Red Hat OpenStack Platform 16.0

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

Last Updated: 2022-09-13

Red Hat OpenStack Platform 16.0 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

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

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

前書き

Red Hat OpenStack Platform は、Red Hat Enterprise Linux をベースとして、プライベートまたはパブリックの Infrastructure-as-a-Service (IaaS) クラウドを構築するための基盤を提供します。これにより、スケーラビリティーが極めて高く、耐障害性に優れたプラットフォームをクラウド対応のワークロード開発にご利用いただくことができます。

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。OpenStack 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。

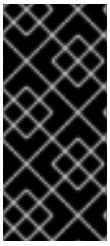
- Block Storage (**openstack-cinder**)
- Object Storage (**openstack-swift**)
- Shared File Systems ストレージ (**openstack-manila**)

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。

OpenStack Dashboard またはコマンドラインクライアント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。大半の手順は、これらのいずれかの方法を使用することができますが、一部の高度な手順はコマンドラインのみで実行可能となっています。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
```

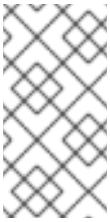
第1章 OPENSTACK での永続ストレージの概要

OpenStack 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。

一方、**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。OpenStack 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。
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、セキュリティーが強化されます。ボリュームに関する情報は、「*2章Block Storage とボリューム*」を参照してください。



注記

一時ストレージは絶対に使用しないように、インスタンスを設定することも可能です。このような場合は、Block Storage サービスによりボリュームにイメージが書き込まれ、そのボリュームはインスタンスのブータブル root ボリュームとして利用することができます。

コンテナー

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、コンテナー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、コンテナー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ー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、クラウド内のプロジェク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。コンテナーは、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。コンテナー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、コンテナー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、コンテナーに比べ、機能セットが幅広く、ネイティブのセキュリティー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. スケーラビリティー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ーラビリティーが高くなっています。Red Hat Ceph を Block Storage のバックエンドとして使用する場合は、Ceph OSD (Object Storage Daemon) 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ーリングできます。Block Storage も Object Storage サービスも、バックエンドとして使用する Red Hat Ceph 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、容量をスケーリングすることができます。Block Storage サービスは、数多くのサポートバックエンドソリューションをサポートしており、その一部には追加のスケーラビリティー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ーリングして、下層にあるブロックストレージで利用可能な容量を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、容量をスケーリングすることも可能です。

Shared File Systems サービスは、別の **ストレージプール** からのストレージでバッキングされるファイル共有をプロビジョニングします。通常はサードパーティーのバックエンドサービスによって管理されるこのプールは、ファイルシステムレベルでストレージをファイル共有に提供します。Shared File Systems サービスでは NetApp および CephFS の両方をサポートしており、事前作成済みのボリューム (プロビジョニングしたファイル共有でストレージとして使用可能) のストレージプールを使用するように設定できます。いずれのデプロイメントにおいても、プールにボリュームを追加してスケーリングを行います。

1.2. アクセシビリティーと管理

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。

コンテナー内のオブジェクトは、API 経由でアクセスでき、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ーとして理想的です。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ー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したボリュームすべてを暗号化する際に使用されます。詳しくは、「ボリューム暗号化の設定」を参照してください。

一方で、オブジェクトとコンテナー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 サービスは、コンテナー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 サービスでは、インスタンスの IP アドレス、ユーザー/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。

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、セキュリティー、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ーとして 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 を使用した Ceph のデプロイ手順については、**『コンテナー化された Red Hat Ceph を持つオーバークラウドのデプロイ』** 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、『オーバークラウドの **高度なカスタマイズ』** の「NFS ストレージの設定」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一覧は、「Component, Plug-In, and Driver Support in Red Hat OpenStack Platform」を参照してください。一部のバックエンドには個別のガイドがあり、Red Hat OpenStack Storage のドキュメントサイト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。

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

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

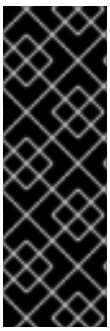
部分的な障害や単一または複数ノードでのハードウェア異常が発生している間に管理レイヤーが維持されます。アクティブ/アクティブのデプロイメントにより、クラスターはノードに障害が発生している間 Block Storage サービスの提供を続けることができます。

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
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、サービスの一覧を表示する。 **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。	**cinder service-list**

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。	**cinder cluster-list** **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。
特定クラスターについての詳細情報を表示する。	**cinder cluster-show <cluster_name>** **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。
無効にしたサービスを有効にする。	**cinder cluster-enable <cluster_name>** **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。
クラスター化したサービスを無効にする。	**cinder cluster-disable <cluster_name>** **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。

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一覧を確認することができます。これらの一覧を表示するには、**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一覧
- 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一覧

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
```

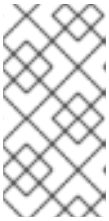
あるいは、**cluster show** コマンドを実行します。

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、一覧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」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「バックエンド間の移行」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一覧表示

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一覧については、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一覧を返します。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                  |
```

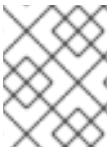
```
| volume_backend_name |                    lvm            |
+---------------------+-------------------------------------+
+---------------------+-------------------------------------+
| Backend properties |                   Value             |
+---------------------+-------------------------------------+
|    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一覧が、**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. **名前** フィールドにボリューム種別の名前を入力します。
4. **ボリューム種別の作成** をクリックします。**ボリューム種別** の表に新しい種別が表示されます。
5. 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。
6. **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. **作成** 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 の全プロジェク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の関連付け」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、全プロジェク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、特定のプロジェク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一覧表示します。一覧には、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、プロジェクトレベルで許可されます。プロジェク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。

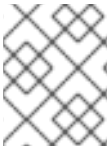
```
$ cinder  type-access-add --volume-type <TYPE-ID> --project-id <TENANT-ID>
```

プライベートのボリューム種別にアクセス可能なプロジェクトを確認するには、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プロジェクトを削除するには、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 サービス用の内部プロジェク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/プロジェクトを使用して、通常の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 Project" 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          |
   +----------+----------------------------------+
```

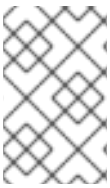
新たな設定オプションを追加する手順により、内部プロジェクトが作成されます。「Image-Volume キャッシュの設定および有効化」を参照してください。

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 サービス用の内部プロジェク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: 1
        value: True
      BACKEND/image_volume_cache_max_size_gb:
        value: MAXSIZE 2
      BACKEND/image_volume_cache_max_count:
        value: MAXNUMBER 3
```

1. **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)。
2. 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す。**MAXSIZE** を GB 単位の数値に変更します。
3. **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 コマンドに関する補足情報は、『**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

種別	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. **作成** 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。
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、1 GB 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 スペックと同様に構成されます。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にのみ適用されます。

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。

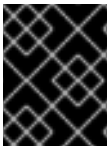
2.2.5.5. ボリューム種別からの QoS スペックの関連付け解除

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. ボリューム暗号化の設定

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ーを侵害されたり、完全に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活用するには、Barbican をデプロイする必要があります。



重要

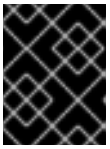
現在、ボリュームの暗号化は、ファイルベースのボリューム (例: NFS) ではサポートされていません。

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構成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。

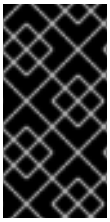
- **cinder_encryption_key_id**: Key Management サービスが特定のイメージ用に保存する暗号鍵の識別子

- **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ジ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ー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. アベイラビリティーゾーンのデプロイ

アベイラビリティーゾーンは、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 ボリュームのバックエンドごとに異なるアベイラビリティー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

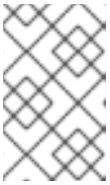
Block Storage ボリュームのバックエンドごとに異なるアベイラビリティーゾーンをデプロイするには、以下の手順を実施します。

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ー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/ ディレクトリー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 サービスでは、**整合性グループ** を設定することができます。これにより、複数のボリュームを単一のエンティティーとしてグループ化することができます。その結果、複数のボリュームに対する操作を、個別に実施せずに 1 度に実施することができます。特に本リリースでは、整合性グループを使用して、複数ボリュ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。その延長で、これらのボリュームを同時に復元またはクローン作成することも可能です。

1 つのボリュ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、ボリュームを整合性グループに追加したら、そのボリュームを削除、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 (**openstack-cinder-api**) をホストするノードの **/etc/cinder/policy.json** の関連整合性グループエントリーに、デフォルト設定の一覧が表示されます。

-

```
"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",
```

これらの設定を環境ファイルで変更し、**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すると、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ます。

整合性グループを有効にするには、環境ファイルを編集して、**parameter_defaults** セクションに新規エントリーを追加します。これにより、エントリーがコンテナーで更新され、**openstack overcloud deploy** コマンドを使用して director で環境を再デプロイするたびにエントリーが維持されるようになります。

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 ファイルからのデフォルト設定を示す同等の parameter_defaults セクションは、以下のようになります。

```
parameter_defauts:
  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' }, \
  }
```

'group:nobody' の値により、どのグループもこの機能を使用できないことが定義され、結果的にその機能が無効化されます。この機能を有効にするには、グループを別の値に変更する必要があります。

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 **"rule:admin_or_owner"** に設定されています (同じ **/etc/cinder/policy.json** ファイルで)。

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

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_cgsnapshot: { key: 'consistencygroup:create_cgsnapshot', value:
'rule:admin_or_owner' }, \
    cinder-consistencygroup_delete_cgsnapshot: { key: 'consistencygroup:delete_cgsnapshot', value:
'rule:admin_or_owner' }, \
    cinder-consistencygroup_get_cgsnapshot: { key: 'consistencygroup:get_cgsnapshot', value:
'rule:admin_or_owner' }, \
    cinder-consistencygroup_get_all_cgsnapshots: { key: 'consistencygroup:get_all_cgsnapshots',
value: 'rule:admin_or_owner' }, \
 }
```

/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ます。続いて、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 コマンドに関する補足情報は、**『director のインストールと** 使用方法 』の「CLI ツールを使用したオーバークラウドの作成 」セクションを参照してください。

2.2.9.2. 整合性グループの作成および管理

整合性グループの API を有効にした後には、整合性グループの作成を開始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 **アベイラビリティーゾーン** 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** 一覧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。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るはずです。

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。

さらに、整合性グループにボリュームを直接追加または削除することもできます。そのためには、以下の手順を実施します。

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. 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** 一覧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** 一覧から削除するボリュームの - ボタンをクリックします。
3. **整合性グループの編集** をクリックします。

2.2.9.3. 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後に、そこからスナップショットを作成することができます。その前に、まず **openstack-cinder-api** をホストするノード上のコマンドラインから **admin** としてログインして、以下のコマンドを実行します。

```
# export OS_VOLUME_API_VERSION=2
```

このコマンドを実行すると、クライアントが **openstack-cinder-api** のバージョン **2** を使用するように設定されます。

利用可能な整合性グループおよびその ID (後で必要となります) をすべて表示するには、以下のコマンドを実行します。

```
# cinder consisgroup-list
```

整合性グループを使用してスナップショットを作成するには、以下のコマンドを実行します。

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

各オプションについての説明は以下のとおりです。

- **CGSNAPNAME** はスナップショットの名前に置き換えてください (任意)。
- **DESCRIPTION** はスナップショットの説明に置き換えてください (任意)。
- **CGNAMEID** は整合性グループの名前または ID に置き換えてください。

利用可能な整合性グループのスナップショットの全一覧を表示するには、以下のコマンドを実行します。

```
` # cinder cgsnapshot-list `
```

2.2.9.4. 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。

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"
```

上記のコマンドで、**CGNAMEID** はクローン作成する整合性グループの名前または ID に、**CGNAME** は整合性グループの名前 (任意) に、**DESCRIPTION** は整合性グループの説明 (任意) に置き換えてください。

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。

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	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) 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。

フィールド	説明
容量 (GB)	ボリュームの容量 (ギガバイト単位)
アベイラビリティーゾーン	アベイラビリティーゾーン (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。アベイラビリティーゾーンは、インストール中に定義されます。アベイラビリティーゾーンとホスト 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管理 」を参照してください。

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)	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。
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** の一覧が新たに表示され、スナップショットを選択できるようになります。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** の一覧が新たに表示され、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** の一覧が新たに表示され、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

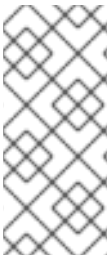
複数の Block Storage バックエンドが設定された場合には、必ず、バックエンドごとにボリューム種別を作成する必要があります。その種別を使用して、作成したボリュームに、どのバックエンドを使用すべきかを指定することができます。ボリューム種別の詳しい情報は、「グループボリューム設定とボリューム種別」を参照してください。

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」のドロップダウン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。詳しくは「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に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、「ボリューム暗号化の設定」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. ボリュームを一覧表示し、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、以下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 および新しいサイズ (現在のサイズより大きい値) をパラメーターとして渡します。

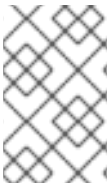
```
$ OS_VOLUME_API_VERSION=<API microversion>
$ cinder extend <volume ID> <size>
```

<API microversion>、<volume ID>、および <size> を適切な値に置き換えます。以下の例を目安にしてください。

```
$ 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。インスタンスに関する詳しい情報は、『インスタンス **&イメージガイド』の「インスタンスの** 管理 **」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** の一覧から、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。この機能は、Block Storage (cinder) のバックエンドに Ceph を使用する場合に機能します。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。それ以外の構成では、データの破損が生じます。

警告

オプションのマルチ接続機能を使用するには、cinder ドライバーがこの機能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の認定についての詳しい情報は、以下のアーティクルおよび Web サイトを参照してください。

- https://access.redhat.com/articles/1535373#Cinder
- https://access.redhat.com/ecosystem/search/#/category/Software?sort=sortTitle%20asc&softwareCategories=Storage&ecosystem=Red%20Hat%20C

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。

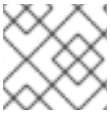
警告

マルチ接続ボリュームの制約は以下のとおりです。

- 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。
- マルチ接続のボリュームでは、ボリュームの暗号化はサポートされません。
- 接続済みボリュームの種別をマルチ接続から非マルチ接続に (あるいは非マルチ接続からマルチ接続に) 変更することはできません。
- 読み取り専用のマルチ接続はサポートされません。
- 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。

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。



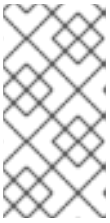
注記

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接続する方法は、「インスタンスへのボリュームの 接続解除」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。そのためには、以下のコマンドを実行して、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、ボリュームの譲渡を行います。ボリュームの譲渡は、ボリュームの所有者が開始し、ボリュームの新しい所有者が譲渡を承認すると、そのボリュームの所有権の変更が完了します。

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一覧表示します。

   ```
   # cinder list
   ```

3. 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。

4. 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たボリュームの譲渡

Dashboard を使用したボリューム譲渡の作成

1. 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。
2. 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。
3. **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、ボリュームの一覧に戻ります。
   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一覧に、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。また、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ぐ目的で使用されるのに対してスナップショットはクローン作成を円滑に行う目的で使用されます。

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ーは通常、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. 対象のボリュームの **スナップショットの作成** アクションを選択します。
3. 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。この操作を行うには、対象のスナップショットの **ボリュームの作成** 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。

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。

関連する手順については、『Red Hat Ceph Storage 1.2.3 Ceph Block Device』の「Protecting a Snapshot」および「Unprotecting a Snapshot」を参照してください。

2.3.11. 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、一覧から **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。イメージの使用方法や設定方法に関する情報は、Red Hat OpenStack Platform の製品ドキュメント から『**Instances and Images Guide**』の「**Manage Images**」を参照してください。

2.3.12. ボリューム種別の変更

ボリューム種別の変更 は、ボリューム種別 (およびその設定) を既存のボリュームに適用するプロセスです。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。

既存のボリューム種別の有無に拘らず、ボリューム種別は変更できます。いずれの場合も、ボリューム種別の追加スペックがボリュームに適用できる場合のみ、ボリューム種別の変更は成功します。ボリューム種別の変更は、事前定義の設定やストレージの属性を既存のボリュームに適用する際に役立ちます。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 (「バックエンド間の移行」)
- ボリュームのストレージクラス/層を変更する場合

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種別しか変更できません。ボリューム種別を変更するには、以下のステップを実行します。

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. 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** のドロップダウン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** のドロップダウンリストから **要求時** を選択します。詳しくは、「バックエンド間の移行」を参照してください。

注記

本リリースでは、2 つの異なるバックエンド種別間でボリューム種別を変更する操作はサポートされません。

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。

2.4. 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。

2.4.1. ボリュームの移行

Block Storage サービスでは、同一または異なるアベイラビリティーゾーン (AZ) にあるホストまたはバックエンド間で、ボリュームを移行することができます。ボリュームの移行には、一部制限があります。

- 使用中のボリュームの移行は、ドライバーがサポートしているかどうかによります。
- スナップショットのあるボリュームは移行できません。
- 使用中のボリュームの移行のターゲットには、ISCSI またはファイバーチャネル (FC) のブロックバックエンドデバイスが必要で、Ceph RADOS Block Device (RBD) などの非ブロックデバイスは使用できません。

移行のスピードは、ホストの設定と構成によって異なります。ドライバーを使用した移行の場合は、データの移動は OpenStack Block Storage サービス内ではなく、ストレージバックプレーンで実行されます。ボリューム種別の変更が必要なければ、使用中ではない RBD ボリュームで、ドライバーを使用する最適なコピー操作を利用することができます。そうでない場合には、データは Block Storage サービスを使用してホスト間でコピーされます。

2.4.1.1. ホスト間の移行

ホスト間でボリュームを移行する場合には、両方のホストが同じバックエンド上にある必要があります。ダッシュボードを使用してホスト間でボリュームを移行するには、以下の手順を実施します。

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. バックエンド間の移行

バックエンド間でボリュームを移行するには、ボリューム種別の変更 を行う必要があります。新規バックエンドに移行するには、以下が True である必要があります。

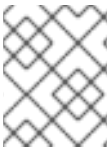
- 新規バックエンドは、別の 移行先のボリューム種別 の 追加スペック として指定する必要があります。
- 移行先のボリューム種別に定義されるその他の追加スペックはすべて、そのボリュームの元のボリューム種別と互換性がある必要があります。

詳細は以下を参照してください。

- 「グループボリューム設定とボリューム種別」
- 「ボリュームを作成するバックエンドの指定」

追加スペックとしてバックエンドを定義する場合は、キー として **volume_backend_name** を使用します。これに対応する値は、Block Storage 設定ファイル **/etc/cinder/cinder.conf** で定義されるバックエンドの名前です。このファイルでは、各バックエンドは独自のセクションに定義され、対応する名前は **volume_backend_name** パラメーターで設定されます。

移行先のボリューム種別にバックエンドを定義したら、ボリューム種別の変更 の手順を使用して、バックエンドにボリュームを移行することができます。この際には、移行先のボリューム種別をボリュームに再適用し、それにより新しいバックエンドの設定が適用されます。詳細は、「「ボリューム種別の変更」」を参照してください。

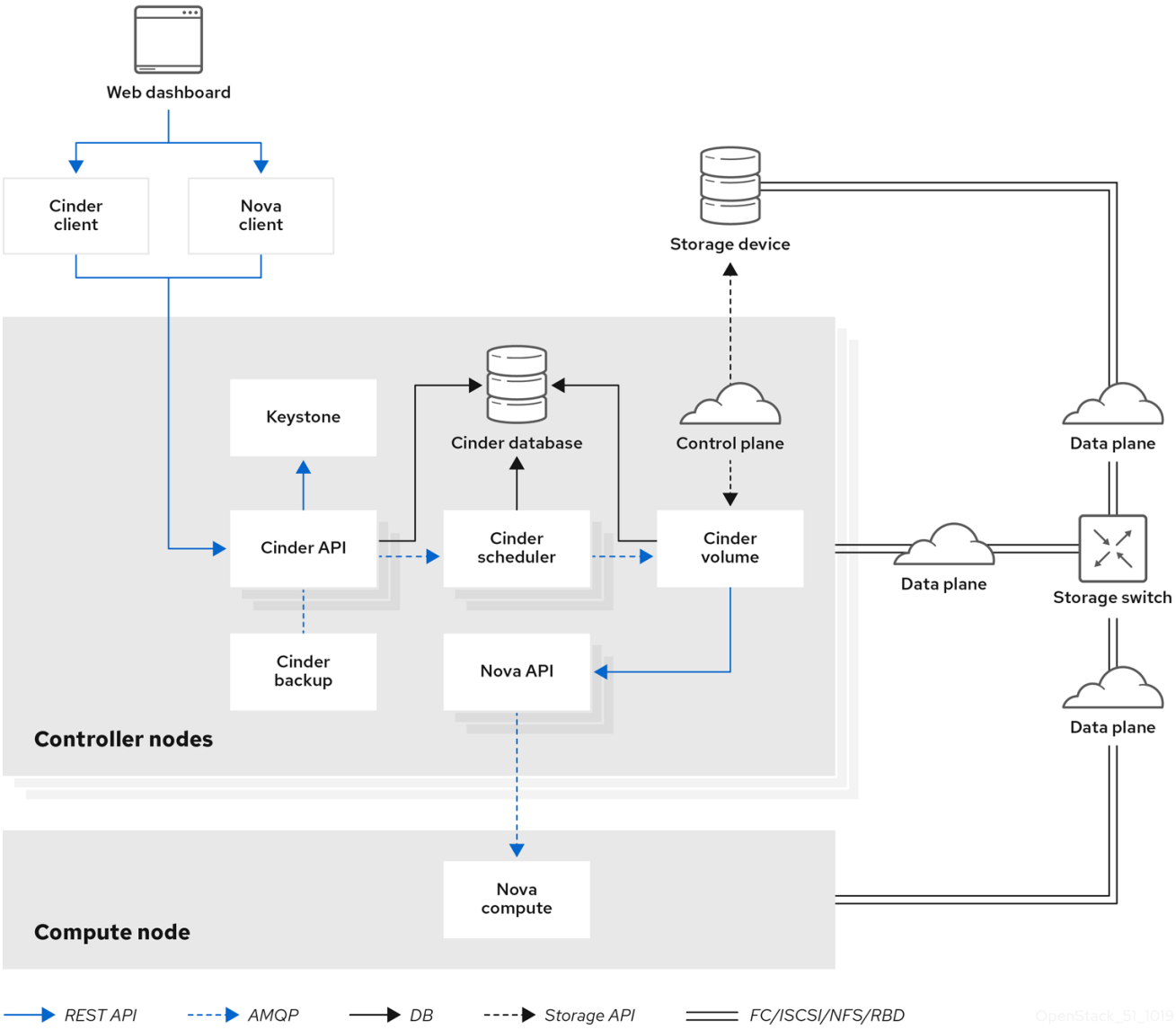


注記

本リリースでは、2 つの異なるバックエンド種別間でボリューム種別を変更する操作はサポートされません。

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。

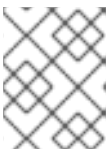
既存のオーバークラウドデプロイメントでマルチパスを設定する方法は、「既存デプロイメントでのマルチパス設定」を参照してください。

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トノードは、Red Hat Enterprise Linux Server のリポジトリーにアクセスできる必要があります。詳しくは、『**director のインストールと使用方法**』の「**ベースのクラウドイメージ** のダウンロード」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。

2. 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
```

3. (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** 構成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** heat テンプレート両方のデフォルト値が一致するようにしてください。

4. 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
```

5. 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** オーバークラウドの作成」を参照してください。

6. コンテナーを実行する前に、すべてのコントローラーノードおよびコンピュート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ト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ので、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、『DM Multipath』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ト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/" /etc/multipath.conf
```

注記

skip_kpartx を **yes** に設定すると、コンピュートノード上の 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。設定の属性についての詳細は、**『DM Multipath』**の「設定ファイルの multipaths セクション」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ト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

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には、以下の手順を実施します。

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トノードは、Red Hat Enterprise Linux Server のリポジトリーにアクセスできる必要があります。詳しくは、『**director のインストールと使用方法**』の「**ベースのクラウドイメージ** のダウンロード」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ト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ト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トノードでマルチパスデーモンを設定します。

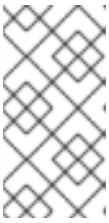
```
$ mpathconf --enable --with_multipathd y --user_friendly_names n --find_multipaths y
```

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ので、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、『DM Multipath』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ト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/" /etc/multipath.conf
```



注記

skip_kpartx を **yes** に設定すると、コンピュートノード上の 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。設定の属性についての詳細は、**『DM Multipath』**の「設定ファイルの multipaths セクション」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ト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** 構成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

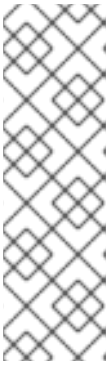
   default_backend と **GlanceBackendID** heat テンプレート両方のデフォルト値が一致するようにしてください。

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 オーバークラウドの作成」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ーにログインします。

   ```
   $ 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。

第3章 OBJECT STORAGE サービス

OpenStack Object Storage (**swift**) は、コンテナー内にそのオブジェクト (データ) を格納します。コンテナーは、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。コンテナーは、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) コンテナー用、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を比較して、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 -arqlT --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す。
- 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す。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、コンテナー、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 プロキシーサービスの **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、セキュリティーリスクとなってしまいます。

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。詳細は、「 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。『オーバークラウド の高度なカスタマイズ』の「 **roles_data** ファイル の作成」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、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 サービスオファ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**) を使用する方法の例を、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. 基本的なコンテナー管理

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、画像を保管する **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ーを作成することができます。また、各コンテナーには、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。

3.3.1. コンテナーの作成

1. Dashboard で **プロジェクト > オブジェクトストア > コンテナー**を選択します。
2. **コンテナーの作成** をクリックします。
3. **コンテナー名** を指定して、**コンテナーアクセス** 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ーおよびデータは表示されません。

4. **コンテナーの作成** をクリックします。

新しいコンテナー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ー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、コンテナー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、コンテナーの名前に置き換えてください。

3.3.2. コンテナー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** > **コンテナー**を選択します。
2. 擬似フォルダーを追加するコンテナーの名前をクリックします。
3. **疑似フォルダーの作成** をクリックします。
4. **疑似フォルダー名** フィールドに名前を指定し、**作成** をクリックします。

3.3.3. コンテナーの削除

1. Dashboard で **プロジェクト** > **オブジェクトストア** > **コンテナー**を選択します。
2. **コンテナー** のセクションの一覧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. 対象のコンテナーの矢印メニューで **コンテナーの削除** を選択します。
4. **コンテナーの削除** をクリックして、コンテナーを削除する操作を確定します。

3.3.4. 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 > コンテナー**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ーの名前をクリックします。そのコンテナー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。

3.3.5. 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 > コンテナー**を選択します。
2. オブジェクトのコンテナー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
宛先コンテナー	新規プロジェクトの宛先コンテナー
パス	宛先コンテナーの擬似フォルダー。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。

3.3.6. 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 > コンテナー**を選択します。
2. 一覧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

OpenStack 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。

4.1. 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は、サポートされる 2 つのバックエンドのいずれかを選択することができます。

- 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
- NetApp

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一覧は、「Component, Plug-In, and Driver Support in Red Hat OpenStack Platform」を参照してください。

4.2. ファイル共有の作成と管理

Shared File Systems サービスを使用して、ファイル共有の作成および管理を行います。

4.2.1. 共有種別の作成

共有種別は、ファイル共有のストレージバックエンドを選択するのに使用されます。Red Hat OpenStack Platform 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。

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 を使用する構成では、値は **false** です。
- NetApp バックエンドの場合には、値は **true** または **false** です。**ManilaNetappDriverHandlesShareServers** 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。詳しくは、『NetApp Back End Guide for the Shared File Systems service』を参照してください。

2. デフォルトの共有種別に仕様を追加するか、または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、CephFS バックエンドを選択する **default** 共有種別および NetApp (**driver_handles_share_servers=True**) バックエンドを選択する追加の共有種別を設定するには、以下のコマンドを実行します。

```
(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。

4.2.2. ファイル共有の作成

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
```

詳細は以下のようになります。

- **SHARETYPE** で指定した共有種別に関連する設定が適用されす。
  - 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、ファイル共有名に置き換えます。
  - 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、使用するファイル共有プロトコルに置き換えます。
  - CephFS を使用した NFS の場合には、PROTO は **nfs** です。
  - NetApp の場合には、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、「ファイル共有の作成と管理」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。

```
(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
```

4.2.3. ファイル共有とエクスポート情報の一覧表示

ファイル共有が正常に作成されたことを確認するには、以下の手順を実施します。

1. 以下のコマンドを実行します。

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

+--------------------------------------+----------+-----+-----------+          | ID                                   |
Name     | ... | Status    ...
+--------------------------------------+----------+-----+-----------+
| 8c3bedd8-bc82-4100-a65d-53ec51b5fe81 | share-01 | ... | available ...
+--------------------------------------+----------+-----+-----------+
```

2. **manila share-export-location-list** コマンドを実行して、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
 +-----------------------------------------------------------------
```

注記

この情報は、「IPv4 ネットワークでの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。

4.2.4. ファイル共有へのネットワーク接続の確保

Shared File Systems サービスは、ネットワークを介してストレージを提供します。したがって、ファイル共有をマウントする予定のコンピュートインスタンスには、そのファイル共有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 OpenStack の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（詳しくは『 Networking requirements for manila 』を参照）を含め多数あります。

driver_handles_share_servers=True と設定したバックエンドの場合には、クラウドユーザーはコンピュートインスタンスのアタッチ先となる ネットワークの詳細を使用して ファイル共有 ネットワークを作成し、ファイル共有の作成時にこのネットワークを参照することができます。

- **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 Shared File Systems 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 を使用する構成では、クラウド管理者は分離ネットワークおよび環境引数を使用して OpenStack director をデプロイし（詳しくは『NFS バックエンドに CephFS を使用した場合のガイド』の「NFS バックエンドに CephFS を使用した Red Hat OpenStack Platform のインストール」および「カスタム network_data ファイル を使用した Red Hat OpenStack Platform のインストール」を参照）、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。

注記

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。

4.2.4.1. IPv4 ネットワーク接続の確保

ファイル共有への IPv4 ネットワーク接続を確保するには、以下の手順を実施します。

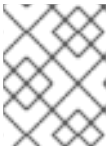
1. StorageNFS ポート用のセキュリティーグループを作成し、(NFS マウントの開始に必要な) ポートから外部に出るパケットは許可する一方、未確立接続の着信パケットは拒否します。

```
(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

コンピュートインスタンスがこのアドレスのインターフェー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。

4.2.4.2. IPv6 ネットワーク接続の確保

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、実稼働環境にデプロイすべきではありません。テクノロジープレビュー機能についての詳しい情報は、「対象範囲の詳細」を参照してください。

ファイル共有への IPv6 ネットワーク接続を確保するには、以下の手順を実施します。

1. StorageNFS ポート用のセキュリティーグループを作成し、(NFS マウントの開始に必要な) ポートから外部に出るパケットは許可する一方、未確立接続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。

```
$ 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: '2019-10-08T16:05:12Z'
data_plane_status: null
description: ''
device_id: ''
device_owner: ''
dns_assignment: null
dns_name: null
extra_dhcp_opts: ''
fixed_ips: ip_address='fd00:fd00:fd00:7000::c', subnet_id='ac5ad772-cd4a-4bb9-876b-
2ceb77762209'
id: 0561cb6b-9aa6-40d3-8b74-b8232c34af9f
ip_address: null
mac_address: fa:16:3e:44:32:46
name: nfs-port0
network_id: d41dcb27-3a3c-44f7-aebb-2e7b9b579ee6
option_name: null
option_value: null
port_security_enabled: true
project_id: 2e9876e230ab4bf5b0170d7f41cc8c92
qos_policy_id: null
revision_number: 6
security_group_ids: 9a3255df-0b54-4c0a-ba39-a6c192f282af
status: DOWN
subnet_id: null
tags: ''
trunk_details: null
updated_at: '2019-10-08T16:05:12Z'
```

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

```
$ openstack server add port demo-instance-0 nfs-port0
$ openstack server list -f yaml
- Flavor: m1.small
  ID: cb35cdaf-d8fa-4d7d-8b58-0f4fb8edd3f9
  Image: fedora27
  Name: demo-instance-0
  Networks: StorageNFS=fd00:fd00:fd00:7000::c; demo-net=172.20.0.7, 10.0.0.228
  Status: ACTIVE
```

4. 以下のコマンドを実行して、StorageNFS ネットワークサブネットからの IPv6 アドレスがコンピュートインスタンスに割り当てられていることを確認します。

```
$ openstack server list
```

4.2.5. ネットワークとインスタンス間の IPv6 インターフェースの設定

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、実稼働環境にデプロイすべきではありません。テクノロジープレビュー機能についての詳しい情報は、「対象範囲の詳細」を参照してください。

1. インスタンスにログインします。
2. 以下のコマンドを実行して、StorageNFS ネットワークとインスタンス間の IPv6 インターフェースを作成します。

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

3. 以下のコマンドを実行して、インターフェースの接続性をテストします。

```
$ ping -6  fd00:fd00:fd00:7000::21
$ dnf install -y telnet
$ telnet fd00:fd00:fd00:7000::21 2049
```

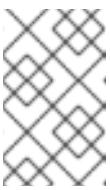
4.2.6. ファイル共有に対するアクセス権限の付与

インスタンス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、インスタンス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>
```

詳細は以下のようになります。

- **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。



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。

- **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** の設定により異なります。
  - **ACCESSTYPE** を **ip** に設定した場合には、IP アドレスを使用します。
  - **ACCESSTYPE** を **user** に設定した場合には、CIFS ユーザーまたはグループを使用します。

4.2.6.1. IPv4 ネットワーク上のファイル共有に対するアクセス権限の付与

IPv4 ネットワーク上のファイル共有へのアクセス権限を付与するには、以下の手順を実施します。

1. StorageNFS ネットワークのポート (IP アドレス: 172.17.5.160) が割り当てられたコンピュートインスタンスに **share-01** への読み取り/書き込みアクセス権限を付与するには、以下のコマンドを実行します。

```
(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. 正常にアクセス設定が行われたことを確認するには、以下のコマンドを実行します。

```
(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 | ...
+--------------+-------------+--------------+--------------+--------+ ...
```

4.2.6.2. IPv6 ネットワーク上のファイル共有に対するアクセス権限の付与

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、実稼働環境にデプロイすべきではありません。テクノロジープレビュー機能についての詳しい情報は、「対象範囲の詳細」を参照してください。

IPv6 ネットワーク上のファイル共有へのアクセス権限を付与するには、以下の手順を実施します。

1. StorageNFS ネットワークのポート (IP アドレス: **fd00:fd00:fd00:7000**) が割り当てられたコンピュートインスタンスに **share-01** への読み取り/書き込みアクセス権限を付与するには、以下のコマンドを実行します。

```
(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=fd00:fd00:fd00:7000
  Status: ACTIVE

(user) [stack@undercloud-0 ~]$ manila access-allow share-01 ip fd00:fd00:fd00:7000
```

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       | fd00:fd00:fd00:7000
| access_level    | rw
| state           | queued_to_apply
| id              | 875c6251-c17e-4c45-8516-fe0928004fff
+-----------------+--------------------------------------+
```

2. 正常にアクセス設定が行われたことを確認するには、以下のコマンドを実行します。

```
(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 | ...
+--------------+-------------+--------------+--------------+---------+ ...
```

4.2.7. ファイル共有へのアクセスの取り消し

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。

1. 以下のコマンドを実行します。

```
# manila access-deny <SHARE> <ACCESSID>
```

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。

```
(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  | ...
+--------------+------------+--------------+--------------+--------+ ...
+--------------+------------+--------------+--------------+--------+ ...
```

4.3. インスタンスへのファイル共有のマウント

インスタンスを認証するようにファイル共有を設定したら、環境の機能を確認し、続いてファイル共有をマウントします。

注記

ファイル共有をマウントするコンピュートインスタンスに、バージョン 4.1 対応の NFS 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。

4.3.1. 環境の確認

環境の機能を確認するには、以下の手順を実施します。

1. 以下のコマンドを実行して、NFS-Ganesha サービスの仮想 IP (VIP) を取得します。

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

2. ファイル共有をマウントする仮想マシンから、ping から VIP にアクセスできることを確認します。

```
# ping 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
64 bytes from 172.17.5.13: icmp_seq=2 ttl=64 time=0.061 ms
^C
--- 172.17.5.13 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.048/0.054/0.061/0.009 ms
```

3. VIP が適切なポートの NFS rpc に応答する準備ができていることを確認します。

```
# rpcinfo -T tcp -a 172.17.5.13.8.1 100003 4
```

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。

4.3.2. IPv4 ネットワークでのファイル共有のマウント

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」

ファイル共有をマウントするには、以下の手順を実施します。

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. 「ファイル共有とエクスポート情報の一覧表示」で取得したエクスポート場所を使用して、ファイル共有をマウントします。

```
# mount.nfs -v  172.17.5.13:/volumes/_nogroup/e840b4ae-6a04-49ee-9d6e-67d4999fbc01
/mnt mount.nfs: timeout set for Wed Sep 19 09:14:46 2018                mount.nfs: trying
text-based options 'vers=4.2,addr=172.17.5.13,clientaddr=172.17.5.160'
172.17.5.13:/volumes/_nogroup/e840b4ae-6a04-49ee-9d6e-67d4999fbc01 on /mnt type nfs
# mount | grep mnt       172.17.5.13:/volumes/_nogroup/e840b4ae-6a04-49ee-9d6e-
67d4999fbc01 on /mnt type nfs4
(rw,relatime,vers=4.2,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,port=0,timeo
=600,retrans=2,sec=sys,clientaddr=172.17.5.160,local_lock=none,addr=172.17.5.13)
```

4.3.3. IPv6 ネットワークでのファイル共有のマウント

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、実稼働環境にデプロイすべきではありません。テクノロジープレビュー機能についての詳しい情報は、「対象範囲の詳細」を参照してください。

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」

ファイル共有をマウントするには、以下の手順を実施します。

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. 「ファイル共有とエクスポート情報の一覧表示」で取得したエクスポート場所を使用して、ファイル共有をマウントします。

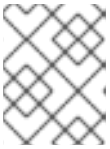
```
# mount.nfs -v [fd00:fd00:fd00:7000::21]:/volumes/_nogroup/74bc2c33-151a-469e-8e8e-d4ffabc9a477 /mnt
mount.nfs: timeout set for Mon Oct 14 14:59:37 2019
mount.nfs: trying text-based options
'vers=4.2,addr=fd00:fd00:fd00:7000::21,clientaddr=fd00:fd00:fd00:7000::c'
```

4.3.4. ファイル共有の削除

ファイル共有を削除するには、以下の手順を実施します。

1. 次のコマンドを実行します。

```
# 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
```

4.4. SHARED FILE SYSTEMS サービスのクォータ

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。プロジェクトまたはユーザーに設定されたクォータの一覧を表示するには、manila quota-show コマンドを使用します。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。

使用方法に関する説明を表示するには、以下のコマンドを使用します。

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

4.5. 非同期障害のトラブルシューティング

ファイル共有の作成 または **ファイル共有グループの作成** などの manila 操作が非同期に失敗する場合、コマンドラインを使用してエラーに関する詳細情報のクエリーを行うことができます。

4.5.1. シナリオ

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。

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** コマンドを使用します。--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. メッセージの詳細情報を表示するには、**message-list** コマンドで取得したメッセージの ID を指定して、**message-show** コマンドを使用します。

```
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把握しているので、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 |
+--------------------------------------+-------------+-----------------------------------+-----------+---
-----------------------------------------------------------------------------------------------+--------
---+---------------------------+
```

サービスが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ないことが分かります。

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するには、**manila list** コマンドを使用します。

```
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      |      | nova              |
| 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              |
+--------------------------------------+---------------+------+-------------+-----------+-----------+-------
----------+------+-------------------+
```

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 |
+----+--------------+-------------+-----------+-------------+-----------+------------+
+----+--------------+-------------+-----------+-------------+-----------+------------+
```