



Red Hat OpenStack Platform 16.2

アンダークラウドおよびコントロールプレーン ノードのバックアップおよび復元

アンダークラウドおよびオーバークラウドコントロールプレーンノードのバック
アップの作成および復元

Red Hat OpenStack Platform 16.2 アンダークラウドおよびコントロールプレーンノードのバックアップおよび復元

アンダークラウドおよびオーバークラウドコントロールプレーンノードのバックアップの作成および復元

Enter your first name here. Enter your surname here.

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

Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Backing_up_and_restoring_the_undercloud_and_control_plane_nodes.ent file |.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本ガイドでは、アンダークラウドおよびコントロールプレーンノードのバックアップを作成し、復元する方法を説明します。バックアップは、Red Hat OpenStack Platform をアップグレードまたは行進する際に必要です。また、任意で環境の定期的なバックアップを作成することで、問題発生時のダウンタイムを最小限に抑えることができます。

目次

前書き	3
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の提供	5
第1章 アンダークラウドノードのバックアップ	6
1.1. サポート対象のバックアップ形式およびプロトコル	6
1.2. バックアップノードへの NFS サーバーのインストールと構成	6
1.3. アンダークラウドノードへの REAR のインストール	7
1.4. アンダークラウドノードのスタンドアロンデータベースバックアップの作成	8
1.5. バックアップ用の OPEN VSWITCH (OVS) インターフェースの設定	10
1.6. アンダークラウドノードのバックアップの作成	10
第2章 コントロールプレーンノードのバックアップ	12
2.1. サポート対象のバックアップ形式およびプロトコル	12
2.2. バックアップノードへの NFS サーバーのインストールと構成	12
2.3. コントロールプレーンノードへの REAR のインストール	13
2.4. バックアップ用の OPEN VSWITCH (OVS) インターフェースの設定	14
2.5. コントロールプレーンノードのバックアップの作成	15
2.6. CRON を使用したコントロールプレーンノードバックアップのスケジューリング	16
第3章 アンダークラウドおよびコントロールプレーンノードの復元	18
3.1. 復元プロセスのために、同じ場所に配置された CEPH モニターを備えたコントロールプレーンを準備	18
3.2. アンダークラウドノードの復元	19
3.3. コントロールプレーンノードの復元	20
3.4. GALERA クラスターの手動による復元	22
3.5. アンダークラウドのノードデータベースを手動で復元する	25

前書き

多様性を受け入れるオープンソースの強化

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

RED HAT ドキュメントへのフィードバックの提供

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

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

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

1. **Multi-page HTML** 形式でドキュメントを表示します。
2. ドキュメントの右上隅に **Feedback** ボタンが表示されていることを確認してください。
3. コメントするテキスト部分をハイライト表示します。
4. **Add Feedback** をクリックします。
5. **Add Feedback** フィールドにコメントを入力します。
6. (オプション) ドキュメントチームが連絡を取り問題についてお伺いできるように、ご自分のメールアドレスを追加します。
7. **送信** をクリックします。

第1章 アンダークラウドノードのバックアップ

アンダークラウドノードをバックアップするには、バックアップノードを設定し、アンダークラウドノードに Relax-and-Recover ツールをインストールし、バックアップイメージを作成します。バックアップは、通常的环境メンテナンスの一環として作成できます。

さらに、更新またはアップグレードを実行する前にアンダークラウドノードをバックアップする必要があります。バックアップを使用して、更新またはアップグレード時にエラーが発生した場合に、アンダークラウドノードを以前の状態に復元することができます。

1.1. サポート対象のバックアップ形式およびプロトコル

アンダークラウドおよびバックアップ/復元プロセスは、オープンソースのツールである Relax-and-Recover (ReaR)を使用してブート可能なバックアップイメージを作成し、復元します。ReaR は Bash で記述されており、複数のイメージ形式と複数の転送プロトコルをサポートします。

次の一覧は、ReaR を使用してアンダークラウドとコントロールプレーンをバックアップおよび復元するときに Red Hat OpenStack Platform がサポートするバックアップ形式とプロトコルを示しています。

ブート可能なメディア形式

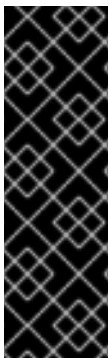
- ISO

ファイルトランスポートプロトコル

- SFTP
- NFS

1.2. バックアップノードへの NFS サーバーのインストールと構成

バックアップファイルを保存するために、新しい NFS サーバーをインストールして構成できます。バックアップノードに NFS サーバーをインストールして設定するには、インベントリーファイルを作成して SSH キーを設定して、NFS サーバーオプションを指定して **openstack undercloud backup** コマンドを実行します。



重要

- NFS サーバーまたは SFTP サーバーをインストールして設定している場合は、この手順を実行する必要はありません。バックアップするノードに ReaR を設定するときに、サーバー情報を入力します。
- デフォルトでは、Relax and Recover (ReaR) 設定は、NFS サーバーの IP アドレスが **192.168.24.1** であることを前提としています。NFS サーバーの IP アドレスが異なる場合は、設定 ReaR コマンドにパラメーター **tripleo_backup_and_restore_nfs_server** を追加する必要があります。

手順

1. アンダークラウドノードにおいて、source コマンドでアンダークラウドの認証情報を読み込みます。

```
[stack@undercloud-0 ~]$ source stackrc
(undercloud) [stack@undercloud ~]$
```

- アンダークラウドノードで、バックアップノードのインベントリーファイルを作成します。

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF'> ~/nfs-inventory.yaml
[BackupNode]
<backup_node> ansible_host=<ip_address> ansible_user=<user>
EOF
```

<ip_address> および **<user>** を、実際の環境に該当する値に置き換えます。

- 公開 SSH 鍵をアンダークラウドノードからバックアップノードにコピーします。

```
(undercloud) [stack@undercloud ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub <backup_node>
```

<backup_node> をバックアップノードのパスおよび名前に置き換えます。

- バックアップノードに NFS サーバーを設定します。

```
(undercloud) [stack@undercloud ~]$ openstack undercloud backup --setup-nfs --extra-vars
/home/stack/bar-vars.yaml --inventory /home/stack/nfs-inventory.yaml
```



注記

- デフォルトでは、**tripleo_backup_and_restore_nfs_server** パラメーターは **bar-vars.yaml** 環境ファイルで **192.168.24.1** に設定されています。独自の NFS サーバーをインストールして設定している場合は、**tripleo_backup_and_restore_nfs_server** パラメーターを **bar-vars.yaml** に追加し、値をご自分の NFS サーバーの IP アドレスに設定します。
- デフォルトでは、**tripleo_backup_and_restore_nfs_clients_nets** パラメーターは **bar-vars.yaml** 環境ファイルで **['192.168.24.0/24', '10.0.0.0/24', '172.16.0.0/24']** に設定されています。NFS バックアップディレクトリーが正しい場所にエクスポートされるように、アンダークラウドおよびオーバークラウドがデプロイされるネットワーク範囲を追加します。

1.3. アンダークラウドノードへの REAR のインストール

アンダークラウドノードのバックアップを作成する前に、アンダークラウドに Relax and Recover (ReaR) をインストールして設定します。

前提条件

- バックアップノードに NFS または SFTP サーバーがインストールおよび構成されています。新しい NFS サーバーの作成方法は「[バックアップノードへの NFS サーバーのインストールと構成](#)」を参照してください。

手順

- アンダークラウドノードにおいて、source コマンドでアンダークラウドの認証情報を読み込みます。

-

```
[stack@undercloud-0 ~]$ source stackrc
```

カスタムのスタック名を使用する場合は、**--stack <stack_name>** オプションを **tripleo-ansible-inventory** コマンドに追加します。

2. 事前に実行していない場合は、インベントリーファイルを作成し、**tripleo-ansible-inventory** コマンドを使用して、すべてのオーバークラウドノードのホストおよび変数が含まれる静的なインベントリーファイルを生成します。

```
(undercloud) [stack@undercloud ~]$ tripleo-ansible-inventory \
--ansible_ssh_user heat-admin \
--static-yaml-inventory /home/stack/tripleo-inventory.yaml
```

3. アンダークラウドノードに ReaR をインストールします。

```
(undercloud) [stack@undercloud ~]$ openstack undercloud backup --setup-rear --extra-vars \
/home/stack/bar-vars.yaml --inventory /home/stack/tripleo-inventory.yaml
```

4. **bar-vars.yaml** ファイルで、バックアップの保存場所を構成します。

- a. NFS サーバーを使用し、独自の NFS サーバーをインストールして設定している場合は、**tripleo_backup_and_restore_nfs_server** パラメーターを追加し、値を自身の NFS サーバーの IP アドレスに設定します。

```
tripleo_backup_and_restore_nfs_server: <ip_address>
```

デフォルトでは、**tripleo_backup_and_restore_nfs_server** パラメーターの値は **192.168.24.1** に設定されています。

- b. バックアップの保存に SFTP サーバーを使用する場合は、**tripleo_backup_and_restore_output_url** パラメーターを追加し、SFTP サーバーの URL と認証情報の値を設定します。

```
tripleo_backup_and_restore_output_url: sftp://<user>:<password>@<backup_node>/
tripleo_backup_and_restore_backup_url: iso:///backup/
```

<user>、**<password>**、および **<backup_node>** を、バックアップノードの URL および認証情報に置き換えます。

5. システムで UEFI ブートローダーを使用している場合は、アンダークラウドノードで以下の手順を実施します。

- a. 以下のツールをインストールします。

```
$ sudo dnf install dosfstools efibootmgr
```

- b. **USING_UEFI_BOOTLOADER** パラメーターの値 **0** を値 **1** に置き換えて、**/etc/rear/local.conf** にある ReaR 設定ファイルで UEFI バックアップを有効にします。

1.4. アンダークラウドノードのスタンドアロンデータベースバックアップの作成

Red Hat OpenStack Platform 環境を 13 から 16.2 にアップグレードする場合は、アンダークラウドのアップグレードを実行してから、オーバークラウドノードで Leapp によるアップグレードプロセスを

実施する前に、スタンドアロンデータベースのバックアップを作成する必要があります。その他のシナリオでは、任意として、アンダークラウドノードのフルバックアップに加えて、スタンドアロンのデータベースのバックアップを作成することができます。



注記

アンダークラウドノードの完全なバックアップを作成する場合は、アンダークラウドノードのデータベースバックアップが含まれます。したがって、フルバックアップからアンダークラウドを復元する予定がある場合は、スタンドアロンのデータベースバックアップを作成する必要がありません。ただし、アンダークラウドの完全復元プロセスが失敗し、アンダークラウドデータベースを手動で復元する場合は、この手順で作成したスタンドアロンデータベースのバックアップにアクセスできる必要があります。

手順

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

```
(undercloud) [stack@undercloud ~]$ source ~/stackrc
```

3. 事前に実行していない場合は、インベントリーファイルを作成し、**tripleo-ansible-inventory** コマンドを使用して、すべてのオーバークラウドノードのホストおよび変数が含まれる静的なインベントリーファイルを生成します。

```
(undercloud) [stack@undercloud ~]$ tripleo-ansible-inventory \
--ansible_ssh_user heat-admin \
--static-yaml-inventory /home/stack/tripleo-inventory.yaml
```

4. アンダークラウドノードで、以下の Ansible Playbook を作成します。

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF' >
~/bar_create_undercloud_db_backup.yaml
# Playbook
# DB backup on the undercloud node.
- become: true
  hosts: undercloud
  tasks:
    - name: Create DB Backup for the undercloud
      include_role:
        name: backup-and-restore
      tasks_from: ../backup/tasks/db_backup.yml
      vars:
        tripleo_backup_and_restore_service_manager: false
EOF
```

5. アンダークラウドノードのデータベースのバックアップを作成します。

```
(undercloud) [stack@undercloud ~]$ ansible-playbook \
-v -i ~/tripleo-inventory.yaml \
--extra="ansible_ssh_common_args=-o StrictHostKeyChecking=no" \
--become \
--become-user root \
~/bar_create_undercloud_db_backup.yaml
```

関連情報

- [Framework for Upgrades \(13 to 16.2\)](#)
- [「アンダークラウドノードのバックアップの作成」](#)
- [「アンダークラウドのノードデータベースを手動で復元する」](#)

1.5. バックアップ用の OPEN VSWITCH (OVS) インターフェースの設定

お使いの環境で Open vSwitch (OVS) ブリッジを使用する場合は、アンダークラウドまたはコントロールプレーンノードをバックアップする前に OVS インターフェースを手動で設定する必要があります。復元プロセスでは、この情報を使用してネットワークインターフェースを復元します。

手順

- `/etc/rear/local.conf` ファイルに、以下の形式で **NETWORKING_PREPARATION_COMMANDS** パラメーターを追加します。

```
NETWORKING_PREPARATION_COMMANDS=('<command_1>' '<command_2>' ...)
```

<command_1> および **<command_2>** を、ネットワークインターフェース名または IP アドレスを設定するコマンドに置き換えます。たとえば、`ip link add br-ctlplane type bridge` コマンドを追加してコントロールプレーンのブリッジ名を設定するか、`ip link set eth0 up` コマンドを追加してインターフェースの名前を設定できます。ネットワーク設定に基づいて、パラメーターにさらにコマンドを追加します。

1.6. アンダークラウドノードのバックアップの作成

アンダークラウドノードのバックアップを作成するには、**openstack undercloud backup** コマンドを使用します。その後、ノードが破損したりアクセスできなくなったりした場合に備えて、バックアップを使用して、アンダークラウドノードを以前の状態に復元できます。アンダークラウドノードのバックアップには、アンダークラウドノードで実行されるデータベースのバックアップが含まれます。

Red Hat OpenStack Platform 環境を 13 から 16.2 にアップグレードする場合は、アンダークラウドのアップグレードを実行してから、オーバークラウドノードで Leapp によるアップグレードプロセスを実施する前に、データベースのバックアップを別に作成する必要があります。詳細は、[「アンダークラウドノードのスタンドアロンデータベースバックアップの作成」](#)を参照してください。



注記

Red Hat OpenStack Platform 環境を 13 から 16.2 にアップグレードする場合は、アンダークラウドのアップグレードを実行してから、オーバークラウドノードで Leapp によるアップグレードプロセスを実施する前に、データベースのバックアップを別に作成する必要があります。詳しくは、[「アンダークラウドノードのデータベースバックアップの作成」](#)を参照してください。

前提条件

- バックアップノードに NFS または SFTP サーバーがインストールおよび構成されています。新しい NFS サーバーの作成方法は [「バックアップノードへの NFS サーバーのインストールと構成」](#) を参照してください。

- コントロールプレーンノードに ReaR がインストールされている。詳細は、[「アンダークラウドノードへの ReaR のインストール」](#)を参照してください。
- ネットワークインターフェースに OVS ブリッジを使用する場合は、OVS インターフェースを設定することができます。詳細は、[「バックアップ用の Open vSwitch \(OVS\) インターフェースの設定」](#)を参照してください。

手順

1. アンダークラウドノードにおいて、source コマンドでアンダークラウドの認証情報を読み込みます。

```
[stack@undercloud-0 ~]$ source stackrc
```

2. 事前に実行していない場合は、インベントリーファイルを作成し、**tripleo-ansible-inventory** コマンドを使用して、すべてのオーバークラウドノードのホストおよび変数が含まれる静的なインベントリーファイルを生成します。

```
(undercloud) [stack@undercloud ~]$ tripleo-ansible-inventory \  
--ansible_ssh_user heat-admin \  
--static-yaml-inventory /home/stack/tripleo-inventory.yaml
```

3. アンダークラウドノードのバックアップを作成します。

```
(undercloud) [stack@undercloud ~]$ openstack undercloud backup --inventory \  
/home/stack/tripleo-inventory.yaml
```

第2章 コントロールプレーンノードのバックアップ

コントロールプレーンノードをバックアップするには、バックアップノードを設定し、コントロールプレーンノードに Relax-and-Recover ツールをインストールし、バックアップイメージを作成します。バックアップは、通常的环境メンテナンスの一環として作成できます。

さらに、更新またはアップグレードを実行する前にコントロールプレーンノードをバックアップする必要があります。更新またはアップグレード時にエラーが発生した場合に、バックアップを使用して、コントロールプレーンノードを以前の状態に復元することができます。

2.1. サポート対象のバックアップ形式およびプロトコル

アンダークラウドおよびバックアップ/復元プロセスは、オープンソースのツールである Relax-and-Recover (ReaR)を使用してブート可能なバックアップイメージを作成し、復元します。ReaR は Bash で記述されており、複数のイメージ形式と複数の転送プロトコルをサポートします。

次の一覧は、ReaR を使用してアンダークラウドとコントロールプレーンをバックアップおよび復元するときに Red Hat OpenStack Platform がサポートするバックアップ形式とプロトコルを示しています。

ブート可能なメディア形式

- ISO

ファイルトランスポートプロトコル

- SFTP
- NFS

2.2. バックアップノードへの NFS サーバーのインストールと構成

バックアップファイルを保存するために、新しい NFS サーバーをインストールして構成できます。バックアップノードに NFS サーバーをインストールして設定するには、インベントリーファイルを作成して SSH キーを設定して、NFS サーバーオプションを指定して **openstack undercloud backup** コマンドを実行します。



重要

- NFS サーバーまたは SFTP サーバーをインストールして設定している場合は、この手順を実行する必要はありません。バックアップするノードに ReaR を設定するときに、サーバー情報を入力します。
- デフォルトでは、Relax and Recover (ReaR) 設定は、NFS サーバーの IP アドレスが **192.168.24.1** であることを前提としています。NFS サーバーの IP アドレスが異なる場合は、設定 ReaR コマンドにパラメーター **tripleo_backup_and_restore_nfs_server** を追加する必要があります。

手順

1. アンダークラウドノードにおいて、source コマンドでアンダークラウドの認証情報を読み込みます。


```
[stack@undercloud-0 ~]$ source stackrc
(undercloud) [stack@undercloud ~]$
```

- アンダークラウドノードで、バックアップノードのインベントリーファイルを作成します。

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF'> ~/nfs-inventory.yaml
[BackupNode]
<backup_node> ansible_host=<ip_address> ansible_user=<user>
EOF
```

<ip_address> および **<user>** を、実際の環境に該当する値に置き換えます。

- 公開 SSH 鍵をアンダークラウドノードからバックアップノードにコピーします。

```
(undercloud) [stack@undercloud ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub <backup_node>
```

<backup_node> をバックアップノードのパスおよび名前に置き換えます。

- バックアップノードに NFS サーバーを設定します。

```
(undercloud) [stack@undercloud ~]$ openstack undercloud backup --setup-nfs --extra-vars
/home/stack/bar-vars.yaml --inventory /home/stack/nfs-inventory.yaml
```



注記

- デフォルトでは、**tripleo_backup_and_restore_nfs_server** パラメーターは **bar-vars.yaml** 環境ファイルで **192.168.24.1** に設定されています。独自の NFS サーバーをインストールして設定している場合は、**tripleo_backup_and_restore_nfs_server** パラメーターを **bar-vars.yaml** に追加し、値をご自分の NFS サーバーの IP アドレスに設定します。
- デフォルトでは、**tripleo_backup_and_restore_nfs_clients_nets** パラメーターは **bar-vars.yaml** 環境ファイルで **['192.168.24.0/24', '10.0.0.0/24', '172.16.0.0/24']** に設定されています。NFS バックアップディレクトリーが正しい場所にエクスポートされるように、アンダークラウドおよびオーバークラウドがデプロイされるネットワーク範囲を追加します。

2.3. コントロールプレーンノードへの REAR のインストール

オーバークラウドコントロールプレーンのバックアップを作成する前に、各コントロールプレーンノードに Relax and Recover (ReaR) をインストールして設定します。

前提条件

- バックアップノードに NFS または SFTP サーバーがインストールおよび構成されています。新しい NFS サーバーの作成方法は「[バックアップノードへの NFS サーバーのインストールと構成](#)」を参照してください。

手順

- アンダークラウドノードにおいて、source コマンドでアンダークラウドの認証情報を読み込みます。

-

```
[stack@undercloud-0 ~]$ source stackrc
```

2. 事前に実行していない場合は、インベントリーファイルを作成し、**tripleo-ansible-inventory** コマンドを使用して、すべてのオーバークラウドノードのホストおよび変数が含まれる静的なインベントリーファイルを生成します。

```
(undercloud) [stack@undercloud ~]$ tripleo-ansible-inventory \
--ansible_ssh_user heat-admin \
--static-yaml-inventory /home/stack/tripleo-inventory.yaml
```

3. コントロールプレーンノードに ReaR をインストールします。

```
(undercloud) [stack@undercloud ~]$ openstack overcloud backup --setup-rear --extra-vars
/home/stack/bar-vars.yaml --inventory /home/stack/tripleo-inventory.yaml
```

4. **bar-vars.yaml** ファイルで、バックアップの保存場所を構成します。

- a. NFS サーバーを使用し、独自の NFS サーバーをインストールして設定している場合は、**tripleo_backup_and_restore_nfs_server** パラメーターを追加し、値を自身の NFS サーバーの IP アドレスに設定します。

```
tripleo_backup_and_restore_nfs_server: <ip_address>
```

デフォルトでは、**tripleo_backup_and_restore_nfs_server** パラメーターの値は **192.168.24.1** に設定されています。

- b. SFTP サーバーを使用する場合は、**tripleo_backup_and_restore_output_url** パラメーターを追加し、SFTP サーバーの URL と認証情報の値を設定します。

```
tripleo_backup_and_restore_output_url: sftp://<user>:<password>@<backup_node>/
tripleo_backup_and_restore_backup_url: iso:///backup/
```

<user>、**<password>**、および **<backup_node>** を、バックアップノードの URL および認証情報に置き換えます。

5. システムで UEFI ブートローダーを使用している場合は、コントロールプレーンノードで以下の手順を実施します。

- a. 以下のツールをインストールします。

```
$ sudo dnf install dosfstools efibootmgr
```

- b. **USING_UEFI_BOOTLOADER** パラメーターの値 **0** を値 **1** に置き換えて、**/etc/rear/local.conf** にある ReaR 設定ファイルで UEFI バックアップを有効にします。

2.4. バックアップ用の OPEN VSWITCH (OVS) インターフェースの設定

お使いの環境で Open vSwitch (OVS) ブリッジを使用する場合は、アンダークラウドまたはコントロールプレーンノードをバックアップする前に OVS インターフェースを手動で設定する必要があります。復元プロセスでは、この情報を使用してネットワークインターフェースを復元します。

手順

- `/etc/rear/local.conf` ファイルに、以下の形式で **NETWORKING_PREPARATION_COMMANDS** パラメーターを追加します。

```
NETWORKING_PREPARATION_COMMANDS=('<command_1>' '<command_2>' ...')
```

<command_1> および **<command_2>** を、ネットワークインターフェース名または IP アドレスを設定するコマンドに置き換えます。たとえば、`ip link add br-ctlplane type bridge` コマンドを追加してコントロールプレーンのブリッジ名を設定するか、`ip link set eth0 up` コマンドを追加してインターフェースの名前を設定できます。ネットワーク設定に基づいて、パラメーターにさらにコマンドを追加します。

2.5. コントロールプレーンノードのバックアップの作成

コントロールプレーンノードのバックアップを作成するには、**openstack overcloud backup** コマンドを使用します。その後、ノードが破損したりアクセスできなくなったりした場合に備えて、バックアップを使用して、コントロールプレーンノードを以前の状態に復元できます。コントロールプレーンのバックアップには、コントロールプレーンノードで実行されるデータベースのバックアップが含まれます。

前提条件

- バックアップノードに NFS または SFTP サーバーがインストールおよび構成されています。新しい NFS サーバーの作成方法は「[バックアップノードへの NFS サーバーのインストールと構成](#)」を参照してください。
- コントロールプレーンノードに ReaR がインストールされている。詳細は、「[コントロールプレーンノードへの ReaR のインストール](#)」を参照してください。
- ネットワークインターフェースに OVS ブリッジを使用する場合は、OVS インターフェースを設定することができます。詳細は、「[バックアップ用の Open vSwitch \(OVS\) インターフェースの設定](#)」を参照してください。

手順

1. 各コントロールプレーンノードで、各ノードの **config-drive** パーティションを **root** ユーザーとしてバックアップします。

```
[root@controller-x ~]$ dd if=<config_drive_partition> of=/mnt/config-drive
```

2. アンダークラウドノードにおいて、`source` コマンドでアンダークラウドの認証情報を読み込みます。

```
[stack@undercloud-0 ~]$ source stackrc
```

3. 事前に実行していない場合は、**tripleo-ansible-inventory** コマンドを使用して、すべてのオーバークラウドノードのホストおよび変数が含まれる静的なインベントリーファイルを生成します。

```
(undercloud) [stack@undercloud ~]$ tripleo-ansible-inventory \
--ansible_ssh_user heat-admin \
--static-yaml-inventory /home/stack/tripleo-inventory.yaml
```

4. コントロールプレーンノードのバックアップを作成します。

```
(undercloud) [stack@undercloud ~]$ openstack overcloud backup --inventory
/home/stack/tripleo-inventory.yaml
```

バックアッププロセスは、環境へのサービスを中断することなく、各コントロールプレーンノードで順次実行されます。

2.6. CRON を使用したコントロールプレーンノードバックアップのスケジューリング



重要

この機能は、本リリースでは **テクノロジープレビュー** として提供しているため、Red Hat では全面的にはサポートしていません。これは、テスト用途にのみご利用いただく機能で、実稼働環境にデプロイすべきではありません。テクノロジープレビュー機能についての詳しい情報は、「[対象範囲の詳細](#)」を参照してください。

cron ジョブを設定し、Ansible **backup-and-restore** ロールを使用して ReaR でコントロールプレーンノードのバックアップを作成することができます。/var/log/rear-cron ディレクトリーでログを確認することができます。

前提条件

- バックアップノードに NFS または SFTP サーバーがインストールおよび構成されています。新しい NFS サーバーの作成方法は「[バックアップノードへの NFS サーバーのインストールと構成](#)」を参照してください。
- アンダークラウドおよびコントロールプレーンノードに ReaR がインストールされている。詳細は、「[コントロールプレーンノードへの ReaR のインストール](#)」を参照してください。
- バックアップの保存用に、バックアップの場所に十分なディスク領域がある。

手順

1. アンダークラウドノードで、以下のコマンドを入力してバックアップ用のスクリプトを作成します。

```
[stack@undercloud ~]$ cat <<'EOF' > /home/stack/execute-rear-cron.sh

#!/bin/bash

OWNER="stack"
TODAY=`date +%Y%m%d`
FILE="/var/log/rear-cron.${TODAY}"
sudo touch ${FILE}
sudo chown ${OWNER}:${OWNER} ${FILE}

CURRENTTIME=`date`
echo "[${CURRENTTIME}] rear start" >> ${FILE}
source /home/stack/stackrc && /usr/bin/openstack overcloud backup 2>&1 >> ${FILE}
CURRENTTIME=`date`
echo "[${CURRENTTIME}] rear end" >> ${FILE}
EOF
```

2. **/home/stack/execute-rear-cron.sh** スクリプトに実行可能な権限を設定します。

```
[stack@undercloud ~]$ chmod 755 /home/stack/execute-rear-cron.sh
```

3. **crontab -e** コマンドで crontab ファイルを編集し、任意のエディターで以下の cron ジョブを追加します。ファイルに加えた変更を保存するようにしてください。

```
[stack@undercloud ~]# $ crontab -e
#adding the following line
0 0 * * * /home/stack/execute-rear-cron.sh
```

/home/stack/execute-rear-cron.sh スクリプトは、stack ユーザーにより午前 0 時に実行されるようにスケジュールされます。

4. cron ジョブがスケジュールされていることを確認するには、以下のコマンドを入力します。

```
[stack@undercloud ~]$ crontab -l
```

コマンド出力には、スケジュールされている cron ジョブが表示されます。

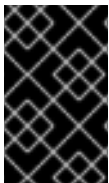
```
0 0 * * * /home/stack/execute-rear-cron.sh
```

第3章 アンダークラウドおよびコントロールプレーンノードの復元

アンダークラウドまたはコントロールプレーンノードが破損している場合、もしくは更新またはアップグレード中にエラーが発生した場合は、コントロールプレーンノードをバックアップから以前の状態に復元することができます。復元プロセスが Galera クラスターまたは共存する Ceph モニターを持つノードを自動的に復元できない場合は、これらのコンポーネントを手動で復元できます。

3.1. 復元プロセスのために、同じ場所に配置された CEPH モニターを備えたコントロールプレーンを準備

同じ場所に配置された Ceph モニターを使用してコントロールプレーンノードを復元する前に、Ceph モニターのバックアップファイルをノードファイルシステムにマウントするスクリプトと、ReaR がバックアップファイルを見つけるために使用する別のスクリプトを作成して、環境を準備します。



重要

`/var/lib/ceph` ディレクトリーのバックアップが作成できない場合は、Red Hat 技術サポートチームにお問い合わせいただき、**ceph-mon** インデックスを再ビルドする必要があります。詳細は、[Red Hat Technical Support Team](#) を参照してください。

前提条件

- アンダークラウドノードのバックアップを作成している。詳細は、「[アンダークラウドノードのバックアップの作成](#)」を参照してください。
- コントロールプレーンノードのバックアップを作成している。詳細は、「[コントロールプレーンノードのバックアップの作成](#)」を参照してください。
- バックアップノードにアクセスすることができる。
- ネットワークインターフェースの OVS ブリッジを使用する場合は、**NETWORKING_PREPARATION_COMMANDS** パラメーターで設定したネットワーク設定情報にアクセスできます。詳細は、「[バックアップ用の Open vSwitch \(OVS\) インターフェースの設定](#)」を参照してください。

手順

1. 復元する各ノードで、スクリプト `/usr/share/rear/setup/default/011_backup_ceph.sh` を作成し、次のコンテンツを追加します。

```
mount -t <file_type> <device_disk> /mnt/local
cd /mnt/local
[ -d "var/lib/ceph" ] && tar cvfz /tmp/ceph.tar.gz var/lib/ceph --xattrs --xattrs-include='*' --acls
cd /
umount <device_disk>
```

<file_type> および **<device_disk>** は、バックアップファイルの種類と場所に置き換えてください。通常、ファイルタイプは **xfs** で、場所は **/dev/vda2** です。

2. 同じノードで、スクリプト `/usr/share/rear/wrapup/default/501_restore_ceph.sh` を作成し、以下の出力を追加します。

```
if [ -f "/tmp/ceph.tar.gz" ]; then
  rm -rf /mnt/local/var/lib/ceph/*
```

```
tar xvC /mnt/local -f /tmp/ceph.tar.gz var/lib/ceph --xattrs --xattrs-include='.'
fi
```

関連情報

- [「アンダークラウドノードの復元」](#)
- [「コントロールプレーンノードの復元」](#)

3.2. アンダークラウドノードの復元

ReaR を使用して作成したバックアップの ISO イメージを使用して、アンダークラウドノードを以前の状態に復元することができます。バックアップの ISO イメージは、バックアップノードにあります。ブート可能な ISO イメージを DVD に書き込むか、Integrated Lights-Out (iLO) リモートアクセスを通じてアンダークラウドノードにダウンロードします。

前提条件

- アンダークラウドノードのバックアップを作成している。詳細は、[「コントロールプレーンノードのバックアップの作成」](#)を参照してください。
- バックアップノードにアクセスすることができる。
- ネットワークインターフェースの OVS ブリッジを使用する場合は、**NETWORKING_PREPARATION_COMMANDS** パラメーターで設定したネットワーク設定情報にアクセスできます。詳細は、[「バックアップ用の Open vSwitch \(OVS\) インターフェースの設定」](#)を参照してください。

手順

1. アンダークラウドノードの電源をオフにします。次のステップに進む前に、アンダークラウドノードの電源が完全にオフになっていることを確認します。
2. バックアップの ISO イメージでアンダークラウドノードをブートします。
3. **Relax-and-Recover** ブートメニューが表示されたら、**Recover <undercloud_node>** を選択します。<undercloud_node> をアンダークラウドノードの名前に置き換えます。



注記

システムで UEFI を使用している場合は、**Relax-and-Recover (no Secure Boot)** オプションを選択します。

4. **root** ユーザーとしてログインし、ノードを復元します。以下のメッセージが表示されます。

```
Welcome to Relax-and-Recover. Run "rear recover" to restore your system!
RESCUE <undercloud_node>:~ # rear recover
```

アンダークラウドノードの復元プロセスが完了すると、コンソールに以下のメッセージが表示されます。

```
Finished recovering your system
Exiting rear recover
Running exit tasks
```

5. ノードの電源を切ります。

```
RESCUE <undercloud_node>:~ # poweroff
```

ノードをブートすると、以前の状態で再開されます。

3.3. コントロールプレーンノードの復元

更新またはアップグレード中にエラーが発生した場合には、ReaR を使用して作成したバックアップの ISO イメージを使用して、コントロールプレーンノードを以前の状態に復元することができます。

コントロールプレーンを復元する場合は、状態の整合性を確保するために、すべてのコントロールプレーンノードを復元する必要があります。

バックアップの ISO イメージは、バックアップノードにあります。ブート可能な ISO イメージを DVD に書き込むか、Integrated Lights-Out (iLO) リモートアクセスを通じてアンダークラウドノードにダウンロードします。



注記

Red Hat は、Open vSwitch (OVS) およびデフォルトの Open Virtual Network (OVN) などのネイティブ SDN を使用する Red Hat OpenStack Platform のバックアップをサポートします。サードパーティーの SDN についての詳細は、サードパーティーの SDN ドキュメントを参照してください。

前提条件

- コントロールプレーンノードのバックアップを作成している。詳細は、[「コントロールプレーンノードのバックアップの作成」](#)を参照してください。
- バックアップノードにアクセスすることができる。
- ネットワークインターフェースの OVS ブリッジを使用する場合は、**NETWORKING_PREPARATION_COMMANDS** パラメーターで設定したネットワーク設定情報にアクセスできます。詳細は、[「バックアップ用の Open vSwitch \(OVS\) インターフェースの設定」](#)を参照してください。

手順

1. 各コントロールプレーンノードの電源をオフにします。次のステップに進む前に、コントロールプレーンノードの電源が完全にオフになっていることを確認します。
2. 対応するバックアップの ISO イメージで各コントロールプレーンノードをブートします。
3. **Relax-and-Recover** ブートメニューが表示されたら、各コントロールプレーンノードで **Recover <control_plane_node>** を選択します。<control_plane_node> を対応するコントロールプレーンノードの名前に置き換えます。



注記

システムで UEFI を使用している場合は、**Relax-and-Recover (no Secure Boot)** オプションを選択します。

- それぞれのコントロールプレーンノードで、**root** ユーザーとしてログインしてノードを復元します。
以下のメッセージが表示されます。

```
Welcome to Relax-and-Recover. Run "rear recover" to restore your system!
RESCUE <control_plane_node>:~ # rear recover
```

コントロールプレーンノードの復元プロセスが完了すると、コンソールに以下のメッセージが表示されます。

```
Finished recovering your system
Exiting rear recover
Running exit tasks
```

- コマンドラインコンソールが利用可能な場合は、各コントロールプレーンノードの **config-drive** パーティションを復元します。

```
# once completed, restore the config-drive partition (which is ISO9660)
RESCUE <control_plane_node>:~ $ dd if=/mnt/local/mnt/config-drive of=
<config_drive_partition>
```

- ノードの電源を切ります。

```
RESCUE <control_plane_node>:~ # poweroff
```

- ブートシーケンスを通常のブートデバイスに設定します。ノードをブートすると、以前の状態で再開されます。
- サービスが正常に実行されていることを確認するには、**pacemaker** のステータスを確認します。**root** ユーザーとしてコントローラーノードにログインし、以下のコマンドを入力します。

```
# pcs status
```

- オーバークラウドのステータスを確認するには、OpenStack Integration Test Suite (**tempest**) を使用します。詳細は、[Validating your OpenStack cloud with the Integration Test Suite \(tempest\)](#) を参照してください。

トラブルシューティング

- pcs status** で表示されるリソースアラームを以下のコマンドで解除します。

```
# pcs resource clean
```

- pcs status** で表示される STONITH フェンシングの動作エラーを以下のコマンドで解除します。

```
# pcs resource clean
# pcs stonith history cleanup
```

3.4. GALERA クラスターの手動による復元

復元手順の一部として Galera クラスターが復元されない場合は、Galera を手動で復元する必要があります。



注記

以下の手順では、1つのコントローラーノードでいくつかのステップを実施する必要があります。手順の実施と同じコントローラーノードで、これらのステップを実施するようにしてください。

手順

1. **Controller-0** で、Galera クラスターの仮想 IP を取得します。

```
$ sudo hiera -c /etc/puppet/hiera.yaml mysql_vip
```

2. すべてのコントローラーノードで、仮想 IP を通じたデータベース接続を無効にします。

```
$ sudo iptables -I INPUT -p tcp --destination-port 3306 -d $MYSQL_VIP -j DROP
```

3. **Controller-0** で MySQL の root パスワードを取得します。

```
$ sudo hiera -c /etc/puppet/hiera.yaml mysql::server::root_password
```

4. **Controller-0** で、Galera リソースを **unmanaged** モードに設定します。

```
$ sudo pcs resource unmanage galera-bundle
```

5. すべてのコントローラーノードで、MySQL コンテナを停止します。

```
$ sudo podman container stop $(sudo podman container ls --all --format "{{.Names}}" --filter=name=galera-bundle)
```

6. すべてのコントローラーノードで、現在のディレクトリーを移動します。

```
$ sudo mv /var/lib/mysql /var/lib/mysql-save
```

7. すべてのコントローラーノードで、新規ディレクトリー **/var/lib/mysq** を作成します。

```
$ sudo mkdir /var/lib/mysql
$ sudo chown 42434:42434 /var/lib/mysql
$ sudo chcon -t container_file_t /var/lib/mysql
$ sudo chmod 0755 /var/lib/mysql
$ sudo chcon -r object_r /var/lib/mysql
$ sudo chcon -u system_u /var/lib/mysql
```

8. すべてのコントローラーノードで、MySQL コンテナを起動します。

```
$ sudo podman container start $(sudo podman container ls --all --format "{{.Names }}" --filter=name=galera-bundle)
```

9. すべてのコントローラーノードで、MySQL データベースを作成します。

```
$ sudo podman exec -i $(sudo podman container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysql_install_db --datadir=/var/lib/mysql --
  user=mysql --log_error=/var/log/mysql/mysql_init.log"
```

10. すべてのコントローラーノードで、データベースを起動します。

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysqld_safe --skip-networking --wsrep-on=OFF --
  log-error=/var/log/mysql/mysql_safe.log" &
```

11. すべてのコントローラーノードで、**.my.cnf** Galera 設定ファイルを移動します。

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mv /root/.my.cnf /root/.my.cnf.bck"
```

12. すべてのコントローラーノードで、Galera root パスワードをリセットします。

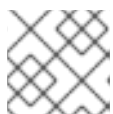
```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysql -uroot -e'use mysql;update user set
  password=PASSWORD(\"$ROOTPASSWORD\")where User='root';flush privileges;""
```

13. すべてのコントローラーノード上の Galera コンテナ内で、**.my.cnf** Galera 設定ファイルを復元します。

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mv /root/.my.cnf.bck /root/.my.cnf"
```

14. **Controller-0** で、バックアップデータベースファイルを **/var/lib/MySQL** にコピーします。

```
$ sudo cp $BACKUP_FILE /var/lib/mysql
$ sudo cp $BACKUP_GRANT_FILE /var/lib/mysql
```



注記

これらのファイルへのパスは **/home/heat-admin/** です。

15. **Controller-0** で、MySQL データベースを復元します。

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysql -u root -p$ROOT_PASSWORD <
  \"/var/lib/mysql/$BACKUP_FILE\" "
```

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysql -u root -p$ROOT_PASSWORD <
  \"/var/lib/mysql/$BACKUP_GRANT_FILE\" "
```

16. すべてのコントローラーノードで、データベースをシャットダウンします。

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
--filter=name=galera-bundle) bash -c "mysqladmin shutdown"
```

17. **Controller-0** で、ブートストラップノードを起動します。

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
filter=name=galera-bundle) \
/usr/bin/mysqld_safe --pid-file=/var/run/mysql/mysqld.pid --
socket=/var/lib/mysql/mysql.sock --datadir=/var/lib/mysql \
--log-error=/var/log/mysql/mysql_cluster.log --user=mysql --open-files-limit=16384 \
--wsrep-cluster-address=gcomm:// &
```

18. 検証: Controller-0 で、クラスターのステータスを確認します。

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
--filter=name=galera-bundle) bash -c "clustercheck"
```

「Galera cluster node is synced」のメッセージが表示されるのを確認してください。表示されない場合は、ノードを再作成する必要があります。

19. **Controller-0** で、設定からクラスターアドレスを取得します。

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
--filter=name=galera-bundle) bash -c "grep wsrep_cluster_address /etc/my.cnf.d/galera.cnf" |
awk '{print $3}'
```

20. 残りの各コントローラーノードでデータベースを起動し、クラスターを検証します。

- a. データベースを起動します。

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
--filter=name=galera-bundle) /usr/bin/mysqld_safe --pid-
file=/var/run/mysql/mysqld.pid --socket=/var/lib/mysql/mysql.sock \
--datadir=/var/lib/mysql --log-error=/var/log/mysql/mysql_cluster.log --user=mysql --
open-files-limit=16384 \
--wsrep-cluster-address=$CLUSTER_ADDRESS &
```

- b. MYSQL クラスターのステータスを確認します。

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
--filter=name=galera-bundle) bash -c "clustercheck"
```

「Galera cluster node is synced」のメッセージが表示されるのを確認してください。表示されない場合は、ノードを再作成する必要があります。

21. すべてのコントローラーノードで MySQL コンテナを停止します。

```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" \
filter=name=galera-bundle) \
/usr/bin/mysqladmin -u root shutdown
```

22. すべてのコントローラーノードで以下のファイアウォールルールを削除して、仮想 IP アドレス経由のデータベース接続を許可します。

■

```
$ sudo iptables -D INPUT -p tcp --destination-port 3306 -d $MYSQL_VIP -j DROP
```

23. すべてのコントローラーノードで MySQL コンテナを再起動します。

```
$ sudo podman container restart $(sudo podman container ls --all --format "{{ .Names }}" --filter=name=galera-bundle)
```

24. すべてのコントローラーノードで **clustercheck** コンテナを再起動します。

```
$ sudo podman container restart $(sudo podman container ls --all --format "{{ .Names }}" --filter=name=clustercheck)
```

25. **Controller-0** で、Galera リソースを **managed** モードに設定します。

```
$ sudo pcs resource manage galera-bundle
```

検証

1. サービスが正常に実行されていることを確認するには、pacemaker のステータスを確認します。

```
$ sudo pcs status
```

2. オーバークラウドのステータスを確認するには、OpenStack Integration Test Suite (tempest) を使用します。詳細は、[Validating your OpenStack cloud with the Integration Test Suite \(tempest\)](#) を参照してください。
3. 特定のノードで問題が疑われる場合は、**clustercheck** でクラスターの状態を確認します。

```
$ sudo podman exec clustercheck /usr/bin/clustercheck
```

3.5. アンダークラウドのノードデータベースを手動で復元する

アンダークラウドのデータベースがアンダークラウドの復元プロセスの一部として復元されない場合は、データベースを手動で復元することができます。データベースを復元できるのは、以前にスタンドアロンのデータベースバックアップを作成した場合のみです。

前提条件

- アンダークラウドデータベースのスタンドアロンバックアップを作成しています。詳細は、「[アンダークラウドノードのスタンドアロンデータベースバックアップの作成](#)」を参照してください。

手順

1. director アンダークラウドノードに **root** ユーザーとしてログインします。
2. すべての tripleo サービスを停止します。

```
[root@director ~]# systemctl stop tripleo_*
```

3. 以下のコマンドを入力して、サーバーでコンテナが実行されていないことを確認します。

```
[root@director ~]# podman ps
```

コンテナが実行している場合は、以下のコマンドを実行してコンテナを停止します。

```
[root@director ~]# podman stop <container_name>
```

- 現在の `/var/lib/mysql` ディレクトリーのバックアップを作成してから、そのディレクトリーを削除します。

```
[root@director ~]# cp -a /var/lib/mysql /var/lib/mysql_bck
[root@director ~]# rm -rf /var/lib/mysql
```

- データベースディレクトリーを再作成し、新しいディレクトリーに SELinux の属性を設定します。

```
[root@director ~]# mkdir /var/lib/mysql
[root@director ~]# chown 42434:42434 /var/lib/mysql
[root@director ~]# chmod 0755 /var/lib/mysql
[root@director ~]# chcon -t container_file_t /var/lib/mysql
[root@director ~]# chcon -r object_r /var/lib/mysql
[root@director ~]# chcon -u system_u /var/lib/mysql
```

- mariadb** イメージのローカルタグを作成します。 `<image_id>` および `<undercloud.ctlplane.example.com>` を、ご自身の環境に該当する値に置き換えます。

```
[root@director ~]# podman images | grep mariadb
<undercloud.ctlplane.example.com>:8787/rh-osbs/rhosp16-openstack-mariadb
16.2_20210322.1 <image_id> 3 weeks ago 718 MB
```

```
[root@director ~]# podman tag <image_id> mariadb
```

```
[root@director ~]# podman images | grep maria
localhost/mariadb latest <image_id> 3
weeks ago 718 MB
<undercloud.ctlplane.example.com>:8787/rh-osbs/rhosp16-openstack-mariadb
16.2_20210322.1 <image_id> 3 weeks ago 718 MB
```

- `/var/lib/mysql` ディレクトリーをコンテナで初期化します。

```
[root@director ~]# podman run --net=host -v /var/lib/mysql:/var/lib/mysql localhost/mariadb
mysql_install_db --datadir=/var/lib/mysql --user=mysql
```

- データベースにインポートするデータベースのバックアップファイルをコピーします。

```
[root@director ~]# cp /home/stack/openstack-backup-mysql.sql /var/lib/mysql
```

- データベースサービスを起動してデータをインポートします。

```
[root@director ~]# podman run --net=host -dt -v /var/lib/mysql:/var/lib/mysql
localhost/mariadb /usr/libexec/mysqld
```

- データを読み込み、`max_allowed_packet` パラメーターを設定します。

- a. コンテナにログインし、設定を行います。

```
[root@director ~]# podman exec -it <container_id> /bin/bash
()[mysql@5a4e429c6f40 /]$ mysql -u root -e "set global max_allowed_packet =
1073741824;"
- ()[mysql@5a4e429c6f40 /]$ mysql -u root < /var/lib/mysql/openstack-backup-mysql.sql
- ()[mysql@5a4e429c6f40 /]$ mysql -u root < /var/lib/mysql/openstack-backup-mysql-
grants.sql
()[mysql@5a4e429c6f40 /]$ mysql -u root -e 'flush privileges'
()[mysql@5a4e429c6f40 /]$ exit
exit
```



注記

インポート後、GRANT **USAGE ON *.* TO 'mysql@localhost'** で示されているように、ユーザーテーブルに **mysql@localhost** の特権が含まれていないため、ERROR 1133 が発生する場合があります。

このエラーは無視してかまいません。このエラーは、既存のパーミッションには影響しません。

- b. コンテナを停止します。

```
[root@director ~]# podman stop <container_id>
```

- c. コンテナが動いていないことを確認します。

```
[root@director ~]# podman ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[root@director ~]#
```

11. すべての tripleo サービスを再起動します。

```
[root@director ~]# systemctl start multi-user.target
```