Red Hat OpenStack Platform 17.1

Configuring persistent storage

OpenStack Block Storage、Object Storage、および Shared File Systems サービスの設定と管理

Last Updated: 2024-09-06

Red Hat OpenStack Platform 17.1 Configuring persistent storage

OpenStack Block Storage、Object Storage、および Shared File Systems サービスの設定と管理

OpenStack Team
rhos-docs@redhat.com

法律上の通知

概要

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

目次

多様性を受け入れるオープンソースの強化 4

RED HAT ドキュメントへのフィードバック (英語のみ) 5

第1章 RED HAT OPENSTACK PLATFORM (RHOSP) での永続ストレージの概要 6
1.1. スケーラビリティーおよびバックエンドストレージ 7
1.2. ストレージへのアクセシビリティーと管理 7
1.3. ストレージのセキュリティー 8
1.4. ストレージの冗長性と障害復旧 8

第2章 BLOCK STORAGE サービス (CINDER) の設定 10
2.1. BLOCK STORAGE サービスのバックエンド 10
2.2. BLOCK STORAGE ボリュームサービスの高可用性 13
2.3. ボリューム種別によるボリューム設定のグループ化 16
2.4. BLOCK STORAGE サービス (CINDER) の内部プロジェクトの作成および設定 23
2.5. IMAGE-VOLUME キャッシュの設定 23
2.6. BLOCK STORAGE サービス (CINDER) のサービス品質仕様 25
2.7. BLOCK STORAGE サービス (CINDER) ボリュームの暗号化 35
2.8. BLOCK STORAGE ボリュームのバックエンド用のアベイラビリティーゾーンのデプロイ 38
2.9. BLOCK STORAGE サービス (CINDER) の整合性グループ 39
2.10. デフォルトの BLOCK STORAGE スケジューラーフィルターの設定 43
2.11. オーバークラウドノードでの LVM2 フィルターの有効化 44
2.12. マルチパス設定 46

第3章 BLOCK STORAGE サービス (CINDER) を使用した基本的な操作の実行 50
3.1. BLOCK STORAGE ボリュームの作成 50
3.2. ボリュームの名前と説明の編集 52
3.3. BLOCK STORAGE サービスボリュームのサイズ変更 (拡張) 52
3.4. BLOCK STORAGE サービスボリュームの削除 53
3.5. 複数のバックエンドのボリューム割り当て 54
3.6. インスタンスへのボリュームの接続 54
3.7. インスタンスからのボリュームの切断 55
3.8. ボリュームへのアクセス権の設定 55
3.9. DASHBOARD を使用したボリューム所有者の変更 56
3.10. CLI を使用したボリューム所有者の変更 57

第4章 BLOCK STORAGE サービス (CINDER) を使用した高度な操作の実行 59
4.1. ボリュームスナップショットの作成 59
4.2. スナップショットからの新しいボリュームの作成 60
4.3. ボリュームスナップショットの削除 61
4.4. スナップショットからのボリュームの復元 61
4.5. IMAGE サービス (GLANCE) へのボリュームのアップロード 63
4.6. 複数のインスタンスに接続できるボリューム 63
4.7. バックエンド間でのボリュームの移動 66
4.8. BLOCK STORAGE のボリューム種別の変更 68
4.9. DASHBOARD を使用したバックエンド間でのボリュームの移行 70
4.10. CLI を使用したバックエンド間でのボリュームの移行 71
4.11. ボリュームとそのスナップショットの管理と管理解除 72
4.12. 暗号化されていないボリュームの暗号化 73
4.13. 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 75

第5章 OBJECT STORAGE サービス (SWIFT) の設定 76
5.1. オブジェクトストレージリング 76

5.2. OBJECT STORAGE サービスのカスタマイズ 79
5.3. OBJECT STORAGE ノードの追加または削除 84
5.4. OBJECT STORAGE サービスにおけるコンテナー管理 92

第6章 SHARED FILE SYSTEMS サービス (MANILA) の設定 96
6.1. SHARED FILE SYSTEMS サービスバックエンドの設定 96
6.2. 共有タイプの作成 104
6.3. 共有タイプの共通機能の比較 105
6.4. 管理/管理解除を使用した共有の追加と削除 106
6.5. 共有ファイルシステムのネットワークの計画 106
6.6. ファイル共有へのネットワーク接続の確保 107
6.7. SHARED FILE SYSTEMS サービスのデフォルトクォータの変更 108

第7章 SHARED FILE SYSTEMS サービスを使用した操作の実行 (MANILA) 114
7.1. 共有タイプのリスト 114
7.2. NFS、CEPHFS、または CIFS 共有の作成 114
7.3. ファイル共有とエクスポート情報のリスト表示 116
7.4. 共有ファイルシステムでのデータのスナップショット作成 117
7.5. 共有にアクセスするための共有ネットワークへの接続 120
7.6. ネットワークとインスタンス間の IPV6 インターフェイスの設定 121
7.7. エンドユーザークライアントに共有アクセスを許可する 122
7.8. コンピューティングインスタンスでの共有のマウント 126
7.9. 共有の削除 128
7.10. SHARED FILE SYSTEMS サービスのリソース制限のリスト表示 128
7.11. 操作エラーのトラブルシューティング 129

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、用語の置き換えは、今後の複数のリリースにわたって段階的に実施されます。詳細は、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

RED HAT ドキュメントへのフィードバック (英語のみ)

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

Jira でドキュメントのフィードバックを提供する

問題の作成 フォームを使用して、Red Hat OpenStack Services on OpenShift (RHOSO) または Red Hat OpenStack Platform (RHOSP) の以前のリリースのドキュメントに関するフィードバックを提供します。RHOSO または RHOSP ドキュメントの問題を作成すると、その問題は RHOSO Jira プロジェクトに記録され、フィードバックの進行状況を追跡できるようになります。

問題の作成 フォームを完了するには、Jira にログインしていることを確認してください。Red Hat Jira アカウントをお持ちでない場合は、https://issues.redhat.com でアカウントを作成できます。

1. 次のリンクをクリックして、**問題の作成** ページを開きます (問題の作成)。
2. **Summary** フィールドと **Description** フィールドに入力します。**Description** フィールドに、ドキュメントの URL、章またはセクション番号、および問題の詳しい説明を入力します。フォーム内の他のフィールドは変更しないでください。
3. **Create** をクリックします。

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

Red Hat OpenStack Platform では、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 System 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。このガイドでは、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を説明します。

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

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

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

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

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

ボリューム

OpenStack Block Storage サービス (**openstack-cinder**) により、ユーザーは **ボリューム** を使用してブロックストレージデバイスにアクセスすることができます。一時ストレージを汎用の永続ストレージで拡張するために、インスタンスにボリュームを接続することができます。ボリュームは、任意でインスタンスからデタッチすることも、再度接続することもできます。接続したインスタンス経由でないと、ボリュームにはアクセスできません。
一時ストレージを使用しないようにインスタンスを設定することもできます。一時ストレージを使用する代わりに、Block Storage サービスがイメージをボリュームに書き込むように設定できます。その後、インスタンスのブート可能なルートボリュームとしてボリュームを使用することができます。

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

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

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ストレージ

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 (cinder) のバックエンドとして使用する場合は、Ceph Object Storage Daemon (OSD) ノードをさらに追加することで、ストレージの容量および冗長性をスケーリングできます。Block Storage, Object Storage (swift) および Shared File Systems Storage (manila) サービスは、バックエンドとして Red Hat Ceph Storage をサポートします。

Block Storage サービスは、個別のバックエンドとして複数のストレージソリューションを使用できます。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 サービスには、多くのサポート対象バックエンドソリューションリストも含まれており、その一部には追加のスケ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容量を利用可能な分だけ消費することができます。ストレージノードにストレージデバイスを追加することで、容量を拡張することもできます。

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

1.2. ストレージへの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。詳細は、「ストレージの冗長性と障害復旧」 を参照してください。プロジェクト管理者は、Block Storage サービスを使用して、サイズやバックエンドなどのボリューム設定を集約する **ボリュームタイプ** を作成できます。ボリュームタイプを **QoS (Quality of Service)** 仕様に関連付けることで、クラウドユーザーにさまざまなレベルのパフォーマンスを提供できます。ユーザーは、新しいボリュームを作成するときに必要なボリュームタイプを指定できます。たとえば、より高いパフォーマンスの QoS 仕様を使用するボリュームは、より多くの IOPS をユーザーに提供できます。また、ユーザーは、リソースを節約するために、より低いパフォーマンスの QoS 仕様を使用するボリュームに軽いワークロードを割り当てることができます。

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 経由でアクセスすることができ、クラウド内のインスタンスやサービスからアクセスすることができます。したがって、サービスのオブジェクトリポジトリ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. ストレージのセキュリティー

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

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

イメージの署名および検証ならびにメタデータ定義 (metadef) API の制限など、Image サービス (glance) のセキュリティーを保護する方法の詳細は、**イメージの作成および管理** の Image サービス (glance) を参照してください。

1.4. ストレージの冗長性と障害復旧

Block Storage (cinder) 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。バックアップ機能を使用して、ボリュームのコンテンツを保護します。サービスは、スナップショットもサポートします。クローン作成に加えて、スナップショットを使用してボリュームを以前の状態に復元することもできます。

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。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、ボリュームをグループ化して同時にスナップショットを作成するために、整合性グループの作成もサポートしています。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が向上します。詳細は、「Block Storage サービス (cinder) の整合性グループ」 を参照してください。

Object Storage (swift) 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。すべてのバックアップを、ファイルシステムまたはノードレベルで実行する必要があります。Object Storage 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 サービス (CINDER) の設定

Block Storage サービス (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。Compute 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。

ボリュームバックアップの詳細は、**Block Storage ボリュームのバックアップ** ガイドを参照してください。

重要

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

Block Storage は、Block Storage REST API を使用して設定されます。

注記

Block Storage はバージョン 2 をサポートしていないため、Block Storage REST API バージョン 3 を使用していることを確認してください。デフォルトのオーバークラウドデプロイメントでは、環境変数 **OS_VOLUME_API_VERSION=3.0** を設定することで、この確認が行われます。

Block Storage REST API は、マイクロバージョンを使用して拡張機能を追加することにより、後方互換性を維持します。**cinder** CLI は、特定のマイクロバージョンが指定されない限り、REST API バージョン 3.0 を使用します。たとえば、**cinder** コマンドに 3.17 マイクロバージョンを指定するには、**--os-volume-api-version 3.17** 引数を追加します。

注記

openstack CLI は、これらのマイクロバージョンをサポートしていないため、Block Storage REST API のバージョン 3.0 しか使用できません。

2.1. BLOCK STORAGE サービスのバックエンド

Red Hat OpenStack Platform (RHOSP) は director を使用してデプロイされます。これを行うことで、Block Storage サービス (cinder) およびそのバックエンドなど、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。

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。RHOSP は、Block Storage サービスのバックエンドとして Red Hat Ceph Storage および NFS をサポートします。RHOSP を使用して Red Hat Ceph Storage をデプロイする方法は、**director を使用した Red Hat Ceph Storage と OpenStack Platform のデプロイ** を参照してください。Block Storage でサポートされている NFS とその設定の詳細は、NFS ストレージの設定 を参照してください。

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。director には、異なるバックエンドソリューションをデプロイするのに必要なコンポーネントが含まれています。

サポートされている Block Storage サービスのバックエンドアプライアンスとドライバーの完全なリストについては、**Component, Plug-In, and Driver Support in Red Hat OpenStack Platform**の Cinder を参照してください。すべてのサードパーティーのバックエンドアプライアンスおよびドライバーに

は、追加のデプロイメントガイドがあります。適切なデプロイメントガイドを確認し、バックエンドアプライアンスまたはドライバーにプラグインが必要かどうかを判断します。

複数のバックエンドを使用するように Block Storage を設定した場合は、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。ボリュームの作成時にバックエンドを指定しない場合、Block Storage スケジューラーはフィルターを使用して適切なバックエンドを選択します。詳細は、デフォルトの Block Storage スケジューラーフィルターの設定 を参照してください。

関連情報

- ボリューム種別の作成および設定

2.1.1. NFS ストレージの設定

共有 NFS ストレージを使用するようにオーバークラウドを設定できます。

2.1.1.1. サポートされる設定および制限

サポートされる NFS ストレージ

- Red Hat では、認定済みのストレージバックエンドおよびドライバーを使用することを推奨します。Red Hat では、汎用 NFS バックエンドの NFS ストレージを使用することを推奨していません。認定済みのストレージバックエンドおよびドライバーと比較すると、その機能に制限があるためです。たとえば、汎用 NFS バックエンドは、ボリュームの暗号化やボリュームのマルチアタッチなどの機能をサポートしません。サポート対象のドライバーの情報は、Red Hat Ecosystem Catalog を参照してください。

- Block Storage (cinder) サービスおよび Compute (nova) サービスには、NFS バージョン 4.0 以降を使用する必要があります。Red Hat OpenStack Platform (RHOSP) は、以前のバージョンの NFS をサポートしません。

サポートされていない NFS 設定

- RHOSP は、通常のボリューム操作を妨げるため、NetApp 機能の NAS セキュアをサポートしていません。Director はデフォルトでこの機能を無効にします。したがって、NFS バックエンドまたは NetApp NFS Block Storage バックエンドが NAS セキュアをサポートするかどうかを制御する次の heat パラメーターは編集しないでください。
  - **CinderNetappNasSecureFileOperations**
  - **CinderNetappNasSecureFilePermissions**
  - **CinderNasSecureFileOperations**
  - **CinderNasSecureFilePermissions**

NFS 共有を使用する場合の制限

- バックエンドが NFS 共有の場合、スワップディスクを持つインスタンスはサイズ変更または再構築できません。

2.1.1.2. NFS ストレージの設定

共有 NFS ストレージを使用するようにオーバークラウドを設定できます。

手順

1. **nfs_storage.yaml** などの NFS ストレージを設定するための環境ファイルを作成します。
2. 次のパラメーターを新しい環境ファイルに追加して、NFS ストレージを設定します。

```
parameter_defaults:
  CinderEnableIscsiBackend: false
  CinderEnableNfsBackend: true
  GlanceBackend: file
  CinderNfsServers: 192.0.2.230:/cinder
  GlanceNfsEnabled: true
  GlanceNfsShare: 192.0.2.230:/glance
```

注記

CinderNfsMountOptions パラメーターおよび **GlanceNfsOptions** パラメーターは設定しないでください。これらのパラメーターのデフォルト値は、ほとんどの Red Hat OpenStack Platform (RHOSP) 環境に適した NFS マウントオプションを有効にするためです。**environment/storage/glance-nfs.yaml** ファイルで **GlanceNfsOptions** パラメーターの値を確認できます。同じ NFS サーバーを共有するように複数のサービスを設定する際に問題が発生した場合は、Red Hat サポートにお問い合わせください。

3. その他の環境ファイルと共に NFS ストレージ環境ファイルをスタックに追加して、オーバークラウドをデプロイします。

```
(undercloud)$ openstack overcloud deploy --templates \
 -e [your environment files] \
 -e /home/stack/templates/nfs_storage.yaml
```

2.1.1.3. 変換用の外部 NFS 共有の設定

Block Storage サービス (cinder) がオーバークラウドのコントローラーノードでイメージ形式の変換を実行し、スペースが限られている場合は、大きな Image Service (glance) のイメージを変換すると、ノードのルートディスクスペースが完全に使用される可能性があります。変換に外部 NFS 共有を使用して、ノードのスペースが完全にいっぱいになるのを防ぐことができます。

外部 NFS 共有設定を制御する 2 つの director heat パラメーターがあります。

- **CinderImageConversionNfsShare**
- **CinderImageConversionNfsOptions**

手順

1. アンダークラウドに **stack** ユーザーとしてログインし、**stackrc** 認証情報ファイルを読み込みます。

```
$ source ~/stackrc
```

2. 新規または既存のストレージ関連の環境ファイルに、外部 NFS 共有に関する情報を追加します。

```
parameter_defaults:
  CinderImageConversionNfsShare: 192.168.10.1:/convert
```

注記

NFS マウントオプションを制御する **CinderImageConversionNfsOptions** パラメーターのデフォルト値は、ほとんどの環境で十分です。

3. ご自分の環境に該当するその他の環境ファイルと共に、新しい設定が含まれる環境ファイルを openstack overcloud deploy コマンドに追加します。

```
$ openstack overcloud deploy \
--templates \
…
-e <existing_overcloud_environment_files> \
-e <new_environment_file> \
…
```

- **<existing_overcloud_environment_files>** を既存のデプロイメントに含まれる環境ファイルのリストに置き換えます。
- **<new_environment_file>** を、NFS 共有設定を含む新規または編集済みの環境ファイルに置き換えます。

2.2. BLOCK STORAGE ボリュームサービスの高可用性

Block Storage ボリュームサービス (**cinder-volume**) は、アクティブ/パッシブモードでコントローラーノードにデプロイされます。この場合、Pacemaker はこのサービスの高可用性 (HA) を維持します。

分散コンピュートノード (DCN) デプロイメントの場合、Block Storage ボリュームサービスはアクティブ/パッシブモードで中央サイトにデプロイされます。この場合、Pacemaker はこのサービスの HA を維持します。Block Storage ボリュームサービスは、ストレージを必要とするエッジサイトにのみデプロイしてください。Pacemaker はエッジサイトにデプロイできないため、Block Storage ボリュームサービスをアクティブ/アクティブモードでデプロイして、このサービスの HA を確保する必要があります。**dcn-storage.yaml** heat テンプレートは、この設定を実行します。この場合、Block Storage ボリュームサービスは、通常 3 つの個別のホスト上で実行される Block Storage クラスターとして展開されます。

Block Storage クラスターを管理する必要があります。Block Storage クラスターは、すべて同じ設定を持ち、同じ Ceph クラスターの同じプールを制御する Block Storage ボリュームサービスのグループです。詳細は、エッジサイトでの Block Storage クラスターの管理 を参照してください。

注記

Red Hat Ceph Storage バックエンドのデフォルトのクラスター名は **tripleo@tripleo_ceph** です。

2.2.1. エッジサイトで Block Storage クラスターの管理

ストレージを必要とするエッジサイトで Block Storage ボリュームサービス (**cinder-volume**) をアクティブ/アクティブモードでデプロイすると、Block Storage クラスターとしてデプロイされます。このクラスターは、すべて同じ設定を持ち、同じ Ceph クラスターの同じプールを制御する Block Storage

ボリュームサービスのグループです。Red Hat Ceph Storage バックエンドのデフォルトのクラスター名は **tripleo@tripleo_ceph** です。

次のコマンドを使用して、このクラスターとそのサービスを管理できます。

クラスターを保守できます。詳細は、Block Storage クラスターのメンテナンスの開始 を参照してください。

注記

これらのクラスター管理コマンドには、Block Storage (cinder) REST API マイクロバージョン 3.17 以降が必要です。

ユーザーのアクション	コマンド
クラスターを監視するには、名前、バイナリー、状態、ステータスの列を使用します。 一部のサービスがダウンしている場合は、**cinder --os-volume-api-version 3.17 service-list** コマンドを使用して、影響を受けるサービスを特定します。	**cinder --os-volume-api-version 3.17 cluster-list**
バイナリー、ホスト、ゾーン、ステータス、状態、クラスター、無効化の理由、クラスター名の各列を使用して、クラスターのすべての Block Storage サービスのステータスと詳細情報を確認します。 **host** 列は、クラスター内で実行している Block Storage ボリュームサービスを識別します。	**cinder --os-volume-api-version 3.17 service-list**
特定のクラスター化されたサービスに関する詳細情報を表示するには。	**cinder --os-volume-api-version 3.17 cluster-show <clustered_service>** ● **<clustered_service>** を、クラスター化されたサービスの名前に置き換えます。
クラスター化されたサービスを有効にするには。	**cinder --os-volume-api-version 3.17 cluster-enable <clustered_service>**
クラスター化したサービスを無効にする。	**cinder — os-volume-api-version 3.17 cluster-disable <clustered_service>**

Block Storage クラスターで管理できるボリュームをリスト表示します。詳細は、ボリュームとそのスナップショットの管理と管理解除 を参照してください。	**cinder — os-volume-api-version 3.17 manageable-list --cluster <cluster_name>** ● **<cluster_name>** をクラスターの名前に置き換えます。たとえば、tripleo@tripleo_ceph などです。
Block Storage クラスターで管理できるスナップショットをリスト表示します。	**cinder — os-volume-api-version 3.17 snapshot-manageable-list --cluster <cluster_name>**
管理されていないボリュームを Block Storage クラスターに追加します。	**cinder — os-volume-api-version 3.17 manage <unmanaged_volume> --cluster <cluster_name>** ● **<unmanaged_volume>** を、管理されていないボリュームを指定するために必要な引数に置き換えます。
Block Storage クラスター化サービスのボリュームを移行します。詳細は、CLI を使用してバックエンド間でボリュームを移行する を参照してください。	**cinder — os-volume-api-version 3.17 migrate <volume> --cluster <cluster_name>** ● **<volume>** を必要なボリュームの名前または ID に置き換えます。

2.2.2. Block Storage クラスターのメンテナンス開始

クラスター内で複数の Block Storage ボリュームサービス (**cinder-volume**) がグループ化されている場合は、実行していないサービスをクリーンアップするには、これらのサービスの少なくとも 1 つが実行されている必要があります。この場合は、**work-cleanup** コマンドを使用してこのクラスターのメンテナンスを実行できます。

注記

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。

前提条件

- プロジェクト管理者である (Block Storage クラスターのメンテナンスを開始するため)。
- Block Storage (cinder) REST API マイクロバージョン 3.24 以降。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

   - **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. 次のコマンドを実行して、Block Storage クラスターのすべてのサービスが実行されているかどうかを確認します。

```
$ cinder --os-volume-api-version 3.17 cluster-list --detailed
```

3. 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 --os-volume-api-version 3.17 service-list
```

4. クラスターのメンテナンスをトリガーするには、次のコマンドを実行します。

```
$ cinder --os-volume-api-version 3.24 work-cleanup --cluster <cluster_name>
```

   - **<cluster_name>** をクラスターの名前に置き換えます。たとえば、tripleo@tripleo_ceph などです。

2.3. ボリューム種別によるボリューム設定のグループ化

Red Hat OpenStack Platform では、ボリューム種別を作成することができ、関連する設定を各ボリューム種別に適用することができます。必要なボリューム種別は、ボリュームの作成前と作成後に割り当てることができます。詳細は、Block Storage ボリュームの作成 および Block Storage の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、Block Storage サービス (cinder) ボリューム暗号化 を参照してください。
- ボリュームが使用するバックエンド。詳細は、複数のバックエンドのボリューム割り当て および バックエンド間でのボリュームの移動 を参照してください。
- ボリュームの QoS (Quality of Service) フォーマンス制限または QoS 仕様の関連リスト。詳細は、Block Storage サービス (cinder) QoS (Quality of Service) の仕様 を参照してください。

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。

ボリュームタイプを作成して、クラウドユーザーにさまざまなレベルのパフォーマンスを提供できます。

- 特定のパフォーマンス、復元力、その他の追加スペックをキーと値のペアとして各ボリュームタイプに追加します。
- QoS パフォーマンス制限または QoS 仕様のさまざまなリストをボリュームタイプに関連付けます。

ユーザーはボリュームを作成するときに、パフォーマンス要件を満たす適切なボリュームタイプを選択できます。

ボリュームを作成し、ボリュームタイプを指定しない場合、ブロックストレージはデフォルトのボリュームタイプを使用します。ブロックストレージ (cinder) 設定ファイルを使用して、すべてのプロジェクト (テナント) に適用される一般的なデフォルトのボリュームタイプを定義できます。ただし、展開でプロジェクト固有のボリュームタイプを使用する場合は、プロジェクトごとにデフォルトのボリュームタイプを定義してください。この場合、ブロックストレージは、一般的なデフォルトのボリュームタイプではなく、プロジェクト固有のボリュームタイプを使用します。詳細は、プロジェクト固有のデフォルトボリュームタイプの定義 を参照してください。

関連情報

- ボリューム種別の作成および設定
- Block Storage サービスのバックエンド
- ダッシュボードを使用した QoS 仕様とボリュームタイプの関連付け
- Dashboard を使用した Block Storage サービスボリューム暗号化の設定

2.3.1. バックエンドドライバーのプロパティーのリスト表示

ボリュームタイプに関連付けられたプロパティーでは、Extra Specs と呼ばれるキーと値のペアが使用されます。各ボリュームタイプのバックエンドドライバーは、独自の追加スペックのセットをサポートします。ドライバーがサポートする追加仕様の詳細は、バックエンドドライバーのドキュメントを参照してください。

あるいは、Block Storage ホストに直接クエリーを実行して、そのバックエンドドライバーの明確に定義された標準追加仕様をリスト表示することもできます。

前提条件

- Block Storage ホストに直接クエリーを実行するには、プロジェクト管理者である必要があります。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

   - **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. **cinder-volume** のホストを特定します。

```
$ cinder service-list
```

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

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

3. ドライバーの機能を表示して、Block Storage サービスでサポートされている追加スペックを特定します。

```
$ cinder get-capabilities <volsvchost>
```

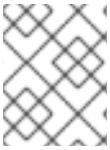
- **<volsvchost>** は **cinder-volume** のホストに置き換えます。以下に例を示します。

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

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

2.3.2. ボリューム種別の作成および設定

ボリュームタイプを作成して、関連する設定を各ボリュームタイプに適用できます。



注記

Block Storage サービス (cinder) 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、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。

たとえば、ボリュームタイプを作成して、クラウドユーザーにさまざまなレベルのパフォーマンスを提供できます。

- 特定のパフォーマンス、復元力、その他の追加スペックをキーと値のペアとして各ボリュームタイプに追加します。
- QoS パフォーマンス制限または QoS 仕様のさまざまなリストをボリュームタイプに関連付けます。詳細は、Block Storage サービス (cinder) QoS (Quality of Service) の仕様 を参照してください。

ユーザーはボリュームを作成するときに、パフォーマンス要件を満たす適切なボリュームタイプを選択できます。

前提条件

- ボリュームタイプを作成および設定するには、プロジェクト管理者である必要があります。
- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. 管理ユーザーとして Dashboard にログインします。
2. **Admin** > **Volumes** > **Volume Types**を選択します。
3. **Create Volume Type**をクリックします。
4. **Name** フィールドにボリューム種別の名前を入力します。
5. **Create Volume Type**をクリックします。**Volume Types** の表に新しい種別が表示されます。
6. ボリューム種別の **View Extra Specs** のアクションを選択します。
7. **Create** をクリックして **Key** と **Value** を指定します。キーと値のペアは有効である必要があります。有効でない場合には、ボリュームの作成時にそのボリューム種別を指定するとエラーが発生してしまいます。たとえば、このボリューム種別のバックエンドを指定するには、**volume_backend_name Key** を追加し、**Value** を必要なバックエンドの名前に設定します。
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 **Extra Specs** 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。手順については、プライベートボリューム種別の作成および設定 を参照してください。

次のステップ

- ダッシュボードを使用した QoS 仕様とボリュームタイプの関連付け

2.3.3. ボリューム種別の編集

Dashboard でボリューム種別を編集して、ボリューム種別の **追加スペック** 設定を変更します。ボリューム種別を削除することもできます。

前提条件

- ボリュームタイプを編集または削除するには、プロジェクト管理者である必要があります。
- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. 管理ユーザーとして Dashboard にログインします。
2. **Admin** > **Volumes** > **Volume Types**を選択します。
3. **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。
4. 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。

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。

2.3.4. プライベートボリューム種別の作成および設定

デフォルトでは、全プロジェクト (テナント) ですべてのボリューム種別を使用することができます。アクセスが制限されたボリューム種別を作成するには、ボリューム種別を **プライベート** に指定します。これを行うには、ボリュームタイプの **is-public** フラグは、デフォルト値が true であるため **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。通常、これらは特定のプロジェクトでのみ使用できる設定です。たとえば、テスト中の新しいバックエンドや超高性能設定などです。

前提条件

- プライベートボリュームタイプのアクセスを作成、表示、または設定するには、プロジェクト管理者である必要があります。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

   - **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. 新しい cinder ボリュームタイプを作成し、**is-public** フラグを **false** に設定します。

```
$ cinder type-create --is-public false <type_name>
```

- **<type_name>** は、この新しいプライベートボリュームタイプに付ける名前に置き換えます。

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
```

このコマンドは、パブリックボリュームタイプとプライベートボリュームタイプの両方の名前と ID をリスト表示します。ボリュームにアクセスできるようにするには、ボリュームタイプの ID が必要です。

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。したがって、必要なプロジェクトの ID を知っている必要があります。このテナント ID がわからないが、このプロジェクトのユーザーの名前はわかっている場合は、次を実行します。

注記

このユーザー名が不明な場合は、**openstack user list** コマンドを実行すると、設定されたすべてのユーザーの名前と ID がリスト表示されます。

```
$ openstack user show <user_name>
```

- **<user_name>** は必要なプロジェクトのユーザーの名前に置き換えて、このユーザーが関連付けられているプロジェクトの **tenantId** を含むユーザーの詳細のリストを表示します。

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>
```

- **<type_id>** は、必要なプライベートボリュームタイプの ID に置き換えます。
- **<tenant_id>** は、必要なテナント ID に置き換えます。

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>
```

2.3.5. プロジェクト固有のデフォルトボリュームタイプの定義

オプション: 複雑な展開の場合、プロジェクト管理者は各プロジェクト (テナント) のデフォルトのボリュームタイプを定義できます。

ボリュームを作成し、ボリュームタイプを指定しない場合、ブロックストレージはデフォルトのボリュームタイプを使用します。

Block Storage (cinder) 設定ファイル **cinder.conf** の **default_volume_type** オプションを使用して、すべてのプロジェクトに適用される一般的なデフォルトのボリュームタイプを定義できます。

ただし、Red Hat OpenStack Platform (RHOSP) デプロイメントでプロジェクト固有のボリュームタイプを使用する場合は、プロジェクトごとにデフォルトのボリュームタイプを定義してください。この場合、ブロックストレージは、一般的なデフォルトのボリュームタイプではなく、プロジェクト固有のボリュームタイプを使用します。次の RHOSP デプロイメントの例では、プロジェクト固有のデフォルトのボリュームタイプが必要です。

- 多くのアベイラビリティーゾーン (AZ) にまたがる分散 RHOSP デプロイメント。各 AZ は独自のプロジェクトにあり、独自のボリュームタイプがあります。
- 会社の 3 つの異なる部門の RHOSP デプロイメント。各部門は独自のプロジェクトにあり、独自の専門的なボリュームタイプがあります。

前提条件

- プロジェクト固有のデフォルトのボリュームタイプとなる各プロジェクトの少なくとも 1 つのボリュームタイプ。詳細は、ボリューム種別の作成および設定 を参照してください。
- Block Storage REST API マイクロバージョン 3.62 以降。
- プロジェクト管理者のみが、プロジェクトのデフォルトのボリュームタイプを定義、クリア、または一覧表示できます。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

   - **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. プロジェクトのデフォルトのボリュームタイプを定義、クリア、または一覧表示します。

> **注記**
>
> これらのコマンドの **<project_id>** を必要なプロジェクトの ID に置き換える必要があります。各テナントの ID と名前を見つけるには、**openstack project list** コマンドを実行します。

   - プロジェクトのデフォルトのボリュームタイプを定義するには:

```
$ cinder --os-volume-api-version 3.62 default-type-set <volume_type> <project_id>
```

     - **<volume_type>** を、必要なボリュームタイプの名前または ID に置き換えます。**cinder type-list** コマンドを実行して、すべてのボリュームタイプの名前と ID を一覧表示できます。

   - プロジェクトのデフォルトのボリュームタイプをクリアするには:

```
$ cinder --os-volume-api-version 3.62 default-type-unset <project_id>
```

–

- プロジェクトのデフォルトのボリュームタイプを一覧表示するには:

```
$ cinder --os-volume-api-version 3.62 default-type-list --project <project_id>
```

2.4. BLOCK STORAGE サービス (CINDER) 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。

手順

1. 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          |
  +----------+----------------------------------+
```

2.5. IMAGE-VOLUME キャッシュの設定

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、Block Storage 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。

前提条件

- **内部テナント** が Block Storage サービス用に設定されています。詳細は、 Block Storage サービス (cinder) の内部プロジェクトの作成と設定 を参照してください。
- アンダークラウドがインストールされる。詳細は、director を **使用した Red Hat OpenStack Platform のインストールと管理の director** のインストールを参照してください。

手順

1. アンダークラウドホストに **stack** ユーザーとしてログインします。
2. **stackrc** アンダークラウド認証情報ファイルを入手します。

```
$ source ~/stackrc
```

3. バックエンドでイメージボリュームキャッシュを有効にして設定するには、オーバークラウドのデプロイメントコマンドに含まれる環境ファイルの **ExtraConfig** セクションに次の値を追加する必要があります。

```
parameter_defaults:
  ExtraConfig:
    cinder::config::cinder_config:
      DEFAULT/cinder_internal_tenant_project_id:
        value: TENANTID (1)
      DEFAULT/cinder_internal_tenant_user_id:
        value: USERID (2)
      BACKEND/image_volume_cache_enabled: (3)
        value: True
      BACKEND/image_volume_cache_max_size_gb:
        value: MAXSIZE (4)
      BACKEND/image_volume_cache_max_count:
        value: MAXNUMBER (5)
```

(1) **TENANTID** を **cinder-internal** プロジェクトの ID に置き換えます。

(2) **USERID** を **cinder-internal** ユーザーの ID に置き換えます。

(3) **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)。

(4) 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** を必要なサイズ (GB 単位) に設定します。

(5) **MAXNUMBER** をイメージの最大数に設定します。

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。

4. 更新を環境ファイルに保存します。

5. その他の環境ファイルと共に環境ファイルをスタックに追加して、オーバークラウドをデプロイします。

2.6. BLOCK STORAGE サービス (CINDER) のサービス品質仕様

QoS (Quality of Service) 仕様を作成して各ボリュームタイプに関連付けることにより、クラウドユーザーが作成したボリュームにパフォーマンス制限を適用できます。たとえば、より高いパフォーマンスの QoS 仕様を使用するボリュームは、より多くの IOPS をユーザーに提供できます。また、ユーザーは、リソースを節約するために、より低いパフォーマンスの QoS 仕様を使用するボリュームに軽いワークロードを割り当てることができます。

注記

QoS 仕様を作成、設定、関連付け、および関連付け解除するには、プロジェクト管理者である必要があります。

QoS 仕様を作成するときは、必要なコンシューマーを選択する必要があります。コンシューマーは、QoS 制限を適用する場所を決定し、QoS 制限を定義するためにどの QoS プロパティーキーが使用できるかを決定します。利用可能なコンシューマーの詳細は、QoS 仕様のコンシューマー を参照してください。

必要な QoS プロパティーキーをデプロイメント固有の値に設定することで、ボリュームのパフォーマンス制限を作成できます。Block Storage サービス (cinder) によって提供される QoS プロパティーキーの詳細は、Block Storage QoS プロパティーキー を参照してください。

QoS 仕様を作成してボリュームタイプに関連付けるには、次のタスクを実行します。

1. QoS 仕様を作成して設定します。

2. QoS 仕様をボリュームタイプに関連付けます。

ダッシュボードまたは CLI を使用して、QoS 仕様を作成、設定し、ボリュームタイプに関連付けることができます。

2.6.1. QoS 仕様のコンシューマー

QoS 仕様を作成するときは、必要なコンシューマーを選択する必要があります。コンシューマーは、QoS 制限を適用する場所を決定し、QoS 制限を定義するためにどの QoS プロパティーキーが使用できるかを決定します。Block Storage サービス (cinder) は、次の QoS 仕様のコンシューマーをサポートします。

- **front-end**: Compute サービス (nova) は、ボリュームがインスタンスに接続されるときに QoS 制限を適用します。Compute サービスは、Block Storage サービスによって提供されるすべての QoS プロパティーキーをサポートします。

- **back-end**: 関連付けられたボリュームタイプのバックエンドドライバーは、QoS 制限を適用します。各バックエンドドライバーは、独自の QoS プロパティーキーのセットをサポートします。ドライバーがサポートする QoS プロパティーキーの詳細は、バックエンドドライバーのドキュメントを参照してください。

back-end コンシューマーがサポートされていない場合は、**front-end** コンシューマーを使用します。たとえば、Bare Metal Provisioning サービスを通じてボリュームをベアメタルノードに接続する場合 (皮肉)。

- **both**: 可能であれば、両方のコンシューマーが QoS 制限を適用します。したがって、このコンシューマータイプは次の QoS プロパティーキーをサポートします。
  - ボリュームがインスタンスにアタッチされている場合は、Compute サービスとバックエンドドライバーの両方がサポートするすべての QoS プロパティーキーを使用できます。
  - ボリュームがインスタンスにアタッチされていない場合は、バックエンドドライバーがサポートする QoS プロパティーキーのみを使用できます。

2.6.2. Block Storage QoS プロパティーキー

Block Storage サービスは、クラウドユーザーが作成するボリュームのパフォーマンスを制限できるように、QoS プロパティーキーを提供します。これらの制限では、ストレージボリュームのパフォーマンスに関する次の 2 つの業界標準測定値が使用されます。

- 1 秒あたりの入出力操作数 (IOPS)
- データ転送速度 (バイト/秒で測定)

QoS 仕様の利用者は、どの QoS プロパティーキーがサポートされるかを決定します。詳細は、QoS 仕様の利用者 を参照してください。

一部の QoS プロパティーキーはバックエンドドライバーによって外部的に定義されているため、Block Storage は QoS プロパティーキーのエラーチェックを実行できません。したがって、Block Storage は、無効またはサポートされていない QoS プロパティーキーを無視します。

重要

QoS プロパティーキーのスペルが正しいことを確認してください。スペルが間違っているプロパティーキーを含むボリュームのパフォーマンス制限は無視されます。

IOPS とデータ転送速度の両方の測定について、次のパフォーマンス制限を設定できます。

固定制限

通常、固定制限はボリュームパフォーマンス測定の平均使用量を定義する必要があります。

バースト制限

通常、バースト制限は、ボリュームパフォーマンス測定の激しいアクティビティーの期間を定義する必要があります。バースト制限により、平均的な使用量に対して固定制限を低く保ちながら、特定の時間におけるアクティビティーの増加率が考慮されます。

注記

バースト制限はすべて 1 秒のバースト長を使用します。

制限合計

total_* QoS プロパティーキーを使用して、必要なパフォーマンス制限の読み取り操作と書き込み操作の両方に対するグローバル制限を指定します。

注記

合計制限を使用する代わりに、読み取り操作と書き込み操作に個別の制限を適用したり、読み取り操作または書き込み操作のみを制限することを選択したりできます。

読み取り制限

read_* QoS プロパティーキーを使用して、必要なパフォーマンス制限の読み取り操作にのみ適用される制限を指定します。

注記

合計制限を指定した場合、この制限は無視されます。

書き込み制限

write_* QoS プロパティーキーを使用して、必要なパフォーマンス制限の書き込み操作にのみ適用される制限を指定します。

注記

合計制限を指定した場合、この制限は無視されます。

次の Block Storage QoS プロパティーキーを使用して、デプロイメントのボリュームパフォーマンス制限を作成できます。

注記

すべての QoS プロパティーキーのデフォルト値は **0** で、制限が無制限であることを意味します。

表2.1 Block Storage QoS プロパティーキー

パフォーマンス制限	測定単位	QoS プロパティーキー
固定 IOPS	IOPS	**total_iops_sec** **read_iops_sec** **write_iops_sec**
ボリュームのサイズによって計算される固定 IOPS。 これらの制限の使用制限の詳細は、ボリュームサイズに応じて拡張される QoS 制限 を参照してください。	GB あたりの IOPS	**total_iops_sec_per_gb** **read_iops_sec_per_gb** **write_iops_sec_per_gb**

パフォーマンス制限	測定単位	QoS プロパティーキー
バースト IOPS	IOPS	**total_iops_sec_max** **read_iops_sec_max** **write_iops_sec_max**
固定データ転送率	1 秒あたりのバイト数	**total_bytes_sec** **read_bytes_sec** **write_bytes_sec**
バーストデータ転送率	1 秒あたりのバイト数	**total_bytes_sec_max** **read_bytes_sec_max** **write_bytes_sec_max**
IOPS 制限を計算するときの IO リクエストのサイズ。 詳細は、IOPS 制限に対する IO リクエストサイズの設定 を参照してください。	Bytes	**size_iops_sec**

2.6.2.1. IOPS 制限の IO リクエストサイズを設定する

IOPS ボリュームのパフォーマンス制限を実装する場合は、ユーザーがこれらの制限を回避できないように、一般的な IO リクエストサイズも指定する必要があります。そうしないと、ユーザーは多数の小さな IO リクエストではなく、いくつかの大規模な IO リクエストを送信する可能性があります。

size_iops_sec QoS プロパティーキーを使用して、一般的な IO 要求の最大サイズをバイト単位で指定します。Block Storage サービスは、このサイズを使用して、送信される各 IO リクエストに対する一般的な IO リクエストの比例数を計算します。次に例を示します。

size_iops_sec=4096

- 8 KB のリクエストは 2 リクエストとしてカウントされます。
- 6 KB のリクエストは 1.5 リクエストとしてカウントされます。
- 4 KB 未満のリクエストは 1 リクエストとしてカウントされます。

Block Storage サービスは、IOPS 制限を計算するときに、この IO リクエストサイズ制限のみを使用します。

注記

size_iops_sec のデフォルト値は **0** で、IOPS 制限を適用するときに IO リクエストのサイズは無視されます。

2.6.2.2. ボリュームサイズに応じて拡張される IOPS 制限

ユーザーが作成するボリュームの容量によって決定される IOPS ボリュームのパフォーマンス制限を作成できます。これらの QoS (Quality of Service) の制限は、プロビジョニングされたボリュームのサイズに応じて拡張されます。たとえば、ボリュームタイプに読み取り操作のボリュームサイズ 1 GB あたり 500 の IOPS 制限がある場合、このボリュームタイプのプロビジョニングされた 3 GB ボリュームの読み取り IOPS 制限は 1500 になります。

重要

ボリュームのサイズは、ボリュームがインスタンスにアタッチされるときに決定されます。したがって、ボリュームがインスタンスにアタッチされているときにボリュームのサイズが変更された場合、これらの制限は、このボリュームがデタッチされてからインスタンスに再アタッチされるときにのみ、新しいボリュームサイズに対して再計算されます。

GB あたりの IOPS で指定された次の QoS プロパティーキーを使用して、スケーラブルなボリュームのパフォーマンス制限を作成できます。

- **total_iops_sec_per_gb**: 読み取り操作と書き込み操作の両方について、ボリュームサイズの GB ごとのグローバル IOPS 制限を指定します。

  注記

  合計制限を使用する代わりに、読み取り操作と書き込み操作に個別の制限を適用したり、読み取り操作または書き込み操作のみを制限することを選択したりできます。

- **read_iops_sec_per_gb**: 読み取り操作にのみ適用される、ボリュームサイズの GB あたりの IOPS 制限を指定します。

  注記

  合計制限を指定した場合、この制限は無視されます。

- **write_iops_sec_per_gb**: 書き込み操作にのみ適用される、ボリュームサイズの GB あたりの IOPS 制限を指定します。

  注記

  合計制限を指定した場合、この制限は無視されます。

重要

これらの QoS 制限を含む QoS 仕様のコンシューマーは **front-end** または **both** にすることができますが、**back-end** にすることはできません。詳細は、QoS 仕様の利用者 を参照してください。

2.6.3. ダッシュボードを使用した QoS 仕様の作成と設定

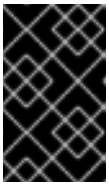
QoS (Quality of Service) 仕様は、ボリュームパフォーマンスの QoS 制限のリストです。各 QoS 制限を作成するには、QoS プロパティーキーをデプロイメント固有の値に設定します。QoS パフォーマンス制限をボリュームに適用するには、QoS 仕様を必要なボリュームタイプに関連付ける必要があります。

前提条件

- QoS 仕様を作成、設定、関連付け、および関連付け解除するには、プロジェクト管理者である必要があります。
- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. 管理ユーザーとして Dashboard にログインします。
2. **Admin** > **Volumes** > **Volume Types**を選択します。
3. **QoS Specs** の表で **Create QoS Spec** をクリックします。
4. **QoS Spec** の名前を入力します。
5. **Consumer** フィールドで、この QoS 仕様のコンシューマーを選択します。詳細は、Consumers of QoS specifications を参照してください。
6. **Create** をクリックします。新しい QoS 仕様が **QoS Specs** テーブルに表示されます。
7. **QoS Specs** テーブルで、新しい QoS 仕様の **Manage Specs** アクションを選択して **Specs** ウィンドウを開き、QoS パフォーマンス制限を追加します。
8. **Specs** ウィンドウで **Create** をクリックして、**Create Extra Specs** ウィンドウを開きます。
9. **Key** フィールドに QoS パフォーマンス制限の QoS プロパティーキーを指定し、 **Value** フィールドにパフォーマンス制限値を設定します。使用可能なプロパティーキーの詳細は、Block Storage QoS プロパティーキー を参照してください。



重要

QoS プロパティーキーのスペルが正しいことを確認してください。スペルが間違っているプロパティーキーを含むボリュームのパフォーマンス制限は無視されます。

10. **Create** をクリックして、QoS 制限を QoS 仕様に追加します。
11. QoS 仕様に追加する QoS 制限ごとに手順 7 ~ 10 を繰り返します。

次のステップ

- ダッシュボードを使用した QoS 仕様とボリュームタイプの関連付け

2.6.4. CLI を使用した QoS 仕様の作成および設定

QoS (Quality of Service) 仕様は、ボリュームパフォーマンスの QoS 制限のリストです。各 QoS 制限を作成するには、QoS プロパティーキーをデプロイメント固有の値に設定します。QoS パフォーマンス制限をボリュームに適用するには、QoS 仕様を必要なボリュームタイプに関連付ける必要があります。

前提条件

- QoS 仕様を作成、設定、関連付け、および関連付け解除するには、プロジェクト管理者である必要があります。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

   - **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. QoS 仕様を作成します。

```
$ openstack volume qos create [--consumer <qos_spec_consumer>] <qos_spec_name>
```

   - オプション: **<qos_spec_consumer>** を、この QoS 仕様の必要なコンシューマーに置き換えます。指定しない場合、コンシューマーはデフォルトで **Both** を使用します。詳細は、Consumers of QoS specifications を参照してください。
   - **<qos_spec_name>** を QoS 仕様の名前に置き換えます。

3. 追加する QoS 制限ごとに個別の **--property <key=value>** 引数を指定して、QoS 仕様にパフォーマンス制限を追加します。

```
$ openstack volume qos set --property <key>=<value> <qos_spec_name>
```

   - **<key>** を必要なパフォーマンス制約の QoS プロパティーキーに置き換えます。詳細は、Block Storage QoS プロパティーキー を参照してください。

     重要

     QoS プロパティーキーのスペルが正しいことを確認してください。スペルが間違っているプロパティーキーを含むボリュームのパフォーマンス制限は無視されます。

   - **<value>** を QoS プロパティーキーで必要な測定単位での、このパフォーマンス制約に対するデプロイメント固有の制限に置き換えます。
   - **<qos_spec_name>** を QoS 仕様の名前または ID に置き換えます。
     以下に例を示します。

```
$ openstack volume qos set \
 --property read_iops_sec=5000 \
 --property write_iops_sec=7000 \
 myqoslimits
```

4. QoS 仕様を確認してください。

```
$ openstack volume qos list
+--------------------------------------+---------+-----------+--------------+-------------------------------
--------------------+
| ID                                   | Name    | Consumer  | Associations | Properties
|
+--------------------------------------+---------+-----------+--------------+-------------------------------
--------------------+
| 204c6ba2-c67c-4ac8-918a-03f101811235 | myqoslimits | front-end |              |
```

```
read_iops_sec='5000', write_iops_sec='7000' |
+-------------------------------------+---------+-----------+--------------+-------------------------------
-------------------+
```

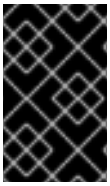
このコマンドは、設定されているすべての QoS 仕様の設定詳細のテーブルを提供します。

次のステップ

- CLI を使用した QoS 仕様とボリュームタイプの関連付け

2.6.5. ダッシュボードを使用した QoS 仕様とボリュームタイプの関連付け

QoS 制限をボリュームに適用するには、QoS (Quality of Service) 仕様を既存のボリュームタイプに関連付ける必要があります。



重要

ボリュームがすでにインスタンスにアタッチされている場合、QoS 制限は、ボリュームがデタッチされてからこのインスタンスに再アタッチされたときにのみこのボリュームに適用されます。

前提条件

- QoS 仕様を作成、設定、関連付け、および関連付け解除するには、プロジェクト管理者である必要があります。
- 必要なボリュームタイプが作成されます。詳細は、ボリューム種別の作成および設定 を参照してください。
- 必要な QoS 仕様が作成されます。詳細は、ダッシュボードを使用した QoS 仕様の作成と設定 を参照してください。
- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. 管理ユーザーとして Dashboard にログインします。
2. **Admin** > **Volumes** > **Volume Types**を選択します。
3. **Volume Types** テーブルで、必要なボリュームタイプの **Manage QoS Spec Association** アクションを選択します。
4. **QoS Spec to be associated**リストから必要な QoS 仕様を選択します。
5. **Associate** をクリックします。QoS 仕様が、編集したボリュームタイプの **Associated QoS Spec** 列に追加されます。

2.6.6. CLI を使用した QoS 仕様とボリュームタイプの関連付け

QoS 制限をボリュームに適用するには、QoS (Quality of Service) 仕様を既存のボリュームタイプに関連付ける必要があります。

重要

ボリュームがすでにインスタンスにアタッチされている場合、QoS 制限は、ボリュームがデタッチされてからこのインスタンスに再アタッチされたときにのみこのボリュームに適用されます。

前提条件

- QoS 仕様を作成、設定、関連付け、および関連付け解除するには、プロジェクト管理者である必要があります。
- 必要なボリュームタイプが作成されます。詳細は、ボリューム種別の作成および設定 を参照してください。
- 必要な QoS 仕様が作成されます。詳細は、CLI を使用した QoS 仕様の作成と設定 を参照してください。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

   - **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. 必要な QoS 仕様を必要なボリュームタイプに関連付けます。

```
$ openstack volume qos associate <qos_spec_name> <volume_type>
```

   - **<qos_spec_name>** を QoS 仕様の名前または ID に置き換えます。 **openstack volume qos list** コマンドを実行すると、すべての QoS 仕様の名前と ID をリスト表示できます。
   - **<volume_type>** をボリュームタイプの名前または ID に置き換えます。 **cinder type-list** コマンドを実行して、すべてのボリュームタイプの名前と ID を一覧表示できます。

3. QoS 仕様が関連付けられていることを確認します。

```
$ openstack volume qos list
```

   出力テーブルの **Associations** 列には、どのボリュームタイプがこの QoS 仕様に関連付けられているかが表示されます。

2.6.7. ダッシュボードを使用して QoS 仕様とボリュームタイプの関連付けを解除する

ボリュームタイプのボリュームに QoS 制限を適用したくない場合は、ボリュームタイプからサービス品質 (QoS) 仕様の関連付けを解除できます。

重要

ボリュームがすでにインスタンスにアタッチされている場合、QoS 制限は、ボリュームがデタッチされてからこのインスタンスに再アタッチされたときにのみ、このボリュームから削除されます。

前提条件

- QoS 仕様を作成、設定、関連付け、および関連付け解除するには、プロジェクト管理者である必要があります。
- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. 管理ユーザーとして Dashboard にログインします。
2. **Admin** > **Volumes** > **Volume Types**を選択します。
3. **Volume Types** テーブルで、必要なボリュームタイプの **Manage QoS Spec Association** アクションを選択します。
4. **QoS Spec to be associated**リストから **None** を選択します。
5. **Associate** をクリックします。
   QoS 仕様は、編集されたボリュームタイプの **Associated QoS Spec** 列から削除する必要があります。

2.6.8. CLI を使用した QoS 仕様とボリュームタイプの関連付けの解除

ボリュームタイプのボリュームに QoS 制限を適用したくない場合は、ボリュームタイプからサービス品質 (QoS) 仕様の関連付けを解除できます。

重要

ボリュームがすでにインスタンスにアタッチされている場合、QoS 制限は、ボリュームがデタッチされてからこのインスタンスに再アタッチされたときにのみ、このボリュームから削除されます。

前提条件

- QoS 仕様を作成、設定、関連付け、および関連付け解除するには、プロジェクト管理者である必要があります。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

   - **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. QoS 仕様に関連付けられているボリュームタイプの関連付けを解除します。特定のボリュームタイプの関連付けを解除することも、複数のボリュームタイプが同じ QoS 仕様に関連付けられている場合はすべてのボリュームタイプの関連付けを解除することもできます。
   - QoS 仕様に関連付けられた特定のボリュームタイプの関連付けを解除するには、次の手順を実行します。

```
$ openstack volume qos disassociate <qos_spec_name> --volume-type <volume_type>
```

  - **<qos_spec_name>** を QoS 仕様の名前または ID に置き換えます。 **openstack volume qos list** コマンドを実行すると、すべての QoS 仕様の名前と ID をリスト表示できます。
  - **<volume_type>** を、この QoS 仕様に関連付けられたボリュームタイプの名前または ID に置き換えます。**cinder type-list** コマンドを実行して、すべてのボリュームタイプの名前と ID を一覧表示できます。

- QoS 仕様に関連付けられているすべてのボリュームタイプの関連付けを解除するには、次の手順を実行します。

```
$ openstack volume qos disassociate <qos_spec_name> --all
```

3. QoS 仕様の関連付けが解除されていることを確認します。

```
$ openstack volume qos list
```

この QoS 仕様の **Associations** 列は、ボリュームタイプを指定しないか、空にする必要があります。

2.7. BLOCK STORAGE サービス (CINDER) ボリュームの暗号化

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

- 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。
- ボリューム暗号化は LUKS1 のみをサポートし、LUKS2 はサポートしません。
- 暗号化されていないボリュームを同じサイズの暗号化されたボリュームに種別変更する操作はサポートされません。暗号化したボリュームには、暗号化データを格納するための追加領域が必要なためです。暗号化されていないボリュームの暗号化に関する詳細は、暗号化されていないボリュームの暗号化 を参照してください。

ボリュームの暗号化は、ボリューム種別を使用して適用されます。暗号化されたボリューム種別の詳細は、Dashboard を使用した Block Storage サービスボリューム暗号化の設定 または CLI を使用した Block Storage サービスボリューム暗号化の設定 を参照してください。

OpenStack Key Manager (barbican) を使用して Block Storage (cinder) 暗号化キーを管理する方法の詳細は、Block Storage (cinder) ボリュームの暗号化 を参照してください。

2.7.1. Dashboard を使用した Block Storage サービスボリューム暗号化の設定

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。暗号化されたボリューム種別の暗号化設定を再設定することも可能です。

暗号化されたボリューム種別を呼び出すことで、暗号化されたボリュームを自動的に作成できます。

前提条件

- 暗号化ボリュームを作成するには、プロジェクト管理者である必要があります。
- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. 管理ユーザーとして Dashboard にログインします。
2. **Admin** > **Volumes** > **Volume Types**を選択します。
3. 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。
4. 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** のコラムで各設定を説明されています。

   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。

5. **ボリューム種別の暗号化設定の作成** をクリックします。

暗号化されたボリューム種別の暗号化設定を再設定することも可能です。

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

関連情報

- CLI を使用した Block Storage サービスボリューム暗号化の設定

2.7.2. CLI を使用した Block Storage サービスボリューム暗号化の設定

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。

前提条件

- 暗号化ボリュームを作成するには、プロジェクト管理者である必要があります。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

   - **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. ボリューム種別を作成します。

```
$ cinder type-create myEncType
```

3. 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 myEncType luks
```

4. 暗号化されたボリュームを作成します。

```
$ cinder --debug create 1 --volume-type myEncType --name myEncVol
```

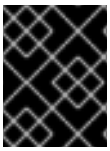
2.7.3. ボリュームイメージ暗号化キーの自動削除

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。

2.8. BLOCK STORAGE ボリュームのバックエンド用の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)。

前提条件

- アンダークラウドがインストールされる。詳細は、director を **使用した Red Hat OpenStack Platform のインストールと管理の director** のインストールを参照してください。

手順

1. アンダークラウドホストに **stack** ユーザーとしてログインします。
2. **stackrc** アンダークラウド認証情報ファイルを入手します。

```
$ source ~/stackrc
```

3. 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
```

4. 更新を環境ファイルに保存します。

5. その他の環境ファイルと共に環境ファイルをスタックに追加して、オーバークラウドをデプロイします。

2.9. BLOCK STORAGE サービス (CINDER) の整合性グループ

Block Storage (cinder) サービスを使用して、整合性グループを設定して複数の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。つまり、複数のボリュームに対して個別に操作を実行するのではなく、同時に複数のボリュームに対して操作を実行することができます。整合性グループを使用して、複数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。また、これらのボリュームを同時にリストアまたはクローン作成することも可能です。

1 つの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後に、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。

2.9.1. Block Storage サービスの整合性グループの設定

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

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

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

前提条件

- アンダークラウドがインストールされる。詳細は、director を **使用した Red Hat OpenStack Platform のインストールと管理の director** のインストールを参照してください。

手順

1. アンダークラウドホストに **stack** ユーザーとしてログインします。

2. **stackrc** アンダークラウド認証情報ファイルを入手します。

```
$ source ~/stackrc
```

3. 環境ファイルを編集し、**parameter_defaults** セクションに新しいエントリーを追加します。これにより、**openstack overcloud deploy** コマンドを使用して環境が再デプロイされるたびに、エントリーがコンテナーで更新され保持されるようになります。

4. **CinderApiPolicies** を使用して環境ファイルに新規セクションを追加し、整合性グループの設定を定義します。JSON ファイルのデフォルト設定を持つ同等の **parameter_defaults** セクションは、次のように表示されます。

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

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

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

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

7. 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' }, \
 }
```

8. 更新を環境ファイルに保存します。

9. その他の環境ファイルと共に環境ファイルをスタックに追加して、オーバークラウドをデプロイします。

2.9.2. Dashboard を使用した Block Storage 整合性グループの作成

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

前提条件

- 整合性グループを作成するには、プロジェクト管理者またはボリューム所有者である必要があります。
- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. 管理者ユーザーまたはボリューム所有者としてダッシュボードにログインします。
2. **Project > Compute > Volumes > Volume Consistency Groups**を選択します。
3. **整合性グループの作成** をクリックします。
4. 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。
5. 整合性グループにボリューム種別を追加することもできます。整合性グループにボリュームを作成する際には、Block Storage サービスにより、これらのボリューム種別から互換性のある設定が適用されます。ボリューム種別を追加するには、**利用可能な全ボリューム種別** リストから追加するボリューム種別の + ボタンをクリックします。
6. **整合性グループの作成** をクリックします。次回、作成した整合性グループが **ボリュームの整合性グループ** テーブルに表示されます。

2.9.3. Dashboard を使用した Block Storage サービスの整合性グループの管理

Dashboard で Block Storage ボリュームの整合性グループを管理できます。

前提条件

- 整合性グループを管理するには、プロジェクト管理者である必要があります。
- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. 管理ユーザーとして Dashboard にログインします。
2. **Project > Compute > Volumes > Volume Consistency Groups**を選択します。
3. (オプション) **アクション** コラムから **整合性グループの編集** を選択して、整合性グループの名前または説明を変更することができます。
4. 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には、設定する整合性グループを見つけます。その整合性グループの **アクション** コラムで、**ボリュームの管理** を選択します。これにより、**整合性グループボリュームの追加/削除** ウィザードが起動します。
   a. 整合性グループにボリュームを追加するには、**利用可能な全ボリューム** リストから追加するボリュームの + ボタンをクリックします。
   b. 整合性グループからボリュームを削除するには、**選択済みのボリューム** リストから削除するボリュームの - ボタンをクリックします。
5. **整合性グループの編集** をクリックします。

2.9.4. Block Storage サービス用の整合性グループのスナップショットの作成および管理

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

前提条件

- 整合性グループのスナップショットを作成および管理するには、プロジェクト管理者である必要があります。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

   - **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. 利用可能な整合性グループおよびその ID をすべて表示します。

```
$ cinder consisgroup-list
```

3. 整合性グループを使用してスナップショットを作成します。

```
$ cinder cgsnapshot-create [--name <cgsnapname>] [--description "<description>"]
<cgnameid>
```

   - **<cgsnapname>** は、スナップショットの名前に置き換えます。
   - **<description>** は、スナップショットの説明に置き換えます。
   - **<CGNAMEID>** は、整合性グループの名前または ID に置き換えます。

4. 利用可能な整合性グループのスナップショットの全リストを表示します。

```
# cinder cgsnapshot-list
```

2.9.5. Block Storage サービスの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。

前提条件

- 整合性グループのクローンを作成し、整合性グループのスナップショットを復元するには、プロジェクト管理者である必要があります。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

   - **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. 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>"]
```

   - **<cgnameid>** は、複製する整合性グループの名前または ID に置き換えます。
   - **<cgname>** は、整合性グループの名前に置き換えます。
   - **<description>** は、整合性グループの説明に置き換えます。

3. 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。

2.10. デフォルトの BLOCK STORAGE スケジューラーフィルターの設定

ボリュームの作成時にボリュームのバックエンドが指定されていないと、Block Storage スケジューラーはフィルターを使用して適切なバックエンドを選択します。次のデフォルトフィルターを設定していることを確認します。

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。

前提条件

- アンダークラウドがインストールされる。詳細は、director を **使用した Red Hat OpenStack Platform のインストールと管理の director** のインストールを参照してください。

手順

1. アンダークラウドホストに **stack** ユーザーとしてログインします。
2. **stackrc** アンダークラウド認証情報ファイルを入手します。

```
$ source ~/stackrc
```

3. 以下のパラメーターが含まれる環境ファイルをオーバークラウドのデプロイメントコマンドに追加します。

```
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。

4. 更新を環境ファイルに保存します。
5. その他の環境ファイルと共に環境ファイルをスタックに追加して、オーバークラウドをデプロイします。

2.11. オーバークラウドノードでの LVM2 フィルターの有効化

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

- LVM がゲストからのボリュームグループを表示するように見える
- LVM が重複するボリュームグループ名を報告する
- LVM がストレージにアクセスしているため、ボリュームの切り離しが失敗する
- LVM の問題が原因でゲストがブートに失敗する
- ゲストマシン上の LVM は、実際に存在するディスクが見つからないため、部分的な状態にある

- LVM を持つデバイスで Block Storage サービス (cinder) のアクションが失敗する
- Block Storage サービス (cinder) のスナップショットが正しく削除されない
- ライブマイグレーション中のエラー: **/etc/multipath.conf** が存在しない

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

前提条件

- アンダークラウドがインストールされる。詳細は、director を **使用した Red Hat OpenStack Platform のインストールと管理の director** のインストールを参照してください。

手順

1. アンダークラウドホストに **stack** ユーザーとしてログインします。
2. **stackrc** アンダークラウド認証情報ファイルを入手します。

```
$ source ~/stackrc
```

3. 以下のパラメーターが含まれる環境ファイルをオーバークラウドのデプロイメントコマンドに追加します。

```
parameter_defaults:
  LVMFilterEnabled: true
```

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

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

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

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

4. 更新を環境ファイルに保存します。
5. その他の環境ファイルと共に環境ファイルをスタックに追加して、オーバークラウドをデプロイします。

2.12. 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。

2.12.1. director を使用したマルチパスの設定

Red Hat OpenStack Platform (RHOSP) オーバークラウドデプロイメントでマルチパスを設定して、帯域幅とネットワークの耐障害性を向上させることができます。

重要

既存のデプロイメントでマルチパスを設定すると、新しいワークロードはマルチパスに対応します。既存のワークロードがある場合は、これらのインスタンスでマルチパスを有効にするには、インスタンスを退避して復元する必要があります。

前提条件

- アンダークラウドがインストールされる。詳細は、director を **使用した Red Hat OpenStack Platform のインストールと管理の director** のインストールを参照してください。

手順

1. アンダークラウドホストに **stack** ユーザーとしてログインします。
2. **stackrc** アンダークラウド認証情報ファイルを入手します。

```
$ source ~/stackrc
```

3. 上書き環境ファイルを使用するか、**multipath_overrides.yaml** などの新しい環境ファイルを作成します。以下のパラメーターを追加して設定します。

```
parameter_defaults:
  ExtraConfig:
    cinder::config::cinder_config:
      backend_defaults/use_multipath_for_image_xfer:
        value: true
```

注記

デフォルト設定では、ほとんどの環境で機能する基本的なマルチパス設定が生成されます。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。マルチパスの詳細は、**Device Mapper Multipath の設定** を参照してください。

4. オプション: オーバークラウドデプロイメント用のマルチパス設定ファイルがある場合、**MultipathdCustomConfigFile** パラメーターを使用してこのファイルの場所を指定できます。

```
parameter_defaults:
  MultipathdCustomConfigFile: <config_file_directory>/<config_file_name>
```

次の例では、**/home/stack** はマルチパス設定ファイルのディレクトリーで、**multipath.conf** はこのファイルの名前です。

```
parameter_defaults:
  MultipathdCustomConfigFile: /home/stack/multipath.conf
```

注記

他の TripleO マルチパスパラメーターは、ローカルのカスタム設定ファイル内の対応する値を上書きします。たとえば、**MultipathdEnableUserFriendlyNames** が **False** の場合、ローカルのカスタムファイルで設定が有効になっている場合でも、オーバークラウドノードのファイルが一致するように更新されます。

マルチパスパラメーターの詳細は、Multipath heat template parameters を参照してください。

5. 更新をオーバーライド環境ファイルに保存します。
6. オーバーライド環境ファイルを、次のような他の環境ファイルと共にスタックに追加します。

```
----
/usr/share/openstack-tripleo-heat-templates/environments/multipathd.yaml
----
```

7. オーバークラウドをデプロイする。

関連情報

- **インスタンスの作成と管理** の インスタンスの退避

2.12.1.1. マルチパス heat テンプレートパラメーター

マルチパスを有効にする以下のパラメーターについては、これを使用します。

パラメーター	説明	Default value
MultipathdEnable	マルチパスデーモンを有効にするかどうかを定義します。このパラメーターは、**multipathd.yaml** ファイルに含まれる設定で、デフォルトで **True** に設定されます。	**True**
MultipathdEnableUserFriendlyNames	各パスに対してユーザーフレンドリーな名前の割り当てを有効にするかどうかを定義します。	**False**
MultipathdEnableFindMultipaths	パスごとにマルチパスデバイスを自動的に作成するかどうかを定義します。	**True**
MultipathdSkipKpartx	デバイスで自動的にパーティションの作成を省略するかどうかを定義します。	**True**

パラメーター	説明	Default value
MultipathdCustomConfigFile	オーバークラウドノードのカスタムマルチパス設定ファイルが含まれています。デフォルトでは、最小の **multipath.conf** ファイルがインストールされます。 **注意:** 他の TripleO マルチパスパラメーターは、追加するローカルのカスタム設定ファイルの対応する値を上書きします。たとえば、**MultipathdEnableUserFriendlyNames** が **False** の場合、ローカルのカスタムファイルで設定が有効になっている場合でも、オーバークラウドノードのファイルが一致するように更新されます。	

2.12.2. マルチパス設定の確認

新規または既存のオーバークラウドデプロイメントでマルチパス設定を確認できます。

手順

1. インスタンスを作成します。
2. 暗号化されていないボリュームをインスタンスにアタッチします。
3. インスタンスが含まれる Compute ノードの名前を取得します。

```
$ nova show <instance> | grep OS-EXT-SRV-ATTR:host
```

   <instance> は、作成したインスタンス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
```

5. Compute ノードの IP アドレスを取得します。

```
$ . stackrc
$ metalsmith list | grep <compute_name>
```

   <compute_name> は、**nova show <instance>** コマンドの出力からの名前に置き換えて、6 列のテーブルから 2 行を表示します。

   <compute_name> が 4 列目にある行を見つけます。 **<compute_name>** の IP アドレスは、この行の最後の列にあります。

   次の例では、compute-0 は 2 行目の 4 列目にあるため、compute-0 の IP アドレスは 192.02.24.15 です。

```
$ . stackrc
$ metalsmith list | grep compute-0
| 3b1bf72e-c425-494c-9717-d0b89bb66580 | compute-0    | 95b21d3e-36be-470d-ba5c-
70d5dcd6d0b3 | compute-1    | ACTIVE | ctlplane=192.02.24.49 |
| 72a24883-25f9-435c-bf71-a20e66be172d | compute-1    | a59f79f7-006e-4f38-a9ad-
8164da47d58e | compute-0    | ACTIVE | ctlplane=192.02.24.15 |
```

6. インスタンスを実行する Compute ノードに SSH 接続します。

```
$ ssh tripleo-admin@<compute_node_ip>
```

<compute_node_ip> は、Compute 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
```

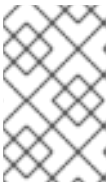
8. Compute ノードインスタンスで以下のコマンドを入力し、cinder ボリュームホストの場所でマルチパスが使用されていることを確認します。

```
virsh domblklist <virsh_instance_name> | grep /dev/dm
```

<virsh_instance_name> は、**nova show <instance> | grep instance_name** コマンドの出力に置き換えます。

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

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



注記

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

第3章 BLOCK STORAGE サービス (CINDER) を使用した基本的な操作の実行

オーバークラウド内の Compute インスタンスのプライマリー永続ストレージとして Block Storage ボリュームを作成して設定します。ボリュームを作成し、ボリュームをインスタンスにアタッチし、ボリュームの編集およびサイズ変更を行い、ボリュームの所有権を変更します。

3.1. BLOCK STORAGE ボリュームの作成

ボリュームを作成して、オーバークラウドの Compute サービス (nova) を使用して起動するインスタンスの永続ストレージを提供します。

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関する説明は、Block Storage サービス (cinder) ボリュームの暗号化 を参照してください。

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。

前提条件

- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. Dashboard にログインします。
2. **Project** > **Compute** > **Volumes**を選択します。
3. **Create Volume** 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)

フィールド	説明
タイプ	オプションのボリューム種別。詳細は、ボリューム種別によるボリューム設定のグループ化 を参照してください。 ボリュームを作成し、ボリュームタイプを指定しない場合、ブロックストレージはデフォルトのボリュームタイプを使用します。デフォルトのボリューム種別を定義する方法の詳細は、プロジェクト固有のデフォルトボリュームタイプの定義 を参照してください。 バックエンドを指定しないと、Block Storage スケジューラーは適切なバックエンドを選択しようとします。詳細は、複数のバックエンドのボリューム割り当て を参照してください。 **注記** 適切なバックエンドがない場合、ボリュームは作成されません。 ボリュームの作成後にボリューム種別を変更することもできます。詳細は、Block Storage のボリューム種別の変更 を参照してください。
容量 (GB)	ボリュームの容量 (ギガバイト単位) 暗号化されていないイメージから暗号化されたボリュームを作成する場合は、暗号化データがボリュームデータを切り捨てないように、ボリュームのサイズがイメージサイズより大きいようにする必要があります。
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関するさらに詳しい説明は、**インスタンス作成のための Compute サービスの設定** の ホストアグリゲートの作成と管理 を参照してください。

4. **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。

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

5. **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。

3.2. ボリュームの名前と説明の編集

Dashboard でボリュームの名前と説明を変更できます。

前提条件

- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. Dashboard にログインします。
2. **Project** > **Compute** > **Volumes**を選択します。
3. 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。
4. 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。
5. **ボリュームの編集** をクリックして、変更を保存します。

3.3. BLOCK STORAGE サービスボリュームのサイズ変更 (拡張)

ボリュームのサイズを変更して、ボリュームのストレージ容量を増やします。

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の詳細は、Red Hat のサポートにお問い合わせください。

手順

1. Source コマンドで認証情報ファイルを読み込みます。
2. ボリュームをリスト表示し、拡張するボリュームの ID を取得します。

```
$ cinder list
```

3. ボリュームのサイズを増やします。

```
$ cinder extend <volume_id> <size>
```

- **<volume_id>** は、拡張するボリュームの ID に置き換えます。
- **<size>** は、このボリュームの必要なサイズ (ギガバイト単位) に置き換えます。

注記

指定されたサイズが、このボリュームの既存のサイズよりも大きいことを確認してください。

以下に例を示します。

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

3.4. BLOCK STORAGE サービスボリュームの削除

不要になったボリュームは削除できます。

注記

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。スナップショットの削除の詳細は、ボリュームスナップショットの削除 を参照してください。

前提条件

- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. Dashboard にログインします。

2. **Project** > **Compute** > **Volumes**を選択します。
3. **ボリューム** の表で、削除するボリュームを選択します。
4. **ボリュームの削除** をクリックします。

3.5. 複数のバックエンドのボリューム割り当て

ボリュームを作成するときに、必要なバックエンドのボリューム種別を種別のリストから選択できます。詳細は、Block Storage ボリュームの作成 を参照してください。

注記

Block Storage サービス (cinder) 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、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。

ボリュームの作成時にバックエンドを指定しない場合、Block Storage スケジューラーは適切なバックエンドを選択しようとします。

スケジューラーは、以下に示すボリュームのデフォルトの関連設定に対してフィルターを使用して、適切なバックエンドを選択します。

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。

適切なバックエンドが複数ある場合、スケジューラーは重み付け方式を使用して最適なバックエンドを選択します。デフォルトでは、CapacityWeigher 方式が使用されるため、使用可能な空き領域が最も多いバックエンドがフィルタリングされて選択されます。

注記

適切なバックエンドがない場合、ボリュームは作成されません。

関連情報

- ボリューム種別の作成および設定
- Block Storage のボリューム種別の変更
- デフォルトの Block Storage スケジューラーフィルターの設定

3.6. インスタンスへのボリュームの接続

インスタンスを閉じると、すべてのデータが失われます。永続ストレージ用にボリュームを接続することができます。ボリュームがマルチ接続ボリューム種別でない限り、ボリュームは一度に1つのインスタンスにしか接続することができません。マルチ接続ボリュームの作成の詳細は、複数のインスタンス

に接続できるボリューム を参照してください。

前提条件

- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. Dashboard にログインします。
2. **Project** > **Compute** > **Volumes**を選択します。
3. **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。
4. **インスタンスへの接続** のリストから、ボリュームの接続先となるインスタンスを選択します。
5. **ボリュームの接続** をクリックします。

3.7. インスタンスからのボリュームの切断

このボリュームを別のインスタンスに接続する場合は、インスタンスからボリュームをデタッチする必要があります (マルチ接続ボリューム種別でない場合)。また、ボリュームへのアクセスパーミッションを変更したり、ボリュームを削除したりするには、ボリュームをデタッチする必要があります。

前提条件

- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. Dashboard にログインします。
2. **Project** > **Compute** > **Volumes**を選択します。
3. 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。
4. 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。

次のステップ

- インスタンスへのボリュームの接続

3.8. ボリュームへのアクセス権の設定

ボリュームのデフォルトの状態は read-write で、データの書き込みおよびそこからの読み取りを可能にします。ボリュームを読み取り専用としてマークし、そのデータが誤って上書きまたは削除されないようにすることができます。

注記

ボリュームを読み取り専用に変更すると、再度読み取り/書き込みに戻すことができます。

前提条件

- ボリュームがすでにインスタンスに接続されている場合は、このボリュームをデタッチします。詳細は、インスタンスからのボリュームの切断 を参照してください。

手順

1. Source コマンドで認証情報ファイルを読み込みます。
2. ボリュームをリスト表示し、設定するボリュームの ID を取得します。

```
$ cinder list
```

3. このボリュームに必要なアクセス権限を設定します。
   - ボリュームのアクセス権限を読み取り専用に設定するには、以下の手順を実施します。

```
$ cinder readonly-mode-update <volume_id> true
```

     - **<volume_id>** は、必要なボリュームの ID に置き換えます。
   - ボリュームのアクセス権限を読み書きに設定するには、以下の手順を実施します。

```
$ cinder readonly-mode-update <volume_id> false
```

4. アクセス権限を変更するためにインスタンスからこのボリュームをデタッチした場合は、ボリュームを再接続します。詳細は、インスタンスへのボリュームの接続 を参照してください。

3.9. DASHBOARD を使用したボリューム所有者の変更

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。

前提条件

- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. ボリューム所有者として Dashboard にログインします。
2. **Projects** > **Volumes** を選択します。
3. 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。
4. **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。

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

Dashboard を使用したボリューム譲渡の受理

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

3.10. CLI を使用したボリューム所有者の変更

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。

手順

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

```
$ cinder list
```

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、転送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 <transfer_id> <transfer_key>
```

- **<transfer_id>** は **cinder transfer-create** コマンドによって返された **id** 値に置き換えます。
- **<transfer_key>** は **cinder transfer-create** コマンドによって返された **auth_key** 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
```

第4章 BLOCK STORAGE サービス (CINDER) を使用した高度な操作の実行

Block Storage ボリュームは、オーバークラウド内の Compute インスタンス用の永続ストレージを形成します。ボリュームスナップショットの使用、マルチ接続ボリュームの作成、ボリュームの種別変更、ボリュームの移行など、ボリュームの高度な機能を設定します。

4.1. ボリュームスナップショットの作成

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

注記

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

このため、スナップショットのバックエンドは、クローン作成中のレイテンシーを最小限に抑えるように、通常ボリュームのバックエンドと同じ場所に配置されます。一方、バックアップのリポジトリーは通常、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。

ボリュームバックアップの詳細は、**Block Storage ボリュームのバックアップ** ガイドを参照してください。

前提条件

- スナップショットを作成するボリューム。ボリュームの作成の詳細は、Block Storage ボリュームの作成 を参照してください。
- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. Dashboard にログインします。
2. **Project** > **Compute** > **Volumes**を選択します。
3. ターゲットボリュームの **スナップショットの作成** アクションを選択します。
4. 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。

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。

検証

- 新しいスナップショットが **ボリュームのスナップショット** タブに表示されていることを確認するか、CLI を使用してボリュームのスナップショットをリスト表示し、スナップショットが作成されていることを確認します。

```
$ openstack volume snapshot list
```

4.2. スナップショットからの新しいボリュームの作成

ボリュームスナップショットのクローンとして、新しいボリュームを作成できます。これらのスナップショットは、特定の時点でのボリュームの状態を保持します。

前提条件

- 新しいボリュームのクローンして作成するボリュームスナップショット。ボリュームスナップショットの作成の詳細は、ボリュームスナップショットの 作成 を参照してください。
- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. Dashboard にログインします。
2. **Project** > **Compute** > **Volumes**を選択します。
3. **ボリュームスナップショット** テーブルで、新しいボリュームを作成する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。ボリュームの作成の詳細は、Block Storage ボリュームの作成 を参照してください。

重要

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

検証

- **Volumes** タブに新しいボリュームが存在することを確認するか、CLI を使用してボリュームをリスト表示し、新しいボリュームが作成されたことを確認します。

```
$ openstack volume list
```

4.3. ボリュームスナップショットの削除

Red Hat OpenStack Platform (RHOSP) 17.1 は RBD CloneV2 API を使用します。これは、依存関係がある場合でも、ボリュームスナップショットを削除できることを意味します。RHOSP デプロイメントが director によってデプロイされた Ceph バックエンドを使用する場合、Ceph クラスターは director によって正しく設定されます。

外部の Ceph バックエンドを使用する場合は、Ceph クラスターで最小限のクライアントを設定する必要があります。外部 Ceph クラスターの設定に関する詳細は、**オーバークラウドの既存 Red Hat Ceph Storage クラスターとの統合** の 既存の Red Hat Ceph Storage クラスターの設定 を参照してください。

前提条件

- 削除するボリュームスナップショット。
- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. Dashboard にログインします。
2. **Project** > **Compute** > **Volumes**を選択します。
3. **ボリュームスナップショット** テーブルで、削除するスナップショットの **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の詳細情報を参照してください。

検証

- **ボリュームスナップショット** タブにスナップショットが表示されなくなったことを確認するか、CLI を使用してボリュームスナップショットをリスト表示し、スナップショットが削除されていることを確認します。

```
$ openstack volume snapshot list
```

4.4. スナップショットからのボリュームの復元

ボリュームデータを最新のスナップショットに戻すことで、ボリュームの最新のスナップショットを復元できます。

警告

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

制限事項

- マルチ接続のボリュームの場合、スナップショットの状態に戻す機能の使用には制限が適用される可能性があります。この機能を使用する前に、このような制限が適用されるかどうかを確認してください。
- スナップショットの作成後にサイズを変更 (拡張) したボリュームを元に戻すことはできません。
- 接続済みまたは使用中のボリュームに対して、スナップショットの状態に戻す機能を使用することはできません。
- デフォルトでは、起動可能なルートボリュームではスナップショットに戻す機能を使用できません。この機能を使用するには、インスタンスが終了した場合にブートボリュームを保持するために、**delete_on_termination=false** プロパティーを使用してインスタンスを起動する必要があります。この場合、スナップショットに戻すには、以下を実行する必要があります。
  - インスタンスを削除してボリュームを使用可能にし、
  - ボリュームを元に戻してから、
  - ボリュームから新しいインスタンスを作成します。

前提条件

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

手順

1. Source コマンドで認証情報ファイルを読み込みます。
2. ボリュームを切断します。

```
$ openstack server remove volume <instance_id> <vol_id>
```

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

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

```
$ cinder snapshot-list
```

4. スナップショットの状態に戻します。

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

   - **<snapshot_id>** を、スナップショット ID に置き換えます。

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

```
$  cinder snapshot-list
```

6. ボリュームを再接続します。

```
$ openstack server add volume <instance_id> <vol_id>
```

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

検証

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

```
$ cinder list
```

4.5. IMAGE サービス (GLANCE) 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。

前提条件

- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. Dashboard にログインします。
2. **Project** > **Compute** > **Volumes**を選択します。
3. 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。
4. ボリュームの **イメージ名** を指定して、リストから **ディスク形式** を選択します。
5. **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、**Project** > **Compute** > **Images**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と設定方法は、**イメージの作成および管理** を参照してください。

4.6. 複数のインスタンスに接続できるボリューム

複数のインスタンスに接続できるマルチ接続 Block Storage ボリュームを作成できます。これらのインスタンスは、同時に読み取りと書き込みを行うことができます。マルチ接続ボリュームには、マルチ接続ボリューム種別が必要です。

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

マルチ接続ボリュームの制限

- Block Storage (cinder) のバックエンドは、マルチ接続ボリュームをサポートしている必要があります。サポートされるバックエンドの情報は、Red Hat のサポートにお問い合わせください。
- Block Storage (cinder) のドライバーは、マルチ接続ボリュームをサポートしている必要があります。Ceph RBD ドライバーがサポートされます。ベンダープラグインでマルチ接続がサポートされることを確認するには、Red Hat のサポートにお問い合わせください。ベンダープラグインの認定の詳細は、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%20OpenStack%
- 読み取り専用のマルチ接続ボリュームはサポートされていません。
- マルチ接続ボリュームのライブマイグレーションは利用できません。
- マルチ接続ボリュームの暗号化はサポートされていません。
- マルチ接続ボリュームは、ベアメタルプロビジョニングサービス (ironic) virt ドライバーではサポートされていません。マルチ接続ボリュームは、libvirt virt ドライバーによってのみサポートされます。
- アタッチされたボリュームをマルチ接続タイプから非マルチ接続タイプに再入力したり、非マルチ接続タイプをマルチ接続タイプに再入力したりすることはできません。
- アタッチされたボリュームの移行中に、複数の読み取り/書き込みアタッチメントを持つマルチ接続ボリュームをソースボリュームまたは宛先ボリュームとして使用することはできません。
- 見送られていたオフロードされたインスタンスにマルチ接続ボリュームをアタッチすることはできません。

4.6.1. 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。

前提条件

- ボリュームタイプを作成するには、プロジェクト管理者である必要があります。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

   - **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. マルチ接続ボリューム用の新しいボリューム種別を作成します。

```
$ cinder type-create multiattach
```

3. このマルチ接続ボリューム種別の **multiattach** プロパティーを有効にします。

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

4. 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>
```

4.6.2. マルチ接続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。

4.6.3. マルチ接続ボリュームの作成

複数のインスタンスに接続できる Block Storage ボリュームを作成できます。これらのインスタンスは、同時に読み取りと書き込みを行うことができます。

注記

この手順では、**multiattach** 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、**multiattach** をサポートするバックエンドが 2 つある場合は、スケジューラーがどちらのバックエンドを使用するかを決定します。詳細は、複数のバックエンドのボリューム割り当て を参照してください。

前提条件

- プロジェクトでマルチ接続ボリューム種別を使用できる。

手順

1. Source コマンドで認証情報ファイルを読み込みます。
2. 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
```

3. 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 |
```

次のステップ

- インスタンスへのボリュームの接続

4.7. バックエンド間でのボリュームの移動

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

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

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

- 再入力: ボリューム種別を変更できるのはボリュームの所有者と管理者だけです。種別変更の操作は、バックエンド間でボリュームを移動する最も一般的な方法です。詳細は、Block Storage のボリューム種別の変更 を参照してください。
- 移行: ボリュームを移行できるのは管理者だけです。ボリュームの移行は特定のユースケース用に制限されます。これは、移行に制約があるため、またデプロイメントの動作について明確に理解する必要があるためです。詳細は、Dashboard を使用したバックエンド間でのボリュームの移行 または CLI を使用したバックエンド間でのボリュームの移行 を参照してください。

制約

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

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

4.7.1. 利用可能なボリュームの移動

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

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

注記

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

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

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

重要

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

4.7.2. 使用中のボリュームの移動

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

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

重要

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

制約

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

4.8. BLOCK STORAGE のボリューム種別の変更

ボリューム種別を変更する際には、ボリューム種別とその設定を既存のボリュームに適用します。ボリューム種別の詳細は、ボリューム種別によるボリューム設定のグループ化 を参照してください。

注記

ボリューム種別を変更できるのはボリュームの所有者と管理者だけです。

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

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

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

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

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

制約

- すべてのボリューム種別を変更することはできません。バックエンド間のボリュームの移動に関する詳細は、バックエンド間でのボリュームの移動 を参照してください。
- 暗号化されていないボリュームは、暗号化されたボリューム種別に変更することはできませんが、暗号化されたボリュームは暗号化されていないボリューム種別に変更できます。
- 暗号化されていないボリュームを同じサイズの暗号化されたボリュームに種別変更する操作はサポートされません。暗号化したボリュームには、暗号化データを格納するための追加領域が必要なためです。暗号化されていないボリュームの暗号化に関する詳細は、暗号化されていないボリュームの暗号化 を参照してください。
- 管理者権限のないユーザーは、自分が所有するボリュームの種別しか変更できません。

関連情報

- ボリューム種別の作成および設定

4.8.1. Dashboard からのボリューム種別の変更

ボリューム種別を変更して、ボリューム種別とその設定を既存のボリュームに適用します。

重要

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

前提条件

- ボリューム種別を変更できるのはボリュームの所有者と管理者だけです。
- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. 管理ユーザーまたはボリューム所有者として Dashboard にログインします。
2. **Project** > **Compute** > **Volumes**を選択します。
3. 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。
4. **ボリューム種別の変更** ダイアログで、対象のボリューム種別を選択し、**種別** のリストから新しいバックエンドを定義します。
5. 別のバックエンドにボリュームを移行する場合は、**移行ポリシー** のリストから **要求時** を選択します。詳細は、バックエンド間でのボリュームの移動 を参照してください。
6. **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。

4.8.2. CLI からのボリューム種別の変更

ボリューム種別を変更して、ボリューム種別とその設定を既存のボリュームに適用します。

重要

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

前提条件

- ボリューム種別を変更できるのはボリュームの所有者と管理者だけです。

手順

1. Source コマンドで認証情報ファイルを読み込みます。
2. 以下のコマンドを入力してボリューム種別を変更します。

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

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

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

4.9. DASHBOARD を使用したバックエンド間でのボリュームの移行

Block Storage サービス (cinder) を使用して、同一または異なるアベイラビリティーゾーン (AZ) にあるバックエンド間で、ボリュームを移行することができます。これは、あるバックエンドから別のバックエンドにボリュームを移行する方法としては、まったく一般的ではありません。

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

制約

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

注記

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

前提条件

- ボリュームを移行するには、プロジェクト管理者である必要があります。
- Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) へのアクセス。詳細は、**OpenStack ダッシュボードを使用したクラウドリソースの管理** を参照してください。

手順

1. 管理ユーザーとして Dashboard にログインします。
2. **Admin** > **Volumes** を選択します。

3. 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。

4. **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。

5. **マイグレーション** をクリックして移行を開始します。

4.10. CLI を使用したバックエンド間でのボリュームの移行

Block Storage サービス (cinder) を使用して、同一または異なるアベイラビリティーゾーン (AZ) にあるバックエンド間で、ボリュームを移行することができます。これは、あるバックエンドから別のバックエンドにボリュームを移行する方法としては、まったく一般的ではありません。

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

制約

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

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

前提条件

- ボリュームを移行するには、プロジェクト管理者である必要があります。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

- **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. 以下のコマンドを入力して、宛先のバックエンドの名前を取得します。

```
$ cinder get-pools --detail

Property                    | Value

...

| name                      | localdomain@lvmdriver-1#lvmdriver-1
| pool_name                 | lvmdriver-1

...

| volume_backend_name       | lvmdriver-1

...

Property                    | Value

...
                                          |
| name                      | localdomain@lvmdriver-2#lvmdriver-1
| pool_name                 | lvmdriver-1

...

| volume_backend_name       | lvmdriver-1

...
```

宛先バックエンド名には、**host@volume_backend_name#pool** という構文が使用されます。

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

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

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

4.11. ボリュームとそのスナップショットの管理と管理解除

cinder manage コマンドと **cinder unmanage** コマンドを使用して、Block Storage ボリュームサービス (**cinder-volume**) にボリュームを追加したり、このサービスからボリュームを削除したりできます。通常、Block Storage ボリュームサービスは、作成したボリュームを管理し、たとえば、これらのボリュームのリスト表示、割り当て、削除などを行うことができます。ただし、**cinder unmanage** コマンドを使用して、Block Storage ボリュームサービスからボリュームを削除し、このボリュームがリス

ト表示されたり、割り当てされたり、削除されたりしないようにすることができます。同様に、**cinder manage** コマンドを使用して、Block Storage ボリュームサービスにボリュームを追加し、たとえば、このボリュームのリスト表示、割り当て、削除を行うことができます。

注記

スナップショットがある場合は、ボリュームを管理解除することはできません。この場合、ボリュームを管理解除する前に、**cinder snapshot-unmanage** コマンドを使用してすべてのスナップショットを管理解除する必要があります。同様に、スナップショットを持つボリュームを管理する場合は、まずボリュームを管理し、次に **cinder snapshot-manage** コマンドを使用してスナップショットを管理する必要があります。

新しいバージョンの RHOSP をデプロイする間、既存の RHOSP バージョンを実行したままにして、Red Hat OpenStack Platform (RHOSP) デプロイメントを並行してアップグレードするときに、これらの Block Storage コマンドを使用できます。このシナリオでは、既存の RHOSP からボリュームを削除するには、すべてのスナップショットを管理解除してからボリュームを管理解除し、その後、このボリュームとそのすべてのスナップショットを管理して、このボリュームとそのスナップショットを新しいバージョンの RHOSP に追加する必要があります。この方法では、既存のクラウドを実行しながら、ボリュームとそのスナップショットを新しい RHOSP バージョンに移動できます。

もう 1 つの考えられるシナリオは、ストレージアレイの 1 つでボリュームを使用しているベアメタルマシンがある場合です。次に、このマシンで実行しているソフトウェアをクラウドに移動することにしますが、このボリュームは引き続き使用する必要があります。このシナリオでは、**cinder manage** コマンドを使用して、このボリュームを Block Storage ボリュームサービスに追加します。

cinder manageable-list コマンドを使用すると、Block Storage ボリュームサービスのストレージアレイ内に管理されていないボリュームがあるかどうかを確認できます。このリスト内のボリュームは通常、ユーザーが管理していないボリューム、または Block Storage ボリュームサービスを使用せずにストレージアレイ上に手動で作成されたボリュームです。同様に、**cinder snapshot-manageable-list** コマンドは、管理可能なすべてのスナップショットをリスト表示します。

ボリュームを識別するために必要なプロパティーはバックエンドに固有であるため、**cinder manage** コマンドの構文はバックエンドに固有です。ほとんどのバックエンドは、**source-name** プロパティーと **source-id** プロパティーのいずれかまたは両方をサポートしていますが、その他のバックエンドでは追加のプロパティーを設定する必要があります。一部のバックエンドでは、管理可能なボリュームと渡す必要があるパラメーターをリストできます。そうでないバックエンドについては、ベンダーのドキュメントを参照してください。**cinder unmanage** コマンドの構文はバックエンドに固有ではないため、必要なボリューム名またはボリューム ID を指定する必要があります。

同様に、スナップショットを識別するために必要なプロパティーはバックエンド固有であるため、**cinder snapshot-manage** コマンドの構文はバックエンド固有です。**cinder snapshot-unmanage** コマンドの構文はバックエンドに固有ではないため、必要なスナップショット名またはスナップショット ID を指定する必要があります。

4.12. 暗号化されていないボリュームの暗号化

暗号化されていないボリュームを暗号化できます。

cinder-backup サービスが利用可能な場合は、暗号化されていないボリュームをバックアップし、それを新しい暗号化されたボリュームに復元します。

cinder-backup サービスが使用できない場合は、暗号化されていないボリュームからグランスイメージを作成し、このイメージから新しい暗号化されたボリュームを作成します。

前提条件

- 暗号化ボリュームを作成するには、プロジェクト管理者である必要があります。
- 暗号化する暗号化されていないボリューム

手順

cinder-backup サービスは次の場合に利用できます。

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

   - **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. 現在の暗号化されていないボリュームをバックアップします。

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

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

3. 暗号化された新しいボリュームを作成します。

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

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

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

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

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

cinder-backup サービスは利用できません:

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

   - **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. 暗号化されていないボリュームの glance イメージを作成します。

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

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

- **<new_image>** を新しいイメージの名前に置き換えます。

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

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

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

4.13. 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 OpenStack Platform (RHOSP) デプロイメントのバックエンドとして Red Hat Ceph Storage (RHCS) を使用する場合は、スナップショットをバックエンドで **保護する** ように設定できます。削除が失敗するため、RHOSP ダッシュボードまたは **cinder snapshot-delete** コマンドを使用して保護されたスナップショットを削除しないでください。

これが発生した場合は、最初に RHCS バックエンドでスナップショットを**保護されていない** 状態に設定します。その後、通常どおり RHOSP を使用してスナップショットを削除できます。

スナップショットの保護の詳細は、Red Hat Ceph Storage **ブロックデバイスガイド** の ブロックデバイススナップショットの保護 および ブロックデバイススナップショットの保護解除 を参照してください。

第5章 OBJECT STORAGE サービス (SWIFT) の設定

Red Hat OpenStack Platform (RHOSP) Object Storage サービス (swift) は、そのオブジェクトまたはデータをコンテナーに保存します。コンテナーはファイルシステムのディレクトリーと似ていますが、ネスト化することはできません。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。

5.1. オブジェクトストレージリング

Object Storage service (swift) 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。リングを使用すると、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関する情報が含まれます。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、次のタイプのデータを格納するための 3 つのリングがあります。

- アカウント情報
- アカウントの下でオブジェクトを整理しやすくするためのコンテナー
- オブジェクトレプリカ

5.1.1. 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 (swift) 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。

手順

1. コントローラーノードのいずれかにログインします。
2. 以下のコマンドを実行します。

```
[tripleo-admin@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 198.51.100.186:6000.
Most recent completion was 2018-12-14 14:55:42 (4 seconds ago) by 198.51.100.174:6000.
===============================================================================
[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.
===============================================================================
```

3. (オプション) **--all** オプションを使用して、追加の出力を返します。
   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します。

   - Replication metrics: レプリケーションのタイムスタンプを確認します。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ません。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設定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。

5.1.2. 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等) 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。

5.1.3. Object Storage サービスにおけるパーティションのべき乗に関する推奨事項

Red Hat OpenStack Platform (RHOSP) Object Storage サービス (swift) 用に独立したノードを使用する場合には、パーティションのべき乗により大きな値を使用します。

Object Storage サービスは、変更した **ハッシュリング** を使用して、データをディスクとノードに分散します。デフォルトでは、アカウント用、コンテナー用、およびオブジェクト用の 3 つのリングがあります。各リングは、**パーティションのべき乗** と呼ばれる固定パラメーターを使用します。このパラメーターは、作成可能なパーティションの最大数を設定します。

パーティションのべき乗パラメーターは重要で、新規コンテナーとそのオブジェクトについてしか変更できません。そのため、**初回デプロイメント** の前にこの値を設定することが重要になります。

RHOSP director がデプロイする環境のデフォルトのパーティションのべき乗値は **10** です。小規模なデプロイメント、特に Object Storage サービスにコントローラーノード上のディスクだけを使用する計画の場合には、これが妥当な値です。

以下の表は、3 つのレプリカを使用する場合に適切なパーティションのべき乗を選択するのに役立ちます。

表5.1 利用可能なディスクの数に対する適切なパーティションのべき乗値

パーティションのべき乗	ディスクの最大数
10	35 まで
11	75 まで
12	150 まで
13	250 まで
14	500 まで

重要

パーティションのべき乗に過剰に大きな値を設定すると (例: 40 ディスクに対して **14**)、レプリケーション時間に悪影響を及ぼします。

パーティションのべき乗を設定するには、以下のリソースを使用します。

```
parameter_defaults:
  SwiftPartPower: 11
```

ヒント

新しいコンテナーに追加のオブジェクトサーバーリングを設定することもできます。これは、当初小さなパーティションのべき乗値を使用する Object Storage サービスのデプロイメントにディスクを追加する場合に便利です。

関連情報

- swift のアップストリームドキュメントの The Rings
- **director を使用した Red Hat OpenStack Platform のインストールと管理**の オーバークラウド環境の変更

5.1.4. カスタムリング

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、大容量の新規ノードが使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。

5.2. OBJECT STORAGE サービスのカスタマイズ

Red Hat OpenStack Platform (RHOSP) 環境の要件によっては、デプロイメントのパフォーマンスを最適化するために、Object Storage サービス (swift) のデフォルト設定の一部をカスタマイズする必要がある場合があります。

5.2.1. 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 (swift) 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、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。

手順

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
```

5.2.2. 保存データ暗号化の有効化

デフォルトでは、Object Storage サービス (swift) にアップロードされるオブジェクトは暗号化されません。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。Object Storage オブジェクト詳細は、**Key Manager サービスを使用したシークレットの管理** の Object Storage (swift) 保存オブジェクトの暗号化 を参照してください。

5.2.3. スタンドアロン Object Storage サービスクラスターのデプロイ

コンポーザブルロールの概念を使用して、OpenStack Identity サービス (keystone) や HAProxy などの最小限の追加サービスを備えたスタンドアロンの Object Storage サービス (swift) クラスターをデプロイできます。

手順

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*、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
```

7. 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 [...]
```

5.2.4. Object Storage サービスのディスク設定に関する推奨事項

Red Hat OpenStack Platform (RHOSP) Object Storage サービス (swift) 用に、1 つまたは複数の独立したローカルディスクを使用します。

デフォルトでは、RHOSP director は、Object Storage サービス用にシステムディスクの **/srv/node/d1** ディレクトリーを使用します。コントローラーでは、このディスクは他のサービスでも使用され、ディスクがパフォーマンスのボトルネックになる可能性があります。

以下の例は、RHOSP Orchestration サービス (heat) のカスタム環境ファイルからの抜粋です。各コントローラーノードで、Object Storage サービスは 2 つの独立したディスクを使用します。両方のディスク全体には XFS ファイルシステムが含まれています。

```
parameter_defaults:
  SwiftRawDisks: {"sdb": {}, "sdc": {}}
```

SwiftRawDisks は、ノード上の各ストレージディスクを定義します。以下の例では、各コントローラーノードの **sdb** ディスクと **sdc** ディスクの両方を定義します。

RHEL 9 では、ハードウェア設定が同じであっても、オーバークラウドノード間で **sdx** 名が異なる可能性があります。RHEL 9 を使用している場合は、**SwiftRawDisks** パラメーターで使用するディスクの定義については、Red Hat ナレッジベースの記事、Red Hat OpenStack Platform 17 の OpenStack Swift で使用するディスク の定義 を参照してください。

重要

複数のディスクを設定する場合は、Bare Metal サービス (ironic) が必ず目的のルートディスクを使用するようにします。

関連情報

- **director を使用した Red Hat OpenStack Platform のインストールと管理**ガイドの マルチディスククラスターのルートディスクの定義。

5.2.5. 外部 SAN ディスクの使用

デフォルトでは、Object Storage サービス(swift)は、独立したローカルディスクを使用するように設定および最適化されています。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およぼすイベント発生時に、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。そのため、オブジェクトストレージ用に追加のローカルディスクを使用して、パフォーマンスとディスク容量の要件を満たすようにしてください。

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

重要

Object Storage 用に外部 SAN を使用する場合は、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。

Red Hat では、以下の問題に対するサポートを提供しません。

- Object Storage 用に外部 SAN を使用した結果のパフォーマンスに関連する問題。
- コアの Object Storage サービスオファリングの外部で発生する問題。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。

手順

- Object Storage 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/"}}
```

5.3. OBJECT STORAGE ノードの追加または削除

クラスターに新規 Object Storage (swift) ノードを追加するには、ノード数を増やし、リングを更新し、変更を同期させる必要があります。オーバークラウドにノードを追加するか、ベアメタルノードをスケールアップすることで、ノード数を増やすことができます。

クラスターから Object Storage ノードを削除するには、クラスター内のデータ量に応じて、単純な削除または増分削除を実行できます。

5.3.1. オーバークラウドへのノード追加

オーバークラウドにノードを追加できます。

注記

Red Hat OpenStack Platform (RHOSP) の新規インストールには、セキュリティーエラータやバグ修正などの特定の更新が含まれていません。その結果、Red Hat Customer Portal または Red Hat Satellite Server を使用する接続環境をスケールアップすると、RPM 更新は新しいノードに適用されません。最新の更新をオーバークラウドノードに適用するには、以下のいずれかを実行する必要があります。

- スケールアウト操作後にノードのオーバークラウド更新を完了します。
- **virt-customize** ツールを使用して、スケールアウト操作の前にパッケージをベースのオーバークラウドイメージに変更します。詳細は、Red Hat ナレッジベースで Modifying the Red Hat Linux OpenStack Platform Overcloud Image with virt-customize のソリューションを参照してください。

手順

1. 登録する新規ノードの詳細を記載した新しい JSON ファイル (**newnodes.json**) を作成します。

```
{
  "nodes":[
    {
        "mac":[
            "dd:dd:dd:dd:dd:dd"
        ],
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
        "pm_type":"ipmi",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.02.24.207"
    },
```

```
    {
        "mac":[
            "ee:ee:ee:ee:ee:ee"
        ],
        "cpu":"4",
        "memory":"6144",
        "disk":"40",
        "arch":"x86_64",
        "pm_type":"ipmi",
        "pm_user":"admin",
        "pm_password":"p@55w0rd!",
        "pm_addr":"192.02.24.208"
    }
  ]
}
```

2. アンダークラウドホストに **stack** ユーザーとしてログインします。

3. **stackrc** アンダークラウド認証情報ファイルを入手します。

```
$ source ~/stackrc
```

4. 新しいノードを登録します。

```
$ openstack overcloud node import newnodes.json
```

5. 新しいノードごとにイントロスペクションプロセスを開始します。

```
$ openstack overcloud node introspect \
 --provide <node_1> [<node_2>] [<node_n>]
```

   - **--provide** オプションを使用して、イントロスペクション後に指定されたすべてのノードを **available** 状態にリセットします。
   - **<node_1>**、**<node_2>**、および **<node_n>** までの全ノードを、イントロスペクトする各ノードの UUID に置き換えます。

6. 新しいノードごとにイメージのプロパティーを設定します。

```
$ openstack overcloud node configure <node>
```

5.3.2. ベアメタルノードのスケールアップ

既存オーバークラウドのベアメタルノード数を増やすには、**overcloud-baremetal-deploy.yaml** ファイルのノード数を増やして、オーバークラウドを再デプロイします。

前提条件

- 新しいベアメタルノードが登録され、イントロスペクトされ、プロビジョニングとデプロイメントに使用できる。詳細は、オーバークラウドのノードの登録 と ベアメタルノードハードウェアのインベントリーの作成 を参照してください。

手順

1. アンダークラウドホストに **stack** ユーザーとしてログインします。

2. **stackrc** アンダークラウド認証情報ファイルを入手します。

```
$ source ~/stackrc
```

3. ベアメタルノードのプロビジョニングに使用する **overcloud-baremetal-deploy.yaml** ノード定義ファイルを開きます。

4. スケールアップするロールの **count** パラメーターを増やします。たとえば、以下の設定では、Object Storage ノード数を 4 に増やします。

```
- name: Controller
  count: 3
- name: Compute
  count: 10
- name: ObjectStorage
  count: 4
```

5. オプション: 新規ノードに予測可能なノード配置を設定します。たとえば、以下の設定を使用して、**node03** に新しい Object Storage ノードをプロビジョニングします。

```
- name: ObjectStorage
  count: 4
  instances:
  - hostname: overcloud-objectstorage-0
    name: node00
  - hostname: overcloud-objectstorage-1
    name: node01
  - hostname: overcloud-objectstorage-2
    name: node02
  - hostname: overcloud-objectstorage-3
    name: node03
```

6. オプション: 新しいノードに割り当てるその他の属性を定義します。ノード定義ファイルでノード属性を設定するために使用できるプロパティーの詳細は、ベアメタルノードのプロビジョニング属性 を参照してください。

7. Object Storage サービス (swift) とディスク全体のオーバークラウドイメージ **overcloud-hardened-uefi-full** を使用する場合に、ディスクのサイズと **/var** および **/srv** のストレージ要件に基づいて **/srv** パーティションのサイズを設定します。詳細は、Object Storage サービスのディスクパーティション全体の設定 を参照してください。

8. オーバークラウドノードをプロビジョニングします。

```
$ openstack overcloud node provision \
  --stack <stack> \
  --network-config \
  --output <deployment_file> \
  /home/stack/templates/overcloud-baremetal-deploy.yaml
```

- **<stack>** を、ベアメタルノードがプロビジョニングされるスタックの名前に置き換えます。指定しない場合、デフォルトは **overcloud** です。

- **--network-config** 引数を含めて、**cli-overcloud-node-network-config.yaml** Ansible Playbook にネットワーク定義を提供します。
- **<deployment_file>** は、デプロイメントコマンドに含めるために生成する heat 環境ファイルの名前に置き換えます (例 **:/home/stack/templates/overcloud-baremetal-deployed.yaml)**。

> **注記**
>
> Red Hat OpenStack Platform 16.2 から 17.1 にアップグレードした場合は、**openstack overcloud node provision** コマンドでアップグレードプロセス中に作成または更新した YAML ファイルを含める必要があります。たとえば、**/home/stack/templates/overcloud-baremetal-deployed.yaml** ファイルの代わりに、**/home/stack/tripleo-[stack]-baremetal-deploy.yaml** ファイルを使用します。詳細は、**16.2 から 17.1 へのアップグレードフレームワーク** での オーバークラウドの導入と準備の実行 を参照してください。

9. 別のターミナルでプロビジョニングの進捗をモニタリングします。プロビジョニングが成功すると、ノードの状態が **available** から **active** に変わります。

```
$ watch openstack baremetal node list
```

10. 生成された **overcloud-baremetal-deployed.yaml** ファイルを他の環境ファイルと共にスタックに追加し、オーバークラウドをデプロイします。

```
$ openstack overcloud deploy --templates \
  -e [your environment files] \
  -e /home/stack/templates/overcloud-baremetal-deployed.yaml \
  --deployed-server \
  --disable-validations \
  ...
```

5.3.3. Object Storage の専用ノードの定義

追加のノードを Red Hat OpenStack Platform (RHOSP) Object Storage サービス専用にして、パフォーマンスを向上させます。

追加のノードをオブジェクトストレージサービス専用にする場合は、カスタム **roles_data.yaml** ファイルを編集して、コントローラーノードからオブジェクトストレージサービスのエントリーを削除します。具体的には、**Controller** ロールの **ServicesDefault** リストから以下の行を削除します。

```
- OS::TripleO::Services::SwiftStorage
```

5.3.4. オブジェクトストレージリングの更新およびリバランス

Object Storage サービス (swift) では、すべてのコントローラーノードと Object Storage ノードで同じリングファイルが必要です。コントローラーノードまたはオブジェクトストレージノードが交換、追加、または削除された場合、適切な機能を確保するために、オーバークラウドの更新後にこれらを同期する必要があります。

手順

1. **stack** ユーザーとしてアンダークラウドにログインし、一時ディレクトリーを作成します。

```
$ mkdir temp && cd temp/
```

2. 既存のノードの 1 つ (この例ではコントローラー 0) から新しいディレクトリーにオーバークラウドのリングファイルをダウンロードします。

```
$ ssh tripleo-admin@overcloud-controller-0.ctlplane 'sudo tar -czvf - \
  /var/lib/config-data/puppet-generated/swift_ringbuilder/etc/swift \
  /{*.builder,*.ring.gz,backups/*.builder}' > swift-rings.tar.gz
```

3. リングを抽出し、リングサブディレクトリーに変更します。

```
$ tar xzvf swift-rings.tar.gz && cd \
var/lib/config-data/puppet-generated/swift_ringbuilder/etc/swift/
```

4. デバイス情報にしたがって、以下の変数の値を収集します。

- **<device_name>**:

```
$ openstack baremetal introspection data save \
  <node_name> | jq ".inventory.disks"
```

- **<node_ip>**:

```
$ metalsmith <node_name> show
```

- **<port>**: デフォルトポートは **600x** です。デフォルトを変更している場合は、該当するポートを使用します。
- **<builder_file>**: 手順 3 のビルダーファイル名。
- **<weight>** および **<zone>** 変数はユーザーが定義します。

5. **swift-ring-builder** を使用して、新しいノードを既存のリングに追加して更新します。デバイス情報に応じて、変数を置き換えます。

> **注記**
>
> **swift-ring-builder** コマンドを使用するには、**python3-swift** RPM をインストールする必要があります。

```
$ swift-ring-builder etc/swift/<builder_file> \
  add <zone>-<node_ip>:<port>/<device_name> <weight>
```

6. リングを再調整して、新しいデバイスが使用されるようにします。

```
$ swift-ring-builder etc/swift/<builder_file> rebalance
```

7. 変更したリングファイルをコントローラーノードにアップロードし、これらのリングファイルが使用されていることを確認します。次の例のようなスクリプトを使用して、リングファイルを配布します。

```
#!/bin/sh
```

```
set -xe

ALL="tripleo-admin@overcloud-controller-0.ctlplane \
tripleo-admin@overcloud-controller-1.ctlplane \
tripleo-admin@overcloud-controller-2.ctlplane"
```

- リングをすべてのノードにアップロードし、Object Storage サービスを再起動します。

```
for DST in ${ALL}; do
  cat swift-rings.tar.gz | ssh "${DST}" 'sudo tar -C / -xvzf -'
  ssh "${DST}" 'sudo podman restart swift_copy_rings'
  ssh "${DST}" 'sudo systemctl restart tripleo_swift*'
done
```

5.3.5. ノード変更の同期およびデータの移行

変更したリングファイルを正しいフォルダーにコピーしたら、新しいリングファイルを Object Storage (swift) コンテナーに配布する必要があります。

重要

- すべてのデータを同時に移行しないでください。10% ずつデータを移行してください。たとえば、移行元デバイスの重みを 90.0 に、移行先デバイスを 10.0 に設定します。次に、移行元デバイスの重みを 80.0 に、移行先デバイスを 20.0 に設定します。プロセスが完了するまで、段階的にデータの移行を続けます。移行時にすべてのデータを同時に移動すると、古いデータは移行元デバイスにありますが、すべてのレプリカについてリングは新しい移行先デバイスをポイントします。レプリケーターがすべてのデータを移行先デバイスに移動するまで、データにアクセスすることができません。

- 移行時に、オブジェクトストレージリングはデータの場所を再割当てし、続いてレプリケーターがデータを新しい場所に移動します。クラスターのアクティビティーが増えると、負荷が増加するため、レプリケーションの処理が遅くなります。クラスターが大きいほど、レプリケーションのパスが完了するのにかかる時間が長くなります。これは想定された動作ですが、クライアントが現在移動しているデータにアクセスする場合に、ログファイルに 404 エラーが記録される可能性があります。プロキシーが新しい場所からデータの取得を試みる際に、データがまだ新しい場所にない場合、**swift-proxy** レポートはログファイルに 404 エラーを記録します。
  移行が段階的な場合には、プロキシーは移動していないレプリカにアクセスし、エラーは発生しません。プロキシーが代替レプリカからデータの取得を試みると、ログファイルの 404 エラーは解決されます。レプリケーションプロセスが実行中であることを確認するには、レプリケーションログを参照してください。Object Storage サービス (swift) は、5 分ごとにレプリケーションログを発行します。

手順

1. 次の例のようなスクリプトを使用して、リングファイルを既存のコントローラーノードからすべてのコントローラーノードに配布し、それらのノードで Object Storage サービスコンテナーを再起動します。

```
#!/bin/sh
set -xe

SRC="tripleo-admin@overcloud-controller-0.ctlplane"
```

```
ALL="tripleo-admin@overcloud-controller-0.ctlplane \
tripleo-admin@overcloud-controller-1.ctlplane \
tripleo-admin@overcloud-controller-2.ctlplane"
```

- リングファイルの現在のセットを取得します。

```
ssh "${SRC}" 'sudo tar -czvf - \
  /var/lib/config-data/puppet-generated/swift_ringbuilder/etc/swift \
  /{*.builder,*.ring.gz,backups/*.builder}' > swift-rings.tar.gz
```

- リングをすべてのノードにアップロードし、Object Storage サービスを再起動します。

```
for DST in ${ALL}; do
  cat swift-rings.tar.gz | ssh "${DST}" 'sudo tar -C / -xvzf -'
  ssh "${DST}" 'sudo podman restart swift_copy_rings'
  ssh "${DST}" 'sudo systemctl restart tripleo_swift*'
done
```

2. データが新しいディスクに移動されていることを確認するには、新規ストレージノードで以下のコマンドを実行してください。

```
$ sudo grep -i replication  /var/log/container/swift/swift.log
```

5.3.6. Object Storage ノードの削除

Object Storage (swift) ノードを削除するには、次の 2 つの方法があります。

- 単純な削除: この方法は 1 つのアクションでノードを削除し、データの量が少なく効率的に動作しているクラスターに適しています。
- 段階的削除: 削除するノード上のディスクの重みを減らすようにリングを変更します。この方法は、ストレージネットワークの使用への影響を最小限に抑える場合、またはクラスターに大量のデータが含まれる場合に適しています。

どちらの方法でも、**ベアメタルノードのスケールダウン** 手順に従います。ただし、増分削除の場合は、次の前提条件を満たしてストレージリングを変更し、削除するノードのディスクの重量を減らします。

前提条件

- Object Storage リングが更新され、再調整されます。詳細は、Object Storage リングの更新と再調整 を参照してください。
- Object Storage リングの変更が同期されている。詳細は、ノードの変更の同期とデータの移行 を参照してください。

Object Storage ノードの交換については、**ベアメタルノードのスケールダウン** 手順の冒頭にある前提条件を参照してください。

5.3.7. ベアメタルノードのスケールダウン

オーバークラウド内のベアメタルノードの数を縮小するには、ノード定義ファイルでスタックから削除するノードにタグを付け、オーバークラウドを再デプロイしてから、オーバークラウドからベアメタルノードを削除します。

前提条件

- アンダークラウドの正常なインストール。詳細は、Installing director on the undercloud を参照してください。
- オーバークラウドの正常なデプロイメント。詳細は、プリプロビジョニングされたノード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。
- Object Storage ノードを置き換える場合は、削除するノードから新しい置き換えノードにデータを複製します。新しいノードでレプリケーションのパスが完了するまで待機します。**/var/log/swift/swift.log** ファイルで複製パスの進捗を確認することができます。パスが完了すると、Object Storage サービス (swift) は、以下の例のようにエントリーをログに追加します。

```
Mar 29 08:49:05 localhost object-server: Object replication complete.
Mar 29 08:49:11 localhost container-server: Replication run OVER
Mar 29 08:49:13 localhost account-server: Replication run OVER
```

手順

1. アンダークラウドホストに **stack** ユーザーとしてログインします。
2. **stackrc** アンダークラウド認証情報ファイルを入手します。

```
$ source ~/stackrc
```

3. スケールダウンするロールについて、**overcloud-baremetal-deploy.yaml** ファイルの **count** パラメーターの数を減らします。
4. スタックから削除する各ノードの **hostname** と **name** を定義します (ロールの **instances** 属性で定義されていない場合)。
5. 削除するノードに属性 **provisioned: false** を追加します。たとえば、スタックからノード **overcloud-objectstorage-1** を削除するには、**overcloud-baremetal-deploy.yaml** ファイルに以下のスニペットを追加します。

```
- name: ObjectStorage
  count: 3
  instances:
  - hostname: overcloud-objectstorage-0
    name: node00
  - hostname: overcloud-objectstorage-1
    name: node01
    # Removed from cluster due to disk failure
    provisioned: false
  - hostname: overcloud-objectstorage-2
    name: node02
  - hostname: overcloud-objectstorage-3
    name: node03
```

オーバークラウドの再デプロイ後、**provisioned: false** 属性で定義したノードがスタックには存在しなくなります。ただし、これらのノードは provisioned の状態で稼働したままです。

注記

スタックから一時的にノードを削除するには、属性 **provisioned: false** でオーバークラウドをデプロイし、属性 **provisioned: true** でオーバークラウドを再デプロイして、ノードをスタックに戻します。

6. オーバークラウドからノードを削除します。

```
$ openstack overcloud node delete \
  --stack <stack> \
  --baremetal-deployment \
   /home/stack/templates/overcloud-baremetal-deploy.yaml
```

- **<stack>** を、ベアメタルノードがプロビジョニングされるスタックの名前に置き換えます。指定しない場合、デフォルトは **overcloud** です。

注記

スタックから削除するノードを、**openstack overcloud node delete** コマンドのコマンド引数に含めないでください。

7. ironic ノードを削除します。

```
$ openstack baremetal node delete <IRONIC_NODE_UUID>
```

IRONIC_NODE_UUID は、ノードの UUID に置き換えます。

8. オーバークラウドノードをプロビジョニングして、デプロイメントコマンドに含める heat 環境ファイルを更新して生成します。

```
$ openstack overcloud node provision \
  --stack <stack> \
  --output <deployment_file> \
  /home/stack/templates/overcloud-baremetal-deploy.yaml
```

- **<deployment_file>** は、デプロイメントコマンドに含めるために生成する heat 環境ファイルの名前に置き換えます (例 **:/home/stack/templates/overcloud-baremetal-deployed.yaml)**。

9. プロビジョニングコマンドによって生成された **overcloud-baremetal-deployed.yaml** ファイルを他の環境ファイルと共にスタックに追加し、オーバークラウドをデプロイします。

```
$ openstack overcloud deploy \
  ...
  -e /usr/share/openstack-tripleo-heat-templates/environments \
  -e /home/stack/templates/overcloud-baremetal-deployed.yaml \
  --deployed-server \
  --disable-validations \
  ...
```

5.4. OBJECT STORAGE サービスにおけるコンテナー管理

Object Storage サービス (swift) の整理を容易にするには、擬似フォルダーを使用できます。これらの

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、入れ子が可能です。たとえば、イメージを保管する **Images** フォルダーや、ビデオを保管する **Media** フォルダーなどを作成することができます。

各プロジェクトに 1 つまたは複数のコンテナ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。

5.4.1. プライベートコンテナーとパブリックコンテナーの作成

ダッシュボードを使用して、Object Storage サービス (swift) にコンテナーを作成します。

手順

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。

タイプ	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。
5. (オプション) 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、デフォルトポリシーと Erasure Coding を有効にする別のポリシーなど)、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。

```
$ swift post -H "X-Storage-Policy:<policy>" <container_name>
```

- **<policy>** を、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。
- **<container_name>** は、コンテナーの名前に置き換えます。

5.4.2. コンテナーの疑似フォルダーの作成

ダッシュボードを使用して、Object Storage サービス (swift) にコンテナーの疑似フォルダーを作成します。

手順

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。

5.4.3. Object Storage サービスからのコンテナーの削除

ダッシュボードを使用して、Object Storage サービス (swift) からコンテナーを削除します。

手順

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** のセクションのリストを参照して全オブジェクトが削除されていることを確認します。詳細は、Object Storage サービスからのオブジェクトの削除 を参照してください。
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。

5.4.4. コンテナーへのオブジェクトのアップロード

実際のファイルを Object Storage サービス (swift) に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ために使用できます。

手順

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. アップロードされたオブジェクトを配置するコンテナーの名前をクリックします。コンテナーに擬似フォルダーがすでに存在する場合は、その名前をクリックします。
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。
   - **/** の文字 (例: **Images/myImage.jpg**) を使用して、名前に擬似フォルダーを指定できます。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。

5.4.5. コンテナー間でのオブジェクトのコピー

ダッシュボードを使用して Object Storage サービス (swift) にオブジェクトをコピーします。

手順

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。

5.4.6. Object Storage サービスからのオブジェクトの削除

ダッシュボードを使用して、Object Storage サービス (swift) からオブジェクトを削除します。

手順

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. リストを参照して対象のオブジェクトを特定し、矢印メニューで **オブジェクトの削除** を選択します。

3. **オブジェクトの削除** をクリックして、オブジェクトを削除する操作を確定します。

第6章 SHARED FILE SYSTEMS サービス (MANILA) の設定

Shared File Systems サービス (manila) により、複数のクラウドユーザーインスタンス、ベアメタルノード、またはコンテナーで消費可能な共有ファイルシステムをプロビジョニングすることができます。クラウド管理者は共有種別を作成してファイル共有サービスの準備を行い、エンドユーザーがファイル共有を作成および管理できるようにします。共有ファイルシステムを管理するには、Shared File Systems サービスコマンドラインクライアントを使用します。

前提条件

- Shared File Systems サービスを使用するために、エンドユーザーには少なくとも 1 つの共有種別が必要です。
- **driver_handles_share_servers=false** と設定したバックエンドの場合、クラウド管理者は共有ファイルシステムバックエンドで動的にネットワークを設定するのではなく、必要なネットワークを事前に設定します。
- CephFS-NFS バックエンドの場合、クラウド管理者は、分離されたネットワークと環境引数、およびカスタム **network_data** ファイルを使用して Red Hat OpenStack Platform (RHOSP) director をデプロイし、NFS エクスポート用の分離された 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。
- Compute 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。

6.1. SHARED FILE SYSTEMS サービスバックエンドの設定

クラウド管理者は、Red Hat OpenStack Platform (RHOSP) director を使用して Shared File Systems サービス (manila) をデプロイする場合、ネイティブ CephFS、CephFS-NFS、NetApp、Dell EMC Unity などのサポートされるバックエンドを 1 つ以上選択できます。。

ネイティブ CephFS および CephFS-NFS の詳細は、**director を使用した Red Hat Ceph Storage および Red Hat OpenStack Platform のデプロイ** を参照してください。

サポートされているバックエンドアプライアンスとドライバーの完全なリストについては、Red Hat ナレッジベーシのアーティクル記事 Component, Plug-In, and Driver Support in Red Hat OpenStack Platform の Manila セクションを参照してください。

6.1.1. 複数のバックエンドの設定

バックエンドは、ファイルシステムをエクスポートする Shared File Systems サービス (manila) ドライバーとペアとなるストレージシステムまたはテクノロジーです。Shared File Systems サービスでは、少なくとも 1 つのバックエンドが動作している必要があります。多くの場合、1 つのバックエンドで十分です。ただし、単一の Shared File Systems サービスインストールで複数のバックエンドを使用することもできます。

重要

現在、Red Hat OpenStack Platform (RHOSP) では、Shared File Systems サービスのデプロイメントに対する同じバックエンドの複数インスタンスはサポートされていません。たとえば、同じデプロイメント内に 2 つの Red Hat Ceph Storage クラスターをバックエンドとして追加することはできません。CephFS ネイティブと CephFS-NFS は、異なるプロトコルを備えた 1 つのバックエンドとみなされます。

Shared File Systems サービスのスケジューラーは、ファイル共有作成要求の宛先バックエンドを決定します。Shared File Systems サービスの 1 つのバックエンドは、複数のストレージプールを公開することができます。

複数のバックエンドを設定する場合、スケジューラーは 1 つのストレージプールを選択し、設定されたすべてのバックエンドで公開されるすべてのプールからリソースを作成します。このプロセスはエンドユーザーから抽象化されます。エンドユーザーは、クラウド管理者が公開する機能のみを認識します。

6.1.2. 複数のバックエンドのデプロイ

デフォルトでは、Shared File Systems サービス (manila) のデプロイに使用する環境ファイルには、単一のバックエンドがあります。次の手順例を使用して、2 つのバックエンド (CephFS-NFS バックエンドと NetApp バックエンド) をデプロイします。

重要

同じベンダーの複数のストレージバックエンドをデプロイメントに追加する場合は、この手順を使用して 1 つのバックエンドを設定します。他のバックエンドをカスタムバックエンドとして設定します。詳細は、カスタムバックエンドの設定 および カスタムバックエンドのデプロイ を参照してください。

前提条件

- 2 つ以上のバックエンド
- 外部ソフトウェアコンポーネントを必要とするストレージベンダーのバックエンドドライバーを使用する場合は、デプロイメント中に Shared File Systems サービスの標準コンテナーイメージをオーバーライドする必要があります。カスタムコンテナーイメージ (たとえば、Dell EMC Unity ストレージシステム用の Dell EMC Unity コンテナーイメージ) は、Red Hat Ecosystem Catalog で見つけることができます。

手順

1. ストレージカスタマイズ YAML ファイルを作成して、環境に適した値またはオーバーライドを提供します。

```
$ vi /home/stack/templates/<multiple_backends>.yaml
```

   - **<multiple_backends>** をファイルの名前に置き換えます。

2. ストレージカスタマイズ YAML ファイルを設定して、オーバーライドを含め、複数のバックエンドを有効にします。

```
parameter_defaults:
  ManilaEnabledShareProtocols:
    - NFS
  ManilaNetappLogin: '<login_name>'
  ManilaNetappPassword: '<password>'
  ManilaNetappServerHostname: '<netapp-hostname>'
  ManilaNetappVserver: ‘<netapp-vserver>’
  ManilaNetappDriverHandlesShareServers: 'false'
```

   - 山かっこ **<>** 内の値を、YAML ファイルに応じた正しい値に置き換えます。

3. アンダークラウドホストに **stack** ユーザーとしてログインします。

4. **stackrc** アンダークラウド認証情報ファイルを入手します。

```
$ source ~/stackrc
```

5. ストレージカスタマイズ YAML ファイルを他の環境ファイルと共にスタックに追加し、オーバークラウドをデプロイします。この設定例では、NetApp バックエンドと CephFS-NFS バックエンドを使用して Shared File Systems サービスを有効にします。

```
$ openstack overcloud deploy \
  --timeout 100 \
  --stack overcloud \
  --templates /usr/share/openstack-tripleo-heat-templates \
  -n /usr/share/openstack-tripleo-heat-templates/network_data_ganesha.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-mds.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-
ansible.yaml \
  -r /home/stack/templates/roles/roles_data.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/manila-cephfsganesha-
config.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/manila-netapp-config.yaml \
  -e /home/stack/templates/storage_customizations.yaml \
  ...
```

関連情報

- **ManilaEnabledShareProtocols** パラメーターの詳細は、「許可される NAS プロトコルの変更」を参照してください。

6.1.3. 複数バックエンドのデプロイメントの確認

manila service-list コマンドを使用して、Shared File Systems サービス (manila) のバックエンドが正常にデプロイされていることを確認します。複数のバックエンドでヘルスチェックを使用する場合、バックエンドの 1 つが応答しなくても ping テストは応答を返すため、ping テストはデプロイメントを検証する信頼できる方法ではありません。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

- **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. Shared File Systems サービスのバックエンドのリストを確認します。

```
$ manila service-list

+----+--------+--------+------+---------+-------+----------------------------+
| Id | Binary | Host | Zone | Status | State | Updated_at |
+----+--------+--------+------+---------+-------+----------------------------+
| 2 | manila-scheduler | hostgroup | nova | enabled | up | 2021-03-24T16:49:09.000000 |
| 5 | manila-share | hostgroup@cephfs | nova | enabled | up | 2021-03-24T16:49:12.000000 |
| 8 | manila-share | hostgroup@tripleo_netapp | nova | enabled | up | 2021-03-
24T16:49:06.000000 |
```

正常にデプロイされた各バックエンドのステータスは **enabled** と表示され、状態は **up** が表示されます。

6.1.4. カスタムバックエンドの設定

Red Hat OpenStack Platform (RHOSP) director インストーラーに対応する実装がないカスタムストレージバックエンドをデプロイできます。

同じベンダーの複数のストレージバックエンドをデプロイメントに追加する場合は、複数のバックエンドのデプロイ の手順を使用して、1 つのバックエンドを設定します。他のバックエンドをカスタムバックエンドとして設定します。

カスタムバックエンドをデプロイする際は、ストレージバックエンドベンダーが Shared File Systems サービス (manila) の標準コンテナーイメージではなく、カスタムコンテナーイメージを必要としているか確認してください。カスタムコンテナーイメージは Red Hat Ecosystem Catalog で見つけることができます。

手順

1. ストレージカスタマイズ YAML ファイルを作成して、環境に適した値またはオーバーライドを提供します。

```
$ vi /home/stack/templates/<custom_backend_data>.yaml
```

   - **<custom_backend_data>** をファイルの名前に置き換えます。

2. Shared File Systems サービスに必要なコンテナーイメージを使用して、ストレージカスタマイズ YAML ファイルを設定します。

```
parameter_defaults:
  ContainerManilaShareImage: <custom_container_image_url>
```

   - **<custom_container_image_url>** をカスタムコンテナーイメージの URL に置き換えます。

3. **ControllerExtraConfig** パラメーターを使用して、Shared File Systems サービスに必要なバックエンドを設定します。このパラメーターにより、設定がすべての Controller ノードに確実に適用されます。

```
parameter_defaults:
  ...
  ControllerExtraConfig:
    manila::config::manila_config:
      <backend_name>/<parameter>:
        value: '<parameter_value>'
```

   - **<backend_name>** をカスタムバックエンドの名前に置き換えます。
   - **<parameter>** を、このバックエンドに設定するパラメーターの名前 (**netapp_server_hostname** や **netapp_password** など) に置き換えます。
   - **<parameter_value>** を、パラメーターに設定する値 (**203.0.113.20** や **admin_password** など) に置き換えます。

注記

カスタムロールを使用する場合は、**ControllerExtraConfig** パラメーターの代わりに **[role_name]ExtraConfig** を使用します。**[role_name]** を、カスタムロールの名前に置き換えます。

6.1.5. カスタムバックエンドのデプロイ

カスタムバックエンドの設定 の説明に従って、Red Hat OpenStack Platform (RHOSP) デプロイメント用のカスタムストレージバックエンドを設定したら、それらをデプロイメントに追加できます。

手順

1. ストレージカスタマイズ YAML ファイルを作成して、環境に適した値またはオーバーライドを提供します。

```
$ vi /home/stack/templates/<manila_enabled_share_backends>.yaml
```

   - **<manila_enabled_share_backends>** をファイルの名前に置き換えます。

2. ストレージカスタマイズ YAML ファイルを設定して、有効な共有バックエンドのリストにカスタムバックエンドを追加します。

```
parameter_defaults:
  ControllerExtraConfig:
    manila_user_enabled_backends:
      - '<backend_name>'
```

   - **<backend_name>** を、カスタムバックエンドの名前に置き換えます。

3. ストレージカスタマイズ YAML ファイルを他の環境ファイルと共にスタックに追加し、オーバークラウドをデプロイします。

```
$ openstack overcloud deploy --templates \
-e [your environment files] \
  -e /home/stack/templates/<manila_enabled_share_backends>.yaml
```

6.1.6. バックエンド用のアベイラビリティーゾーンのデプロイメント

アベイラビリティーゾーン (AZ) を作成して、クラウドユーザー向けにクラウドインフラストラクチャーとサービスを論理的にグループ化できます。AZ を障害ドメインにマップし、高可用性、フォールトトレランス、およびリソーススケジューリングのためにリソースを計算できます。たとえば、クラウドユーザーがそのハードウェアを必要とするインスタンスを作成するときに指定できる特定のハードウェアを持つ Compute ノードの AZ を作成できます。

共有は常に AZ に関連付けられます。共有の作成時に AZ パラメーターを設定しない場合、Shared File Systems サービス (manila) は共有を **nova** というデフォルトの AZ に関連付けます。**ManilaXXXAvailabilityZone** パラメーター (**XXX** は特定のバックエンドに関連付けられています) を使用して、Shared File Systems サービスバックエンドに異なる AZ を設定します。AZ の詳細は、**インスタンス作成のためのコンピューティングサービスの設定** の ホストアグリゲートの作成と管理 を参照してください。

重要

既存の共有の AZ パラメーターを変更すると、これらの共有は引き続き元の AZ に関連付けられますが、バックエンドは新しい AZ にマッピングされます。現在、既存の共有の AZ パラメーターを変更するときに、元の AZ と新しくマップされた AZ の間の競合を調整する方法はありません。

手順

1. 以下のパラメーターを Shared File Systems サービス環境ファイルに追加して、2 つの AZ を作成します。

```
parameter_defaults:
 ManilaXXXAvailabilityZone: zone1
 ManilaYYYAvailabilityZone: 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。

```
ManilaCephFSAvailabilityZone
ManilaNetAppAvailabilityZone
```

     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、**CephFS** がゾーン 1 で **NetApp** がゾーン 2 です。

```
parameter_defaults:
 ManilaCephFSAvailabilityZone: zone1
 ManilaNetAppAvailabilityZone: zone2
```

     注記

     /usr/share/openstack-tripleo-heat-templates/deployment/manila/ 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. アンダークラウドホストに **stack** ユーザーとしてログインします。

3. **stackrc** アンダークラウド認証情報ファイルを入手します。

```
$ source ~/stackrc
```

4. 更新された Shared File Systems サービス環境ファイルを他の環境ファイルと共にスタックに追加し、オーバークラウドをデプロイします。

```
$ openstack overcloud deploy --templates \
  -e [your environment files] \
  -e /home/stack/templates/<updated_environment_file>.yaml
```

5. デプロイ後、**availability_zones** 共有タイプの追加仕様を使用して、共有タイプを 1 つ以上の AZ に制限します。クラウドユーザーは、共有タイプによって制限されない限り、AZ に共有を直接作成できます。

関連情報

- 共有タイプの作成
- **Red Hat OpenStack Platform のハードニング** の 共有タイプのアクセス制御。

6.1.7. 許可される NAS プロトコルの変更

Shared File Systems サービス (manila) を使用して、NFS、CephFS、または CIFS ネットワーク接続ストレージ (NAS) プロトコルで共有をエクスポートできます。デフォルトでは、Shared File Systems サービスは、デプロイメント内のバックエンドがサポートする NAS プロトコルをすべて有効にします。

Red Hat OpenStack Platform (RHOSP) 管理者は、**ManilaEnabledShareProtocols** パラメーターを変更して、クラウドで許可するプロトコルのみをリストすることができます。たとえば、デプロイメント内のバックエンドが NFS と CIFS の両方をサポートしている場合、デフォルト値を変更して 1 つのプロトコルのみを有効にすることができます。

すべてのストレージバックエンドドライバーが、CIFS プロトコルをサポートしているわけではありません。CIFS をサポートする認定ストレージシステムの詳細は、Red Hat エコシステムカタログ を参照してください。RHOSP director を使用したサービスの設定に関する詳細は、ベンダーストレージに関するドキュメントを参照してください。

手順

1. source コマンドでオーバークラウドの認証情報ファイルを読み込みます。

```
$ source ~/<credentials_file>
```

   - **<credentials_file>** を認証情報ファイルの名前 (**overcloudrc** など) に置き換えます。

2. ストレージカスタマイズ YAML ファイルを作成して、環境に適した値またはオーバーライドを提供します。

```
$ vi /home/stack/templates/<share_protocols>.yaml
```

   - **<share_protocols>** をファイルの名前に置き換えます。

3. 有効にする NAS プロトコルを使用して **ManilaEnabledShareProtocols** パラメーターを設定します。

```
parameter_defaults:
  ManilaEnabledShareProtocols:
    - NFS
    - CEPHFS
```

4. ストレージカスタマイズ YAML ファイルを他の環境ファイルと共にスタックに追加し、オーバークラウドをデプロイします。

```
$ openstack overcloud deploy --templates \
-e [your environment files] \
  -e /home/stack/templates/<share_protocols>.yaml
```

注記

デプロイメントは設定を検証しません。割り当てる NAS プロトコルは、Shared File Systems サービスのデプロイメントのバックエンドによりサポートされる必要があります。

6.1.8. バックエンド機能の表示

Shared File Systems サービス (manila) のスケジューラーコンポーネントは、複数の要素に基づいて高度な配置決定を行います。これには、容量、プロビジョニング設定、配置ヒント、およびバックエンドストレージシステムドライバーが検出および公開する機能などが含まれます。

手順

- 利用可能な機能を表示するには、以下のコマンドを実行します。

```
$ manila pool-list --detail
+-----------------------------------+---------------------------+
| Property                          | Value                     |
+-----------------------------------+---------------------------+
| name                              | hostgroup@cephfs#cephfs   |
| pool_name                         | cephfs                    |
| total_capacity_gb                 | 1978                      |
| free_capacity_gb                  | 1812                      |
...
| driver_handles_share_servers      | False                     |
| snapshot_support                  | True                      |
| create_share_from_snapshot_support | False                     |
| revert_to_snapshot_support        | False                     |
| mount_snapshot_support            | False                     |
...
+-----------------------------------+---------------------------+
+-----------------------------------+-----------------------------------+
| Property                          | Value                             |
+-----------------------------------+-----------------------------------+
| name                              | hostgroup@tripleo_netapp#aggr1_n1 |
| pool_name                         | aggr1_n1                          |
| total_capacity_gb                 | 6342.1                            |
| free_capacity_gb                  | 6161.99                           |
...
| driver_handles_share_servers      | False                             |
| mount_snapshot_support            | False                             |
| replication_type                  | None                              |
| replication_domain                | None                              |
| sg_consistent_snapshot_support    | host                              |
| ipv4_support                      | True                              |
| ipv6_support                      | False                             |
+-----------------------------------+-----------------------------------+
+-----------------------------------+-----------------------------------+
| Property                          | Value                             |
+-----------------------------------+-----------------------------------+
| name                              | hostgroup@tripleo_netapp#aggr1_n2 |
| pool_name                         | aggr1_n2                          |
| total_capacity_gb                 | 6342.1                            |
| free_capacity_gb                  | 6209.26                           |
...
| snapshot_support                  | True                              |
| create_share_from_snapshot_support | True                              |
| revert_to_snapshot_support        | True                              |
| driver_handles_share_servers      | False                             |
| mount_snapshot_support            | False                             |
| replication_type                  | None                              |
| replication_domain                | None                              |
| sg_consistent_snapshot_support    | host                              |
```

```
| ipv4_support                    | True                              |
| ipv6_support                    | False                             |
+-----------------------------------+-----------------------------------+
```

関連情報

配置の決定に影響を及ぼすには、管理者はファイル共有種別および追加の仕様を使用できます。共有タイプの詳細は、共有タイプの作成 を参照してください。

6.2. 共有タイプの作成

クラウド管理者は、共有タイプを作成して、Shared File Systems サービス (manila) スケジューラーがスケジュール決定を行うために使用するサービスのタイプ、およびドライバーが共有の作成を制御するために使用するサービスのタイプを定義できます。

共有タイプには、バックエンドをフィルタリングするための説明と追加の仕様 (**driver_handles_share_servers** や **snapshot_support** など) が含まれます。Red Hat OpenStack Platform (RHOSP) director は **default** という名前のデフォルト共有タイプで Shared File Systems サービスを設定しますが、director は共有タイプを作成しません。

クラウドユーザーが Shared File Systems サービスを使用するには、少なくとも 1 つの共有タイプが必要であり、ユーザーは使用可能な共有タイプに一致する共有のみを作成できます。

デフォルトでは、共有タイプはパブリックです。これは、すべてのクラウドプロジェクトで使用できることを意味します。ただし、特定のプロジェクト内で使用するプライベート共有タイプを作成できます。

次の手順例では、**driver_handles_share_servers** パラメーター (DHSS) を使用します。これは **true** または **false** に設定できます。

- CephFS-NFS およびネイティブ CephFS の場合、DHSS を **false** に設定します。
- 他のバックエンドの場合は、DHSS を **true** または **false** に設定できます。ストレージカスタマイズ環境ファイルの **Manila<backend>DriverHandlesShareServers** パラメーターの値と一致するように DHSS 値を設定します。たとえば、NetApp バックエンドを使用する場合、このパラメーターは **ManilaNetappDriverHandlesShareServers** になります。

手順

1. オーバークラウドをデプロイした後、次のコマンドを実行して共有タイプを作成します。

   ```
   $ manila type-create default <driver_handles_share_servers>
   ```

   - **<driver_handles_share_servers>** を **true** または **false** に置き換えます。

2. デフォルトの共有タイプに仕様を追加するか、別のバックエンドで使用する追加の共有タイプを作成します。この例では、CephFS バックエンドを選択するようにデフォルトの共有タイプを設定します。また、NetApp **driver_handles_share_servers=true** バックエンドを使用する追加の共有タイプを設定します。

   ```
   $ manila type-create default false \
     --extra-specs share_backend_name='cephfs'
   $ manila type-create netapp true \
     --extra-specs share_backend_name='tripleo_netapp'
   ```

関連情報

関連情報

- プライベート共有タイプを作成する方法、または追加の共有タイプオプションを設定する方法の詳細は、Red Hat OpenStack Platform のハードニング を参照してください。

6.3. 共有タイプの共通機能の比較

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

表6.1 共有種別の機能

機能	値	説明
driver_handles_share_servers	true または false	共有ネットワークを使用してファイル共有を作成する権限を付与します。
snapshot_support	true または false	ファイル共有のスナップショットを作成する権限を付与します。
create_share_from_snapshot_support	true または false	共有スナップショットのクローンを作成する権限を付与します。
revert_to_snapshot_support	true または false	ファイル共有を最新のスナップショットに戻す権限を付与します。
mount_snapshot_support	true または false	スナップショットをエクスポートおよびマウントする権限を付与します。
replication_type	dr	障害復旧用のレプリカを作成する権限を付与します。1度に許可されるアクティブなエクスポートは1つだけです。
	readable	読み取り専用レプリカを作成する権限を付与します。1度に許可される書き込み可能なアクティブなエクスポートは1つだけです。
	writable	読み取り/書き込みレプリカを作成する権限を付与します。共有ごとに1度に任意の数のアクティブなエクスポートが許可されます。
availability_zones	1つまたは複数のアベイラビリティーゾーンのリスト	リスト表示されるアベイラビリティーゾーンでのみ共有を作成する権限を付与します。

6.4. 管理/管理解除を使用した共有の追加と削除

クラウド管理者は、Shared File Systems サービス (manila) の管理/管理解除機能を使用して、ストレージにすでに存在するファイル共有を管理できます。アクセス権の付与、マウント、サイズ変更などの操作を、Shared File Systems サービス共有に対して実行する場合と同じ方法で、管理された共有に対して実行できます。

driver_handles_share_servers パラメーター (DHSS) が true に設定されている共有と、DHSS が false に設定されている共有のライフサイクルを管理できます。DHSS=true 共有を管理するには、クラウド管理者は共有を含む共有サーバーも管理する必要があります。

共有の管理を解除すると、共有は削除されずに、Shared File Systems サービスの管理から削除されます。共有に依存スナップショットまたは共有レプリカがある場合、スナップショットまたは共有レプリカが削除されている場合にのみ、Shared File Systems サービスから共有を削除できます。

制限事項

- ドライバーは、管理/管理解除機能をサポートする必要があります。
- 管理/管理解除機能は、ネイティブ CephFS または CephFS-NFS バックエンドをサポートしません。CephFS 共有を Shared File Systems サービスの管理から削除できます。ただし、既存の CephFS 共有を Shared File Systems サービスの管理下に置くことはできません。
- 共有を Shared File Systems サービスの管理下に置くと、既存のクライアントは切断されます。Shared File Systems サービスの管理から共有を削除しても、既存のクライアントは接続されたままになります。

手順

1. ファイル共有を管理します。

```
manila manage
--name <name>
--description <description>
--share_type <share-type>
--driver_options [<key=value> [<key=value> ...]]
                 [--public] [--share_server_id <share-server-id>] \
                 [--wait] <service_host> <protocol> <export_path>
```

   - 山かっこ **<>** 内の値を、環境に応じた正しい値に置き換えます。

2. 共有が利用可能であることを確認します。

```
$ manila show <name>
```

3. ファイル共有の管理を解除します。

```
manila unmanage [--wait] <name>
```

6.5. 共有ファイルシステムのネットワークの計画

エンドユーザークライアントが Red Hat OpenStack Platform (RHOSP) 仮想マシン、ベアメタルサーバー、およびコンテナーで実行されるワークロードに共有を接続できるように、クラウド上のネットワークのプランニングを行います。

クラウドユーザーに必要なセキュリティーと分離のレベルに応じて、**driver_handles_share_servers** パラメーター (DHSS) を **true** または **false** に設定できます。

DHSS=true
DHSS パラメーターを **true** に設定すると、Shared File Systems サービス (manila) を使用して、分離された共有サーバーを持つエンドユーザー定義の共有ネットワークに共有をエクスポートできます。ユーザーは、セルフサービス共有ネットワーク上でワークロードをプロビジョニングし、専用ネットワークセグメント上の分離された NAS ファイルサーバーが、確実に共有をエクスポート可能にすることができます。

クラウド管理者は、分離ネットワークをマッピングする物理ネットワークが、ストレージインフラストラクチャーまで拡張されていることを確認する必要があります。また、使用しているストレージシステムが、ネットワークセグメントをサポートしていることを確認する必要もあります。NetApp ONTAP および Dell EMC PowerMax、Unity、ならびに VNX 等のストレージシステムは、GENEVE または VXLAN 等の仮想オーバーレイセグメント化スタイルをサポートしません。

オーバーレイネットワークの代わりに、次のいずれかを実行できます。

- プロジェクトネットワークには VLAN ネットワークを使用します。
- 共有プロバイダーネットワーク上で VLAN セグメントを許可します。
- ストレージシステムにすでに接続されている既存のセグメント化されたネットワークへのアクセスを提供します。

DHSS=false
DHSS パラメーターを **false** に設定すると、クラウドユーザーは自身の共有ネットワーク上に共有を作成できなくなります。クラウド管理者が設定したネットワークにクライアントを接続する必要があります。

クラウド管理者は、director を通じて専用の共有ストレージネットワークを作成できます。たとえば、標準の Director テンプレートを使用してネイティブ CephFS バックエンドをデプロイすると、**Storage** と呼ばれる共有プロバイダーネットワークが生成されます。NFS バックエンド経由で CephFS をデプロイすると、**StorageNFS** と呼ばれる共有プロバイダーネットワークが生成されます。クラウドユーザーは、共有にアクセスするためにクライアントを共有ストレージネットワークに接続する必要があります。

すべての共有ファイルシステムストレージドライバーが DHSS=true と DHSS=false の両方をサポートしているわけではありません。DHSS=true と DHSS=false の両方で、データパスのマルチテナント分離が確実となります。ただし、セルフサービスモデルの一部としてテナントワークロードのネットワークパスマルチテナント分離が必要な場合は、DHSS=true をサポートするバックエンドを備えた Shared File Systems サービスをデプロイメントする必要があります。

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

6.6. ファイル共有へのネットワーク接続の確保

ファイル共有に接続するには、クライアントがその共有の 1 つ以上のエクスポート場所にネットワーク接続できる必要があります。

クラウド管理者が共有タイプの **driver_handles_share_servers** パラメーター (DHSS) を true に設定すると、クラウドユーザーは、Compute インスタンスが接続するネットワークの詳細を使用して共有ネットワークを作成できます。これにより、クラウドユーザーは、共有を作成するときに共有ネットワークを参照できるようになります。

クラウド管理者が共有タイプの DHSS パラメーターを false に設定した場合、クラウドユーザーは、クラウド管理者が設定した共有ストレージネットワークに、Compute インスタンスを接続する必要があります。共有ネットワークへのネットワーク接続を設定して検証する方法の詳細は、「共有にアクセスするための共有ネットワークへの接続」 を参照してください。

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

気付かぬままにシステムリソースをすべて消費してしまうのを防ぐために、クラウド管理者はクォータを設定することができます。クォータとは、運用上の制限です。Shared File Systems サービス (manila) は、デフォルトでいくつかの実質的な制限を強制します。これらの制限はデフォルトクォータと呼ばれます。クラウド管理者はデフォルトのクォータを上書きし、個々のプロジェクトに異なる消費制限を設定することができます。

6.7.1. プロジェクト、ユーザー、および共有タイプのクォータを更新する

クラウド管理者は、**manila quota-show** コマンドを使用して、プロジェクトまたはユーザーのクォータをリスト表示できます。

プロジェクトのすべてのユーザーもしくは特定のプロジェクトユーザー、またはプロジェクトユーザーが使用する共有種別のクォータを更新することができます。選択したターゲットについて、以下のクォータを更新できます。

- **shares**: 作成可能なファイル共有の数
- **snapshots**: 作成可能なスナップショットの数
- **gigabytes**: 全ファイル共有に割り当てることができる総容量 (GB)
- **snapshot-gigabytes**: ファイル共有のすべてのスナップショットに割り当て可能な合計サイズ (GB)
- **share-networks**: 作成可能な共有ネットワークの合計数
- **share_groups**: 作成可能な共有グループの合計数
- **share_group_snapshots**: 作成可能な共有グループスナップショットの合計数
- **share-replicas**: 作成可能な共有レプリカの合計数
- **replica-gigabytes**: すべての共有レプリカに割り当てることのできる合計サイズ (GB)。

注記

share-type のクォータは、プロジェクトレベルでのみ指定することができます。特定のプロジェクトユーザーに **share-type** のクォータを設定することはできません。

重要

以下の手順では、値を慎重に入力してください。Shared File Systems サービスは、誤った値を検出したり報告したりしません。

手順

1. 以下のコマンドを使用してクォータを表示することができます。**--user** オプションを含めると、指定したプロジェクトの特定ユーザーのクォータを表示できます。**--user** オプションを指

定しないと、指定したプロジェクトのすべてのユーザーに適用されるクォータを表示できます。
同様に、オプションの **--share-type** を指定すると、プロジェクトに関連する特定の共有種別のクォータを表示することができます。**--user** オプションおよび **--share-type** オプションは相互排他的です。

```
$ manila quota-show
```

- プロジェクトの例:

```
$ manila quota-show \
  --project af2838436f3f4cf6896399dd97c4c050
+----------------------+---------------------------------+
| Property             | Value                           |
+----------------------+---------------------------------+
| gigabytes            | 1000                            |
| id                   | af2838436f3f4cf6896399dd97c4c050 |
| replica_gigabytes    | 1000                            |
| share_group_snapshots | 50                             |
| share_groups         | 49                              |
| share_networks       | 10                              |
| share_replicas       | 100                             |
| shares               | 50                              |
| snapshot_gigabytes   | 1000                            |
| snapshots            | 50                              |
+----------------------+---------------------------------+
```

- プロジェクトユーザーの例:

```
$ manila quota-show \
  --project af2838436f3f4cf6896399dd97c4c050 \
  --user 81ebb491dd0e4c2aae0775dd564e76d1
+----------------------+---------------------------------+
| Property             | Value                           |
+----------------------+---------------------------------+
| gigabytes            | 500                             |
| id                   | af2838436f3f4cf6896399dd97c4c050 |
| replica_gigabytes    | 1000                            |
| share_group_snapshots | 50                             |
| share_groups         | 49                              |
| share_networks       | 10                              |
| share_replicas       | 100                             |
| shares               | 25                              |
| snapshot_gigabytes   | 1000                            |
| snapshots            | 50                              |
+----------------------+---------------------------------+
```

- 特定の共有種別のプロジェクトの例:

```
$ manila quota-show \
  --project af2838436f3f4cf6896399dd97c4c050 \
  --share-type dhss_false
+--------------------+--------------------------------+
| Property           | Value                          |
+--------------------+--------------------------------+
```

```
| gigabytes          | 1000                                |
| id                 | af2838436f3f4cf6896399dd97c4c050 |
| replica_gigabytes  | 1000                                |
| share_replicas     | 100                                 |
| shares             | 15                                  |
| snapshot_gigabytes | 1000                                |
| snapshots          | 50                                  |
+--------------------+-----------------------------------+
```

2. **manila quota-update** コマンドを使用してクォータを更新します。すべてのプロジェクトユーザー、特定のプロジェクトユーザー、またはプロジェクトの共有種別のクォータを更新できます。

   - プロジェクトのすべてのユーザーについてクォータを更新します。

     ```
     $ manila quota-update <id> \
       [--shares <share_quota> \
       --gigabytes <share_gigabytes_quota> \
       …]
     ```

     <id> をプロジェクト ID に置き換えます。この値はプロジェクト名ではなく、プロジェクト ID である必要があります。

   - プロジェクト内の特定ユーザーのクォータを更新します。

     ```
     $ manila quota-update <id> \
       --user <user_id> \
       [--shares <new_share_quota> \
       --gigabytes <new_share_gigabytes_quota> \
       …]
     ```

     - **<id>** をプロジェクト ID に置き換えます。この値はプロジェクト名ではなく、プロジェクト ID である必要があります。
     - **<user_id>** をユーザー ID に置き換えます。値はユーザー名ではなく、ユーザー ID である必要があります。

   - 特定の共有種別を使用するすべてのユーザーについてクォータを更新します。

     ```
     $ manila quota-update <id> \
       --share-type <share_type> \
       [--shares <new_share_quota>30
       --gigabytes <new-share_gigabytes_quota> \
       …]
     ```

     - **<id>** をプロジェクト ID に置き換えます。この値はプロジェクト名ではなく、プロジェクト ID である必要があります。
     - **<share_type>** を、クォータを適用する共有種別の名前または ID に置き換えてください。

検証

- **quota-update** コマンドは出力を生成しません。**quota-show** コマンドを使用して、クォータが正常に更新されたことを確認します。

6.7.2. プロジェクト、ユーザー、共有タイプのクォータのリセット

クォータのオーバーライドを削除して、クォータをデフォルト値に戻すことができます。ターゲットエンティティーは、オーバーライドのないすべてのプロジェクトに適用されるデフォルトのクォータによって制限されます。

手順

- **manila quota-delete** コマンドを使用して、クォータをデフォルト値に戻します。すべてのプロジェクトユーザー、特定のプロジェクトユーザー、またはプロジェクトの共有種別について、クォータをデフォルト値に戻すことができます。
  - プロジェクトクォータをリセットします。

    ```
    $ manila quota-delete --project <id>
    ```

    <id> をプロジェクト ID に置き換えます。この値はプロジェクト名ではなく、プロジェクト ID である必要があります。

  - 特定ユーザーのクォータをリセットします。

    ```
    $ manila quota-delete --project <id> --user <user_id>
    ```

    - **<id>** をプロジェクト ID に置き換えます。この値はプロジェクト名ではなく、プロジェクト ID である必要があります。
    - **<user_id>** をユーザー ID に置き換えます。値はユーザー名ではなく、ユーザー ID である必要があります。

  - プロジェクトユーザーが使用する共有種別のクォータをリセットします。

    ```
    $ manila quota-delete --project <id> --share-type <share_type>
    ```

    - **<id>** をプロジェクト ID に置き換えます。この値はプロジェクト名ではなく、プロジェクト ID である必要があります。
    - **<share_type>** を、クォータをリセットする必要がある共有種別の名前または ID に置き換えてください。

検証

1. **quota-delete** コマンドは出力を生成しません。**quota-show** コマンドを使用して、クォータが正常にリセットされたかどうかを確認します。
2. すべてのプロジェクトのデフォルトクォータをリスト表示します。デフォルトクォータは、オーバーライドのないプロジェクトに適用されます。

   ```
   $ manila quota-class-show default
   ```

6.7.3. Shared File Systems サービスプロジェクトのデフォルトクォータ値の更新

クラウド管理者は、まだクォータのオーバーライドがないすべてのプロジェクトに適用されるデフォルトのクォータを更新できます。

手順

1. **manila quota-class-update** コマンドの使用状況のステートメントを表示します。

```
$ manila help quota-class-update
usage: manila quota-class-update [--shares <shares>] [--snapshots <snapshots>]
                                 [--gigabytes <gigabytes>]
                                 [--snapshot-gigabytes <snapshot_gigabytes>]
                                 [--share-networks <share_networks>]
                                 [--share-replicas <share_replicas>]
                                 [--replica-gigabytes <replica_gigabytes>]
                                 <class_name>
```

注記

パラメーター **<class_name>** は位置引数です。クォータが設定されるクォータクラスを特定します。このパラメーターの値を **default** に設定します。他のクォータクラスはサポートされません。

以下のオプションパラメーターの値をすべて更新できます。

- **--shares <shares>**: **shares** クォータの新しい値を追加します。
- **--snapshots <snapshots>**: **snapshots** クォータの新しい値を追加します。
- **--gigabytes <gigabytes>**: **gigabytes** クォータの新しい値を追加します。
- **--snapshot-gigabytes <snapshot_gigabytes>** または **--snapshot_gigabytes <snapshot_gigabytes>**: **snapshot_gigabytes** クォータの新しい値を追加します。
- **--share-networks <share_networks>** または **--share_networks <share_networks>**: **share_networks** クォータの新しい値を追加します。
- **--share-replicas <share_replicas>**、**--share_replicas <share_replicas>**、または **--replicas <share_replicas>**: **share_replicas** クォータの新しい値を追加します。
- **--replica-gigabytes <replica_gigabytes>** または **--replica_gigabytes <replica_gigabytes>**: **replica_gigabytes** クォータの新しい値を追加します。

2. 使用状況のステートメントからの情報を使用して、デフォルトのクォータを更新します。以下の例では、ファイル **shares** および **gigabytes** のデフォルトクォータを更新しています。

```
$ manila quota-class-update default \
  --shares 30 \
  --gigabytes 512
$ manila quota-class-show default
+-----------------------+---------+
| Property              | Value   |
+-----------------------+---------+
| gigabytes             | 512     |
| id                    | default |
| replica_gigabytes     | 1000    |
| share_group_snapshots | 50      |
| share_groups          | 50      |
| share_networks        | 10      |
```

```
| share_replicas        | 100     |
| shares                | 30      |
| snapshot_gigabytes    | 1000    |
| snapshots             | 50      |
+----------------------+---------+
```

第7章 SHARED FILE SYSTEMS サービスを使用した操作の実行 (MANILA)

クラウドユーザーは、Shared File Systems サービス (manila) で利用可能な共有タイプから共有を作成および管理できます。共有ファイルシステムを管理するには、Shared File Systems サービスコマンドラインクライアントを使用します。

7.1. 共有タイプのリスト

クラウドユーザーは、ファイル共有の作成時に共有種別を指定する必要があります。Shared File Systems サービス (manila) を使用するには、使用可能な共有タイプが少なくとも 1 つ必要です。また、作成できるのは、使用可能な共有タイプに一致する共有のみです。クラウド管理者は、共有タイプを設定して、Shared File Systems サービススケジューラーがスケジュールの決定を行うために使用するサービスのタイプ、およびドライバーが共有の作成を制御するために使用するサービスのタイプを定義します。

手順

- 利用可能な共有タイプを表示します。

```
$ manila type-list
```

  コマンド出力には、使用可能な共有タイプの名前と ID が表示されます。

7.2. NFS、CEPHFS、または CIFS 共有の作成

クラウドユーザーは、CephFS-NFS、ネイティブ CephFS、または CIFS 共有を作成して、データを読み書きできます。

共有を作成するときは、共有プロトコルと共有のサイズをギガバイト単位で指定する必要があります。また、**share-type**、**share-network**、および **name** コマンドオプションを含めることもできます。

```
$ manila create [--share-type <share_type>] \
      [--share-network <share_network>]  \
      [--name <share_name>] <share_protocol> <GB>
```

コマンド例では、次の値を置き換えます。

- **<share_type>**: 指定された共有タイプに関連付けられた設定を適用します。
  - オプション: 共有タイプを指定しない場合は、**default** の共有タイプが使用されます。
- **<share_network>**: 共有ネットワークの名前:
  - 共有タイプの **driver_handles_share_servers** が、**true** に設定されている場合は必須です。
  - 共有タイプの **driver_handles_share_servers** が、**false** に設定されている場合はサポートされません。
  - CephFS-NFS およびネイティブ CephFS ではサポートされません。これらのプロトコルは、**driver_handles_share_servers** が **true** に設定されている共有タイプをサポートしません。

- **<share_name>**: 共有の名前:
  - オプション: 共有に名前を付ける必要はなく、名前が一意である必要もありません。
- **<share_protocol>**: 使用する共有プロトコル:
  - CephFS-NFS の場合は、**<share_protocol>** を **nfs** に置き換えます。
  - ネイティブ CephFS の場合は、**<share_protocol>** を **cephfs** に置き換えます。
  - NFS または CIFS プロトコルをサポートする他のストレージバックエンド (たとえば、NetApp または Dell EMC ストレージバックエンド) の場合は、**<share_protocol>** を **nfs** または **cifs** に置き換えます。
- **<GB>**: 共有のサイズ (GB 単位)

7.2.1. DHSS=true での NFS または CIFS 共有の作成

クラウド管理者が共有タイプの追加仕様 **driver_handles_share_servers=true** を使用してセルフサービス共有ネットワークをアクティブ化すると、クラウドユーザーは共有ネットワークに独自のセキュリティーサービスを追加して、NFS または CIFS 共有を作成およびエクスポートできます。ネイティブ CephFS プロトコルは共有ネットワークをサポートしません。

セキュリティーサービスを追加するには、Active Directory サーバーを表す共有ネットワークとセキュリティーサービスリソースを作成する必要があります。その後、セキュリティーサービスを共有ネットワークに関連付けて、NFS または CIFS 共有を作成およびエクスポートできます。

手順

1. 共有ネットワークを作成します。

```
$ manila share-network-create --name <network-name> \
   --neutron-net-id <25d1e65c-d961-4f22-9476-1190f55f118f> \
   --neutron-subnet-id <8ba20dce-0ca5-4efd-bf1c-608d6bceffe1>
```

   - **<network-name>** を、NFS または CIFS 共有に使用する共有ネットワーク名に置き換えます。
   - **neutron-net-id** と **neutron-subnet-id** を共有ネットワークの正しい値に置き換えます。

2. Active Directory サーバーを表すセキュリティーサービスリソースを作成します。

```
$ manila security-service-create <active_directory> \
        --dns-ip <192.02.12.10> \
        --domain <domain-name.com> \
        --user <Administrator> \
        --password <password> \
        --name <AD-service>
```

   - 山かっこ **<>** の値を、セキュリティーサービスリソースに応じた適切な詳細情報に置き換えます。

3. セキュリティーサービスリソースを共有ネットワークに関連付けます。

```
$ manila share-network-security-service-add \
  <network-name> <AD-service>
```

4. NFS または CIFS 共有を作成します。
   - 10 GB NFS の例:

     ```
     $ manila create --name <nfs-share> --share-type <netapp> \
       --share-network <nfs-network> nfs 10
     ```

   - 20 GB CIFS の例:

     ```
     $ manila create --name <cifs-share> --share-type dhss_true \
       --share-network <cifs-network> cifs 20
     ```

     - 山かっこ **<>** の値を、NFS または CIFS 共有に応じた適切な詳細情報に置き換えます。

7.2.2. DHSS=false での NFS、CephFS、または CIFS 共有の作成

クラウド管理者が、共有タイプの追加仕様 **driver_handles_share_servers=false** を使用してセルフサービス共有ネットワークを非アクティブ化する場合、ストレージシステムに対して Active Directory サービスを事前に設定する必要があります。この設定を実行する方法は、ストレージベンダーのドキュメントを参照してください。

DHSS=false の場合、共有ストレージネットワークはクラウド管理者によって事前に設定されており、クラウドユーザーは **share-network** コマンドオプションを使用せずに共有を作成できます。

手順

- DHSS=false の場合は、NFS、ネイティブ CephFS、または CIFS 共有を作成します。これらの例では **name** を指定していますが、**share-type** または **share-network** は指定していません。**default** の共有タイプと、クラウド管理者によって設定された共有ストレージネットワークが使用されます。
  - **share-01** という名前の 10 GB NFS 共有を作成します。

    ```
    $ manila create --name share-01 nfs 10
    ```

  - **share-02** という名前の 15 GB のネイティブ CephFS 共有を作成します。

    ```
    $ manila create --name share-02 cephfs 15
    ```

  - **share-03** という名前の 20 GB CIFS 共有を作成します。

    ```
    $ manila create --name share-03 cifs 20
    ```

7.3. ファイル共有とエクスポート情報のリスト表示

Shared File Systems サービス (manila) で NFS、CephFS、または CIFS 共有が正常に作成されたことを確認するには、次の手順を実行して共有をリスト表示し、エクスポート場所とパラメーターを表示します。

手順

1. ファイル共有をリスト表示します。

```
$ manila list

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

2. ファイル共有のエクスポート場所を表示します。

```
$ manila share-export-location-list <share>

 +-------------------------------------------------------------------
 | Path
 |  198.51.100.13:/volumes/_nogroup/e840b4ae-6a04-49ee-9d6e-67d4999fbc01
 +-------------------------------------------------------------------
```

- **<share>** は、共有名または共有 ID に置き換えます。

3. ファイル共有のパラメーターを表示します。

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

注記

「NFS、ネイティブ CephFS、または CIFS 共有のマウント」 で説明されているように、エクスポート場所を使用して共有をマウントします。

7.4. 共有ファイルシステムでのデータのスナップショット作成

スナップショットは、共有上のデータの読み取り専用の特定の時点におけるコピーです。スナップショットを使用して、誤ったデータ削除またはファイルシステムの破損により失われたデータを復元できます。スナップショットはバックアップよりも領域の効率が高く、Shared File Systems サービス (manila) のパフォーマンスには影響を与えません。

前提条件

- 親共有の **snapshot_support** パラメーターが**true** です。以下のコマンドを実行して確認できます。

```
$ manila show | grep snapshot_support
```

手順

1. クラウドユーザーとして、共有のスナップショットを作成します。

```
$ manila snapshot-create [--name <snapshot_name>] <share>
```

- **<share>** を、スナップショットを作成する共有の名前または ID に置き換えてください。
- オプション: **<snapshot_name>** をスナップショットの名前に置き換えてください。

出力例

```
+-------------------+--------------------------------------+
| Property          | Value                                |
+-------------------+--------------------------------------+
| id                | dbdcb91b-82ba-407e-a23d-44ffca4da04c |
| share_id          | ee7059aa-5887-4b87-b03e-d4f0c27ed735 |
| share_size        | 1                                    |
| created_at        | 2022-01-07T14:20:55.541084           |
| status            | creating                             |
| name              | snapshot_name                        |
| description       | None                                 |
| size              | 1                                    |
| share_proto       | NFS                                  |
| provider_location | None                                 |
| user_id           | 6d414c62237841dcbe63d3707c1cdd90     |
| project_id        | 041ff9e24eba469491d770ad8666682d     |
+-------------------+--------------------------------------+
```

2. スナップショットが作成されたことを確認します。

```
$ manila snapshot-list --share-id <share>
```

<share> を、スナップショットを作成した共有の名前または ID に置き換えてください。

7.4.1. スナップショットからの共有の作成

スナップショットから共有を作成できます。スナップショットの作成元である親共有の共有種別の **driver_handles_share_servers** が **true** に設定されている場合、新しい共有が親と同じファイル共有ネットワークに作成されます。

注記

親共有の共有種別の **driver_handles_share_servers** が **true** に設定されている場合には、スナップショットから作成する共有のファイル共有ネットワークを変更することはできません。

前提条件

- **create_share_from_snapshot_support** 共有属性が **true** に設定されている。
  共有タイプの詳細は、共有タイプの一般的な機能の比較 を参照してください。
- スナップショットの **status** 属性が **available** に設定されている。

手順

1. 新規共有に必要なデータが含まれる共有スナップショットの ID を取得します。

```
$ manila snapshot-list
```

2. スナップショットから作成されたファイル共有は、スナップショットよりも大きくなりますが、小さくすることはできません。スナップショットのサイズを取得します。

```
$ manila snapshot-show <snapshot-id>
```

3. スナップショットからファイル共有を作成します。

```
$ manila create <share_protocol> <size> \
  --snapshot-id <snapshot_id> \
  --name <name>
```

- **<share_protocol>** を、NFS 等のプロトコルに置き換えます。
- **<size>** を、作成するファイル共有のサイズ (GiB 単位) に置き換えます。
- **<snapshot_id>** を、スナップショット ID に置き換えます。
- **<name>** を、新しいファイル共有の名前に置き換えます。

4. ファイル共有をリスト表示して、ファイル共有が正常に作成されたことを確認します。

```
$ manila list
```

5. 新しいファイル共有のプロパティーを表示します。

```
$ manila show <name>
```

検証

スナップショットを作成したら、スナップショットが利用可能であることを確認します。

- スナップショットをリスト表示して、スナップショットが利用可能であることを確認します。

```
$ manila snapshot-list
```

7.4.2. スナップショットの削除

ファイル共有のスナップショットを作成する場合、ファイル共有から作成されたスナップショットをすべて削除するまで、そのファイル共有を削除することはできません。

手順

1. 削除するスナップショットを特定し、その ID を取得します。

```
$ manila snapshot-list
```

2. スナップショットを削除します。

```
$ manila snapshot-delete <snapshot>
```

注記

削除するそれぞれのスナップショットについて、この手順を繰り返します。

3. スナップショットを削除したら、以下のコマンドを実行してスナップショットが削除されたことを確認します。

```
$ manila snapshot-list
```

–

7.5. 共有にアクセスするための共有ネットワークへの接続

driver_handles_share_servers パラメーター (DHSS) が **false** の場合、共有は、クラウド管理者が使用可能にした共有プロバイダーネットワークにエクスポートされます。エンドユーザーは、ファイル共有にアクセスするために、Compute インスタンス等のクライアントを共有プロバイダーネットワークに接続する必要があります。

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

注記

この例では IPv4 アドレス指定を使用していますが、手順は IPv6 の場合とまったく同じです。

手順

1. ポートからの Egress パケットは許可するが、未確立接続からの Ingress パケットは拒否する StorageNFS ポート用のセキュリティーグループを作成します。

```
$ 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"
```

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: '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='198.51.100.160', 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

この例では、StorageNFS ネットワーク上の StorageNFS サブネットは、IP アドレス 198.51.100.160 を **nfs-port0** に割り当てています。StorageNFS サブネットの詳細は、**Director を使用した Red Hat Ceph Storage および Red Hat OpenStack Platform のデプロイ** の 共有プロバイダー StorageNFS ネットワークの設定 を参照してください。

3. Compute インスタンスに **nfs-port0** を追加します。

```
$ openstack server add port instance0 nfs-port0
$ openstack server list -f yaml
- Flavor: m1.micro
  ID: 0b878c11-e791-434b-ab63-274ecfc957e8
  Image: manila-test
  Name: demo-instance0
  Networks: demo-network=198.51.100.4, 10.0.0.53; StorageNFS=198.51.100.160
  Status: ACTIVE
```

Compute インスタンスには、プライベートアドレスとフローティングアドレスに加えて、StorageNFS ネットワーク上のポート (IP アドレスは 198.51.100.160) が割り当てられます。NFS 共有のアドレスへのアクセスが許可されている場合、この IP アドレスを使用してその共有をマウントできます。

注記

このアドレスのインターフェイスをアクティブ化するには、Compute 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。

7.6. ネットワークとインスタンス間の IPV6 インターフェイスの設定

ファイル共有をエクスポートする共有ネットワークが IPv6 アドレス設定を使用する場合、セカンダリーインターフェイスの DHCPv6 で問題が発生する可能性があります。この問題が発生した場合は、インスタンスで IPv6 インターフェイスを手動で設定します。

前提条件

- 共有にアクセスするための共有ネットワークへの接続

手順

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

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

3. インターフェイスをアクティベートします。

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

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

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

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

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

7.7. エンドユーザークライアントに共有アクセスを許可する

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

インスタンスの IP アドレスを使用して、クライアントコンピューティングインスタンスに NFS 共有へのアクセスを許可します。CIFS 共有の **user** ルールと CephFS 共有の **cephx** ルールは同様のパターンに従います。**user** および **cephx** アクセスタイプでは、必要に応じて、複数のクライアントで同じ **clientidentifier** を使用できます。

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

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

以下の値を置き換えます。

- **share**: 「NFS、CephFS、または CIFS 共有の作成」 で説明されているように、作成された共有の共有名または ID。
- **accesstype**: ファイル共有で要求されるアクセスの種別。以下に種別の例を示します。

  - **user**: ユーザーまたはグループ名で認証する場合に使用します。
  - **ip**: IP アドレスでインスタンスを認証する場合に使用します。
  - **cephx**: ネイティブ CephFS クライアントユーザー名による認証に使用します。

> **注記**
>
> アクセスの種別は、ファイル共有のプロトコルにより異なります。CIFS の場合、**user** を使用できます。NFS 共有の場合、**ip** を使用する必要があります。ネイティブ CephFS ファイル共有の場合は、**cephx** を使用する必要があります。

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

7.7.1. NFS 共有へのアクセスの許可

クラウドユーザーは、IP アドレスを通じて NFS 共有へのアクセスを提供できます。

> **注記**
>
> IPv4 または IPv6 アドレスでは次の手順を使用できます。

手順

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

```
$ openstack server list -f yaml
- Flavor: m1.micro
  ID: 0b878c11-e791-434b-ab63-274ecfc957e8
  Image: manila-test
  Name: demo-instance0
  Networks: demo-network=198.51.100.4, 10.0.0.53;
  StorageNFS=198.51.100.160
  Status: ACTIVE

$ manila access-allow <share> ip 198.51.100.160
```

注記

共有へのアクセスには、独自の ID **accessid** があります。

```
+-----------------+--------------------------------------+
| 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       | 198.51.100.160
| access_level    | rw
| state           | queued_to_apply
| id              | 875c6251-c17e-4c45-8516-fe0928004fff
+-----------------+--------------------------------------+
```

検証

- アクセス設定が正常に完了したことを確認します。

```
$ manila access-list <share>

+--------------+------------+--------------+--------------+--------+ ...
| id           | access_type | access_to    | access_level | state  | ...
+--------------+------------+--------------+--------------+--------+
| 875c6251-... | ip         | 198.51.100.160 | rw         | active | ...
+--------------+------------+--------------+--------------+---------+ ...
```

7.7.2. ネイティブ CephFS 共有へのアクセスの許可

クラウドユーザーは、Ceph クライアントユーザー名を通じてネイティブ CephFS 共有へのアクセスを提供できます。Shared File Systems サービス (manila) では、既存の Ceph ユーザーを使用できないため、固有の Ceph クライアントユーザー名を作成する必要があります。

共有をマウントするには、Ceph クライアントのユーザー名とアクセスキーが必要です。アクセスキーは、Shared File Systems サービス API を使用して取得できます。デフォルトでは、アクセスキーはプロジェクト namespace 内のすべてのユーザーに表示されます。同じユーザーに、プロジェクト namespace 内の別々の共有へのアクセスを許可できます。その後、ユーザーはクライアントマシンで CephFS カーネルクライアントを使用して共有にアクセスできます。

重要

ネイティブの CephFS ドライバーは、信頼できるクライアントでのみ使用してください。ネイティブ CephFS バックエンドセキュリティーの詳細は、**director と共に Red Hat Ceph Storage および Red Hat OpenStack Platform をデプロイする** の ネイティブ CephFS バックエンドのセキュリティー を参照してください。

手順

1. ユーザーにネイティブ CephFS 共有へのアクセスを許可します。

```
$ manila access-allow <share> cephx <user>
```

- **<share>** は、共有名または共有 ID に置き換えます。
- **<user>** を cephx ユーザーに置き換えます。

2. ユーザーのアクセスキーを収集します。

```
$ manila access-list <share>
```

7.7.3. CIFS 共有へのアクセスの許可

クラウドユーザーは、Active Directory サービスに存在するユーザー名を通じて CIFS 共有へのアクセスを許可します。Shared File Systems サービス (manila) は、Active Directory サーバー上に新しいユーザーを作成しません。セキュリティーサービスを通じてユーザー名のみが検証され、無効なユーザー名を含むアクセスルールは **error** ステータスになります。

クラウド管理者が **driver_handles_share_servers** パラメーター (DHSS) の値を **true** に設定した場合、クラウドユーザーはセキュリティーサービスを追加して Active Directory サービスを設定します。クラウド管理者が DHSS パラメーターの値を **false** に設定した場合、クラウド管理者は Active Directory サービスを設定し、それをストレージネットワークに関連付けます。

共有をマウントするには、ユーザーの Active Directory ユーザー名とパスワードを指定する必要があります。このパスワードは、Shared File Systems サービスを通じて取得することはできません。

手順

- ユーザーに CIFS 共有へのアクセスを許可します。

```
$ manila access-allow <share> user <user>
```

  - **<share>** は、共有名または共有 ID に置き換えます。
  - **<user>** を Active Directory ユーザーに対応するユーザー名に置き換えます。

7.7.4. ファイル共有へのアクセスの取り消し

ファイル共有の所有者は、何らかの理由でファイル共有へのアクセスを無効にすることができます。以前に共有に付与されたアクセスを取り消すには、次の手順を実行します。

手順

- ファイル共有へのアクセスを取り消します。

```
$ manila access-deny <share> <access-id>
```

  - **<share>** は、共有名または共有 ID に置き換えます。
  - **<access-id>** を共有のアクセス ID に置き換えます。
    以下に例を示します。

```
$ manila access-list share-01
+--------------+-------------+--------------+--------------+--------+
| id           | access_type | access_to    | access_level | state  | ...
```

```
+--------------+-------------+--------------+--------------+--------+ ...
| 875c6251-... | ip          | 198.51.100.160 | rw         | active | ...
+--------------+-------------+--------------+--------------+--------+

$ manila access-deny share-01 875c6251-c17e-4c45-8516-fe0928004fff

$ manila access-list share-01

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

注記

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

7.8. コンピューティングインスタンスでの共有のマウント

クライアントに共有アクセスを許可すると、クライアントは共有をマウントして使用できるようになります。クライアントへのネットワーク接続があれば、任意のクライアント種別がファイル共有にアクセスできます。

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

注記

ファイル共有をマウントする Compute インスタンスに、異なるプロトコル用のクライアントパッケージをインストールする必要があります。たとえば、CephFS-NFS を使用した Shared File Systems サービスの場合、NFS クライアントパッケージは NFS 4.1 をサポートする必要があります。

7.8.1. 共有エクスポート場所の一覧表示

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

手順

- 共有のエクスポート場所を取得します。

  ```
  $ manila share-export-location-list share-01
  ```

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

7.8.2. NFS、ネイティブ CephFS、または CIFS 共有のマウント

NFS、ネイティブ CephFS、または CIFS 共有を作成し、エンドユーザークライアントに共有アクセス権を付与すると、ネットワーク接続がある限り、ユーザーはクライアントに共有をマウントしてデータへのアクセスを有効にすることができます。

前提条件

- NFS 共有をマウントするには、**nfs-utils** パッケージがクライアントマシンにインストールされている必要があります。
- ネイティブの CephFS 共有をマウントするには、クライアントマシンに ceph **-common** パッケージをインストールする必要があります。ユーザーは、クライアントマシンで CephFS カーネルクライアントを使用して、ネイティブの CephFS 共有にアクセスします。
- CIFS 共有をマウントするには、**cifs-utils** パッケージがクライアントマシンにインストールされている必要があります。

手順

1. インスタンスにログインします。

```
$ openstack server ssh demo-instance0 --login user
```

2. NFS 共有をマウントします。サンプル構文については、次の例を参照してください。

```
$ mount -t nfs \
  -v <198.51.100.13:/volumes/_nogroup/e840b4ae-6a04-49ee-9d6e-67d4999fbc01>  \
  /mnt
```

   - **<198.51.100.13:/volumes/_nogroup/e840b4ae-6a04-49ee-9d6e-67d4999fbc01>** を共有のエクスポート場所に置き換えます。
   - の説明に従って、エクスポート先を取得します。「共有エクスポート場所の一覧表示」 .

3. ネイティブ CephFS 共有をマウントします。サンプル構文については、次の例を参照してください。

```
$  mount -t ceph \
  <192.0.2.125:6789,192.0.2.126:6789,192.0.2.127:6789:/volumes/_nogroup/4c55ad20-9c55-
4a5e-9233-8ac64566b98c> \
  -o name=<user>,secret='<AQA8+ANW/<4ZWNRAAOtWJMFPEihBA1unFImJczA==>'
```

   - **<192.0.2.125:6789,192.0.2.126:6789,192.0.2.127:6789:/volumes/_nogroup/4c55ad20-9c55-4a5e-9233-8ac64566b98c>** を共有のエクスポート場所に置き換えます。
   - の説明に従って、エクスポート先を取得します。「共有エクスポート場所の一覧表示」 .
   - **<user>** を、共有にアクセスできる cephx ユーザーに置き換えます。
   - **secret** 値を、で収集したアクセスキーに置き換えます。「ネイティブ CephFS 共有へのアクセスの許可」 .

4. CIFS 共有をマウントします。サンプル構文については、次の例を参照してください。

```
$  mount -t cifs \
  -o user=<user>,pass=<password> \
  <\\192.0.2.128/share_11265e8a_200c_4e0a_a40f_b7a1117001ed>
```

- **<user>** を、共有にアクセスできる Active Directory ユーザーに置き換えます。
- **<password>** をユーザーの Active Directory パスワードに置き換えます。
- **<\\192.0.2.128/share_11265e8a_200c_4e0a_a40f_b7a1117001ed>** を共有のエクスポート場所に置き換えます。
- の説明に従って、エクスポート先を取得します。「共有エクスポート場所の一覧表示」.

検証

- マウントコマンドが成功したことを確認します。

```
$ df -k
```

7.9. 共有の削除

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

警告

ファイル共有を削除する前にデータをバックアップします。

前提条件

- ファイル共有からスナップショットを作成した場合は、ファイル共有を削除する前にすべてのスナップショットおよびレプリカを削除している。詳細は、スナップショットの削除 を参照してください。

手順

- ファイル共有を削除します。

```
$ manila delete <share>
```

- **<share>** は、共有名または共有 ID に置き換えます。

7.10. SHARED FILE SYSTEMS サービスのリソース制限のリスト表示

クラウドユーザーは、現在のリソース制限をリスト表示できます。これは、ワークロードのプランニングや、リソース消費に基づくアクションの準備に役立ちます。

手順

手順

- プロジェクトのリソース制限および現在のリソース消費をリスト表示します。

```
$ manila absolute-limits
+-----------------------------+-------+
| Name                        | Value |
+-----------------------------+-------+
| maxTotalReplicaGigabytes    | 1000  |
| maxTotalShareGigabytes      | 1000  |
| maxTotalShareGroupSnapshots | 50    |
| maxTotalShareGroups         | 49    |
| maxTotalShareNetworks       | 10    |
| maxTotalShareReplicas       | 100   |
| maxTotalShareSnapshots      | 50    |
| maxTotalShares              | 50    |
| maxTotalSnapshotGigabytes   | 1000  |
| totalReplicaGigabytesUsed   | 22    |
| totalShareGigabytesUsed     | 25    |
| totalShareGroupSnapshotsUsed | 0     |
| totalShareGroupsUsed        | 9     |
| totalShareNetworksUsed      | 2     |
| totalShareReplicasUsed      | 9     |
| totalShareSnapshotsUsed     | 4     |
| totalSharesUsed             | 12    |
| totalSnapshotGigabytesUsed  | 4     |
+-----------------------------+-------+
```

7.11. 操作エラーのトラブルシューティング

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

7.11.1. ファイル共有作成またはファイル共有グループ作成の失敗の修正

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

手順

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

```
clouduser1@client:~$ manila type-list
+--------------------------------------+-------------+------------+------------+-------------------------------
--------+----------------------------------------------+-------------+
| ID                                   | Name        | visibility | is_default | required_extra_specs                |
optional_extra_specs                         | Description |
+--------------------------------------+-------------+------------+------------+-------------------------------
--------+----------------------------------------------+-------------+
| 1cf5d45a-61b3-44d1-8ec7-89a21f51a4d4 | dhss_false  | public     | YES        |
driver_handles_share_servers : False | create_share_from_snapshot_support : True  | None
```

```
|
|                                      |          |          |          |                                      |
mount_snapshot_support : False          |          |
|                                      |          |          |          |                                      |
revert_to_snapshot_support : False      |          |
|                                      |          |          |          |                                      | snapshot_support :
True                  |          |
| 277c1089-127f-426e-9b12-711845991ea1 | dhss_true   | public     | -         |
driver_handles_share_servers : True  | create_share_from_snapshot_support : True  | None
|
|                                      |          |          |          |                                      |
mount_snapshot_support : False          |          |
|                                      |          |          |          |                                      |
revert_to_snapshot_support : False      |          |
|                                      |          |          |          |                                      | snapshot_support :
True                  |          |
+--------------------------------------+-------------+------------+------------+------------------------------
--------+-------------------------------------------+-------------+
```

この例では、2 つの共有種別が使用可能です。

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

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

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

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

clouduser1@client:~$ manila share-network-list
+--------------------------------------+-------+
| id                                   | name  |
+--------------------------------------+-------+
| 6c7ef9ef-3591-48b6-b18a-71a03059edd5 | mynet |
+--------------------------------------+-------+
```

3. ファイル共有を作成します。

```
clouduser1@client:~$ manila create nfs 1 \
  --name software_share \
  --share-network mynet \
  --share-type dhss_true
+---------------------------------------+--------------------------------------+
| Property                              | Value                                |
+---------------------------------------+--------------------------------------+
| status                                | creating                             |
| share_type_name                       | dhss_true                            |
| description                           | None                                 |
| availability_zone                     | None                                 |
| share_network_id                      | 6c7ef9ef-3591-48b6-b18a-71a03059edd5 |
| share_server_id                       | None                                 |
| share_group_id                        | None                                 |
| host                                  |                                      |
| revert_to_snapshot_support            | False                                |
| access_rules_status                   | active                               |
| snapshot_id                           | None                                 |
| create_share_from_snapshot_support    | False                                |
| is_public                             | False                                |
| task_state                            | None                                 |
| snapshot_support                      | False                                |
| id                                    | 243f3a51-0624-4bdd-950e-7ed190b53b67 |
| size                                  | 1                                    |
| source_share_group_snapshot_member_id | None                                 |
| user_id                               | 61aef4895b0b41619e67ae83fba6defe     |
| name                                  | software_share                       |
| share_type                            | 277c1089-127f-426e-9b12-711845991ea1 |
| has_replicas                          | False                                |
| replication_type                      | None                                 |
| created_at                            | 2018-10-09T21:12:21.000000           |
| share_proto                           | NFS                                  |
| mount_snapshot_support                | False                                |
| project_id                            | cadd7139bc3148b8973df097c0911016     |
| metadata                              | {}                                   |
+---------------------------------------+--------------------------------------+
```

4. ファイル共有のステータスを表示します。

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

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

この例では、ファイル共有の作成中にエラーが発生しています。

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

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

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

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

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

```
| id            | 7d411c3c-46d9-433f-9e21-c04ca30b209c
|
| resource_type | SHARE                                                                         |
| action_id     | 001                                                                           |
+---------------+-----------------------------------------------------------------------------
-----------+
```

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

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

8. 利用可能な他の共有種別でファイル共有を作成します。

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

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

この例では、試みた 2 番目のファイル共有作成は失敗します。

9. ユーザーサポートメッセージを表示します。

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

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

```
+--------------------------------------+---------------+--------------------------------------+-----------+---
-----------------------------------------------------------------------------------------------------------+--------
---+----------------------------+
```

サービスは、使用した共有種別のファイル共有ネットワークに対応していません。

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

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

11. ファイル共有が正常に作成されたことを確認します。

```
clouduser1@client:~$ manila list
+--------------------------------------+----------------+------+-------------+-----------+-----------+-------
----------+------+-------------------+
| ID                                   | Name           | Size | Share Proto | Status    | Is Public | Share Type
Name | Host | Availability Zone |
+--------------------------------------+----------------+------+-------------+-----------+-----------+-------
----------+------+-------------------+
| 4d3d7fcf-5fb7-4209-90eb-9e064659f46d | software_share | 1    | NFS         | available |
False     | dhss_false      |      | 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 |
+----+---------------+-------------+-----------+--------------+-----------+------------+
+----+---------------+-------------+-----------+--------------+-----------+------------+
```

7.11.2. ファイル共有のマウント失敗のデバッグ

共有をマウントするときに問題が発生した場合は、これらの検証手順を使用して根本原因を特定してください。

手順

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

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

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

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

```
$ ping -c 1 198.51.100.13
PING 198.51.100.13 (198.51.100.13) 56(84) bytes of data.
64 bytes from 198.51.100.13: icmp_seq=1 ttl=64 time=0.048 ms--- 198.51.100.13 ping
statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 7.851/7.851/7.851/0.000 ms
If using the NFS protocol, you may verify that the NFS server is ready to respond to NFS rpcs
on the proper port:
$ rpcinfo -T tcp -a 198.51.100.13.8.1 100003 4
program 100003 version 4 ready and waiting
```

注記

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

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