注記

Block Storage (cinder) をブート可能なルートボリュームとして使用した場合、ボリュームは `available` の状態にないため、そのボリュームでスナップショットの状態に戻す機能を使用することはできません。この機能を使用するには、インスタンスが終了した場合にブートボリュームを保持するために、**delete_on_termination=false** (デフォルト) の属性を設定してインスタンスをブートしている必要があります。スナップショットの状態に戻す場合は、ボリュームが利用可能になるように、まず初期インスタンスを削除する必要があります。その後、元に戻してボリュームから新規インスタンスを作成することができます。

2.3.12. Image サービスへのボリュームのアップロード

イメージとして既存のボリュームを Image サービスに直接アップロードすることができます。そのためには、以下の手順を実施します。

1. Dashboard で **プロジェクト > コンピュート > ボリューム** を選択します。
2. 対象のボリュームの **イメージにアップロード** アクションを選択します。
3. ボリュームの **イメージ名** を指定して、リストから **ディスク形式** を選択します。
4. **アップロード** をクリックします。

アップロードしたイメージを表示するには、**プロジェクト > コンピュート > イメージ** を選択します。新しいイメージが **イメージ** の表に表示されます。イメージの使用法と設定方法については、**イメージの作成と管理ガイド**の **イメージの管理** を参照してください。

2.3.13. バックエンド間でのボリュームの移動

ボリュームをあるストレージバックエンドから別のストレージバックエンドに移動する理由には、以下のような理由があります。

- サポートされなくなったストレージシステムの使用を停止するため。
- ボリュームのストレージクラスまたは階層を変更するため。
- ボリュームのアベイラビリティゾーンを変更するため。

Block Storage サービス (cinder) を使用すると、以下の方法でバックエンド間でボリュームを移動することができます。

- 種別の変更: デフォルトのポリシーにより、ボリュームの所有者と管理者はボリューム種別を変更することができます。種別変更の操作は、バックエンド間でボリュームを移動する最も一般的な方法です。
- 移行: デフォルトのポリシーでは、管理者のみがボリュームを移行することができます。ボリュームの移行は特定のユースケース用に制限されます。これは、移行に制約があるため、またデプロイメントの動作について明確に理解する必要があるためです。詳細は、[ボリュームの移行](#) を参照してください。

制約

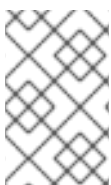
Red Hat は、同一または異なるアベイラビリティゾーン (AZ) にあるバックエンド間のボリュームの移動をサポートしますが、以下の制約があります。

- 移行するボリュームは、利用可能な状態または使用中の状態のいずれかでなければなりません。
- 使用中のボリュームのサポートはドライバーに依存します。
- ボリュームにはスナップショットを含めることができません。
- ボリュームは、グループまたは整合性グループに所属させることはできません。

2.3.14. 利用可能なボリュームの移動

すべてのバックエンド間で利用可能なボリュームを移動できますが、パフォーマンスは使用するバックエンドにより異なります。多くのバックエンドは、アシスト付き移行をサポートします。バックエンドのアシスト付き移行のサポートについての詳細は、ベンダーにお問い合わせください。

アシスト付き移行は、ボリューム種別変更およびボリュームの移行の両方で機能します。アシスト付き移行により、バックエンドはソースバックエンドから移行先バックエンドへのデータの移動を最適化しますが、両方のバックエンドが同じベンダーから取得されている必要があります。



注記

Red Hat は、マルチプールバックエンドを使用する場合、または RBD などのシングルプールバックエンドに cinder の移行操作を使用する場合のみ、バックエンドアシスト付き移行をサポートしています。

2.3.14.1. 通常のボリューム移行

バックエンド間のアシスト付き移行ができない場合には、Block Storage サービスは通常のボリューム移行を実行します。

通常のボリューム移行では、Block Storage (cinder) サービスが移行元ボリュームからコントローラーノードに、およびコントローラーノードから移行先ボリュームにデータを移動する前に、両方のバックエンド上のボリュームを接続する必要があります。Block Storage サービスは、ソースおよび宛先のバックエンドのストレージの種類に関わらず、プロセスをシームレスに実行します。



重要

通常のボリューム移行を実行する前に、十分な帯域幅を確保してください。通常のボリューム移行にかかる時間は、ボリュームのサイズと直接比例するため、操作はアシスト付き移行よりも遅くなります。

2.3.15. 使用中のボリュームの移動

使用中のボリュームを移動する場合、最適化のオプションまたはアシスト付きのオプションはありません。使用中のボリュームを移行する場合には、Compute サービス (nova) がハイパーバイザーを使用して、移行元バックエンドのボリュームから移行先バックエンドのボリュームにデータを転送する必要があります。これには、ボリュームが使用されているインスタンスを実行するハイパーバイザーとの連携が必要です。

Block Storage サービス (cinder) と Compute サービスは、連携してこの操作を実施します。データがコンピュータノードを介してあるボリュームから別のボリュームにコピーされるため、Compute サービスがほとんどの作業を管理します。



重要

使用中のボリュームを移動する前に、十分な帯域幅を確保してください。この操作にかかる時間は、ボリュームのサイズと直接比例するため、操作はアシスト付き移行よりも遅くなります。

制約

- 使用中のマルチ接続ボリュームは、複数の nova インスタンスに接続している間は移動できません。
- ターゲットバックエンドのストレージプロトコルを iSCSI、ファイバーチャネル (FC)、または RBD に制限する、非ブロックデバイスはサポートされません。

2.3.16. ボリューム種別の変更

ボリューム種別の変更は、あるバックエンドから別のバックエンドにボリュームを移動する標準的な方法です。操作を実行するには、管理者はそれぞれのバックエンドに適切なボリューム種別を定義する必要があります。デフォルトのポリシーにより、ボリュームの所有者と管理者はボリューム種別を変更することができます。

ボリューム種別を変更する際には、ボリューム種別とその設定を既存のボリュームに適用します。ボリューム種別の詳しい情報は、「[グループボリューム設定とボリューム種別](#)」を参照してください。

新規ボリューム種別の追加スペックを既存のボリュームに適用できる場合、ボリューム種別を変更することができます。ボリューム種別を変更して、事前定義の設定やストレージ属性を既存ボリュームに適用することができます。以下に例を示します。

- 異なるバックエンドへボリュームを移行する場合
- ボリュームのストレージクラスまたは階層を変更するため。
- レプリケーションなどの機能を有効または無効にする場合

ボリューム種別を変更したからといって、そのボリュームをあるバックエンドから別のバックエンドに移動しなければならない訳ではありません。ただし、種別変更を完了するのにボリュームを移動しなければならない場合があります。

- 新しいボリュームタイプには、異なる **volume_backend_name** 定義されています。
- 現在のボリュームタイプの **volume_backend_name** は未定義であり、ボリュームは新しいボリュームタイプの **volume_backend_name** で指定されたものとは異なるバックエンドに格納されます。

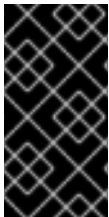
ボリュームをあるバックエンドから別のバックエンドに移動するには、非常に多くの時間とリソースが必要になる場合があります。したがって、種別の変更でデータを移動する必要がある場合には、Block Storage サービスはデフォルトではデータを移動しません。種別変更の要求の一部として移行ポリシーを指定して移動を明示的に許可しない限り、操作は失敗します。詳細は、「[コマンドラインを使用したボリューム種別の変更](#)」を参照してください。

制約

- すべてのボリューム種別を変更することはできません。バックエンド間のボリュームの移動に関する詳細は、「[バックエンド間でのボリュームの移動](#)」を参照してください。
- 暗号化されないボリューム種別を暗号化ボリューム種別に変更することはできませんが、暗号化ボリューム種別を暗号化されないボリューム種別に変更することはできます。
- 管理者権限のないユーザーは、自分が所有するボリュームの種別しか変更できません。

2.3.16.1. Dashboard UI を使用したボリューム種別の変更

Dashboard UI を使用してボリューム種別を変更します。



重要

暗号化されていないボリュームを同じサイズの暗号化されたボリュームに種別変更する操作はサポートされません。暗号化したボリュームには、暗号化データを格納するための追加領域が必要なためです。暗号化されていないボリュームの暗号化に関する詳細は、「[暗号化されていないボリュームの暗号化](#)」を参照してください。

前提条件

- アンダークラウドの正常なインストール。詳細は、[Installing director on the undercloud](#) を参照してください。
- オーバークラウドの正常なデプロイメント。詳細は、[CLI ツールで基本的なオーバークラウドの作成](#) を参照してください。
- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、[Overcloud deployment output](#) を参照してください。

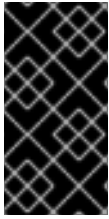
手順

1. Dashboard で **プロジェクト > コンピュート > ボリューム** を選択します。
2. 移行するボリュームの **アクション** のコラムで、**ボリューム種別の変更** を選択します。
3. **ボリューム種別の変更** ダイアログで、対象のボリューム種別を選択し、**種別** のリストから新しいバックエンドを定義します。
4. 別のバックエンドにボリュームを移行する場合は、**移行ポリシー** のリストから **要求時** を選択します。詳細は、「[バックエンド間でのボリュームの移動](#)」を参照してください。

5. **ボリューム種別の変更** をクリックして移行を開始します。

2.3.16.2. コマンドラインを使用したボリューム種別の変更

Dashboard UI の手順と同様に、コマンドラインからボリューム種別を変更することができます。



重要

暗号化されていないボリュームを同じサイズの暗号化されたボリュームに種別変更する操作はサポートされません。暗号化したボリュームには、暗号化データを格納するための追加領域が必要なためです。暗号化されていないボリュームの暗号化に関する詳細は、[暗号化されていないボリュームの暗号化](#) を参照してください。

前提条件

- アンダークラウドの正常なインストール。詳細は、[Installing director on the undercloud](#) を参照してください。
- オーバークラウドの正常なデプロイメント。詳細は、[CLI ツールで基本的なオーバークラウドの作成](#) を参照してください。

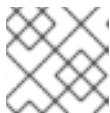
手順

1. 以下のコマンドを入力してボリューム種別を変更します。

```
$ cinder retype <volume id> <new volume type name>
```

2. 種別変更の操作で、あるバックエンドから別のバックエンドにボリュームを移動する必要がある場合は、Block Storage サービスには特定のフラグが必要です。

```
$ cinder retype --migration-policy on-demand <volume id> <new volume type name>
```



注記

種別変更の操作が進むと、ボリュームのステータスが **retyping** に変わります。

3. 次のコマンドを入力し、**volume_type** フィールドを確認して、再入力操作が成功したことを確認します。**volume_type** フィールドには、新しいボリュームタイプが表示されます。

```
$ cinder show <volume id>
```



注記

種別変更の操作を開始すると、ボリューム名が重複します。そのボリューム名で **cinder show** コマンドを実行すると、シンダークライアントは次のようなエラーを返します。**ERROR: Multiple volume matches found for '<volume name>'**このエラーを回避するには、ボリューム名の代わりにボリューム ID を使用します。

2.3.17. オーバークラウドノードでの LVM2 フィルターの有効化

特定の BlockStorage Service (cinder) バックエンドで LVM2 (Logical Volume Management) ボリューム

を使用する場合、Red Hat OpenStack Platform (RHOSP) ゲスト内で作成したボリュームが、**cinder-volume** または **nova-compute** をホストするオーバークラウドノードに表示される場合があります。この場合、ホスト上の LVM2 ツールは、OpenStack ゲストが作成する LVM2 ボリュームをスキャンします。これにより、コンピュータノードまたはコントローラーノードで次の問題が1つ以上発生する可能性があります。

- LVM がゲストからのボリュームグループを表示するようになる
- LVM が重複するボリュームグループ名を報告する
- LVM がストレージにアクセスしているため、ボリュームの切り離しが失敗する
- LVM の問題が原因でゲストがブートに失敗する
- ゲストマシン上の LVM は、実際に存在するディスクが見つからないため、部分的な状態にある
- LVM を持つデバイスで Block Storage サービス (cinder) のアクションが失敗する
- Block Storage サービス (cinder) のスナップショットが正しく削除されない
- ライブマイグレーション中のエラー: `/etc/multipath.conf` が存在しない

この誤ったスキャンを防ぎ、ゲスト LVM2 ボリュームをホストノードから分離するために、オーバークラウドのデプロイまたは更新時に **LVMFilterEnabled** heat パラメーターを使用してフィルターを有効にし、設定できます。このフィルターは、アクティブな LVM2 ボリュームをホストする物理デバイスのリストから計算されます。**LVMFilterAllowlist** および **LVMFilterDenylist** パラメーターを使用して、ブロックデバイスを明示的に許可および拒否することもできます。このフィルタリングは、グローバルに、特定のノードロールに、または特定のデバイスに適用できます。



注記

この機能は、本リリースでは **テクノロジープレビュー** として提供しているため、Red Hat では全面的にはサポートしていません。これは、テスト用途にのみご利用いただく機能です。実稼働環境にはデプロイしないでください。テクノロジープレビュー機能の詳細は、[対象範囲の詳細](#) を参照してください。

前提条件

- アンダークラウドの正常なインストール。詳しくは、[Installing the undercloud](#) を参照してください。

手順

1. アンダークラウドホストに **stack** ユーザーとしてログインします。
2. `source` コマンドでアンダークラウドの認証情報ファイルを読み込みます。

```
$ source ~/stackrc
```

3. 新しい環境ファイルを作成するか、既存の環境ファイルを変更します。この例では、新しいファイル **lvm2-filtering.yaml** を作成します。

```
$ touch ~/lvm2-filtering.yaml
```

4. 環境ファイルに以下のパラメーターを追加します。


```
parameter_defaults:
  LVMFilterEnabled: true
```

LVM2 フィルターの実装はさらにカスタマイズできます。たとえば、コンピュータノードでのみフィルタリングを有効にするには、次の設定を使用します。

```
parameter_defaults:
  ComputeParameters:
    LVMFilterEnabled: true
```

これらのパラメーターは、正規表現もサポートしています。コンピュータノードでのみフィルタリングを有効にし、`/dev/sd` で始まるすべてのデバイスを無視するには、次の設定を使用します。

```
parameter_defaults:
  ComputeParameters:
    LVMFilterEnabled: true
    LVMFilterDenylist:
      - /dev/sd.*
```

5. **openstack overcloud deploy** コマンドを実行し、LVM2 フィルタリング設定を含む環境ファイルと、オーバークラウドデプロイメントに関連するその他の環境ファイルを含めます。

```
$ openstack overcloud deploy --templates \
  <environment-files> \
  -e lvm2-filtering.yaml
```

2.4. ボリュームの高度な設定

以下の手順で、ボリュームの高度な管理方法について説明します。



重要

Block Storage サービス (cinder) およびファイバーチャネル (FC) バックエンドを使用するすべてのデプロイメントにおいて、すべてのコントローラーノードおよびコンピュータノードにホストバスアダプター (HBA) をインストールする必要があります。

2.4.1. ボリュームの移行

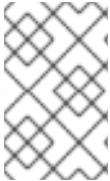
Block Storage サービス (cinder) を使用して、同一または異なるアベイラビリティゾーン (AZ) にあるバックエンド間で、ボリュームを移行することができます。これは、あるバックエンドから別のバックエンドにボリュームを移行する方法としては、まったく一般的ではありません。デフォルトのポリシーでは、管理者のみがボリュームを移行することができます。デフォルトのポリシーは変更しないでください。

高度にカスタマイズされたデプロイメントの場合や、ストレージシステムを廃止する必要がある状況では、管理者はボリュームを移行できます。どちらのユースケースでも、複数のストレージシステムが同じ **volume_backend_name** を共有しているか、未定義です。

制約

- ボリュームは複製できません。

- 移行先バックエンドは、ボリュームの現在のバックエンドとは異なる必要があります。
- 既存のボリューム種別は、新規バックエンドに対して有効である必要があります。つまり、以下の状況でなければなりません。
 - ボリュームタイプの追加仕様で **backend_volume_name** を定義することはできません。または、両方のブロックストレージバックエンドを同じ **backend_volume_name** で設定する必要があります。
 - どちらのバックエンドも、シンプロビジョニングのサポート、シックプロビジョニングのサポート、またはその他の機能設定など、ボリューム種別で設定した同じ機能をサポートする。



注記

ボリュームをあるバックエンドから別のバックエンドに移動するには、非常に多くの時間とリソースが必要になる場合があります。詳細は、「[バックエンド間でのボリュームの移動](#)」を参照してください。

2.4.1.1. バックエンド間の移行

Dashboard UI を使用して、バックエンド間でボリュームを移行します。

手順

1. Dashboard で **管理 > ボリューム** を選択します。
2. 移行するボリュームの **アクション** のコラムで、**ボリュームのマイグレーション** を選択します。
3. **ボリュームのマイグレーション** ダイアログで、**移行先ホスト** ドロップダウンリストからボリュームを移行する先のホストを選択します。



注記

ホストの移行でドライバーの最適化をスキップするには、**強制ホストコピー** のチェックボックスを選択します。

4. **マイグレーション** をクリックして移行を開始します。

2.4.1.2. コマンドラインを使用したバックエンド間の移行

- アンダークラウドの正常なインストール。詳細は、[Installing director on the undercloud](#) を参照してください。

手順

1. 以下のコマンドを入力して、宛先のバックエンドの名前を取得します。

```
$ cinder get-pools --detail
```

```
Property          | Value
```

```
...
```

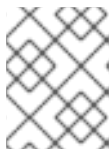
```

| name          | localdomain@lvmdriver-1#lvmdriver-1
| pool_name     | lvmdriver-1
...
| volume_backend_name | lvmdriver-1
...
Property      | Value
...
| name          | localdomain@lvmdriver-2#lvmdriver-1
| pool_name     | lvmdriver-1
...
| volume_backend_name | lvmdriver-1
...

```

バックエンド名は、**host @ volume_backend_name#pool** の形式を取ります。

出力例では、BlockStorage サービスで公開されている2つのLVMバックエンドがあります。**localdomain@lvmdriver-1#lvmdriver-1** と **localdomain@lvmdriver-2#lvmdriver-1** です。両方のバックエンドが同じ **volume_backend_name**、**lvmdriver-1** を共有していることに注意してください。



注記

LVMの使用は、例としてのみ挙げられています。LVMは、実稼働環境ではサポートされません。

2. 以下のコマンドを入力して、ボリュームをあるバックエンドから別のバックエンドに移行します。

```
$ cinder migrate <volume id or name> <new host>
```

2.4.1.3. ボリュームの移行の確認

ボリュームを作成すると、**migration_status** の値は **None** になります。移行を開始すると、ステータスが **migrating** に変わります。移行が完了すると、ステータスは **success** また **error** に変わります。

ブロックストレージサービスが移行要求を受け入れた後、cinder クライアントは、**Request to migrate volume <volume id> has been accepted.** のようなメッセージで応答します。ただし、移行が完了するまでには時間がかかります。管理者は、移行のステータスを確認できます。

手順

1. 次のコマンドを入力し、**migration_status** フィールドを確認します。

```
$ cinder show <volume id>
```



注記

通常のボリューム移行を開始すると、ボリューム名が重複します。そのボリューム名で **cinder show** コマンドを実行すると、シンダークライアントは次のようなエラーを返します。**ERROR: Multiple volume matches found for '<volume name>'**このエラーを回避するには、ボリューム名の代わりにボリューム ID を使用します。

移行が成功すると、**host** フィールドは **cinder migrate** コマンドで設定された **<new host>** 値と一致します。

2.4.2. 暗号化されていないボリュームの暗号化

暗号化されていないボリュームを暗号化するには、暗号化されていないボリュームをバックアップし、それを暗号化された新しいボリュームに復元するか、暗号化されていないボリュームから Image サービス (glance) のイメージを作成し、そのイメージから暗号化された新しいボリュームを作成する必要があります。

前提条件

- 暗号化する暗号化されていないボリューム

手順

1. **cinder-backup** サービスが利用可能な場合は、現在の暗号化されていないボリュームをバックアップします。

```
$ cinder backup-create <unencrypted_volume>
```

- **<unencrypted_volume>** を、暗号化されていないボリュームの名前または ID に置き換えます。

2. 暗号化された新しいボリュームを作成します。

```
$ cinder create <encrypted_volume_size> --volume-type <encrypted_volume_type>
```

- **<encrypted_volume_size>** を新しいボリュームのサイズ (GB) に置き換えます。暗号化メタデータに対応するために、この値は暗号化されていないボリュームのサイズよりも 1GB 大きくする必要があります。
- **<encrypted_volume_type>** を必要な暗号化タイプに置き換えます。

3. 暗号化されていないボリュームのバックアップを暗号化された新しいボリュームに復元します。

```
$ cinder backup-restore <backup> --volume <encrypted_volume>
```

- **<backup>** を、暗号化されていないボリュームバックアップの名前または ID に置き換えます。
- **<encrypted_volume>** を新しい暗号化ボリュームの ID に置き換えます。

cinder-backup サービスが利用できない場合は、**upload-to-image** コマンドを使用して、暗号化されていないボリュームのイメージを作成してから、そのイメージから新しい暗号化されたボリュームを作成します。

1. 暗号化されていないボリュームの Image サービスイメージを作成します。

```
$ cinder upload-to-image <unencrypted_volume> <new_image>
```

- **<unencrypted_volume>** を、暗号化されていないボリュームの名前または ID に置き換えます。
- **<new_image>** を新しいイメージの名前に置き換えます。

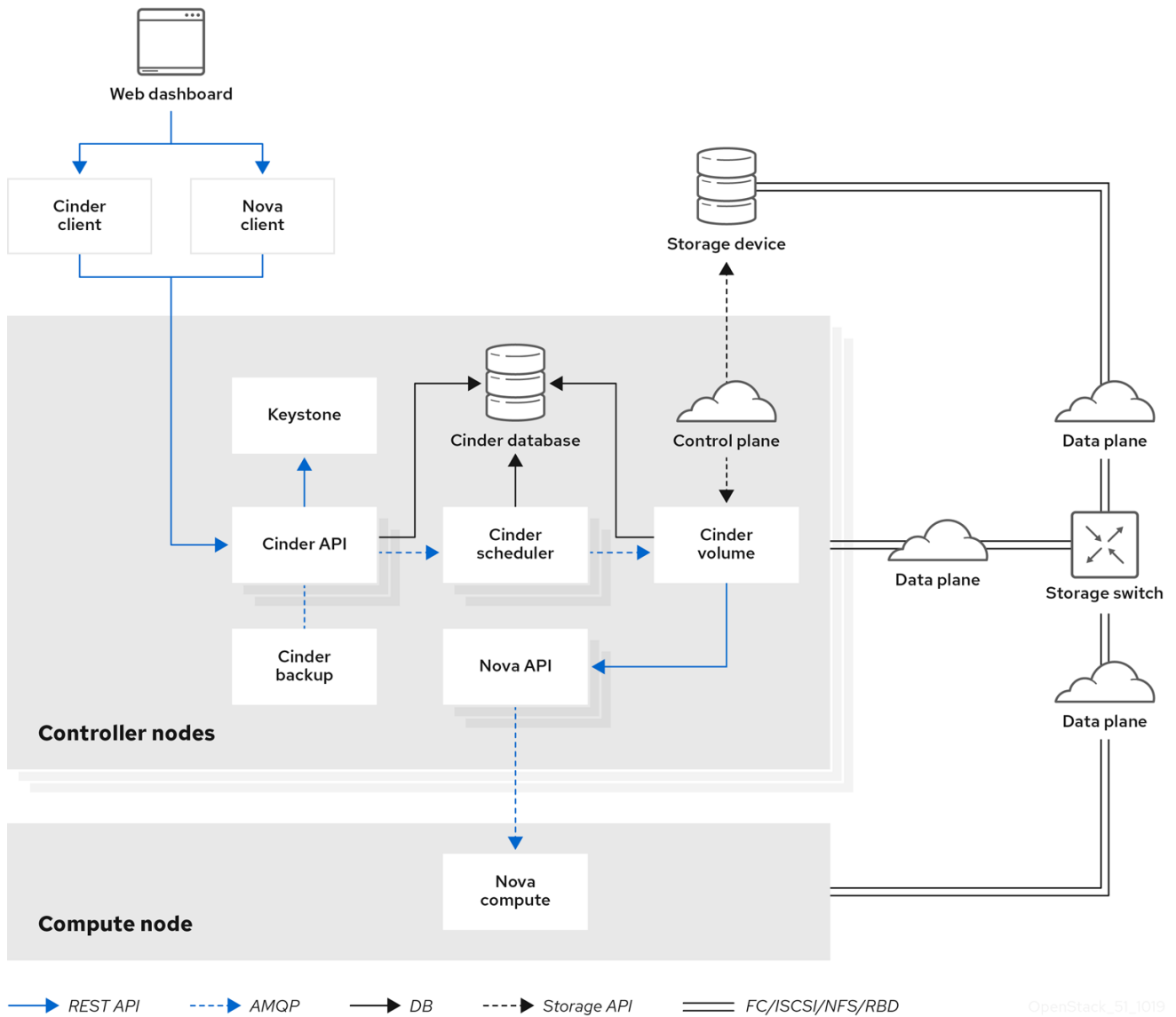
2. イメージから、そのイメージより 1GB 大きい新たなボリュームを作成します。

```
$ cinder volume create --size <size> --volume-type luks --image <new_image>  
<encrypted_volume_name>
```

- **<size>** を新しいボリュームのサイズに置き換えます。この値は、暗号化されていない古いボリュームのサイズよりも 1GB 大きくする必要があります。
- **<new_image>** を、暗号化されていないボリュームから作成したイメージの名前に置き換えます。
- **<encrypted_volume_name>** を新しい暗号化ボリュームの名前に置き換えます。

2.5. マルチパス設定

マルチパスを使用してサーバーノードおよびストレージアレイ間の複数の I/O パスを単一のデバイスに設定することで、冗長性が得られると共にパフォーマンスが向上します。マルチパスは、新規および既存のオーバークラウドデプロイメントに設定することができます。



2.5.1. 新規デプロイメントでのマルチパス設定

新規オーバークラウドデプロイメントでマルチパスを設定するには、以下の手順を実施します。

既存のオーバークラウドデプロイメントでマルチパスを設定する方法は、「[既存デプロイメントでのマルチパス設定](#)」を参照してください。

前提条件

オーバークラウドのコントローラーノードおよびコンピューターノードは、Red Hat Enterprise Linux Server のリポジトリにアクセスする必要があります。詳しくは、**Director Installation and Usage**の [Downloading the base cloud image](#) を参照してください。

手順

1. オーバークラウドを設定します。



注記

詳細は、**director** のインストールと使用方法の [CLI ツールを使用した基本的なオーバークラウドの設定](#) を参照してください。

- heat テンプレートを更新してマルチパスを有効にします。

```
parameter_defaults:
  NovaLibvirtVolumeUseMultipath: true
  NovaComputeOptVolumes:
    - /etc/multipath.conf:/etc/multipath.conf:ro
    - /etc/multipath/;/etc/multipath/:rw
  CinderVolumeOptVolumes:
    - /etc/multipath.conf:/etc/multipath.conf:ro
    - /etc/multipath/;/etc/multipath/:rw
```

- (オプション) Block Storage (cinder) を Image サービス (glance) のバックエンドとして使用する場合には、以下の手順も完了する必要があります。

- 次の **GlanceApiOptVolumes** 設定をヒートテンプレートに追加します。

```
parameter_defaults:
  GlanceApiOptVolumes:
    - /etc/multipath.conf:/etc/multipath.conf:ro
    - /etc/multipath/;/etc/multipath/:rw
```

- 次の方法で **ControllerExtraConfig** パラメーターを設定します。

```
parameter_defaults:
  ControllerExtraConfig:
    glance::config::api_config:
      default_backend/cinder_use_multipath:
        value: true
```

注記

default_backend と **GlanceBackendID** ヒートテンプレートのデフォルト値の両方が一致していることを確認してください。

- 設定したすべてのバックエンドについて、**use_multipath_for_image_xfer** を **true** に設定します。

```
parameter_defaults:
  ExtraConfig:
    cinder::config::cinder_config:
      <backend>/use_multipath_for_image_xfer:
        value: true
```

- オーバークラウドをデプロイします。

```
$ openstack overcloud deploy
```



注記

オーバークラウドパラメーターを使用してオーバークラウドを作成する方法は、**director** のインストールと使用方法の [CLI ツールを使用した基本的なオーバークラウドの設定](#) を参照してください。

6. コンテナを実行する前に、すべてのコントローラーノードおよびコンピューターノードにマルチパスをインストールします。

```
$ sudo dnf install -y device-mapper-multipath
```

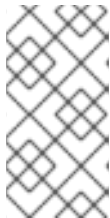


注記

director にはフックのセットが用意されており、初回のブートが完了してコア設定が開始する前に、特定ノードロールのカスタム設定を行うことができます。オーバークラウドのカスタム設定についての詳しい情報は、[オーバークラウドの高度なカスタマイズの事前設定: 特定のオーバークラウドロールのカスタマイズ](#)を参照してください。

7. すべてのコントローラーノードおよびコンピューターノードでマルチパスデーモンを設定します。

```
$ mpathconf --enable --with_multipathd y --user_friendly_names n --find_multipaths y
```



注記

このコード例により、ほとんどの環境で機能する基本的なマルチパス設定が作成されます。ただし、一部のストレージベンダーはハードウェア固有の最適化した設定を使用しているため、ベンダーに推奨事項を問い合わせてください。マルチパスについての詳細は、[Device Mapper Multipath の設定](#)を参照してください。

8. すべてのコントローラーノードおよびコンピューターノードで以下のコマンドを実行して、パーティションが作成されないようにします。

```
$ sed -i "s/^defaults {/defaults {\n\tskip_kpartx yes\n\ttrecheck_wwid yes/" /etc/multipath.conf
```



注記

skip_kpartx を **yes** に設定すると、コンピューターノード上の kpartx がデバイス上に自動的にパーティションを作成しなくなり、不要なデバイスマッパーエントリーを避けることができます。設定属性の詳細は、[Device Mapper Multipath の設定の DM Multipath 設定ファイルの編集](#)を参照してください。

9. すべてのコントローラーノードおよびコンピューターノードでマルチパスデーモンを起動します。

```
$ systemctl enable --now multipathd
```

2.5.2. 既存デプロイメントでのマルチパス設定

ワークロードがマルチパス機能を使用できるように、既存のデプロイメントでマルチパスを設定します。



注記

既存のデプロイメントでマルチパスを設定した後に作成する新しいワークロードは、デフォルトでマルチパスに対応しています。既存のワークロードがある場合は、これらのインスタンスでマルチパスを有効にするには、インスタンスを退避して復元する必要があります。

新規オーバークラウドデプロイメントでマルチパスを設定する方法は、「[新規デプロイメントでのマルチパス設定](#)」を参照してください。

前提条件

オーバークラウドのコントローラーノードおよびコンピューターノードは、Red Hat Enterprise Linux Server のリポジトリにアクセスできる必要があります。詳しくは、**Director Installation and Usage**の [Downloading the base cloud image](#) を参照してください。

手順

1. すべてのコントローラーノードおよびコンピューターノードにマルチパスがインストールされていることを確認します。

```
$ rpm -qa | grep device-mapper-multipath
device-mapper-multipath-0.4.9-127.el8.x86_64
device-mapper-multipath-libs-0.4.9-127.el8.x86_64
```

マルチパスがインストールされていない場合は、すべてのコントローラーノードおよびコンピューターノードにインストールします。

```
$ sudo dnf install -y device-mapper-multipath
```

2. すべてのコントローラーノードおよびコンピューターノードでマルチパスデーモンを設定します。

```
$ mpathconf --enable --with_multipathd y --user_friendly_names n --find_multipaths y
```



注記

このコード例により、ほとんどの環境で機能する基本的なマルチパス設定が作成されます。ただし、一部のストレージベンダーはハードウェア固有の最適化された設定を使用しているため、ベンダーに推奨事項を問い合わせてください。マルチパスについての詳細は、[Device Mapper Multipath の設定](#) を参照してください。

3. すべてのコントローラーノードおよびコンピューターノードで以下のコマンドを実行して、パーティションが作成されないようにします。

```
$ sed -i "s/^defaults {/defaults {\n\tskip_kpartx yes\n\ttrecheck_wwid yes/" /etc/multipath.conf
```



注記

skip_kpartx を **yes** に設定すると、コンピュータノード上の kpartx がデバイス上に自動的にパーティションを作成しなくなり、不要なデバイスマッパーエントリーを避けることができます。設定属性の詳細は、**Device Mapper Multipath の設定**の [DM Multipath 設定ファイルの編集](#) を参照してください。

4. すべてのコントローラーノードおよびコンピュータノードでマルチパスデーモンを起動します。

```
$ systemctl enable --now multipathd
```

5. heat テンプレートを更新してマルチパスを有効にします。

```
parameter_defaults:
  NovaLibvirtVolumeUseMultipath: true
  NovaComputeOptVolumes:
    - /etc/multipath.conf:/etc/multipath.conf:ro
    - /etc/multipath/;/etc/multipath/:rw
  CinderVolumeOptVolumes:
    - /etc/multipath.conf:/etc/multipath.conf:ro
    - /etc/multipath/;/etc/multipath/:rw
```

6. (オプション) Block Storage (cinder) を Image サービス (glance) のバックエンドとして使用する場合には、以下の手順も完了する必要があります。

- a. 次の **GlanceApiOptVolumes** 設定をヒートテンプレートに追加します。

```
parameter_defaults:
  GlanceApiOptVolumes:
    - /etc/multipath.conf:/etc/multipath.conf:ro
    - /etc/multipath/;/etc/multipath/:rw
```

- b. 次の方法で **ControllerExtraConfig** パラメーターを設定します。

```
parameter_defaults:
  ControllerExtraConfig:
    glance::config::api_config:
      default_backend/cinder_use_multipath:
        value: true
```

注記

default_backend と **GlanceBackendID** ヒートテンプレートのデフォルト値の両方が一致していることを確認してください。

7. 設定したすべてのバックエンドについて、**use_multipath_for_image_xfer** を **true** に設定します。

```
parameter_defaults:
  ExtraConfig:
    cinder::config::cinder_config:
      <backend>/use_multipath_for_image_xfer:
        value: true
```

8. 以下のコマンドを実行してオーバークラウドを更新します。

```
$ openstack overcloud deploy
```



注記

openstack overcloud deploy コマンドを実行してマルチパスをインストールおよび設定する場合、**--templates**、**--roles-file**、**-e** (すべての環境ファイル用)、および **--timeout** など、オーバークラウドのデプロイに使用した以前のロールファイルおよび環境ファイルをすべて渡す必要があります。以前のロールファイルおよび環境ファイルをすべて渡さないと、オーバークラウドのデプロイメントで問題が発生する可能性があります。オーバークラウドパラメーターの使用についての詳細は、**director** のインストールと使用方法の [CLI ツールを使用した基本的なオーバークラウドの設定](#) を参照してください。

2.5.3. マルチパス設定の確認

以下の手順で、新規または既存のオーバークラウドデプロイメントのマルチパス設定を確認する方法を説明します。

手順

1. 仮想マシンを作成します。
2. 暗号化されていないボリュームを仮想マシンに割り当てます。
3. インスタンスが含まれるコンピュートノードの名前を取得します。

```
$ nova show INSTANCE | grep OS-EXT-SRV-ATTR:host
```

INSTANCE をブートした仮想マシンの名前に置き換えます。

4. インスタンスの `virsh` 名を取得します。

```
$ nova show INSTANCE | grep instance_name
```

INSTANCE をブートした仮想マシンの名前に置き換えます。

5. コンピュートノードの IP アドレスを取得します。

```
$ . stackrc
$ nova list | grep compute_name
```

compute_name を **nova show INSTANCE** コマンドの出力に表示される名前に置き換えます。

6. 仮想マシンを実行するコンピュートノードに SSH 接続します。

```
$ ssh heat-admin@COMPUTE_NODE_IP
```

COMPUTE_NODE_IP をコンピュートノードの IP アドレスに置き換えます。

7. `virsh` を実行するコンテナにログインします。

```
$ podman exec -it nova_libvirt /bin/bash
```

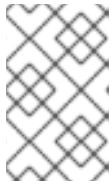
8. コンピュートノードインスタンスで以下のコマンドを入力し、cinder ボリュームホストの場所でマルチパスが使用されていることを確認します。

```
virsh domblklist VIRSH_INSTANCE_NAME | grep /dev/dm
```

VIRSH_INSTANCE_NAME を `nova show INSTANCE | grep instance_name` コマンドの出力に置き換えます。

インスタンスに `/dev/dm-` 以外の値が表示されている場合、接続は非マルチパスであるため、`nova shelve` および `nova unshelve` コマンドを使用して接続情報を更新する必要があります。

```
$ nova shelve <instance>
$ nova unshelve <instance>
```



注記

複数の種別のバックエンドがある場合には、すべてのバックエンド上のインスタンスおよびボリュームを検証する必要があります。これは、各バックエンドが返す接続情報が異なる可能性があるためです。

第3章 OBJECT STORAGE サービス

OpenStack Object Storage (**swift**) は、コンテナ内にそのオブジェクト (データ) を格納します。コンテナは、ファイルシステムにおけるディレクトリーと似ています。ただし、入れ子にはできません。コンテナは、あらゆるタイプの非構造化データを格納する簡単な方法をユーザーに提供します。たとえば、オブジェクトには写真、テキストファイル、イメージなどが含まれます。格納されるオブジェクトは圧縮されません。

3.1. オブジェクトストレージリング

オブジェクトストレージは、**リング** と呼ばれるデータ構造を使用して、パーティション領域をクラスター内に分散します。このパーティション領域は、Object Storage サービスのデータ永続性エンジン (data durability engine) の中核となります。これにより、Object Storage サービスが迅速かつ簡単にクラスター内の各パーティションを同期できるようになります。

リングには、オブジェクトストレージのパーティションについての情報、およびパーティションがさまざまなノードおよびディスクにどのように分散されるかについての情報が含まれます。Object Storage のコンポーネントがデータと対話する場合、リング内をローカルで素早く検索して、各オブジェクトが保管されているはずのパーティションを特定します。

Object Storage サービスには3つのリングがあり、異なるデータ種別が格納されます。1つはアカウント情報用、もう1つは (アカウント内のオブジェクトを整理するのに役立つ) コンテナ用、残りはオブジェクトのレプリカ用です。

3.1.1. リングのリバランス

ストレージ容量、ノード、またはディスクを追加または削除して Object Storage 環境を変更する場合、リングのリバランスが必要になります。**openstack overcloud deploy** を実行することでリングのリバランスが可能ですが、この方法ではオーバークラウド全体が再デプロイされます。この作業は非常に複雑になる可能性があります (特に、オーバークラウドが大規模な場合)。これに代わる方法として、アンダークラウドで以下のコマンドを実行し、リングのリバランスを行います。

```
source ~/stackrc
ansible-playbook -i /usr/bin/tripleo-ansible-inventory
/usr/share/openstack-tripleo-common/playbooks/swift_ring_rebalance.yaml
```

3.1.2. クラスターの健全性の確認

長期のデータ可用性、耐久性、および永続性を確保するために、Object Storage サービスではバックグラウンドで多くのプロセスが実行されます。以下に例を示します。

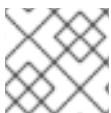
- **auditors** は定期的にデータベースおよびオブジェクトファイルを再読み取りし、チェックサムを使用してそれらと比較して、サイレントビットロットがないことを確認します。チェックサムと一致しなくなったデータベースまたはオブジェクトファイルは隔離され、そのノードでは読み取ることができなくなります。この場合、**replicators** は他のレプリカのいずれかをコピーして、再びローカルコピーが利用できる状態にします。
- ディスクやノードを置き換えた場合、またはオブジェクトが隔離された場合、オブジェクトおよびファイルが消失することがあります。この場合、**replicators** は欠けているオブジェクトまたはデータベースファイルを他のノードのいずれかにコピーします。

Object Storage サービスには **swift-recon** と呼ばれるツールが含まれています。このツールは、すべてのノードからデータを収集してクラスターの全体的な健全性を確認します。

swift-recon を使用するには、コントローラーノードのいずれかにログインして以下のコマンドを実行します。

```
[root@overcloud-controller-2 ~]# sudo podman exec -it -u swift swift_object_server /usr/bin/swift-recon -arqIT --md5

=====>
Starting reconnaissance on 3 hosts (object)
=====[2018-12-14 14:55:47] Checking async pendings
[async_pending] - No hosts returned valid data.
=====[2018-12-14 14:55:47] Checking on replication
[replication_failure] low: 0, high: 0, avg: 0.0, total: 0, Failed: 0.0%, no_result: 0, reported: 3
[replication_success] low: 0, high: 0, avg: 0.0, total: 0, Failed: 0.0%, no_result: 0, reported: 3
[replication_time] low: 0, high: 0, avg: 0.0, total: 0, Failed: 0.0%, no_result: 0, reported: 3
[replication_attempted] low: 0, high: 0, avg: 0.0, total: 0, Failed: 0.0%, no_result: 0, reported: 3
Oldest completion was 2018-12-14 14:55:39 (7 seconds ago) by 172.16.3.186:6000.
Most recent completion was 2018-12-14 14:55:42 (4 seconds ago) by 172.16.3.174:6000.
=====[2018-12-14 14:55:47] Checking load averages
[5m_load_avg] low: 1, high: 2, avg: 2.1, total: 6, Failed: 0.0%, no_result: 0, reported: 3
[15m_load_avg] low: 2, high: 2, avg: 2.6, total: 7, Failed: 0.0%, no_result: 0, reported: 3
[1m_load_avg] low: 0, high: 0, avg: 0.8, total: 2, Failed: 0.0%, no_result: 0, reported: 3
=====[2018-12-14 14:55:47] Checking ring md5sums
3/3 hosts matched, 0 error[s] while checking hosts.
=====[2018-12-14 14:55:47] Checking swift.conf md5sum
3/3 hosts matched, 0 error[s] while checking hosts.
=====[2018-12-14 14:55:47] Checking quarantine
[quarantined_objects] low: 0, high: 0, avg: 0.0, total: 0, Failed: 0.0%, no_result: 0, reported: 3
[quarantined_accounts] low: 0, high: 0, avg: 0.0, total: 0, Failed: 0.0%, no_result: 0, reported: 3
[quarantined_containers] low: 0, high: 0, avg: 0.0, total: 0, Failed: 0.0%, no_result: 0, reported: 3
=====[2018-12-14 14:55:47] Checking time-sync
3/3 hosts matched, 0 error[s] while checking hosts.
=====
```



注記

また、**--all** オプションを使用すると、追加の出力が返されます。

このコマンドは、リング上のすべてのサーバーに対して、以下のデータのクエリーを実行します。

- Async pendings: クラスターの負荷が非常に高くプロセスがデータベースファイルを十分な速度で更新できない場合、一部の更新は非同期で行われます。これらの数は徐々に減少するはずで
 - Replication metrics: レプリケーションのタイムスタンプに注意します。完全なレプリケーションパスは頻繁に行われ、エラーはほとんど発生しないはずで
- 古いエントリー (例: 6 カ月前のタイムスタンプを持つエントリー) ば、そのノードでのレプリケーションが過去 6 カ月間完了していないことを意味します。

- Ring md5sums: これにより、すべてのノードですべてのリングファイルの一貫性が確保されます。
- **swift.conf** md5sums: これにより、すべてのノードですべてのリングファイルの一貫性が確保されます。
- Quarantined files: すべてのノードについて、隔離されたファイルがない (あるいは、ほとんどない) はずです。
- Time-sync: すべてのノードが同期されている必要があります。

3.1.3. リングパーティションのべき乗の増加

リソース (アカウント、コンテナ、またはオブジェクト) がマッピングされるパーティションは、リングパーティションのべき乗により決定されます。パーティションは、リソースがバックエンドのファイルシステムに保管されるパスに含まれます。したがって、パーティションのべき乗を変更すると、リソースをバックエンドファイルシステムの新しいパスに再配置する必要があります。

稼働率の高いクラスターでは、再配置のプロセスに時間がかかります。ダウンタイムを回避するには、クラスターが稼働している間にリソースを再配置します。データへの一時的なアクセス不能やプロセス (レプリケーションや監査) のパフォーマンス低下を起こさずに、この操作を行う必要があります。リングパーティションのべき乗を増やす場合には、Red Hat サポートにお問い合わせください。

3.1.4. カスタムリングの作成

技術が進歩し、ストレージ容量への要求が高まる今日、カスタムリングを作成することが、既存の Object Storage クラスターを更新する手段となっています。

新規ノードをクラスターに追加する場合、それらの特性が元のノードとは異なる可能性があります。カスタムの調整を行わないと、新規ノードの大容量が十分に活用されない可能性があります。あるいは、リングで重みが変わると、データの分散が不均等になり、安全性が低下します。

自動化が将来の技術トレンドと整合しなくなることも考えられます。たとえば、現在使用されている旧式の Object Storage クラスターの中には、SSD が利用可能になる前に作られたものもあります。

リングビルダーは、クラスターの規模拡大および技術の進化に合わせてオブジェクトストレージを管理するのに役立ちます。カスタムリングの作成については、Red Hat サポートにお問い合わせください。

3.2. OBJECT STORAGE サービスの管理

以下の手順で、Object Storage サービスをカスタマイズする方法について説明します。

3.2.1. fast-post の設定

デフォルトでは、オブジェクトメタデータの一部にでも変更があると、Object Storage サービスは必ずオブジェクト全体をコピーします。**fast-post** 機能を使用することでこれを回避できます。fast-post 機能は、複数の大きなオブジェクトのコンテンツ種別を変更する際の時間を短縮します。

fast-post 機能を有効にするには、以下の手順により Object Storage プロキシサービスの **object_post_as_copy** オプションを無効にします。

1. **swift_params.yaml** を編集します。

```
cat > swift_params.yaml << EOF
parameter_defaults:
```

```
ExtraConfig:
  swift::proxy::copy::object_post_as_copy: False
EOF
```

2. オーバークラウドをデプロイまたは更新する際に、パラメーターファイルを指定します。

```
openstack overcloud deploy [... previous args ...] -e swift_params.yaml
```

3.2.2. 保存データ暗号化の有効化

デフォルトでは、オブジェクトストレージにアップロードされるオブジェクトは暗号化されずに保管されます。したがって、ファイルシステムからオブジェクトに直接アクセスすることが可能です。このため、ディスクを破棄する前に適切に消去しなかった場合には、セキュリティリスクとなってしまう。

OpenStack Key Manager (barbican) を使用して、保存されている swift オブジェクトを暗号化することができます。詳しい情報は、Manage Secrets with OpenStack Key Manager の [Encrypt at-rest swift objects](#) を参照してください。

3.2.3. スタンドアロンの Object Storage クラスターのデプロイ

コンポーザブルロールの概念を採用して、最小限の追加のサービス (例: Keystone、HAProxy) を実装したスタンドアロンの Object Storage (openstack-swift) クラスターをデプロイすることができます。Advanced Overcloud Customization の [Creating a roles_data File](#) セクションに、ロールについての情報が記載されています。

3.2.3.1. roles_data.yaml ファイルの作成

1. `/usr/share/openstack-tripleo-heat-templates` から `roles_data.yaml` をコピーします。
2. 新規ファイルを編集します。
3. 不要な Controller ロールを削除します (例: Aodh*、Ceilometer*、Ceph*、Cinder*、Glance*、Heat*、Ironic*、Manila*、Mistral*、Nova*、Octavia*、Swift*)。
4. `roles_data.yaml` 内で ObjectStorage を見つけます。
5. このロールを、同じファイル内の新しいロールにコピーして、**ObjectProxy** という名前を付けます。
6. このロールの **SwiftStorage** は **SwiftProxy** に置き換えます。

以下の `roles_data.yaml` ファイルの例には、サンプルのロールを記載しています。

```
- name: Controller
  description: |
    Controller role that has all the controller services loaded and handles
    Database, Messaging and Network functions.
  CountDefault: 1
  tags:
  - primary
  - controller
  networks:
  - External
  - InternalApi
```



```
- Storage
- StorageMgmt
- Tenant
HostnameFormatDefault: '%stackname%-controller-%index%'
ServicesDefault:
- OS::TripleO::Services::AuditD
- OS::TripleO::Services::CACerts
- OS::TripleO::Services::CertmongerUser
- OS::TripleO::Services::Clustercheck
- OS::TripleO::Services::Docker
- OS::TripleO::Services::Ec2Api
- OS::TripleO::Services::Etcd
- OS::TripleO::Services::HAproxy
- OS::TripleO::Services::Keepalived
- OS::TripleO::Services::Kernel
- OS::TripleO::Services::Keystone
- OS::TripleO::Services::Memcached
- OS::TripleO::Services::MySQL
- OS::TripleO::Services::MySQLClient
- OS::TripleO::Services::Ntp
- OS::TripleO::Services::Pacemaker
- OS::TripleO::Services::RabbitMQ
- OS::TripleO::Services::Securetty
- OS::TripleO::Services::Snmp
- OS::TripleO::Services::Sshd
- OS::TripleO::Services::Timezone
- OS::TripleO::Services::TripleoFirewall
- OS::TripleO::Services::TripleoPackages
- OS::TripleO::Services::Vpp

- name: ObjectStorage
  CountDefault: 1
  description: |
Swift Object Storage node role
  networks:
- InternalApi
- Storage
- StorageMgmt
  disable_upgrade_deployment: True
  ServicesDefault:
- OS::TripleO::Services::AuditD
- OS::TripleO::Services::CACerts
- OS::TripleO::Services::CertmongerUser
- OS::TripleO::Services::Collectd
- OS::TripleO::Services::Docker
- OS::TripleO::Services::FluentdClient
- OS::TripleO::Services::Kernel
- OS::TripleO::Services::MySQLClient
- OS::TripleO::Services::Ntp
- OS::TripleO::Services::Securetty
- OS::TripleO::Services::SensuClient
- OS::TripleO::Services::Snmp
- OS::TripleO::Services::Sshd
- OS::TripleO::Services::SwiftRingBuilder
- OS::TripleO::Services::SwiftStorage
- OS::TripleO::Services::Timezone
```

```

- OS::TripleO::Services::TripleoFirewall
- OS::TripleO::Services::TripleoPackages

- name: ObjectProxy
  CountDefault: 1
  description: |
    Swift Object proxy node role
  networks:
- InternalApi
- Storage
- StorageMgmt
  disable_upgrade_deployment: True
  ServicesDefault:
- OS::TripleO::Services::AuditD
- OS::TripleO::Services::CACerts
- OS::TripleO::Services::CertmongerUser
- OS::TripleO::Services::Collectd
- OS::TripleO::Services::Docker
- OS::TripleO::Services::FluentdClient
- OS::TripleO::Services::Kernel
- OS::TripleO::Services::MySQLClient
- OS::TripleO::Services::Ntp
- OS::TripleO::Services::Securetty
- OS::TripleO::Services::SensuClient
- OS::TripleO::Services::Snmp
- OS::TripleO::Services::Sshd
- OS::TripleO::Services::SwiftRingBuilder
- OS::TripleO::Services::SwiftProxy
- OS::TripleO::Services::Timezone
- OS::TripleO::Services::TripleoFirewall
- OS::TripleO::Services::TripleoPackages

```

3.2.3.2. 新規ロールのデプロイ

通常の **openstack deploy** コマンドで、新規ロールを指定して、オーバークラウドをデプロイします。

```
openstack overcloud deploy --templates -r roles_data.yaml -e [...]
```

3.2.4. 外部 SAN ディスクの使用

デフォルトでは、Red Hat OpenStack Platform director が Object Storage サービス (swift) をデプロイする際に、独立したローカルディスクを使用するように Object Storage が設定、最適化されます。この設定により、負荷がすべてのディスクに分散されるようになります。その結果、ノードに障害が発生した場合やその他のシステム異常時にパフォーマンスへの影響を最小限に抑えることができます。

パフォーマンスに影響を及ぼす類似のイベント発生時に、1つの SAN を使用する環境では、すべての LUN でパフォーマンスが低下する可能性があります。Object Storage サービスは、SAN ディスクを使用する環境で生じるパフォーマンスの問題を軽減することができません。

したがって、Red Hat では、パフォーマンスおよびディスク容量に対する要求を満たすために、Object Storage 用に SAN ディスクの代わりに追加のローカルディスクを使用することを強く推奨します。詳しくは、[Deployment Recommendations for Specific Red Hat OpenStack Platform Servicesの Object Storage](#) を参照してください。

Object Storage 用に外部 SAN を使用する場合は、ケースごとに評価する必要があります。詳細は、Red Hat のサポートにお問い合わせください。

重要

Object Storage 用に外部 SAN を使用する場合は、以下の条件に注意してください。

- デフォルトでは、Object Storage サービスは Telemetry データおよび Image サービス (glance) のイメージを保管します。glance のイメージはより多くのディスク容量を必要としますが、パフォーマンスの観点からは、glance のイメージを保管することの影響は、Telemetry データを保管することの影響よりは軽微です。Telemetry データの保管と処理には、より高いパフォーマンスが必要です。Object Storage 用に外部 SAN を使用した結果パフォーマンスに関する問題が生じた場合、Red Hat はこの問題に対するサポートを提供しません。
- Red Hat は、コアの Object Storage サービスオフリングの外部で生じる問題に対するサポートを提供しません。高可用性およびパフォーマンスに関するサポートは、ストレージベンダーにお問い合わせください。
- Red Hat は、Object Storage サービスと SAN ソリューションの組み合わせをテストしません。サードパーティー製品の互換性、ガイダンス、およびサポートに関する詳細は、ストレージベンダーにお問い合わせください。
- Red Hat では、実際のデプロイメントでパフォーマンスの要求を評価してテストすることを推奨します。お使いの SAN デプロイメントがテストおよびサポートされ、パフォーマンス要求を満たしていることを確認するには、ストレージベンダーにお問い合わせください。

3.2.4.1. SAN ディスクのデプロイメント設定

Object Storage のストレージ用に 2 つのデバイス (`/dev/mapper/vdb` および `/dev/mapper/vdc`) を使用方法の例を、以下のテンプレートに示します。

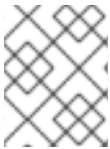
```
parameter_defaults:
  SwiftMountCheck: true
  SwiftUseLocalDir: false
  SwiftRawDisks: {"vdb": {"base_dir": "/dev/mapper/"}, "vdc": {"base_dir": "/dev/mapper/"}}
```

3.3. RED HAT ENTERPRISE LINUX 用ストレージノードのインストールおよび設定

外部ストレージノードで Red Hat OpenStack Platform Object Storage サービス (swift) を使用するには、アカウント、コンテナ、およびオブジェクトサービスのプロセスを処理するストレージノードをインストールして設定する必要があります。この設定では、2 つのストレージノードを参照します。それぞれのノードには、空のローカルブロックストレージデバイスが 2 つ含まれます。

注記

Object Storage サービスの内部ネットワークは認証されません。セキュリティ上の理由から、Red Hat では、ストレージノードを専用のネットワークまたは VLAN 上に維持することを推奨します。



注記

手順ではデバイス名に **/dev/sdb** および **/dev/sdc** を使用していますが、実際の環境内の特定ノードの値に置き換えることができます。

3.3.1. ストレージデバイスの準備

ストレージノードに Object Storage サービスをインストールして設定する前に、ストレージデバイスの準備を行う必要があります。



注記

各ストレージノードでこれらのすべての手順を実施します。

手順

1. 補助ユーティリティーパッケージをインストールします。

```
# yum install xfsprogs rsync
```

2. **/dev/sdb** および **/dev/sdc** デバイスを XFS としてフォーマットします。

```
# mkfs.xfs /dev/sdb
# mkfs.xfs /dev/sdc
```

3. マウントポイントのディレクトリ構造を作成します。

```
# mkdir -p /srv/node/sdb
# mkdir -p /srv/node/sdc
```

4. **/etc/fstab** ファイルを編集し、以下のデータを追加します。

```
/dev/sdb /srv/node/sdb xfs defaults 0 2
/dev/sdc /srv/node/sdc xfs defaults 0 2
```

5. デバイスをマウントします。

```
# mount /srv/node/sdb
# mount /srv/node/sdc
```

6. **/etc/rsyncd.conf** ファイルを作成または編集して、以下のデータを追加します。

```
uid = swift
gid = swift
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid
address = MANAGEMENT_INTERFACE_IP_ADDRESS
```

```
[account]
max connections = 2
path = /srv/node/
read only = False
lock file = /var/lock/account.lock
```

```
[container]
max connections = 2
path = /srv/node/
read only = False
lock file = /var/lock/container.lock
```

```
[object]
max connections = 2
path = /srv/node/
read only = False
lock file = /var/lock/object.lock
```

MANAGEMENT_INTERFACE_IP_ADDRESS は、ストレージノード上の管理ネットワークの IP アドレスに置き換えます。

7. **rsyncd** サービスを開始し、さらにシステムのブート時に開始するように設定します。

```
# systemctl enable rsyncd.service
# systemctl start rsyncd.service
```

3.3.2. コンポーネントの設定

アカウント、コンテナ、およびオブジェクトストレージサーバーを設定します。

手順

1. パッケージをインストールします。

```
# yum install openstack-swift-account openstack-swift-container \
openstack-swift-object
```

2. **/etc/swift/account-server.conf** ファイルを編集し、以下の操作を完了します。

- a. **[DEFAULT]** セクションで、バインド IP アドレス、バインドポート、ユーザー、設定ディレクトリ、およびマウントポイントディレクトリを設定します。

```
[DEFAULT]
...
bind_ip = MANAGEMENT_INTERFACE_IP_ADDRESS
bind_port = 6202
user = swift
swift_dir = /etc/swift
devices = /srv/node
mount_check = True
```

MANAGEMENT_INTERFACE_IP_ADDRESS は、ストレージノード上の管理ネットワークの IP アドレスに置き換えます。

- b. **[pipeline:main]** セクションで、**healthcheck** および **recon** モジュールを有効にします。

```
[pipeline:main]
pipeline = healthcheck recon account-server
```

- c. **[filter:recon]** セクションで、recon キャッシュディレクトリーを設定します。

```
[filter:recon]
use = egg:swift#recon
...
recon_cache_path = /var/cache/swift
```

3. アカウントサービスのデフォルトファイアウォールポートを開放します。

```
# firewall-cmd --permanent --add-port=6202/tcp
```

4. **/etc/swift/container-server.conf** ファイルを編集し、以下の操作を完了します。

- a. **[DEFAULT]** セクションで、バインド IP アドレス、バインドポート、ユーザー、設定ディレクトリー、およびマウントポイントディレクトリーを設定します。

```
[DEFAULT]
...
bind_ip = MANAGEMENT_INTERFACE_IP_ADDRESS
bind_port = 6201
user = swift
swift_dir = /etc/swift
devices = /srv/node
mount_check = True
```

MANAGEMENT_INTERFACE_IP_ADDRESS は、ストレージノード上の管理ネットワークの IP アドレスに置き換えます。

- b. **[pipeline:main]** セクションで、**healthcheck** および **recon** モジュールを有効にします。

```
[pipeline:main]
pipeline = healthcheck recon container-server
```

- c. **[filter:recon]** セクションで、recon キャッシュディレクトリーを設定します。

```
[filter:recon]
use = egg:swift#recon
...
recon_cache_path = /var/cache/swift
```

5. コンテナサービスのデフォルトファイアウォールポートを開放します。

```
# firewall-cmd --permanent --add-port=6201/tcp
```

6. **/etc/swift/object-server.conf** ファイルを編集し、以下の操作を完了します。

- a. **[DEFAULT]** セクションで、バインド IP アドレス、バインドポート、ユーザー、設定ディレクトリー、およびマウントポイントディレクトリーを設定します。

```
[DEFAULT]
...
bind_ip = MANAGEMENT_INTERFACE_IP_ADDRESS
bind_port = 6200
user = swift
```

```
swift_dir = /etc/swift
devices = /srv/node
mount_check = True
```

MANAGEMENT_INTERFACE_IP_ADDRESS は、ストレージノード上の管理ネットワークの IP アドレスに置き換えます。

- b. **[pipeline:main]** セクションで、**healthcheck** および **recon** モジュールを有効にします。

```
[pipeline:main]
pipeline = healthcheck recon object-server
```

- c. **[filter:recon]** セクションで、**recon_cache_path** および **recon_lock_path** ディレクトリーを設定します。

```
[filter:recon]
use = egg:swift#recon
...
recon_cache_path = /var/cache/swift
recon_lock_path = /var/lock
```

7. オブジェクトサービスのデフォルトファイアウォールポートを開放します。

```
# firewall-cmd --permanent --add-port=6200/tcp
```

8. マウントポイントディレクトリー構造の所有権を正しく設定します。

```
# chown -R swift:swift /srv/node
```

9. **recon** ディレクトリーを作成し、その所有権を適切に設定します。

```
# mkdir -p /var/cache/swift
# chown -R root:swift /var/cache/swift
# chmod -R 775 /var/cache/swift
```

3.4. 基本的なコンテナ管理

擬似フォルダーは、オブジェクトを格納することができる論理デバイスで、入れ子が可能となっているので、整理がしやすくなります。たとえば、イメージを保管する **Images** フォルダーや、ビデオを保管する **Media** フォルダーなどを作成することができます。

各プロジェクトに1つまたは複数のコンテナを作成することができます。また、各コンテナには、1つまたは複数のオブジェクトまたは擬似フォルダーを作成することができます。

3.4.1. コンテナの作成

1. Dashboard で **プロジェクト > オブジェクトストア > コンテナ** を選択します。
2. **コンテナの作成** をクリックします。
3. **コンテナ名** を指定して、**コンテナアクセス** フィールドで以下のいずれかのオプションを選択します。

型	説明
プライベート	アクセスを現在のプロジェクトのユーザーに制限します。
パブリック	パブリックの URL を使用して API アクセスを全員に許可します。ただし、Dashboard では、プロジェクトユーザーには、他のプロジェクトのパブリックコンテナおよびデータは表示されません。

4. コンテナの作成 をクリックします。

新しいコンテナはデフォルトのストレージポリシーを使用します。複数のストレージポリシーが定義されている場合には (たとえば、デフォルトと Erasure Coding を有効にするポリシーなど)、コマンドラインからデフォルト以外のストレージポリシーを使用するようにコンテナを設定することができます。そのためには、以下のコマンドを実行します。

```
# swift post -H "X-Storage-Policy:POLICY" CONTAINERNAME
```

各オプションについての説明は以下のとおりです。

- **POLICY** は、コンテナで使用するポリシーの名前またはエイリアスに置き換えます。
- **CONTAINERNAME** は、コンテナの名前に置き換えてください。

3.4.2. コンテナ用の擬似フォルダーの作成

1. Dashboard で **プロジェクト > オブジェクトストア > コンテナ** を選択します。
2. 擬似フォルダーを追加するコンテナの名前をクリックします。
3. **擬似フォルダーの作成** をクリックします。
4. **擬似フォルダー名** フィールドに名前を指定し、**作成** をクリックします。

3.4.3. コンテナの削除

1. Dashboard で **プロジェクト > オブジェクトストア > コンテナ** を選択します。
2. **コンテナ** のセクションのリストを参照して全オブジェクトが削除済みであることを確認します (「[オブジェクトの削除](#)」を参照)。
3. 対象のコンテナの矢印メニューで **コンテナの削除** を選択します。
4. **コンテナの削除** をクリックして、コンテナを削除する操作を確定します。

3.4.4. オブジェクトのアップロード

実際のファイルをアップロードしない場合でも、オブジェクトは (プレースホルダーとして) 作成され、後でファイルをアップロードする際に使用することができます。

1. Dashboard で **プロジェクト > オブジェクトストア > コンテナ** を選択します。

2. アップロードしたオブジェクトの配置先となるコンテナの名前をクリックします。そのコンテナに擬似フォルダーがすでに存在している場合には、擬似フォルダーの名前をクリックすることもできます。
3. ファイルをブラウズして **オブジェクトのアップロード** をクリックします。
4. **オブジェクト名** フィールドに名前を指定します。
 - 擬似フォルダーはスラッシュ (/) の記号を使用して指定することができます (例: Images/myImage.jpg)。指定したフォルダーがまだ存在していない場合には、オブジェクトのアップロード時に作成されます。
 - 名前がその場所で一意ではない場合 (オブジェクトがすでに存在している場合)、そのオブジェクトのコンテンツは上書きされます。
5. **オブジェクトのアップロード** をクリックします。

3.4.5. オブジェクトのコピー

1. Dashboard で **プロジェクト > オブジェクトストア > コンテナ** を選択します。
2. オブジェクトのコンテナまたはフォルダーの名前をクリックします (オブジェクトを表示します)。
3. **オブジェクトのアップロード** をクリックします。
4. コピーするファイルを参照し、矢印メニューで **コピー** を選択します。
5. 以下の項目を設定します。

フィールド	説明
宛先コンテナ	新規プロジェクトの宛先コンテナ
パス	宛先コンテナの擬似フォルダー。フォルダーが存在しない場合は、作成されます。
宛先オブジェクト名	新規オブジェクト名。その場所で一意ではない名前を使用した場合 (オブジェクトがすでに存在している場合)、そのオブジェクトの以前のコンテンツは上書きされます。

6. **オブジェクトのコピー** をクリックします。

3.4.6. オブジェクトの削除

1. Dashboard で **プロジェクト > オブジェクトストア > コンテナ** を選択します。
2. リストを参照して対象のオブジェクトを特定し、矢印メニューで **オブジェクトの削除** を選択します。
3. **オブジェクトの削除** をクリックして、オブジェクトを削除する操作を確定します。

第4章 SHARED FILE SYSTEMS サービス

Shared File Systems サービス (manila) により、複数のコンピュートインスタンス、ベアメタルノード、またはコンテナで消費可能な共有ファイルシステムをプロビジョニングすることができます。クラウド管理者は共有種別を作成してファイル共有サービスの準備を行い、エンドユーザーがファイル共有を作成および管理できるようにします。

前提条件

- Shared File Systems サービスを使用するために、エンドユーザーには少なくとも1つの共有種別が必要です。
- **driver_handles_share_servers=False** と設定したバックエンドの場合には、クラウド管理者は共有ファイルシステムバックエンドで動的にネットワークを設定するのではなく、必要なネットワークを事前に設定します。
- NFS バックエンドに CephFS を使用する場合、クラウド管理者は分離ネットワーク、環境引数、およびカスタム **network_data** ファイルを使用して Red Hat OpenStack Platform (RHOSP) director をデプロイし、NFS エクスポート用の分離 StorageNFS ネットワークを作成します。デプロイ後オーバークラウドを使用する前に、管理者は対応する Networking サービス (neutron) StorageNFS 共有プロバイダーネットワークを作成して、データセンターの分離 StorageNFS ネットワークにマッピングします。
- コンピュートインスタンスをこの共有プロバイダーネットワークに接続するためには、ユーザーは新たな neutron ポートを追加する必要があります。

Shared File Systems サービスについて理解および使用するには、以下の概念および手順を使用します。

- バックエンドを選択するには、[「Shared File Systems サービス \(manila\) バックエンド」](#) を参照してください。
- ネットワークの選択について理解するには、[「共有ファイルシステムのネットワーク」](#) を参照してください。
- ファイル共有を作成して管理するには、以下のセクションを参照してください。
 - [「共有種別の作成」](#)
 - [「共有種別の検出」](#)
 - [「ファイル共有の作成」](#)
 - [「ファイル共有とエクスポート情報のリスト表示」](#)
- ネットワーク接続を管理するには、以下のセクションを参照してください。
 - [「ファイル共有へのネットワーク接続の確保」](#)
 - [「ファイル共有にアクセスするための共有ネットワークへの接続」](#)
- ファイル共有へのアクセスを管理するには、以下のセクションを参照してください。
 - [「ファイル共有へのアクセスの付与」](#)
 - [「ファイル共有へのアクセスの取り消し」](#)
- ファイル共有を管理するには、以下のセクションを参照してください。

- [「ファイル共有のエクスポート場所のリスト表示」](#)
- [「ファイル共有のマウント」](#)
- [「ファイル共有の削除」](#)
- クォータを管理するには、以下のセクションを参照してください。
 - [「クォータのリスト表示」](#)
- 問題のトラブルシューティングを行うには、以下のセクションを参照してください。
 - [「ファイル共有作成またはファイル共有グループ作成の失敗の修正」](#)
 - [「ファイル共有のマウント失敗のデバッグ」](#)

4.1. SHARED FILE SYSTEMS サービス (MANILA) バックエンド

Red Hat OpenStack Platform (RHOSP) director を使用して Shared File Systems サービスをデプロイする場合、クラウド管理者は以下に示すサポート対象のバックエンドの1つを選択することができます。

- NFS バックエンドへの CephFS の使用。詳細は、[NFS バックエンドに CephFS を使用した Shared File Systems サービスのデプロイ](#) を参照してください。
- ネイティブ CephFS。詳しくは、[CephFS Back End Guide for the Shared File System Service](#) を参照してください。
- NetApp。詳しくは、NetApp のドキュメント [Shared Filesystems Service \(Manila\)](#) を参照してください。
- Dell EMC Unity、Dell VNX、または Dell PowerMax。詳しくは、Dell のドキュメント [Dell EMC Manila Backend Deployment Guide for Red Hat OpenStack Platform 16](#) を参照してください。

サポート対象のバックエンドアプライアンスおよびドライバーの完全なリストは、[Component, Plug-In, and Driver Support in Red Hat OpenStack Platform](#) を参照してください。

4.1.1. 共有ファイルシステムのネットワーク

共有ファイルシステムには、ネットワーク経由でアクセスします。エンドユーザークライアントが Red Hat OpenStack Platform (RHOSP) 仮想マシン、ベアメタルサーバー、およびコンテナで実行されるワークロードに共有を接続できるように、クラウド上のネットワークのプランニングを行うことが重要です。

エンドユーザーに必要なセキュリティと分離のレベルに応じて、管理者は `driver_handles_share_servers` パラメーターを `true` または `false` に設定できます。

`driver_handles_share_servers` パラメーターを `true` に設定すると、サービスは、分離された共有サーバーを使用して、エンドユーザー定義の共有ネットワークに共有をエクスポートできるようになります。

`driver_handles_share_servers` パラメーターが `true` の場合、ユーザーはセルフサービス共有ネットワークでワークロードをプロビジョニングできます。これにより、専用のネットワークセグメント上の完全に分離された NAS ファイルサーバーにより共有がエクスポートされるようになります。

エンドユーザーが使用する共有ネットワークは、作成可能なプライベートプロジェクトネットワークと同じになります。管理者は、分離ネットワークをマッピングする物理ネットワークがストレージインフラストラクチャーに拡張するようする必要があります。

また、プロジェクトネットワークによるネットワークのセグメント化スタイルが、使用するストレージシステムによりサポートされるようにする必要があります。NetApp ONTAP および Dell EMC PowerMax、Unity、ならびに VNX 等のストレージシステムは、GENEVE または VXLAN 等の仮想オーバーレイセグメント化スタイルをサポートしません。

代替として、オーバーレイネットワークをトップオブブラックのスイッチで終端し、プロジェクトネットワークに VLAN 等のよりプリミティブなネットワーク形式を使用することができます。さらに別の方法として、共有プロバイダーネットワーク上の VLAN セグメントを許可するか、ストレージシステムにすでに接続されている既存のセグメント化ネットワークへのアクセスを提供することもできます。

driver_handles_share_servers パラメーターを `false` に設定すると、ユーザーは自分の共有ネットワーク上に共有を作成できなくなります。代わりに、クライアントをクラウド管理者が設定したネットワークに接続する必要があります。

driver_handles_share_servers パラメーターが `false` に等しい場合、director は専用の共有ストレージネットワークを作成できます。たとえば、標準の director テンプレートを使用してネイティブ CephFS バックエンドをデプロイする場合、director は **Storage** と呼ばれる共有プロバイダーネットワークを作成します。NFS バックエンドを介して CephFS をデプロイメントする場合、共有プロバイダーネットワークは **StorageNFS** と呼ばれます。エンドユーザーは、共有にアクセスするためにクライアントを共有ストレージネットワークに接続する必要があります。

すべての共有ファイルシステムのストレージドライバーが、両方のモードの操作に対応しているわけではありません。選択したモードに関係なく、サービスはハードデータパスのマルチテナンシー分離の保証を確保します。

セルフサービスモデルの一部としてテナントワークロードにハードネットワークパスマルチテナンシー分離保証を提供する場合は、**driver_handles_share_servers** ドライバーモードをサポートするバックエンドを使用してデプロイする必要があります。

ファイル共有へのネットワーク接続に関する情報は、「[ファイル共有へのネットワーク接続の確保](#)」を参照してください。

4.1.2. 共有種別の作成

共有種別は、配置の決定を行う Shared File Systems サービスのスケジューラーへのヒントとして機能します。Red Hat OpenStack Platform (RHOSP) director は `default` という名前のデフォルト共有種別で Shared File Systems サービスを設定しますが、共有種別を作成する訳ではありません。

重要

Shared File Systems サービスを使用するために、エンドユーザーには少なくとも1つの共有種別が必要です。

手順

1. オーバークラウドをデプロイしたら、クラウド管理者として以下のコマンドを実行して共有種別を作成します。

```
# manila type-create default <spec_driver_handles_share_servers>
```

<spec_driver_handles_share_servers> パラメーターはブール値で、以下のように設定します。

- NFS バックエンドに CephFS を使用する設定またはネイティブ CephFS の場合、値は false です。
 - 他のバックエンドの場合、値は true または false になります。 **Manila<backend>DriverHandlesShareServers** パラメーターの値と一致するように **<spec_driver_handles_share_servers>** を設定します。たとえば、NetApp バックエンドを使用する場合、パラメーターは **ManilaNetappDriverHandlesShareServers** と呼ばれます。
2. デフォルトの共有種別に仕様を追加するか、複数のバックエンドが設定された環境で使用するために追加の共有種別を作成します。たとえば、デフォルトの共有タイプを設定して、CephFS バックエンドと、NetApp **driver_handles_share_servers=True** バックエンドを使用する追加の共有タイプを選択します。

```
(overcloud) [stack@undercloud-0 ~]$ manila type-create default false --extra-specs
share_backend_name='cephfs'
(overcloud) [stack@undercloud-0 ~]$ manila type-create netapp true --extra-specs
share_backend_name='tripleo_netapp'
```

注記

デフォルトでは共有種別はパブリックなので、すべてのクラウドプロジェクトが認識して使用することができます。ただし、特定プロジェクト用にプライベートの共有種別を作成することもできます。プライベートの共有種別を作成する、または追加の共有種別オプションを設定する方法についての詳細は、[Security and Hardening Guide](#) を参照してください。

4.1.3. 共有種別の一般的な機能

ファイル共有の種別は、ファイル共有の一般的な機能を定義します。共有種別の一般的な機能を確認し、ファイル共有での操作について理解するようにしてください。

表4.1 共有種別の機能

機能	値	説明
driver_handles_share_servers	true または false	共有ネットワークを使用してファイル共有を作成する権限を付与します。
snapshot_support	true または false	ファイル共有のスナップショットを作成する権限を付与します。
create_share_from_snapshot_support	true または false	共有スナップショットのクローンを作成する権限を付与します。
revert_to_snapshot_support	true または false	ファイル共有を最新のスナップショットに戻す権限を付与します。
mount_snapshot_support	true または false	スナップショットをエクスポートおよびマウントする権限を付与します。

機能	値	説明
replication_type	dr	障害復旧用のレプリカを作成する権限を付与します。1度に許可されるアクティブなエクスポートは1つだけです。
	readable	読み取り専用レプリカを作成する権限を付与します。1度に許可される書き込み可能なアクティブなエクスポートは1つだけです。
	writable	読み取り/書き込みレプリカを作成する権限を付与します。共有ごとに1度に任意の数のアクティブなエクスポートが許可されます。
availability_zones	1つまたは複数のアベイラビリティゾーンのリスト	リスト表示されるアベイラビリティゾーンでのみ共有を作成する権限を付与します。

4.1.4. 共有種別の検出

クラウドユーザーは、ファイル共有の作成時に共有種別を指定する必要があります。

手順

1. 利用可能な共有種別を検出します。

```
$ manila type-list
```

コマンド出力には、共有種別の名前と ID がリスト表示されます。

4.1.5. ファイル共有の作成

データの読み取りと書き込みを行うための共有を作成します。

ファイル共有を作成するには、以下のようなコマンドを使用します。

```
$ manila create [--share-type <sharetype>] [--name <sharename>] proto GB
```

以下の値を置き換えます。

- **sharetype** は、指定された共有タイプに関連付けられた設定を適用します。
 - オプション: 指定しない場合には、**default** 共有種別が使用されます。
- **sharename** は共有の名前です。
 - (オプション) ファイル共有名は必須ではありません。また、名前は一意でなくても構いません。

- **proto** は、使用する共有プロトコルです。
 - NFS を使用する CephFS の場合、**proto** は **nfs** です。
 - CephFS ネイティブの場合、**proto** は **cephfs** です。
 - NetApp および DellEMC ストレージバックエンドの場合、**proto** は **nfs** または **cifs** です。
- **GB** は、ファイル共有のサイズ (GB 単位) に置き換えます。

たとえば、「共有種別の作成」では、クラウド管理者は CephFS バックエンドを選択する **default** 共有種別および NetApp バックエンドを選択する **netapp** という名前の別の共有種別を作成しました。

手順

1. 共有種別の例を使用して CephFS バックエンド上に **share-01** という名前の 10 GB の NFS 共有を作成します。以下の例では、NFS バックエンドに CephFS を使用しています。

```
(user) [stack@undercloud-0 ~]$ manila create --name share-01 nfs 10
```

2. オプションとして、NetApp バックエンド上に **share-02** という名前の 20 GB の NFS 共有を作成します。

```
(user) [stack@undercloud-0 ~]$ manila create --name share-02 --share-type netapp --share-network mynet nfs 20
```

4.1.6. ファイル共有とエクスポート情報のリスト表示

ファイル共有が正常に作成されていることを確認するには、以下の手順を実施します。

手順

1. ファイル共有をリスト表示します。

```
(user) [stack@undercloud-0 ~]$ manila list
+-----+-----+-----+-----+ | ID |
Name | ... | Status | ... |
+-----+-----+-----+-----+
| 8c3bedd8-bc82-4100-a65d-53ec51b5fe81 | share-01 | ... | available ...
+-----+-----+-----+-----+
```

2. ファイル共有のエクスポート場所を表示します。

```
(user) [stack@undercloud-0 ~]$ manila share-export-location-list share-01
+-----+
| Path |
| 172.17.5.13:/volumes/_nogroup/e840b4ae-6a04-49ee-9d6e-67d4999fbc01 |
+-----+
```

3. ファイル共有のパラメーターを表示します。

```
manila share-export-location-show <id>
```



注記

この情報は、「[ファイル共有のマウント](#)」でファイル共有をマウントする際に使用します。

4.2. ファイル共有へのネットワーク接続の確保

ファイル共有に接続する必要があるクライアントには、そのファイル共有の1つまたは複数のエクスポート場所へのネットワーク接続が必要です。

Shared File Systems サービスと共にネットワークを設定する方法は、ネットワークプラグインの使用を含め多数あります。

共有タイプの **driver_handles_share_servers** パラメーターが true の場合、クラウドユーザーは、コンピューティングインスタンスが接続されているネットワークの詳細を使用して共有ネットワークを作成し、共有を作成するときにそれを参照できます。

共有タイプの **driver_handles_share_servers** パラメーターが false に等しい場合、クラウドユーザーはコンピューティングインスタンスを共有ストレージネットワークに接続する必要があります。

共有ネットワークへのネットワーク接続を設定して検証する方法の詳細は、「[ファイル共有にアクセスするための共有ネットワークへの接続](#)」を参照してください。

4.2.1. ファイル共有にアクセスするための共有ネットワークへの接続

driver_handles_share_servers パラメーターが false に等しい場合、共有は、管理者が使用可能にした共有プロバイダーネットワークにエクスポートされます。エンドユーザーは、ファイル共有にアクセスするために、コンピューティングインスタンス等のクライアントを共有プロバイダーネットワークに接続する必要があります。

以下の手順例では、共有プロバイダーネットワークは StorageNFS という名前です。StorageNFS は、director が NFS バックエンドに CephFS を使用する Shared File Systems サービスをデプロイする際に設定されます。クラウド管理者が利用可能にするネットワークに接続するには、同様の手順に従います。



注記

この手順の例では、クライアントの IP アドレスファミリーバージョンは重要ではありません。この手順のステップでは IPv4 アドレス設定を使用していますが、IPv6 でもステップは同一です。

手順

1. StorageNFS ポート用のセキュリティーグループを作成し、ポートから外部に出るパケットは許可するが、未確立接続からの着信パケットは拒否します。

```
(user) [stack@undercloud-0 ~]$ openstack security group create no-ingress -f yaml
created_at: '2018-09-19T08:19:58Z'
description: no-ingress
id: 66f67c24-cd8b-45e2-b60f-9eaedc79e3c5
name: no-ingress
project_id: 1e021e8b322a40968484e1af538b8b63
revision_number: 2
rules: 'created_at="2018-09-19T08:19:58Z", direction="egress", ethertype="IPv4",
id="6c7f643f-3715-4df5-9fef-0850fb6eaaf2", updated_at="2018-09-19T08:19:58Z"
```

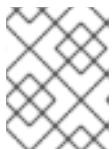


```
created_at="2018-09-19T08:19:58Z", direction="egress", ethertype="IPv6",
id="a8ca1ac2-fbe5-40e9-ab67-3e55b7a8632a", updated_at="2018-09-19T08:19:58Z"
updated_at: '2018-09-19T08:19:58Z'
```

2. **no-ingress** セキュリティーグループによりセキュリティーを確保し、StorageNFS ネットワーク上にポートを作成します。

```
(user) [stack@undercloud-0 ~]$ openstack port create nfs-port0 --network StorageNFS --
security-group no-ingress -f yaml
```

```
admin_state_up: UP
allowed_address_pairs: "
binding_host_id: null
binding_profile: null
binding_vif_details: null
binding_vif_type: null
binding_vnic_type: normal
created_at: '2018-09-19T08:03:02Z'
data_plane_status: null
description: "
device_id: "
device_owner: "
dns_assignment: null
dns_name: null
extra_dhcp_opts: "
fixed_ips: ip_address='172.17.5.160', subnet_id='7bc188ae-aab3-425b-a894-863e4b664192'
id: 7a91cbbc-8821-4d20-a24c-99c07178e5f7
ip_address: null
mac_address: fa:16:3e:be:41:6f
name: nfs-port0
network_id: cb2cbc5f-ea92-4c2d-beb8-d9b10e10efae
option_name: null
option_value: null
port_security_enabled: true
project_id: 1e021e8b322a40968484e1af538b8b63
qos_policy_id: null
revision_number: 6
security_group_ids: 66f67c24-cd8b-45e2-b60f-9eaedc79e3c5
status: DOWN
subnet_id: null
tags: "
trunk_details: null
updated_at: '2018-09-19T08:03:03Z'
```



注記

StorageNFSSubnet では、**nfs-port0** に IP アドレス 172.17.5.160 が割り当てられています。

3. コンピュートインスタンスに **nfs-port0** を追加します。

```
(user) [stack@undercloud-0 ~]$ openstack server add port instance0 nfs-port0
(user) [stack@undercloud-0 ~]$ openstack server list -f yaml
```

```
- Flavor: m1.micro
ID: 0b878c11-e791-434b-ab63-274ecfc957e8
Image: manila-test
Name: demo-instance0
Networks: demo-network=172.20.0.4, 10.0.0.53; StorageNFS=172.17.5.160
Status: ACTIVE
```

プライベートアドレスおよび Floating IP アドレスに加えて、コンピュートインスタンスに StorageNFS ネットワークのポート (IP アドレス: 172.17.5.160) が割り当てられ、該当するファイル共有のアドレスへのアクセスが許可されると、NFS 共有のマウントに使用することができます。



注記

このアドレスのインターフェイスをアクティブ化するには、コンピュートインスタンスのネットワーク設定の調整やサービスの再起動が必要になる場合があります。

4.2.2. ネットワークとインスタンス間の IPv6 インターフェイスの設定

ファイル共有をエクスポートする共有ネットワークが IPv6 アドレス設定を使用する場合、セカンダリーインターフェイスの DHCPv6 で問題が発生する可能性があります。以下の手順に従って、インスタンスで IPv6 インターフェイスを手動で設定します。詳細な情報は、[BZ#1859695](#) を参照してください。

前提条件

- ファイル共有にアクセスするための共有ネットワークへの接続

手順

1. インスタンスにログインします。
2. IPv6 インターフェイスアドレスを設定します。

```
$ ip address add fd00:fd00:fd00:7000::c/64 dev eth1
```

3. インターフェイスをアクティベートします。

```
$ ip link set dev eth1 up
```

4. ファイル共有のエクスポート場所にある IPv6 アドレスに ping を行い、インターフェイスの接続をテストします。

```
$ ping -6 fd00:fd00:fd00:7000::21
```

5. あるいは、Telnet を使用して NFS サーバーに到達できることを確認します。

```
$ dnf install -y telnet
$ telnet fd00:fd00:fd00:7000::21 2049
```

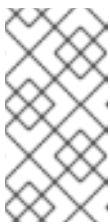
4.3. ファイル共有に対するアクセス権限の付与

コンピュータインスタンス等のクライアントにファイル共有をマウントする前に、以下のようなコマンドを使用して、クライアントがファイル共有にアクセスできるようにする必要があります。

```
# manila access-allow <share> <accesstype> --access-level <accesslevel> <clientidentifier>
```

以下の値を置き換えます。

- **share**: 「[ファイル共有の作成](#)」で作成したファイル共有の名前または ID
- **accesstype**: ファイル共有で要求されるアクセスの種別。以下に種別の例を示します。
 - **user**: ユーザーまたはグループ名で認証する場合に使用します。
 - **ip**: IP アドレスでインスタンスを認証する場合に使用します。
 - **cephx**: ネイティブ CephFS クライアントユーザー名による認証に使用します。



注記

アクセスの種別は、ファイル共有のプロトコルにより異なります。NFS 共有の場合は、許可されるのは **ip** アクセス種別だけです。CIFS の場合は、**user** アクセス種別が適切です。ネイティブ CephFS ファイル共有の場合は、**cephx** を使用する必要があります。

- **accesslevel**: オプションの設定で、デフォルトは **rw** です。
 - **rw**: ファイル共有への読み取り/書き込みアクセスが許可されます。
 - **ro**: ファイル共有への読み取りアクセスのみが許可されます。
- **clientidentifier -accesstype** によって異なります。
 - **ip accesstype** のための IP アドレスを使用してください。
 - **user accesstype** には CIFS ユーザーまたはグループを使用します。
 - **cephx accesstype** にはユーザー名文字列を使用します。

4.3.1. ファイル共有へのアクセスの付与

ユーザーがファイル共有からデータの読み取りと書き込みを可能にするように、エンドユーザークライアントにファイル共有へのアクセス権限を付与する必要があります。

以下の手順に従って、インスタンスの IP アドレスを使用して、クライアントのコンピュータインスタンスに NFS 共有へのアクセス権限を付与します。CIFS 共有の **user** ルールと CephFS 共有の **cephx** ルールは同様のパターンに従います。**user** および **cephx** アクセスタイプでは、必要に応じて、複数のクライアントで同じ **clientidentifier** を使用できます。



注記

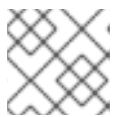
この手順の例では、クライアントの IP アドレスファミリーバージョンは重要ではありません。この手順のステップでは IPv4 アドレス設定を使用していますが、IPv6 でもステップは同一です。

手順

1. ファイル共有をマウントする予定のクライアントコンピュータインスタンスの IP アドレスを取得します。ファイル共有に到達できるネットワークに対応する IP アドレスを選択してください。以下の例では、StorageNFS ネットワークの IP アドレスを使用します。

```
(user) [stack@undercloud-0 ~]$ openstack server list -f yaml
- Flavor: m1.micro
  ID: 0b878c11-e791-434b-ab63-274ecfc957e8
  Image: manila-test
  Name: demo-instance0
  Networks: demo-network=172.20.0.4, 10.0.0.53;
  StorageNFS=172.17.5.160
  Status: ACTIVE
```

```
(user) [stack@undercloud-0 ~]$ manila access-allow share-01 ip 172.17.5.160
```



注記

ファイル共有へのアクセスには、独自の ID (**accessid**) が指定されています。

```
+-----+-----+
| Property | Value |
+-----+-----+
| access_key | None |
| share_id | db3bedd8-bc82-4100-a65d-53ec51b5cba3 |
| created_at | 2018-09-17T21:57:42.000000 |
| updated_at | None |
| access_type | ip |
| access_to | 172.17.5.160 |
| access_level | rw |
| state | queued_to_apply |
| id | 875c6251-c17e-4c45-8516-fe0928004fff |
+-----+-----+
```

2. アクセス設定が正常に完了したことを確認します。

```
(user) [stack@undercloud-0 ~]$ manila access-list share-01
```

```
+-----+-----+-----+-----+-----+ ...
| id      | access_type | access_to | access_level | state | ...
+-----+-----+-----+-----+-----+
| 875c6251-... | ip      | 172.17.5.160 | rw      | active | ...
+-----+-----+-----+-----+-----+ ...
```

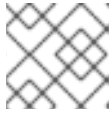
4.3.2. ファイル共有へのアクセスの取り消し

ファイル共有の所有者は、何らかの理由でファイル共有へのアクセスを無効にすることができます。以前に付与したファイル共有へのアクセス権限を取り消すには、以下の手順を実施します。

手順

1. ファイル共有へのアクセスを取り消します。

```
# manila access-deny <share> <accessid>
```

**注記**

<share> を共有名または共有 ID に置き換えます。

以下に例を示します。

```
(user) [stack@undercloud-0 ~]$ manila access-list share-01
+-----+-----+-----+-----+-----+
| id      | access_type | access_to  | access_level | state | ...
+-----+-----+-----+-----+-----+ ...
| 875c6251-... | ip        | 172.17.5.160 | rw          | active | ...
+-----+-----+-----+-----+-----+

(user) [stack@undercloud-0 ~]$ manila access-deny share-01 875c6251-c17e-4c45-8516-
fe0928004fff

(user) [stack@undercloud-0 ~]$ manila access-list share-01

+-----+-----+-----+-----+-----+ ...
| id      | access_type| access_to  | access_level | state | ...
+-----+-----+-----+-----+-----+ ...
+-----+-----+-----+-----+-----+ ...
```

**注記**

読み取り/書き込み権限を持つ既存のクライアントがあり、クライアントに読み取り専用の権限を付与する場合は、ファイル共有へのアクセスを無効にし、読み取り専用のルールを追加する必要があります。

4.4. コンピュートインスタンスへのファイル共有のマウント

クライアントにアクセスを付与したら、共有をマウントしてクライアントによる使用が可能になります。クライアントへのネットワーク接続があれば、任意のクライアント種別がファイル共有にアクセスできます。

仮想コンピューティングインスタンスに NFS 共有をマウントするのに使用する手順は、ベアメタルのコンピューティングインスタンスに NFS 共有をマウントする手順と類似しています。OpenShift コンテナにファイル共有をマウントする方法の詳細は、[OpenShift Container Platform の製品ドキュメント](#) を参照してください。

**注記**

ファイル共有をマウントするコンピューティングインスタンスに、異なるプロトコル用のクライアントパッケージをインストールする必要があります。たとえば、NFS バックエンドに CephFS を使用する Shared File Systems サービスの場合には、NFS クライアントパッケージが NFS 4.1 をサポートしている必要があります。

4.4.1. ファイル共有のエクスポート場所のリスト表示

ファイル共有をマウントできるように、ファイル共有のエクスポート場所を取得します。

手順

1. ファイル共有のエクスポート場所を取得します。

```
(user) [stack@undercloud-0 ~]$ manila share-export-location-list share-01
```

複数のエクスポート場所が存在する場合には、**preferred** メタデータフィールドの値が True のものを選択します。希望するエクスポート場所が存在しない場合は、任意のエクスポート場所を使用できます。

4.4.2. ファイル共有のマウント

クライアントにファイル共有をマウントして、データへのアクセスを可能にします。

ファイル共有の作成およびファイル共有へのアクセス権限の付与についての情報は、以下の手順を参照してください。

- [「ファイル共有の作成」](#)
- [「ファイル共有に対するアクセス権限の付与」](#)

手順

1. インスタンスにログインし、以下のコマンドを実行します。

```
(user) [stack@undercloud-0 ~]$ openstack server ssh demo-instance0 --login root
# hostname
demo-instance0
```

2. エクスポート場所を使用して、IPv4 ネットワークにファイル共有をマウントします。

```
# mount -t nfs -v 172.17.5.13:/volumes/_nogroup/e840b4ae-6a04-49ee-9d6e-67d4999fbc01
/mnt
```

4.5. ファイル共有の削除

Shared File Systems service (manila) は、データの削除を防ぐ保護機能を提供しません。Shared File Systems サービスは、クライアントが接続されているか、ワークロードが実行されているかを確認しません。ファイル共有を削除する場合は、ファイル共有を取得することはできません。

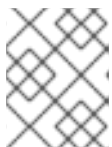
警告

ファイル共有を削除する前にデータをバックアップします。

手順

1. ファイル共有を削除します。

```
# manila delete <share>
```



注記

上記コマンド例の <share> は、ファイル共有の名前または ID のどちらかを指定することができます。

以下に例を示します。

```
# manila delete share-01
```

4.6. SHARED FILE SYSTEMS サービスのデフォルトクォータの変更

気付かぬままにシステムリソースをすべて消費してしまうのを防ぐために、クラウド管理者はクォータを設定することができます。クォータとは、運用上の制限です。

4.6.1. クォータのリスト表示

クラウド管理者は、**manila quote-show** コマンドを使用して、プロジェクトまたはユーザーのクォータをリスト表示できます。オプションの **--user** パラメーターを指定すると、指定したプロジェクトでそのユーザーに設定されたクォータを表示することができます。このパラメーターを省略した場合には、指定したプロジェクトに設定されたクォータが表示されます。

クォータは、更新や削除が可能です。shares、snapshots、gigabytes、snapshot-gigabytes、share-networks、share_groups、share_group_snapshots、および share-type のクォータを更新することができます。

手順

1. 使用方法に関する説明を表示するには、以下のコマンドを実行します。

```
# manila help quota-show
# manila help quota-update
# manila help quota-delete
```

4.7. エラーのトラブルシューティング

ファイル共有の作成やファイル共有グループの作成などの Shared File Systems (manila) の操作が非同期に失敗する場合には、エンドユーザーはコマンドラインからクエリーを実行してエラーについての詳細を取得することができます。

4.7.1. ファイル共有作成またはファイル共有グループ作成の失敗の修正

この例では、ユーザーは複数の仮想マシンにソフトウェアライブラリーをホストするファイル共有を作成しようとしています。この例では、ファイル共有の作成を2度意図的に失敗させ、コマンドラインを使用してユーザーサポートメッセージを取得する方法を説明します。

手順

1. ファイル共有を作成するには、ファイル共有に割り当てる機能を指定する共有種別を使用します。クラウド管理者は共有種別を作成することができます。利用可能な共有種別を表示します。

```
clouduser1@client:~$ manila type-list
+-----+-----+-----+-----+-----+-----+
| ID           | Name       | visibility | is_default | required_extra_specs | optional_extra_specs | Description |
+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+
```

```

| 1cf5d45a-61b3-44d1-8ec7-89a21f51a4d4 | dhss_false | public | YES |
driver_handles_share_servers : False | create_share_from_snapshot_support : True | None
|
|
mount_snapshot_support : False |
|
revert_to_snapshot_support : False |
|
| snapshot_support :
True
|
| 277c1089-127f-426e-9b12-711845991ea1 | dhss_true | public | - |
driver_handles_share_servers : True | create_share_from_snapshot_support : True | None
|
|
mount_snapshot_support : False |
|
revert_to_snapshot_support : False |
|
| snapshot_support :
True
+-----+-----+-----+-----+
+-----+

```

この例では、2つの共有種別が使用可能です。

2. **driver_handles_share_servers=True** 機能を指定する共有種別を使用するには、ファイル共有をエクスポートするファイル共有ネットワークを作成する必要があります。プライベートプロジェクトネットワークから、ファイル共有ネットワークを作成します。

```

clouduser1@client:~$ openstack subnet list
+-----+-----+-----+-----+
+-----+
| ID | Name | Network | Subnet |
+-----+-----+-----+-----+
+-----+
| 78c6ac57-bba7-4922-ab81-16cde31c2d06 | private-subnet | 74d5cfb3-5dd0-43f7-b1b2-5b544cb16212 | 10.0.0.0/26 |
| a344682c-718d-4825-a87a-3622b4d3a771 | ipv6-private-subnet | 74d5cfb3-5dd0-43f7-b1b2-5b544cb16212 | fd36:18fc:a8e9::/64 |
+-----+-----+-----+-----+
+-----+

clouduser1@client:~$ manila share-network-create --name mynet --neutron-net-id 74d5cfb3-5dd0-43f7-b1b2-5b544cb16212 --neutron-subnet-id 78c6ac57-bba7-4922-ab81-16cde31c2d06
+-----+-----+-----+-----+
| Property | Value |
+-----+-----+-----+
| network_type | None |
| name | mynet |
| segmentation_id | None |
| created_at | 2018-10-09T21:32:22.485399 |
| neutron_subnet_id | 78c6ac57-bba7-4922-ab81-16cde31c2d06 |
| updated_at | None |
| mtu | None |
| gateway | None |
| neutron_net_id | 74d5cfb3-5dd0-43f7-b1b2-5b544cb16212 |
| ip_version | None |

```



```

| cidr          | None          |
| project_id    | cadd7139bc3148b8973df097c0911016 |
| id            | 0b0fc320-d4b5-44a1-a1ae-800c56de550c |
| description   | None          |
+-----+

```

```
clouduser1@client:~$ manila share-network-list
```

```

+-----+-----+
| id            | name         |
+-----+-----+
| 6c7ef9ef-3591-48b6-b18a-71a03059edd5 | mynet |
+-----+-----+

```

3. ファイル共有を作成します。

```
clouduser1@client:~$ manila create nfs 1 --name software_share --share-network mynet --share-type dhss_true
```

```

+-----+-----+
| Property      | Value       |
+-----+-----+
| status        | creating    |
| share_type_name | dhss_true   |
| description    | None        |
| availability_zone | None        |
| share_network_id | 6c7ef9ef-3591-48b6-b18a-71a03059edd5 |
| share_server_id | None        |
| share_group_id | None        |
| host          |             |
| revert_to_snapshot_support | False      |
| access_rules_status | active      |
| snapshot_id    | None        |
| create_share_from_snapshot_support | False      |
| is_public      | False       |
| task_state     | None        |
| snapshot_support | False       |
| id            | 243f3a51-0624-4bdd-950e-7ed190b53b67 |
| size          | 1           |
| source_share_group_snapshot_member_id | None       |
| user_id       | 61aef4895b0b41619e67ae83fba6defe |
| name          | software_share |
| share_type     | 277c1089-127f-426e-9b12-711845991ea1 |
| has_replicas  | False       |
| replication_type | None        |
| created_at    | 2018-10-09T21:12:21.000000 |
| share_proto   | NFS         |
| mount_snapshot_support | False       |
| project_id    | cadd7139bc3148b8973df097c0911016 |
| metadata     | {}          |
+-----+-----+

```

4. ファイル共有のステータスを表示します。

```
clouduser1@client:~$ manila list
```

```

+-----+-----+-----+-----+-----+-----+
+-----+-----+

```

```

| ID | Name | Size | Share Proto | Status | Is Public | Share Type
Name | Host | Availability Zone |
+-----+-----+-----+-----+-----+-----+-----+
| 243f3a51-0624-4bdd-950e-7ed190b53b67 | software_share | 1 | NFS | error | False
| dhss_true | | None |
+-----+-----+-----+-----+-----+-----+

```

この例では、ファイル共有の作成中にエラーが発生しています。

5. ユーザーサポートメッセージを表示するには、**message-list** コマンドを実行します。 --**resource-id** を使用して、確認したい特定のファイル共有に絞り込みます。

```

clouduser1@client:~$ manila message-list
+-----+-----+-----+-----+-----+-----+-----+
| ID | Resource Type | Resource ID | Action ID | User
Message | Detail ID | Created At
|
+-----+-----+-----+-----+-----+-----+-----+
| 7d411c3c-46d9-433f-9e21-c04ca30b209c | SHARE | 243f3a51-0624-4bdd-950e-
7ed190b53b67 | 001 | allocate host: No storage could be allocated for this share request,
Capabilities filter didn't succeed. | 008 | 2018-10-09T21:12:21.000000 |
+-----+-----+-----+-----+-----+-----+-----+

```

User Message コラムに、機能の不一致により Shared File Systems サービスがファイル共有の作成に失敗したことが表示されています。

6. より多くのメッセージ情報を表示するには、**message-show** コマンドを実行してから、**message-list** コマンドからのメッセージの ID を実行します。

```

clouduser1@client:~$ manila message-show 7d411c3c-46d9-433f-9e21-c04ca30b209c
+-----+-----+-----+-----+-----+-----+-----+
| Property | Value |
+-----+-----+-----+-----+-----+-----+-----+
| request_id | req-0a875292-6c52-458b-87d4-1f945556feac |
| detail_id | 008 |
| expires_at | 2018-11-08T21:12:21.000000 |
| resource_id | 243f3a51-0624-4bdd-950e-7ed190b53b67 |
| user_message | allocate host: No storage could be allocated for this share request,
Capabilities filter didn't succeed. |
| created_at | 2018-10-09T21:12:21.000000 |
| message_level | ERROR |
| id | 7d411c3c-46d9-433f-9e21-c04ca30b209c |

```

```

|
| resource_type | SHARE
| action_id    | 001
+-----+
-----+

```

7. クラウドユーザーは共有種別により機能を確認することができるので、利用可能な共有種別を確認することができます。2つの共有種別の違いは、**driver_handles_share_servers** の値です。

```

clouduser1@client:~$ manila type-list
+-----+-----+-----+-----+-----+-----+
| ID                | Name      | visibility | is_default | required_extra_specs | optional_extra_specs | Description |
+-----+-----+-----+-----+-----+-----+
| 1cf5d45a-61b3-44d1-8ec7-89a21f51a4d4 | dhss_false | public    | YES       |                       |                       |             |
driver_handles_share_servers : False | create_share_from_snapshot_support : True | None
|
| mount_snapshot_support : False
|
| revert_to_snapshot_support : False
|
| snapshot_support :
True
|
| 277c1089-127f-426e-9b12-711845991ea1 | dhss_true  | public    | -        |                       |                       |             |
driver_handles_share_servers : True  | create_share_from_snapshot_support : True | None
|
| mount_snapshot_support : False
|
| revert_to_snapshot_support : False
|
| snapshot_support :
True
+-----+-----+-----+-----+-----+-----+
-----+

```

8. 利用可能な他の共有種別でファイル共有を作成します。

```

clouduser1@client:~$ manila create nfs 1 --name software_share --share-network mynet --share-type dhss_false
+-----+-----+-----+-----+-----+-----+
| Property          | Value
+-----+-----+-----+-----+-----+-----+
| status            | creating
| share_type_name   | dhss_false
| description       | None
| availability_zone | None
| share_network_id  | 6c7ef9ef-3591-48b6-b18a-71a03059edd5 |
| share_group_id    | None
| revert_to_snapshot_support | False
| access_rules_status | active
| snapshot_id       | None
| create_share_from_snapshot_support | True

```

```

| is_public          | False          |
| task_state        | None           |
| snapshot_support  | True           |
| id                | 2d03d480-7cba-4122-ac9d-edc59c8df698 |
| size              | 1              |
| source_share_group_snapshot_member_id | None          |
| user_id           | 5c7bdb6eb0504d54a619acf8375c08ce |
| name              | software_share |
| share_type        | 1cf5d45a-61b3-44d1-8ec7-89a21f51a4d4 |
| has_replicas      | False         |
| replication_type  | None          |
| created_at        | 2018-10-09T21:24:40.000000 |
| share_proto       | NFS           |
| mount_snapshot_support | False        |
| project_id        | cadd7139bc3148b8973df097c0911016 |
| metadata          | {}            |
+-----+

```

この例では、試みた 2 番目のファイル共有作成は失敗します。

9. ユーザーサポートメッセージを表示します。

```

clouduser1@client:~$ manila list
+-----+
-----+-----+
| ID                | Name          | Size | Share Proto | Status | Is Public | Share Type
Name | Host | Availability Zone |
+-----+-----+-----+-----+-----+-----+-----+
| 2d03d480-7cba-4122-ac9d-edc59c8df698 | software_share | 1    | NFS        | error | False
| dhss_false      | | nova          |
| 243f3a51-0624-4bdd-950e-7ed190b53b67 | software_share | 1    | NFS        | error | False
| dhss_true       | | None          |
+-----+-----+-----+-----+-----+-----+
-----+-----+

clouduser1@client:~$ manila message-list
+-----+
-----+-----+
---+-----+
| ID                | Resource Type | Resource ID                | Action ID | User
Message                                     | Detail ID | Created At
|
+-----+-----+-----+-----+-----+-----+
-----+-----+
---+-----+
| ed7e02a2-0cdb-4ff9-b64f-e4d2ec1ef069 | SHARE        | 2d03d480-7cba-4122-ac9d-
edc59c8df698 | 002    | create: Driver does not expect share-network to be provided with
current configuration. | 003    | 2018-10-09T21:24:40.000000 |
| 7d411c3c-46d9-433f-9e21-c04ca30b209c | SHARE        | 243f3a51-0624-4bdd-950e-
7ed190b53b67 | 001    | allocate host: No storage could be allocated for this share request,
Capabilities filter didn't succeed. | 008    | 2018-10-09T21:12:21.000000 |
+-----+-----+-----+-----+-----+-----+
-----+-----+
---+-----+

```

サービスは、使用した共有種別のファイル共有ネットワークに対応していません。

10. 管理者に相談することなく、管理者が利用可能にしたストレージバックエンドではプライベート neutron ネットワークにファイル共有を直接エクスポートする操作がサポートされない、ということを明らかにすることができます。 **share-network** パラメーターを指定せずに、ファイル共有を作成します。

```
clouduser1@client:~$ manila create nfs 1 --name software_share --share-type dhss_false
+-----+
| Property          | Value          |
+-----+
| status            | creating       |
| share_type_name   | dhss_false     |
| description       | None           |
| availability_zone | None           |
| share_network_id | None           |
| share_group_id    | None           |
| revert_to_snapshot_support | False         |
| access_rules_status | active         |
| snapshot_id       | None           |
| create_share_from_snapshot_support | True         |
| is_public         | False          |
| task_state        | None           |
| snapshot_support  | True           |
| id                | 4d3d7fcf-5fb7-4209-90eb-9e064659f46d |
| size              | 1              |
| source_share_group_snapshot_member_id | None         |
| user_id           | 5c7bdb6eb0504d54a619acf8375c08ce |
| name              | software_share |
| share_type        | 1cf5d45a-61b3-44d1-8ec7-89a21f51a4d4 |
| has_replicas      | False          |
| replication_type  | None           |
| created_at        | 2018-10-09T21:25:40.000000 |
| share_proto       | NFS            |
| mount_snapshot_support | False         |
| project_id        | cadd7139bc3148b8973df097c0911016 |
| metadata          | {}             |
+-----+
```

11. ファイル共有が正常に作成されたことを確認します。

```
clouduser1@client:~$ manila list
+-----+-----+-----+-----+-----+-----+-----+
| ID          | Name          | Size | Share Proto | Status | Is Public | Share Type |
| Name | Host | Availability Zone |
+-----+-----+-----+-----+-----+-----+-----+
| 4d3d7fcf-5fb7-4209-90eb-9e064659f46d | software_share | 1 | NFS | available | False | dhss_false |
| 2d03d480-7cba-4122-ac9d-edc59c8df698 | software_share | 1 | NFS | error | False | dhss_false |
| 243f3a51-0624-4bdd-950e-7ed190b53b67 | software_share | 1 | NFS | error | False | dhss_true |
| None |
+-----+-----+-----+-----+-----+-----+-----+
```

12. ファイル共有およびサポートメッセージを削除します。

```
clouduser1@client:~$ manila message-list
+-----+-----+-----+-----+
| ID | Resource Type | Resource ID | Action ID | User |
| Message | Detail ID | Created At |
+-----+-----+-----+-----+
| ed7e02a2-0cdb-4ff9-b64f-e4d2ec1ef069 | SHARE | 2d03d480-7cba-4122-ac9d- |
| edc59c8df698 | 002 | create: Driver does not expect share-network to be provided with |
| current configuration. | 003 | 2018-10-09T21:24:40.000000 |
| 7d411c3c-46d9-433f-9e21-c04ca30b209c | SHARE | 243f3a51-0624-4bdd-950e- |
| 7ed190b53b67 | 001 | allocate host: No storage could be allocated for this share request, |
| Capabilities filter didn't succeed. | 008 | 2018-10-09T21:12:21.000000 |
+-----+-----+-----+-----+
clouduser1@client:~$ manila delete 2d03d480-7cba-4122-ac9d-edc59c8df698 243f3a51-
0624-4bdd-950e-7ed190b53b67
clouduser1@client:~$ manila message-delete ed7e02a2-0cdb-4ff9-b64f-e4d2ec1ef069
7d411c3c-46d9-433f-9e21-c04ca30b209c

clouduser1@client:~$ manila message-list
+-----+-----+-----+-----+
| ID | Resource Type | Resource ID | Action ID | User Message | Detail ID | Created At |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

4.7.2. ファイル共有のマウント失敗のデバッグ

ファイル共有をマウントする際に問題が発生した場合は、以下の検証手順に従って、問題の根本的な原因を特定します。

手順

1. ファイル共有のアクセス制御リストを確認し、クライアントに対応するルールが正しく適用され、正常に適用されていることを確認します。

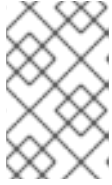
```
$ manila access-list share-01
```

成功したルールでは、**state** 属性は **active** に等しくなります。

2. 共有種別パラメーターが **driver_handles_share_servers=False** に設定されている場合には、エクスポート場所からホスト名または IP アドレスをコピーし、これに ping して NAS サーバーへの接続を確認します。

```
# ping -c 1 172.17.5.13
PING 172.17.5.13 (172.17.5.13) 56(84) bytes of data.
64 bytes from 172.17.5.13: icmp_seq=1 ttl=64 time=0.048 ms--- 172.17.5.13 ping statistics --
```

```
-  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 7.851/7.851/7.851/0.000 ms  
If using the NFS protocol, you may verify that the NFS server is ready to respond to NFS rpcs  
on the proper port:  
# rpcinfo -T tcp -a 172.17.5.13.8.1 100003 4  
program 100003 version 4 ready and waiting
```



注記

IP アドレスは汎用アドレスフォーマット (uaddr) で書かれているので、NFS サービスのポート 2049 を表すのに 2 つのオクテット (8.1) が追加されています。

これらの検証手順が失敗した場合は、ネットワーク接続に問題があるか、共有ファイルシステムのバックエンドストレージが失敗している可能性があります。ログを収集し、Red Hat サポートにお問い合わせください。