Red Hat OpenStack Platform 16.0

Block Storage バックアップガイド

OpenStack での Block Storage のバックアップサービスの理解、使用、管理

Last Updated: 2022-09-13

Red Hat OpenStack Platform 16.0 Block Storage バックアップガイド

OpenStack での Block Storage のバックアップサービスの理解、使用、管理

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

法律上の通知

概要

本書では、OpenStack Block Storage のバックアップサービスをデプロイする方法を説明します。Red Hat OpenStack Platform director は、Red Hat Ceph ストレージ、NFS、およびオブジェクトストレージ (swift) をバックエンドとして設定できます。Google Cloud Storage をバックアップバックエンドとして設定することもできます。

目次

前書き

Red Hat OpenStack Platform は、Red Hat Enterprise Linux 上にプライベートまたはパブリックの Infrastructure-as-a-Service (IaaS) クラウドを構築するための基盤を提供します。RHOSP は、クラウド対応のワークロード開発向けのスケーラビリティーおよび耐障害性に優れたプラットフォームです。

OpenStack Dashboard またはコマンドラインクライアントメソッドのいずれかを使用して、バックアップサービスのほとんどの機能を管理できますが、一部の高度な手順を実行するにはコマンドラインを使用する必要があります。

注記

Red Hat OpenStack Platform の全ドキュメントスイートは Red Hat OpenStack Platform の製品ドキュメント で参照してください。

第1章 概要

Block Storage Service (cinder) には、cinder ボリュームのバックアップをさまざまなストレージバックエンドに提供するために使用できる、水平方向にスケーラブルなバックアップサービスが同梱されています。Block Storage バックアップサービスを使用して、完全バックアップまたは増分バックアップを作成および復元できます。このサービスはボリュームアレイに依存しません。

Red Hat OpenStack Platform (RHOSP) ディレクターは、オーバークラウドと呼ばれる完全な RHOSP 環境をインストールして管理するツールセットです。director の詳細は、『Director のインストールと使用 ガイド』を参照してください。オーバークラウドには、Block Storage など、エンドユーザーにサービスを提供するコンポーネントが含まれています。Block Storage バックアップサービスは、Controller ノードにデプロイするオプションのサービスです。

1.1. バックアップおよびスナップショット

ボリュームバックアップは、ボリュームのコンテンツの永続コピーです。ボリュームのバックアップは通常、オブジェクトストアとして作成され、デフォルトでは OpenStack Object Storage サービス (swift) で管理されます。Red Hat Ceph および NFS をバックアップの代替バックエンドとして使用できます。

ボリュームバックアップを作成すると、バックアップメタデータはすべて Block Storage サービスデータベースに保存されます。cinder-backup サービスは、バックアップからボリュームを復元する際にこのメタデータを使用します。つまり、致命的なデータベース損失からの復旧を実行する場合、Block Storage サービスデータベースの元のボリュームバックアップメタデータがそのままの状態であれば、バックアップからボリュームを復旧する前に、Block Storage サービスデータベースを復旧する必要があります。致命的なデータベースの損失に備えて、ボリュームバックアップのサブセットのみを設定する場合は、バックアップメタデータもエクスポートできます。その後、REST API または cinder クライアントを使用して、Block Storage データベースにメタデータを再インポートし、通常どおりボリュームバックアップを復元できます。

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。バックアップではボリュームに含まれるデータが保持され、スナップショットでは、指定した時点でボリュームの状態が保持されます。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通常、クローン作成時の待ち時間を最小限にとどめるため、ボリュームバックエンドにコロケーションを設定します。一方、バックアップリポジトリは、通常、別のノードまたは別の物理ストレージにあり、バックエンドとは別の場所にあります。これにより、ボリュームバックエンドに発生する可能性がある損傷からバックアップリポジトリーを保護します。

ボリュームのスナップショットについての詳しい情報は、『**ストレージガイド**』の「ボリュームスナップショットの作成、使用、または削除」を参照してください。

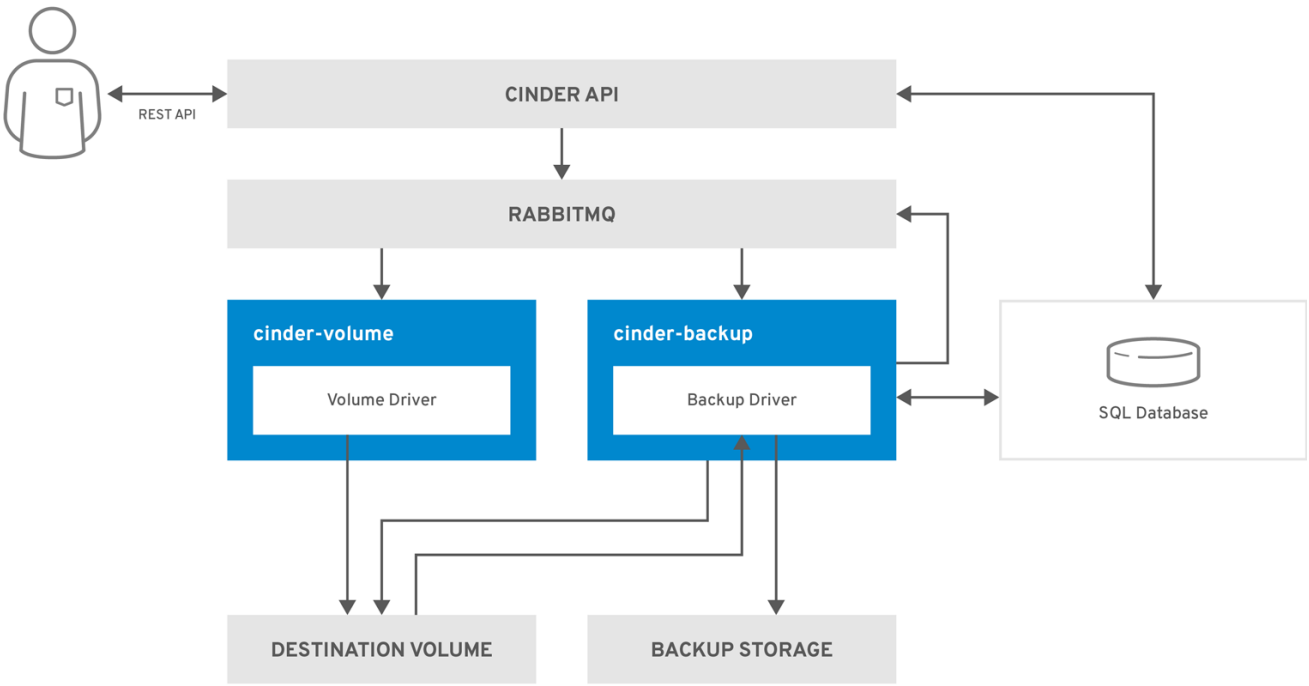
1.2. バックアップおよび復元の仕組み

以下のサブセクションでは、バックアップと復元のワークフローについて説明します。

1.2.1. ボリュームバックアップのワークフロー

Block Storage バックアップサービスがバックアップを実行すると、cinder API から、ターゲットのボリュームのバックアップを作成する要求を受け取ります。バックアップサービスは要求を完了し、コンテンツをバックエンドに保存します。

以下の図は、要求が Block Storage (cinder) サービスと相互作用してバックアップを実行する方法を示しています。

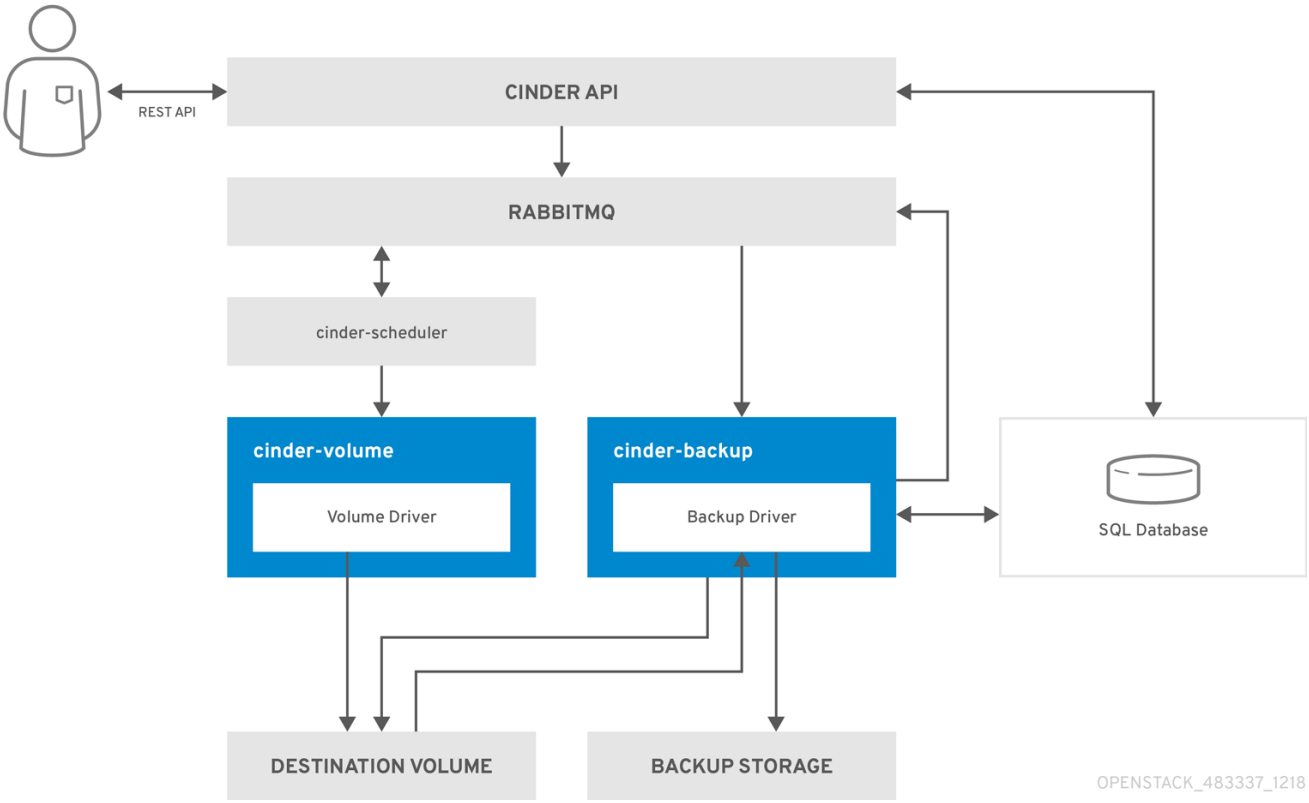


1. クライアントは、cinder API を起動して、Block Storage ボリュームのバックアップを作成するように要求します。
2. cinder API サービスは、HAProxy から要求を受信し、要求、ユーザー認証情報、およびその他の情報を検証します。
3. SQL データベースにバックアップレコードを作成します。
4. AMQP を介して、**cinder-backup** への非同期 RPC 呼び出しを行い、ボリュームのバックアップを作成します。
5. API 呼び出し元に、現在のバックアップレコード (ID) を返します。
6. RPC 作成メッセージは、バックアップサービスのいずれかに届きます。
7. **cinder-backup** は、**get_backup_デバイス** への同期 RPC 呼び出しを実行します。
8. **cinder-volume** は、正しいデバイスが呼び出し元に返されるようにします。通常は同じボリュームですが、ボリュームが使用中の場合は、設定によっては一時クローンボリュームまたは一時スナップショットが返されます。
9. **cinder-backup** は、別の同期 RPC を発行して、ソースデバイスを公開するように **cinder-volume** します。
10. **cinder-volume** サービスは、ソースデバイス (ボリュームまたはスナップショット) をエクスポートしてマッピングし、適切な接続情報を返します。
11. **cinder-backup** は、接続情報を使用してソースボリュームを割り当てます。
12. **cinder-backup** は、デバイスが接続されている状態でバックアップドライバーを呼び出し、バックアップ先へのデータ転送を開始します。
13. ボリュームがバックアップホストから切り離されている。

14. **cinder-backup** は、同期 RPC を **cinder-volume** に発行して、ソースデバイスの接続を解除します。
15. **cinder-volume** サービスは、デバイスのマッピングを解除し、エクスポートを削除します。
16. 一時ボリュームまたは一時スナップショットが作成された場合、**cinder-backup** は **cinder-volume** を呼び出してそのボリュームを削除します。
17. **cinder-volume** により、一時ボリュームが削除されます。
18. バックアップが完了すると、データベースのバックアップレコードが更新されます。

1.2.2. ボリュームの復元のワークフロー

以下の図は、ユーザーが Block Storage Service (cinder) バックアップの復元を要求すると発生する手順を示しています。



1. クライアントは、CinderREST API を呼び出して、Block Storage バックアップを復元する要求を発行します。
2. cinder API は、HAProxy から要求を受信し、要求、ユーザー認証情報、およびその他の情報を検証します。
3. 要求に宛先として既存のボリュームが含まれていない場合、API は非同期 RPC 呼び出しを行って新しいボリュームを作成し、ボリュームのステータスをポーリングして利用可能になります。
4. **cinder-scheduler**がボリュームサービスを選択し、RPC 呼び出しを実行してボリュームを作成します。
5. 選択した**cinder-ボリューム**により、音量が作成されます。

6. **cinder-api** がボリュームが利用可能であることを検出すると、バックアップレコードがデータベースに作成されます。
7. AMQP 経由でバックアップサービスへの非同期 RPC 呼び出しを行い、バックアップを復元します。
8. 現在のボリューム ID、バックアップ ID、およびボリューム名を API 呼び出し元に返します。
9. RPC 作成メッセージは、バックアップサービスのいずれかに届きます。
10. **cinder-backup** は、**cinder-volume** への同期 RPC 呼び出しを実行して、宛先ボリュームを公開します。
11. **cinder-volume** は、適切な接続情報を返す宛先ボリュームをエクスポートしてマッピングします。
12. **cinder-backup** は、接続情報を使用してソースボリュームを接続します。
13. **cinder-backup** サービスは、デバイスが接続されている状態でドライバーを呼び出し、ボリュームデスティネーションへのデータの復元を開始します。
14. ボリュームがバックアップホストから切り離されている。
15. **cinder-backup** は、同期 RPC を **cinder-volume** に発行して、ソースデバイスの接続を解除します。
16. **cinder-volume** サービスは、デバイスのマッピングを解除し、エクスポートを削除します。
17. バックアップが完了すると、データベースのバックアップレコードが更新されます。

1.3. クラウドストレージとローカルストレージ

Google Cloud Storage ドライバーは、Block Storage バックアップサービスで対応している唯一のクラウドドライバーです。デフォルトでは、Google Cloud Storage ドライバーは、このタイプのバックアップに、最も低コストのストレージソリューションである Nearline を使用します。

パフォーマンスを最適化するようにバックアップサービスを設定します。たとえば、ヨーロッパからバックアップを作成する場合は、バックアップリージョンをヨーロッパに変更します。バックアップリージョンをデフォルトの US から変更しないと、2 つのリージョン間の地理的な距離が原因で、パフォーマンスが低下する場合があります。

注記

Google Cloud Storage には、付録A *Google Cloud Storage の設定* セクションで説明されている特殊な設定が必要です。

次の表は、状況に基づくクラウドストレージとローカルストレージの利点と制限を一覧表示します。

状況	クラウドストレージ	ローカルストレージ

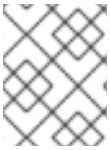
状況	クラウドストレージ	ローカルストレージ
オフサイトバックアップ	クラウドストレージは別の会社のデータセンターにあるため、自動的にオフサイトになります。多くの場所からデータにアクセスできます。リモートコピーは、障害復旧に使用できます。	追加の計画と費用が必要。
ハードウェア制御	別のサービスの利用可能性と専門知識に依存します。	ストレージハードウェアを完全に制御できる。管理および専門知識が必要。
コストに関する考慮事項	ベンダーから使用するサービスに応じて異なる価格ポリシーまたは層。	必要に応じてハードウェアを追加することは、既知のコストとなります。
ネットワーク速度とデータアクセス	全体的なデータアクセスは低速で、インターネットアクセスが必要です。速度とレイテンシーは複数の要因により異なります。	データへのアクセスは速く、即座に行われます。インターネットアクセスは必要ありません。

第2章 BLOCK STORAGE バックアップサービスのデプロイメント

Block Storage バックアップサービスはオプションです。デフォルトではインストールされないため、オーバークラウドデプロイメントに追加する必要があります。

前提条件

- 既存の Red Hat OpenStack Platform (RHOSP) インストール。
- 互換性のあるバックアップドライバーを備えた利用可能なストレージソース - オブジェクトストレージ (swift。デフォルト)、Ceph、NFS、または Google Cloud ストレージ。



注記

Google Cloud Storage には、追加の設定が必要です。詳細は「付録A *Google Cloud Storage の設定*」を参照してください。

2.1. バックアップサービスのバックエンドストレージオプションの設定

バックアップサービスを有効にするには、以下の手順を実行します。

手順

1. **/usr/share/openstack-tripleo-heat-templates/environments/** ディレクトリーにある **cinder-backup.yaml** ファイルのコピーを作成し、別のカスタムテンプレートと同じ場所に保存します。

```
cp /usr/share/openstack-tripleo-heat-templates/environments/cinder-backup.yaml
/home/stack/templates/cinder-backup-settings.yaml
```

2. **cinder-backup.yaml** ファイルのコピーには、Pacemaker で Block Storage バックアップサービスの OpenStack Object Storage (swift) バックエンドを設定するデフォルト設定が含まれています。バックアップに使用しているバックエンドである場合は、このファイルを変更する必要はありません。別のバックエンドを使用している場合は、バックアップバックエンドに応じて **parameter_defaults** を設定します。
   - Red Hat Ceph ストレージを使用している場合は、以下の方法で**parameter_defaults** を設定します。
     - **CinderBackupBackend**: (必須) **セフ**
     - **CinderBackupRbdPoolName**: (必要に応じて) カスタムの RBD プール名を設定します。デフォルト: **backups**
   - NFS を使用している場合は、以下の方法で **parameter_defaults** を設定します。
     - **CinderBackupBackend**: (必須) **nfs**
     - **CinderBackupNfsShare** - (必須) マウントする NFS 共有に設定します。デフォルト値は空です。
     - **CinderBackupNfsOptions** - (必要に応じて) 必要なマウントオプションに設定します。
3. 変更をファイルに保存します。

4. バックアップサービスを有効にし、この設定を適用するには、別の環境ファイルを使用してバックアップ設定環境ファイルをスタックに追加し、オーバークラウドをデプロイします。

```
(undercloud) [stack@undercloud ~]$ openstack overcloud deploy --templates \
  -e [your environment files]
  -e /home/stack/templates/cinder-backup-settings.yaml
```

詳細と追加の設定オプションは、付録A *Google Cloud Storage の設定* を参照してください。

2.2. GOOGLE CLOUD 設定でのオーバークラウドのデプロイメント

`**/home/stack/templates/** で環境ファイルを作成したら、オーバークラウドをデプロイしてから、cinder-backup サービスを再起動します。

手順

1. **stack** ユーザとしてログインします。
2. 設定をデプロイします。

```
$ openstack overcloud deploy --templates \
-e /home/stack/templates/cinder-backup-settings.yaml
```

重要

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

3. デプロイメントが終了したら、**cinder-backup** サービスを再起動します。

詳しい情報は、『 オーバークラウドの **高度なカスタマイズ**』 の「オーバークラウドデプロイメントへの 環境ファイル の 追加 」を参照してください。

第3章 BLOCK STORAGE バックアップサービスの使用

Block Storage バックアップサービスを使用して、完全バックアップまたは増分バックアップを実行し、バックアップをボリュームに復元できます。

3.1. 完全バックアップ

3.1.1. フルボリュームバックアップの作成

ボリュームのバックアップを作成するには、**bakcup-create** コマンドを使用します。デフォルトでは、このコマンドはボリュームの完全バックアップを作成します。ボリュームに既存のバックアップがある場合は、代わりに増分バックアップを作成できます。詳細は、「「増分バックアップの実行」」を参照してください。

注記

Red Hat OpenStack Platform バージョン 16 よりも前のバージョンでは、**cinder backup-create** コマンドは、Ceph Storage バックエンドへの最初の Ceph ボリュームのバックアップ後に増分バックアップを作成していました。RHOSP バージョン 16 以降では、**--incremental** を使用して増分ボリュームバックアップを作成する必要があります。**cinder backup-create** コマンドで **--incremental** オプションを使用しない場合には、デフォルト設定で完全バックアップが作成されます。詳細は、「「増分バックアップの実行」」を参照してください。

アクセス可能なボリュームのバックアップを作成することができます。つまり、管理者特権を持つユーザーは、所有者に関係なく、どのボリュームでもバックアップを作成できます。詳細は「「管理者としてのボリュームバックアップの作成」」を参照してください。

手順

1. バックアップを作成するボリュームの ID または表示名を表示します。

```
# cinder list
```

2. ボリュームをバックアップします。

```
# cinder backup-create _VOLUME_
```

VOLUME の箇所は、バックアップするボリュームの **ID** または **Display Name** に置き換えます。以下に例を示します。

```
+-----------+--------------------------------------+
|  Property |                Value                 |
+-----------+--------------------------------------+
|     id    | e9d15fc7-eeae-4ca4-aa72-d52536dc551d |
|    name   |                None                  |
| volume_id | 5f75430a-abff-4cc7-b74e-f808234fa6c5 |
+-----------+--------------------------------------+
```

作成されるバックアップの**volume_id**は、ソースボリュームの ID と同じです。

3. ボリュームのバックアップ作成が完了したことを確認します。

```
# cinder backup-list
```

4. ボリュームバックアップの作成は、バックアップエントリーの**Status**が利用可能になると完了します。

3.1.2. 管理者としてのボリュームバックアップの作成

管理者特権を持つユーザーは、Red Hat OpenStack Platform が管理するボリュームのバックアップを作成できます。管理者ユーザーが所有しているボリュームのバックアップを作成する場合、バックアップはデフォルトではボリューム所有者に表示されません。

手順

- 管理ユーザーは、以下のコマンドを使用してボリュームをバックアップし、特定のプロジェクトでバックアップできます。

```
# cinder --os-auth-url <KEYSTONEURL> --os-tenant-name <PROJECTNAME> --os-username <USERNAME> --os-password <PASSWD> backup-create <VOLUME>
```

使用環境の要件に従って、次の変数を置き換えます。

- <PROJECTNAME> は、バックアップを利用可能にするプロジェクト（テナント）の名前です。
- <USERNAME> および <PASSWD> は、<PROJECTNAME> 内のユーザーのユーザー名とパスワードの認証情報です。
- <VOLUME> は、バックアップを作成するボリュームの名前または ID です。
- <KEYSTONEURL> は Identity サービスの URL エンドポイントです。通常、**http://<IP>:5000/v2** になります。< **IP** > は Identity サービスホストの IP アドレスに置き換えます。この操作を実行する際、作成されるバックアップのサイズは、プロジェクト管理者のクォータではなく、<PROJECTNAME> のクォータに対してカウントされます。

3.1.3. ボリュームバックアップのメタデータのエクスポート

ボリュームバックアップのメタデータをエクスポートおよび保存して、Block Storage データベースで致命的な損傷が発生した場合でもボリュームバックアップを復元できるようにできます。

手順

- 以下のコマンドを実行します。

```
# cinder backup-export _BACKUPID_
```

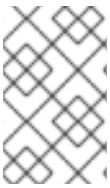
<BACKUPID> を、ボリュームバックアップの ID または名前に置き換えます。

```
+----------------+-----------------------------------------+
|    Property    |                Value                    |
+----------------+-----------------------------------------+
| backup_service |     cinder.backup.drivers.swift         |
|   backup_url   | eyJzdGF0dXMiOiAiYXZhaWxhYmxlIiwgIm9iam...|
|                | ...4NS02ZmY4MzBhZWYwNWUiLCAic2l6ZSI6IDF9 |
+----------------+-----------------------------------------+
```

ボリュームのバックアップメタデータは、**backup_service** および **backup_url** の値で構成されます。

3.1.4. 使用中のボリュームのバックアップ

ブロックストレージバックエンドスナップショットに対応している場合は、**--force** で使用中のボリュームのバックアップを作成できます。



注記

--force を使用するには、ストレージバックエンドスナップショットのブロックに対応している必要があります。使用しているバックエンドのドキュメントを確認することで、スナップショットのサポートを確認できます。

--force を使用することで、ドライブを静止していないことを確認し、バックアップを実行します。この方法を使用すると、クラッシュの一貫性はありますが、アプリケーションの一貫性はありません。バックアップが作成されます。つまり、バックアップの実行時に実行していたアプリケーションを認識していないことを意味します。ただし、データはそのままです。

手順

- 使用中のボリュームのバックアップを作成するには、次のコマンドを実行します。

```
# cinder backup-create _VOLUME_ --incremental --force
```

3.1.5. スナップショットのバックアップ

スナップショットに関連付けられているボリューム ID を使用して、スナップショットから完全バックアップを作成できます。

手順

1. **cinder snapshot list** を使用して、バックアップを作成するスナップショットのスナップショット ID を特定します。

```
# cinder snapshot-list --volume-id _VOLUME_ID_
```

2. スナップショットに名前が付けられている場合は、以下の例を使用して**ID**の場所を特定できます。

```
# cinder snapshot-show _SNAPSHOT_NAME_
```

3. スナップショットのバックアップを作成します。

```
# cinder backup-create _VOLUME_ --snapshot-id=_SNAPSHOT_ID_
```

3.2. 増分バックアップ

Block Storage のバックアップサービスを使用すると、増分バックアップを実行できます。

3.2.1. パフォーマンスに関する考慮事項

増分やデータ圧縮などのバックアップ機能の一部は、パフォーマンスに影響を与える可能性があります。増分バックアップは、ボリューム内のデータをすべて読み込み、完全バックアップと各増分バックアップの両方でサムチェックを行う必要があるため、パフォーマンスに影響を及ぼします。

Ceph 以外のバックエンドではデータ圧縮を使用できます。データ圧縮を有効にするには、追加の CPU 電力が必要になりますが、使用するネットワーク帯域幅とストレージ領域は全体で少なくなります。

マルチパスの設定はパフォーマンスにも影響します。マルチパスを有効にせずに複数のボリュームを接続すると、接続できなくなるか、完全なネットワーク機能が利用できなくなる可能性があります。これはパフォーマンスに影響を及ぼします。

詳細設定オプションを使用して、圧縮の有効化または無効化、プロセス数の定義、およびその他の CPU リソースの追加を行うことができます。詳細は「付録B *高度な Block Storage のバックアップ設定オプション*」を参照してください。

3.2.1.1. スナップショットからのバックアップの影響

バックエンドの中には、スナップショットからのバックアップの作成に対応するものもあります。この機能に対応するドライバーでは、スナップショットを直接割り当てることができます。これにより、スナップショットをボリュームにクローンするのと比較して速くなり、ボリュームに割り当てることができるようになります。通常、この機能はスナップショットからボリュームを作成する手順が追加されたため、パフォーマンスに影響を与える可能性があります。

3.2.2. 増分バックアップの実行

デフォルトでは、**cinder backup-create** はボリュームのフルバックアップを作成します。ただし、ボリュームに既存のバックアップがある場合は、増分バックアップを作成できます。

増分バックアップは、NFS、オブジェクトストレージ (swift)、および Red Hat Ceph ストレージバックアップリポジトリーで完全にサポートされています。

増分バックアップは、最後の完全バックアップまたは増分バックアップ以降のボリュームへの変更をキャプチャーします。ボリュームのサイズは時間とともに増加するため、ボリュームの多数の定期的なフルバックアップの実行はリソースの集中的なものになる可能性があります。増分バックアップを使用すると、ボリュームへの定期的な変更をキャプチャーして、リソースの使用を最小限にとどめることができます。

手順

- 増分ボリュームバックアップを作成するには、次のコマンドで **--incremental** を使用します。

```
# cinder backup-create _VOLUME_ --incremental
```

VOLUME の箇所は、バックアップするボリュームの **ID** または **Display Name** に置き換えます。

注記

増分バックアップがすでに作成されている場合には、完全バックアップを削除することはできません。フルバックアップに複数の増分バックアップがある場合は、最新の増分バックアップのみを削除できます。

3.3. バックアップのキャンセル

バックアップをキャンセルするには、管理者がバックアップで強制削除を要求する必要があります。

重要

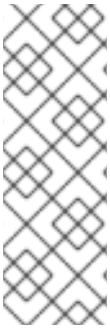
Ceph または RBD のバックエンドを使用する場合は、この操作に対応していません。

手順

- 以下のコマンドを実行します。

```
# openstack volume backup delete --force <backup>
```

キャンセルを完了し、バックアップ一覧にバックアップが表示されなくなった後、バックアップを正常にキャンセルするのに若干の遅延が生じることがあります。バックアップが正常にキャンセルされたことを確認するため、ソースリソースのバックアップステータスが停止します。



注記

Red Hat OpenStack バージョン 12 以前では、スナップショットのバックアップを作成している場合でも、**backing-up**の状態はボリュームに保存されていました。したがって、スナップショットのバックアップを作成する際に、スナップショットに対して削除操作を行ってからキャンセルした場合に、スナップショットが依然としてマッピングされているとエラーが発生することがありました。Red Hat OpenStack Platform バージョン 13 以降では、対応しているバックアップドライバーのいずれかで、継続中の復元操作をキャンセルできます。

3.4. プロジェクトバックアップクォータの表示および変更

通常、ダッシュボードを使用してプロジェクトストレージクォータを変更できます。たとえば、プロジェクトに割り当てることのできるボリューム、ボリュームのストレージ、スナップショット、その他の操作制限などです。ただし、ダッシュボードでバックアップクォータを変更する機能は利用できません。

バックアップのクォータを変更するには、コマンドラインインターフェースを使用する必要があります。

手順

1. 特定のプロジェクト（テナント）のストレージクォータを表示するには、以下のコマンドを入力します。

```
# cinder quota-show <PROJECT_ID>
```

2. 特定のプロジェクトで作成可能なバックアップの最大数 & **lt;MAXNUM** > を更新します。

```
# cinder quota-update --backups <MAXNUM> <PROJECT_ID>
```

3. 特定のプロジェクト内で、すべてのバックアップの最大サイズ(**<MAXGB** >)を更新します。

```
# cinder quota-update --backup-gigabytes MAXGB <PROJECT_ID>
```

4. 特定のプロジェクトのストレージクォータの使用状況を表示します。

```
# cinder quota-usage <PROJECT_ID>
```

3.5. バックアップからの復元

3.5.1. バックアップからのボリュームの復元

バックアップから新しいボリュームを作成するには、以下の手順を完了します。

手順

1. 使用するボリュームバックアップの ID を検索します。

```
# cinder backup-list
```

   ボリューム ID が、復元するボリュームの ID と一致していることを確認します。

2. ボリュームのバックアップを復元します。

```
# cinder backup-restore _BACKUP_ID_
```

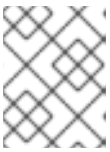
   BACKUP_ID を、使用するボリュームバックアップの ID に置き換えます。

3. バックアップが必要なくなった場合には、削除します。

```
# cinder backup-delete _BACKUP_ID_
```

4. バックアップボリュームを特定タイプのボリュームに復元する必要がある場合は、**--volume** を使用して、バックアップを特定ボリュームに復元します。

```
# cinder backup-restore _BACKUP_ID --volume VOLUME_ID_
```



注記

暗号化したバックアップからボリュームを復元する場合は、復元先ボリュームの種類も暗号化する必要があります。

3.5.2. Block Storage データベースの損失後のボリュームの復元

Block Storage データベースの損失が発生した場合、ボリュームバックアップサービスで必要なメタデータがデータベースに含まれているため、ボリュームバックアップを復元できません。ただし、ボリュームバックアップの作成後に、メタデータ (**backup_service** 値および **backup_url** 値で構成) をエクスポートおよび保存して、データベースが失われた場合にボリュームバックアップを復元できるようにします。詳細は「「フルボリュームバックアップの作成」」を参照してください。

このメタデータをエクスポートして保存した場合は、新しい Block Storage データベースにインポートして、ボリュームのバックアップを復元できます。

注記

増分バックアップの場合は、増分バックアップのいずれかを復元する前に、エクスポートしたデータをすべてインポートする必要があります。

手順

1. 管理者特権を持つユーザーとして、次のコマンドを実行します。

```
# cinder backup-import _backup_service_ _backup_url_
```

backup_service および **backup_url** を、エクスポートしたメタデータに置き換えます。たとえば、「フルボリュームバックアップの作成」 からエクスポートしたメタデータを使用する場合は、次のコマンドを実行します。

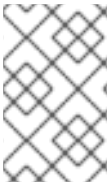
```
# cinder backup-import cinder.backup.drivers.swift eyJzdGF0dXMi...c2l6ZSI6IDF9
+----------+--------------------------------------+
| Property |                Value                 |
+----------+--------------------------------------+
|    id    | 77951e2f-4aff-4365-8c64-f833802eaa43 |
|   name   |                None                  |
+----------+--------------------------------------+
```

2. メタデータをブロックストレージサービスデータベースにインポートしたら、通常どおりボリュームを復元できます。「バックアップからのボリュームの復元」 を参照してください。

3.5.3. バックアップの復元のキャンセル

バックアップの復元操作をキャンセルするには、バックアップのステータスを **restoring** 以外に変更します。**error** ステートを使用すると、復元の成功の有無に関する混同を最小限にとどめることができます。または、値を **available** に変更できます。

```
$ openstack volume backup set --state error BACKUP_ID
```



注記

バックアップのキャンセルは非同期のアクションです。バックアップドライバーは、バックアップをキャンセルする前にステータスの変更を検出する必要があるためです。宛先ボリュームで状況が **available** に変更すると、キャンセルが完了します。

注記

この機能は、現在、RBD バックアップでは使用できません。

警告

復元操作を開始した後に取り消すと、宛先ボリュームが実際に復元されたデータ量 (存在する場合) を把握できなくなるため、宛先ボリュームは役に立ちません。

3.6. トラブルシューティング

バックアップサービスで発生する問題の多くは、以下の 2 つの一般的なシナリオで説明します。

- **cinder-backup** サービスが起動すると、設定したバックエンドに接続し、これをバックアップのターゲットとして使用します。この接続の問題により、サービスが失敗する場合があります。

- バックアップが要求されると、バックアップサービスはボリュームサービスに接続し、要求されたボリュームを割り当てます。この接続の問題は、バックアップ時にのみ明らかになります。

いずれの場合も、ログにはエラーを説明するメッセージが含まれます。

ログファイルおよびサービスに関する情報は、『**Logging, Monitoring and Troubleshooting Guide**』の「Log Files for OpenStack Services」を参照してください。

ログの場所に関する一般的な情報およびトラブルシューティングの提案についての詳細は、『**Logging, Monitoring and Troubleshooting Guide**』の「Block Storage(cinder)ログファイル」を参照してください。

3.6.1. サービスの確認

多くの問題を診断するには、サービスが利用可能であることを確認し、ログファイルでエラーメッセージを確認します。キーサービスおよびその相互作用の詳細は、「バックアップおよび復元の仕組み」を参照してください。

サービスの状態を確認したら、**cinder-backup.log** を確認します。Block Storage Backup サービスログは、**/var/log/containers/cinder]/cinder-backup.log** にあります。

手順

1. ボリュームで **cinder show** コマンドを実行して、ホストに保存されているかどうかを確認します。

```
# cinder show
```

2. **cinder service-list** を実行して、実行中のサービスを表示します。

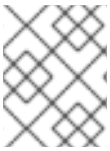
```
# cinder service-list
+------------------+--------------------+------+---------+-------+----------------------------+-----------------
---+
| Binary           | Host               | Zone | Status  | State | Updated_at                 | Disabled
Reason |
+------------------+--------------------+------+---------+-------+----------------------------+-----------------
---+
| cinder-backup    | hostgroup          | nova | enabled | up    | 2017-05-15T02:42:25.000000 | -
|
| cinder-scheduler | hostgroup          | nova | enabled | up    | 2017-05-15T02:42:25.000000 | -
|
| cinder-volume    | hostgroup@sas-pool | nova | enabled | down  | 2017-05-
14T03:04:01.000000 | -              |
| cinder-volume    | hostgroup@ssd-pool | nova | enabled | down  | 2017-05-
14T03:04:01.000000 | -              |
+------------------+--------------------+------+---------+-------+----------------------------+-----------------
---+
```

3. 期待されるサービスが利用可能であることを確認する。

3.6.2. トラブルシューティングのヒント

バックアップは非同期です。ブロックストレージバックアップサービスは、API リクエストを受信すると、不正なボリュームリファレンス (**missing**) や、インスタンスに**in-use**またはアタッチされているボ

リュームの確認など、少数の静的チェックを実行します。**in-use** の場合は、**--force** を使用する必要があります。



注記

--force を使用すると、I/O が静止せず、ボリュームイメージが破損する可能性があります。

API が要求を受け入れると、バックアップがバックグラウンドで発生します。通常、バックアップに失敗したり、障害が近づいていても、CLI はすぐに戻ります。cinder バックアップ API を使用して、バックアップのステータスをクエリーできます。エラーが発生した場合は、ログを確認して原因を特定します。

3.6.3. Pacemaker

デフォルトでは、Pacemaker は Block Storage バックアップサービスをデプロイします。Pacemaker は、定義された OpenStack クラスターリソースセットが実行中で、利用可能になるように、仮想 IP アドレス、コンテナー、サービス、およびその他の機能をクラスターのリソースとして設定します。クラスター内のサービスノードまたは全ノードが停止した場合には、Pacemaker はリソースの再起動、クラスターからのノードの削除、ノードの再起動を実行することができます。大半のサービスへの要求は HAProxy を経由します。

トラブルシューティングに Pacemaker を使用する方法は、『**High Availability Deployment and Usage**』の「Managing high availability services with Pacemaker」を参照してください。

付録A GOOGLE CLOUD STORAGE の設定

Google Cloud Storage をバックアップバックエンドとして使用するように Block Storage サービス (cinder) を設定するには、以下の手順を完了します。

1. Google アカウントのサービスアカウント認証情報を作成してダウンロードします。
   - 「GCS 認証情報ファイルの作成」
   - 「**cinder-backup-gcs.yaml**の作成」
2. 必要な Block Storage 設定をマッピングする環境ファイルを作成します。
   - 「Google Cloud 設定で環境ファイルの作成」
3. 作成した環境ファイルで、オーバークラウドを再デプロイします。
   - 「オーバークラウドのデプロイ」

前提条件

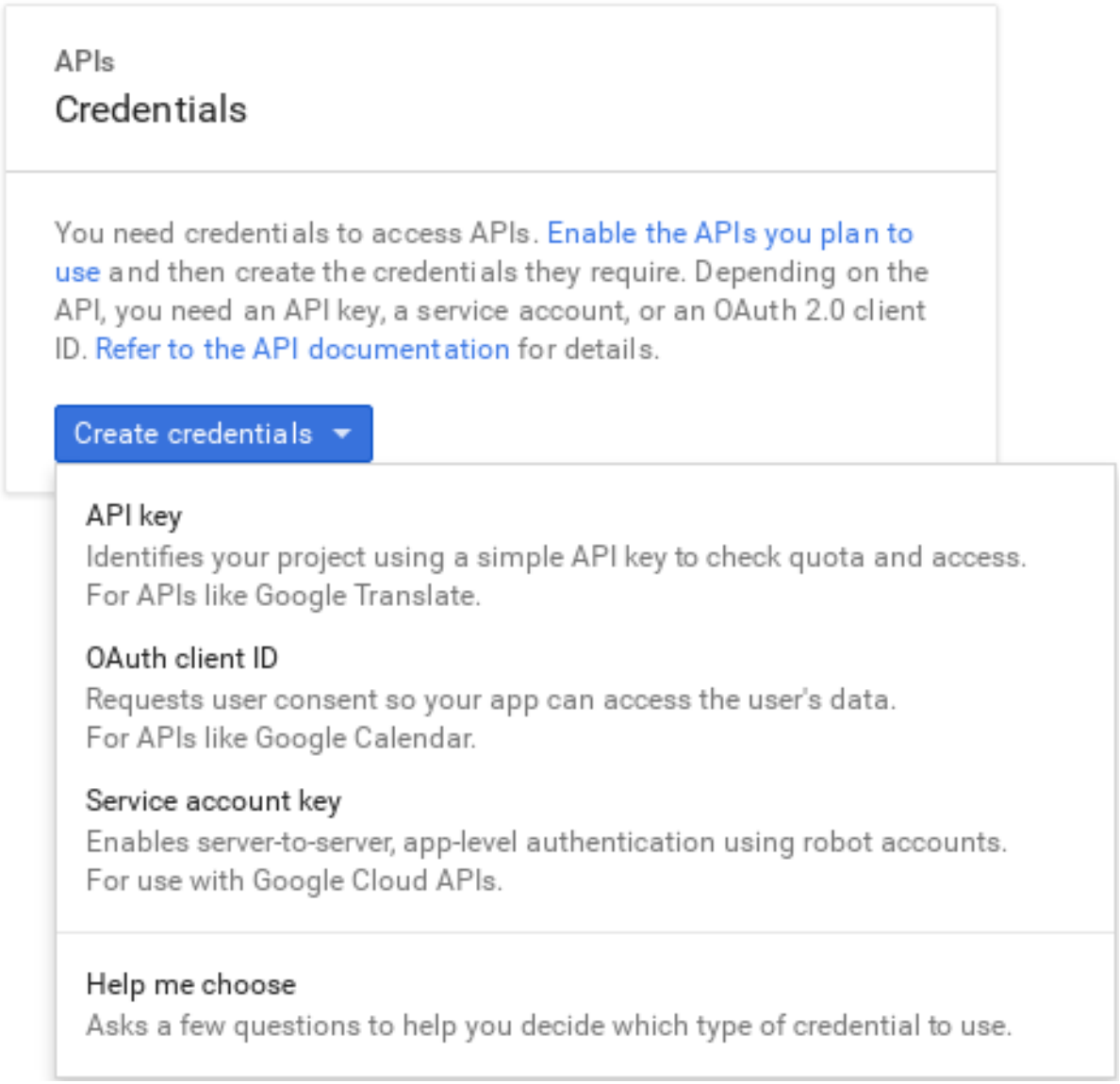
- 昇格した特権を持つアカウントのユーザー名およびパスワードを所有している。作成した **stack** ユーザーアカウントを使用して、オーバークラウドをデプロイできます。詳しくは、『**director のインストールと使用方法**』を参照してください。
- Google Cloud Platform にアクセスできる Google アカウントがある。Block Storage サービスはこのアカウントを使用してアクセスし、Google Cloud を使用してバックアップを保存します。

A.1. GCS 認証情報ファイルの作成

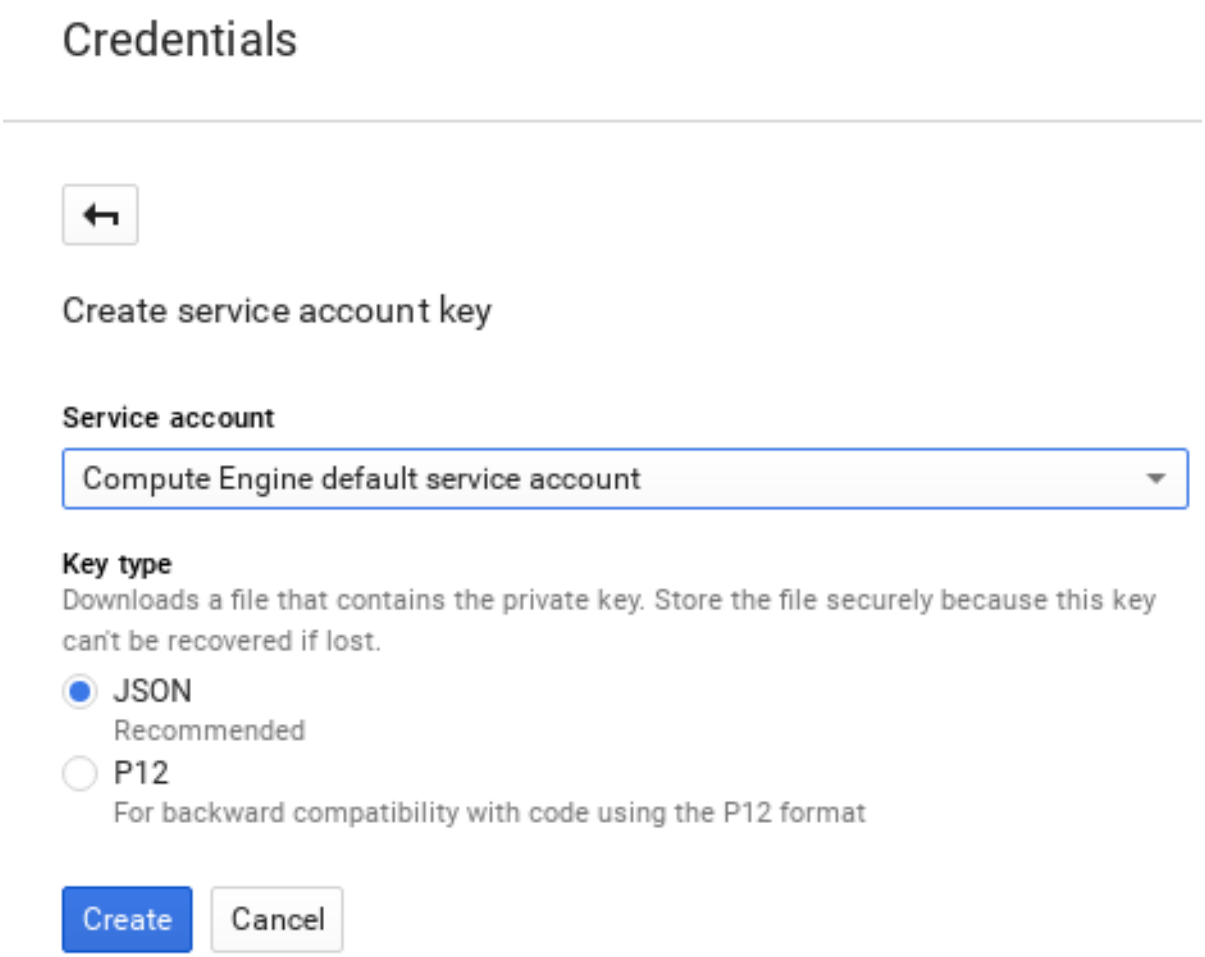
Block Storage サービス (cinder) では、Google クラウドにアクセスしてバックアップに使用するために、Google 認証情報が必要です。サービスアカウントキーを作成することで、この認証情報を Block Storage サービスに提供できます。

手順

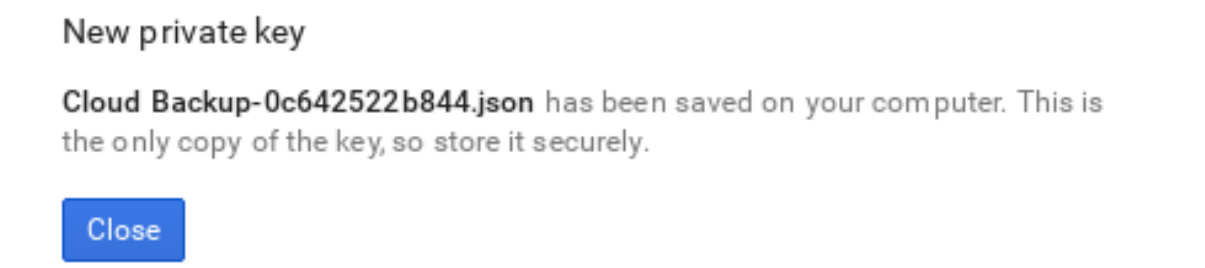
1. Google アカウントで Google デベロッパーコンソール (http://console.developers.google.com) にログインします。
2. **認証情報** タブをクリックし、**認証情報の作成** ドロップダウンメニューから **サービスアカウントの鍵** を選択します。



3. **Create service account key**画面で、Block Storage サービスで使用するサービスアカウントを **Service account** ドロップダウンメニューから選択します。



4. 同じ画面で、**Key**の種類のセクションから **JSON** を選択し、**作成** をクリックします。次に、ブラウザーはキーをデフォルトのダウンロード場所にダウンロードします。



5. ファイルを開き、**project_id** パラメーターの値を書き留めておきます。

```
{
  "type": "service_account",
  "project_id": "*cloud-backup-1370*",
...
```

6. GCS JSON 認証情報のコピーを **/home/stack/templates/Cloud-Backup.json** に保存します。

重要な影響

ファイル**Cloud-Backup.json**に名前を付け、ファイル名は変更しないでください。この JSON ファイルは、「**cinder-backup-gcs.yaml**の作成」 の手順で作成する**cinder-backup-gcs.yaml** ファイルと同じディレクトリーの場所に存在する必要があります。

A.2. CINDER-BACKUP-GCS.YAMLの作成

提供されたサンプルファイルを使用して、**cinder-backup-gcs.yaml** ファイルを作成します。

注記

この例 (およびファイル) で使用する空白と形式は重要です。空白を変更すると、ファイルが期待どおりに機能しなくなることがあります。

手順

1. 以下のテキストをコピーして、新しいファイルに貼り付けます。ファイルの内容を変更しないでください。

```
heat_template_version: rocky

description: >
  Post-deployment for configuration cinder-backup to GCS

parameters:
  servers:
    type: json
  DeployIdentifier:
    type: string

resources:
  CinderBackupGcsExtraConfig:
    type: OS::Heat::SoftwareConfig
    properties:
      group: script
      config:
        str_replace:
          template: |
            #!/bin/bash
            GCS_FILE=/var/lib/config-data/puppet-generated/cinder/etc/cinder/Cloud-
Backup.json
            HOSTNAME=$(hostname -s)
            for NODE in $(hiera -c /etc/puppet/hiera.yaml cinder_backup_short_node_names | tr
-d '[]",'); do
              if [ $NODE == $HOSTNAME ]; then
                cat <<EOF > $GCS_FILE
            GCS_JSON_DATA
            EOF
                chmod 0640 $GCS_FILE
                chown root:42407 $GCS_FILE
              fi
            done
          params:
            GCS_JSON_DATA: {get_file: Cloud-Backup.json}

  CinderBackupGcsDeployment:
```

```
    type: OS::Heat::SoftwareDeploymentGroup
    properties:
      servers:  {get_param: servers}
      config: {get_resource: CinderBackupGcsExtraConfig}
      actions: ['CREATE','UPDATE']
      input_values:
        deploy_identifier: {get_param: DeployIdentifier}
```

2. ファイルを**/home/stack/templates/cinder-backup-gcs.yaml** として保存します。

A.3. GOOGLE CLOUD 設定で環境ファイルの作成

Block Storage サービス (cinder) に適用する設定を含む環境ファイルを作成します。この場合、環境ファイルは、ボリュームバックアップを Google Cloud に保存するように Block Storage サービスを設定します。環境ファイルに関する詳しい情報は、『 director のインストールと使用方法』 を参照してください。

以下のサンプル環境ファイルを使用し、**Cloud-Backup.json** ファイルに一覧表示されているプロジェクト ID で**backup_gcs_project_id**を更新します。また、**backup_gcs_bucket_location**の場所を US から、自分の場所に近い場所に変更することもできます。

Google Cloud Backup Storage バックアップバックエンドの設定オプションの一覧は、表A.1「Google Cloud Storage バックアップバックエンド設定オプション」 を参照してください。

手順

1. 以下の環境ファイルの例をコピーします。空白領域の使用率は保持します。
2. 新しいファイルにコンテンツを貼り付けます。(**/home/stack/templates/cinder-backup-settings.yaml)**
3. **backup_gcs_project_id** の値を **cloud-backup-1370** から、**Cloud-Backup.json** に一覧表示されているプロジェクト ID に変更します。
4. ファイルを保存します。

環境ファイルの例

環境ファイルで各設定を定義します。表A.1「Google Cloud Storage バックアップバックエンド設定オプション」 を使用して、利用可能な設定オプションを選択します。

```
resource_registry:
  OS::TripleO::Services::CinderBackup: /usr/share/openstack-tripleo-heat-
templates/deployment/cinder/cinder-backup-pacemaker-puppet.yaml
  # For non-pcmk managed implementation
  # OS::TripleO::Services::CinderBackup: /usr/share/openstack-tripleo-heat-
templates/deployment/cinder/cinder-backup-container-puppet.yaml
  OS::TripleO::NodeExtraConfigPost: /home/stack/templates/cinder-backup-gcs.yaml

parameter_defaults:
  CinderBackupBackend: swift
  ExtraConfig:
    cinder::backup::swift::backup_driver: cinder.backup.drivers.gcs.GoogleBackupDriver
    cinder::config::cinder_config:
      DEFAULT/backup_gcs_credential_file:
        value: /etc/cinder/Cloud-Backup.json
```

```
DEFAULT/backup_gcs_project_id:
  value: cloud-backup-1370
DEFAULT/backup_gcs_bucket:
  value: cinder-backup-gcs
DEFAULT/backup_gcs_bucket_location:
  value: us
```

表A.1 Google Cloud Storage バックアップバックエンド設定オプション

PARAM	デフォルト	CONFIGの内容
backup_gcs_project_id		必須。使用しているサービスアカウントのプロジェクト ID で、「GCS 認証情報ファイルの作成」 のサービスアカウント鍵の**project_id**に含まれています。
backup_gcs_credential_file		「GCS 認証情報ファイルの作成」 で作成したサービスアカウントキーファイルの絶対パス。
backup_gcs_bucket		使用する GCS バケットまたはオブジェクトストレージリポジトリー (存在する場合とない場合があります)。存在しないバケットを指定すると、Google Cloud Storage バックアップドライバーがバケットを作成し、ここで指定した名前を割り当てます。詳細は「バケット および バケット名の要件」を参照してください。
backup_gcs_bucket_location	**us**	GCS バケットの場所。この値は、**backup_gcs_bucket** に存在しないバケットを指定する場合にのみ使用されます。この場合は、Google Cloud Storage バックアップドライバーが、これを GCS バケットの場所として指定します。
backup_gcs_object_size	**52428800**	GCS バックアップオブジェクトのサイズ (バイト単位)。
backup_gcs_block_size	**32768**	増分バックアップで変更が追跡されるサイズ (バイト単位)。この値は、**backup_gcs_object_size** 値の倍数にする必要があります。
backup_gcs_user_agent	**gcscinder**	GCS API の HTTP ユーザーエージェント文字列。

PARAM	デフォルト	CONFIGの内容
backup_gcs_reader_chunk_size	**2097152**	GCS オブジェクトは、このサイズのチャンク (バイト) でダウンロードされます。
backup_gcs_writer_chunk_size	**2097152**	GCS オブジェクトは、このサイズのチャンク (バイト) でアップロードされます。代わりにファイルを 1 つのチャンクとしてアップロードするには、値 -1 を使用します。
backup_gcs_num_retries	**3**	再試行の回数。
backup_gcs_storage_class	**NEARLINE**	GCS バケットのストレージクラス。この値は、**backup_gcs_bucket** に存在しないバケットを指定する場合にのみ使用されます。この場合、Google Cloud Storage バックアップドライバーは、これを GCS バケットストレージクラスとして指定します。詳細は「ストレージクラス」を参照してください。
backup_gcs_retry_error_codes	**429**	GCS エラーコードの一覧
backup_gcs_enable_progress_timer	**True**	ボリュームのバックアップ中に Telemetry サービス (ceilometer) に定期的な進捗通知を送信するタイマーを有効または無効にするブール値。これはデフォルトで有効になっています (True)。

警告

新しいバケットを作成すると、選択したストレージクラス (**backup_gcs_storage_class**) に基づいて、Google Cloud Storage が課金します。省略時の **NEARLINE** クラスは、バックアップサービスに適しています。

警告

バケットの作成後に、バケットの場所やクラスを編集することはできません。詳細は「バケットのストレージクラスまたは場所の管理」を参照してください。

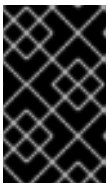
A.4. オーバークラウドのデプロイ

/home/stack/templates/ で環境ファイルファイルを作成したら、オーバークラウドをデプロイしてから、cinder-backup サービスを再起動します。

手順

1. **stack** ユーザとしてログインします。
2. 設定をデプロイします。

```
$ openstack overcloud deploy --templates \
-e /home/stack/templates/cinder-backup-settings.yaml
```



重要

オーバークラウドの作成時に追加の環境ファイルを渡した場合は、-e オプションを使用して再度渡します。これにより、オーバークラウドに望ましくない変更を防ぎます。

3. デプロイメントが終了したら、**cinder-backup** を再起動します。

詳しい情報は、『オーバークラウドの **高度なカスタマイズ』 の「オーバークラウド作成時** の 環境ファイル の追加」セクションを参照 してください。

付録B 高度な BLOCK STORAGE のバックアップ設定オプション

ディレクターによるインストールの前に、**cinder.conf** ファイルにより、ブロックストレージサービスとバックアップサービスが設定されていました。**cinder.conf** の値に、オーケストレーション (ヒート) テンプレートがない場合は、カスタム環境ファイルを使用して値をディレクターに渡すことができます。カスタム環境ファイル (**cinder-backup-settings.yaml** ファイルなど) の **parameter_defaults** セクションの **ExtraConfig** セクションに値を追加します。

ExtraConfig により、追加の hiera 設定を追加して、全ノードのクラスターに挿入することができます。この設定は、専用のバックアップノードに含まれます。ただし、**ExtraConfig** の代わりに **ControllerExtraConfig** を使用した場合、設定は Controller ノードにインストールされ、専用のバックアップノードにはインストールされません。

設定は、**cinder.conf** ファイルの **DEFAULT** セクションの **DEFAULT/[cinder.conf setting]`** で置き換えることができます。以下の例は、**ExtraConfig** エントリーが YAML ファイルにどのように表示されるかを示しています。

```
parameter_defaults:
 ExtraConfig:
   cinder::config::cinder_config:
     DEFAULT/backup_compression_algorithm:
       value: None
```

表 B.1 に、バックアップ関連のサンプルオプションの一覧を示します。

表B.1 ブロックストレージバックアップサービスの設定オプション

オプション	タイプ	デフォルト値	説明
backup_service_inithost_offload	任意	True	バックアップサービスの起動時に、バックアップの削除が保留中であることをオフロードします。false の場合、バックアップサービスは、保留中のバックアップがすべて削除されるまでダウンしたままになります。
use_multipath_for_image_xfer	任意	False	バックアップおよび復元の手順中にマルチパスを使用してボリュームを接続できる場合は、接続します。これは、イメージからのボリュームの作成、一般的なコールド移行、その他の操作など、すべての cinder の接続操作に影響します。
num_volume_device_scan_tries	任意	3	接続時にボリュームを検出するためにターゲットを再スキャンする回数の上限です。

オプション	タイプ	デフォルト値	説明
backup_workers	任意	1	実行するバックアッププロセスの数。複数の同時バックアップまたはリストアを実行し、圧縮を行うとパフォーマンスが大幅に向上します。
backup_native_threads_pool_size	任意	60	バックアップ用のネイティブスレッドプールのサイズ。ほとんどのバックアップドライバーは、これに大きく依存しています。このオプションに依存しない特定のドライバーの値は減らすことができます。
backup_share	必須/任意		**HOST**:_EXPORT_PATH_ に設定します。
backup_container	任意	None	(文字列) バックアップに使用するカスタムディレクトリー。
backup_enable_progress_timer	任意	True	ボリュームをバックエンドストレージにバックアップする際に、Telemetry サービス (ceilometer) に定期的な進捗通知を送信するタイマーを有効 (true) または無効 (false) にします。
backup_mount_options	任意		backup_share で指定した NFS エクスポートをマウントする際に指定できるオプションのコンマ区切りの一覧。
backup_mount_point_base	任意	$state_path/backup_mount	(文字列) NFS 共有のマウントポイントを含むベースディレクトリー。
backup_compression_algorithm	任意	zlib	バックアップデータをリポジトリーに送信する際に使用する圧縮アルゴリズム。有効な値は、**zlib**、**bz2**、および **None** です。

オプション	タイプ	デフォルト値	説明
backup_file_size	任意	1999994880	この値よりも大きな cinder ボリュームのデータは、バックアップリポジトリに複数ファイルとして保存されます。このオプションは、backup_sha_block_size_bytes の倍数にする必要があります。
backup_sha_block_size_bytes	任意	32768	デジタル署名の計算用の cinder ボリュームブロックのサイズ