



Red Hat OpenStack Platform 17.1

インスタンスの高可用性の設定

コンピュータインスタンスの高可用性の設定

Red Hat OpenStack Platform 17.1 インスタンスの高可用性の設定

コンピュータインスタンスの高可用性の設定

OpenStack Team
rhos-docs@redhat.com

法律上の通知

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

このガイドでは、インスタンスの高可用性 (インスタンス HA) を管理する方法を説明します。インスタンス HA が設定された Red Hat OpenStack Platform (RHOSP) では、コンピューターノードに障害が発生した場合に、インスタンスを自動的に退避させ別のコンピューターノードに作成し直すことができます。

目次

多様性を受け入れるオープンソースの強化	3
RED HAT ドキュメントへのフィードバック (英語のみ)	4
第1章 インスタンス HA デプロイメントの導入および計画	5
1.1. インスタンス HA の仕組み	5
1.2. インスタンス HA デプロイメントのプランニング	5
1.3. インスタンス HA リソースエージェント	6
第2章 インスタンス HA のインストールと設定	8
2.1. インスタンス HA のロールおよびプロファイルの設定	8
2.2. インスタンス HA が設定されたオーバークラウドでのフェンシングの有効化	9
2.3. インスタンス HA が設定されたオーバークラウドのデプロイ	10
2.4. インスタンス HA による退避のテスト	11
2.5. インスタンス HA で退避させるインスタンスの指定	12
2.6. 関連情報	12
第3章 インスタンス HA が設定されたアンダークラウドおよびオーバークラウドでのメンテナンスの実行	13
3.1. 前提条件	13
3.2. アンダークラウドおよびオーバークラウドのシャットダウン順序	13
3.3. システムメンテナンスの実施	17
3.4. アンダークラウドおよびオーバークラウドの起動順序	17
第4章 インスタンス HA が設定されたコンピュートノードおよびコントローラーノードでのメンテナンスの実行 ..	23

多様性を受け入れるオープンソースの強化

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

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

ドキュメントに関するフィードバックを提供するには、[Create Issue](#) フォームを使用します。Red Hat OpenStack Platform Jira プロジェクトで Jira Issue が作成され、フィードバックの進行状況を追跡できます。

1. Jira にログインしていることを確認してください。Jira アカウントをお持ちでない場合は、アカウントを作成してフィードバックを送信してください。
2. [Create Issue](#) をクリックして、**Create Issue** ページを開きます。
3. **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。
4. **Create** をクリックします。

第1章 インスタンス HA デプロイメントの導入および計画

コンピュータインスタンスの高可用性 (インスタンス HA) は、障害が発生したコンピュータノードからインスタンスを退避して、別のコンピュータノードにインスタンスを再作成するために使用できるツールです。

インスタンス HA は、共有ストレージまたはローカルストレージ環境で機能します。したがって、退避させたインスタンスは同じネットワーク設定 (静的 IP アドレス、Floating IP アドレス等) を維持します。再作成されたインスタンスも、新しいコンピュータノード内で同じ特性を維持します。

1.1. インスタンス HA の仕組み

コンピュータノードに障害が発生すると、オーバークラウドのフェンシングエージェントはノードをフェンシングします。続いてインスタンス HA エージェントは、障害が発生したコンピュータノードから別のコンピュータノードにインスタンスを退避させます。

コンピュータノードに障害が発生しインスタンス HA がトリガーされると、以下のイベントが発生します。

1. 障害が発生すると、**IPMI** エージェントは最初のレイヤーのフェンシングを実行します。これには、ノードがシャットダウンされるように物理的にリセットする操作や、データの破損やオーバークラウド上に複数の同一インスタンスが存在するのを回避する操作が含まれます。ノードがオフラインである場合、フェンシング済みとみなされます。
2. IPMI による物理的なフェンシングの後に以下のコマンドを実行すると、**fence-nova** エージェントが第二レイヤーのフェンシングを自動的に実施し、フェンシング済みのノードを属性 "**evacuate=yes**" でマーキングします (この属性は、ノードごとに設定する必要があります)。

```
$ attrd_updater -n evacuate -A name="evacuate" host="FAILEDHOST" value="yes"
```

FAILEDHOST は障害が発生したコンピュータノードの名前です。

3. **nova-evacuate** エージェントは常にバックグラウンドで動作し、クラスターに "**evacuate=yes**" 属性のノードがないか定期的に確認します。この属性がフェンシング済みノードに含まれることを **nova-evacuate** が検出すると、エージェントはノードからの全インスタンスの退避を開始します。退避プロセスは、いつでも実行できる手動インスタンス退避プロセスと似ています。
4. IPMI によるリセット後に障害が発生したノードが再起動すると、そのノードの **nova-compute** プロセスも自動的に開始されます。ノードはそれまでフェンシングされていたので、Pacemaker がノードのフェンシングを解除するまで、新しいインスタンスを実行しません。
5. コンピュータノードがオンライン状態にあることを Pacemaker が検出すると、ノードで **compute-unfence-trigger** リソースエージェントが起動します。これにより、ノードが解放され、インスタンスが再び実行できるようになります。

関連情報

- [インスタンスの退避](#)

1.2. インスタンス HA デプロイメントのプランニング

インスタンス HA をデプロイする前に、適合性のためにリソース名を確認し、環境に応じてストレージおよびネットワークを設定します。

- コンピュートノードのホスト名および Pacemaker リモートリソース名は、W3C 命名規則に従う必要があります。詳細は、W3C ドキュメントの [namespace の宣言](#) および [名前とトークン](#) を参照してください。
- 一般的に、インスタンス HA ではインスタンスのディスクイメージ用に共有ストレージを設定する必要があります。したがって、**no-shared-storage** オプションの使用を試みると、退避中に **InvalidSharedStorage** エラーが表示され、インスタンスが別のコンピュートノードで起動しない場合があります。
ただし、すべてのインスタンスが OpenStack Block Storage (**cinder**) ボリュームから起動するように設定されている場合には、インスタンスのディスクイメージ用に共有ストレージを設定する必要はないので、**no-shared-storage** オプションを使用してすべてのインスタンスを退避させることができます。

インスタンスが Block Storage ボリュームから起動するように設定されている場合には、退避させたインスタンスは別のコンピュートノード上の同じボリュームからブートします。したがって、OS イメージおよびアプリケーションデータが OpenStack Block Storage ボリュームに保管されているので、退避させたインスタンスは直ちにジョブを再開します。
- インスタンス HA をスパイン/リーフ環境でデプロイする場合には、コントローラーノードおよびコンピュートノードに単一の **internal_api** ネットワークを定義する必要があります。その後、各リーフのサブネットを定義できます。スパイン/リーフ型ネットワークの設定の詳細は、[スパイン/リーフ型ネットワーク ガイドの ロールデータファイルの作成](#) を参照してください。
- Red Hat OpenStack Platform 13 以降では、オーバークラウドのアップグレードの一環として、director を使用してインスタンス HA をアップグレードします。オーバークラウドのアップグレードの詳細は、[Red Hat OpenStack Platform のマイナー更新の実行](#) を参照してください。
- vTPM デバイスを搭載したインスタンスを退避することはできません。vTPM デバイスを使用してインスタンスをデプロイする場合は、退避する必要がある他のインスタンスがフレーバーを使用するか、**evacuatable** 属性でタグ付けしたイメージを使用するようにしてください。退避するインスタンスの指定の詳細は、[インスタンス HA を使用して退避するインスタンスの指定](#) を参照してください。
- インストール後の director でのインスタンス HA の無効化はサポートされていません。デプロイメントからインスタンス HA コンポーネントを手動で削除する回避策は、[How can I remove Instance HA components from the controller nodes?](#) を参照してください。



重要

この回避策は、実稼働環境用には検証されていません。実稼働環境で実装する前に、テスト環境で手順を検証する必要があります。

1.3. インスタンス HA リソースエージェント

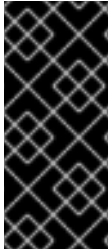
インスタンス HA は **fence_compute**、**NovaEvacuate**、および **comput-undefence-trigger** リソースエージェントを使用して、コンピュートノードに障害が発生した場合にインスタンスの退避および再作成を行います。

エージェント名	クラスター内での名前	ロール
fence_compute	fence-nova	コンピュートノードが使用不能になった場合に、退避のためにそのノードをマーキングします。

エージェント名	クラスター内での名前	ロール
NovaEvacuate	nova-evacuate	障害が発生したノードからインスタンスを退避させます。このエージェントは、コントローラーノードのいずれかで動作します。
Dummy	compute-ufence-trigger	フェンシングされたノードを解放し、再びインスタンスを実行できるようにします。

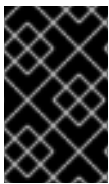
第2章 インスタンス HA のインストールと設定

インスタンスの高可用性 (HA) をデプロイするには、Red Hat OpenStack Platform (RHOSP) director を使用します。ただし、新規オーバークラウドで新たなインスタンス HA デプロイメントを設定するには、追加のステップを実施する必要があります。ステップを完了すると、インスタンス HA はカスタムロールを持つコンピューターノードのサブセット上で実行されます。



重要

インスタンス HA は、RHOSP ハイパーコンバージドインフラストラクチャー (HCI) 環境ではサポートされません。RHOSP HCI 環境でインスタンス HA を使用するには、ComputeInstanceHA ロールを持つコンピューターノードのサブセットで、インスタンス HA を使用するように指定する必要があります。Red Hat Ceph Storage サービスは、インスタンス HA をホストするコンピューティングノード上でホストしないでください。



重要

標準のロールまたはカスタムロールを使用する既存のオーバークラウド等、別の環境でインスタンス HA を有効にするには、デプロイメントに該当する手順のみを実施し、テンプレートを適切に変更します。

2.1. インスタンス HA のロールおよびプロファイルの設定

インスタンス HA をデプロイする前に、インスタンス HA ロールを **roles-data.yaml** ファイルに追加し、インスタンス HA プロファイルを使用してインスタンス HA で管理する各 Compute ノードにタグを付け、これらを **overcloud-baremetal-deploy.yaml** ファイルまたは同等のものに追加します。特定のロール用にオーバークラウドノードを指定する方法については、[プロファイルを一致させることによるオーバークラウドノードのロールの指定](#) を参照してください。例として、**computeiha** プロファイルを使用してノードを設定できます。

手順

1. 登録された各ノードの既存の機能を確認します。

```
(undercloud)$ openstack baremetal node show <node> -f json -c properties | jq -r
.properties.capabilities
```

2. ノードの既存の機能に **profile:computeiha** を追加して、ロールプロファイルに一致させる各ベアメタルノードにプロファイル機能を割り当てます。

```
(undercloud)$ openstack baremetal node set <node> --property
capabilities="profile:computeiha,<capability_1>,...,<capability_n>"
```

- <node> をベアメタルノードの ID に置き換えます。
 - <capability_1> および <capability_n> までのすべての機能は、手順1で取得した各機能に置き換えます。
3. まだ定義されていない場合は、**overcloud-baremetal-deploy.yaml** ファイルにロールを追加します。
 4. **overcloud-baremetal-deploy.yaml** を編集して、ロールのノードに割り当てるプロファイルを定義します。

```
- name: ComputeInstanceHA
  count: 2
  hostname_format: compute-%index%
  defaults:
    profile: computeiha
    network_config:
      template: /home/stack/composable_roles/network/nic-configs/compute.j2
  networks:
    - network: ctlplane
      vif: true
    - network: internal_api
    - network: tenant
    - network: storage
```

5. オーバークラウドノードをプロビジョニングします。

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

- <stack> を、ベアメタルノードをプロビジョニングしたスタックの名前に置き換えます。デフォルト値は **overcloud** です。
- <deployment_file> を、生成された heat 環境ファイルに選択した名前に置き換えて、デプロイコマンドに含めます (例: **:/home/stack/templates/overcloud-baremetal-deployed.yaml**)。

2.2. インスタンス HA が設定されたオーバークラウドでのフェンシングの有効化

フェンシング情報を定義した環境ファイルを作成して、オーバークラウドの全コントローラーノードおよびコンピュートノードのフェンシングを有効にします。

手順

1. **~/templates** などのアクセス可能な場所に環境ファイルを作成し、以下の内容を追加します。

```
parameter_defaults:
  EnableFencing: true
  FencingConfig:
    devices:
      - agent: fence_ipmilan
        host_mac: 00:ec:ad:cb:3c:c7
        params:
          login: admin
          ipaddr: 192.168.24.1
          ipport: 6230
          passwd: password
          lanplus: 1
      - agent: fence_ipmilan
        host_mac: 00:ec:ad:cb:3c:cb
        params:
          login: admin
```

```

    ipaddr: 192.168.24.1
    ipport: 6231
    passwd: password
    lanplus: 1
  - agent: fence_ipmilan
    host_mac: 00:ec:ad:cb:3c:cf
    params:
      login: admin
      ipaddr: 192.168.24.1
      ipport: 6232
      passwd: password
      lanplus: 1
  - agent: fence_ipmilan
    host_mac: 00:ec:ad:cb:3c:d3
    params:
      login: admin
      ipaddr: 192.168.24.1
      ipport: 6233
      passwd: password
      lanplus: 1
  - agent: fence_ipmilan
    host_mac: 00:ec:ad:cb:3c:d7
    params:
      login: admin
      ipaddr: 192.168.24.1
      ipport: 6234
      passwd: password
      lanplus: 1

```

2. コンピュートインスタンスに共有ストレージを使用しない場合は、作成した環境ファイルに以下のパラメーターを追加します。

```

parameter_defaults:
  ExtraConfig:
    tripleo::instanceha::no_shared_storage: true

```

関連情報

- [「インスタンス HA デプロイメントのプランニング」](#)
- [STONITH を使用したコントローラーノードのフェンシング](#)

2.3. インスタンス HA が設定されたオーバークラウドのデプロイ

オーバークラウドをすでにデプロイしている場合は、作成した追加のインスタンス HA ファイルを使用して **openstack overcloud deploy** コマンドを再度実行できます。アンダーグラウンドの作成後は、いつでもオーバークラウドにインスタンス HA を設定することができます。

前提条件

- インスタンス HA ロールおよびプロファイルを設定している。
- オーバークラウドでフェンシングを有効にしている。

手順

- **openstack overcloud deploy** コマンドで **-e** オプションを指定して、**compute-instanceha.yaml** 環境ファイルと追加の環境ファイルを含めます。

```
$ openstack overcloud deploy --templates \
-e <fencing_environment_file> \
-r my_roles_data.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/compute-instanceha.yaml
```

<fencing_environment_file> を環境に適したファイル名に置き換えます。



注記

- **compute-instanceha.yaml** 環境ファイルは変更しないでください。
- オーバークラウドデプロイメントに追加する各環境ファイルへのフルパスを含めます。

デプロイ後、各コンピュータノードには、**STONITH** デバイスと、**pacemaker_remote** サービスが含まれます。

2.4. インスタンス HA による退避のテスト

インスタンス HA がインスタンスを正常に退避することをテストするには、コンピュータノードで退避をトリガーし、インスタンス HA エージェントが正常にインスタンスを退避して別のコンピュータノードに再作成することを確認します。



警告

以下の手順では、コンピュータノードを故意にクラッシュさせます。これにより、インスタンス HA によるインスタンスの自動退避がトリガーされます。

前提条件

- インスタンス HA がコンピュータノードにデプロイされている。

手順

1. オーバークラウドで1つ以上のインスタンスを起動します。

```
stack@director $ . overcloudrc
stack@director $ openstack server create --image cirros --flavor 2 test-failover
stack@director $ openstack server list -c Name -c Status
```

2. インスタンスをホストするコンピュータノードにログインし、**root** ユーザーに変更します。**compute-n** をコンピュータノードの名前に置き換えます。

```
stack@director $ . stackrc
stack@director $ ssh -l tripleo-admin compute-n
tripleo-admin@compute-n $ su -
```

3. コンピュートノードをクラッシュさせます。

```
root@compute-n $ echo c > /proc/sysrq-trigger
```

4. ノードが再起動するまで数分待ってから、クラッシュしたコンピュートノードのインスタンスが別のコンピュートノードで再作成されていることを確認します。

```
stack@director $ openstack server list -c Name -c Status
stack@director $ openstack compute service list
```

2.5. インスタンス HA で退避させるインスタンスの指定

デフォルトでは、インスタンス HA は障害が発生しているノードからすべてのインスタンスを退避させます。インスタンス HA を設定して、特定のイメージまたはフレーバーを持つインスタンスのみを退避させることができます。

前提条件

- インスタンス HA がオーバークラウドにデプロイされている。

手順

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

```
$ source ~/overcloudrc
```

3. 以下のオプションのいずれかを使用します。

- イメージをタグ付けする。

```
(overcloud) $ openstack image set --tag evacuable <image_id>
```

<image_id> を退避させるイメージの ID に置き換えます。

- フレーバーをタグ付けする。

```
(overcloud) $ openstack flavor set --property evacuable=true <flavor_id>
```

<flavor_id> を退避させるフレーバーの ID に置き換えてください。

2.6. 関連情報

- [director を使用した Red Hat OpenStack Platform のインストールと管理](#)
- [コンポーザブルサービスとカスタムロール](#)

第3章 インスタンス HA が設定されたアンダークラウドおよびオーバークラウドでのメンテナンスの実行

アンダークラウドおよびオーバークラウドでメンテナンスを実施するには、オーバークラウド起動時の問題を最小限に抑えるために、アンダークラウドおよびオーバークラウドノードを特定の順序でシャットダウンして起動する必要があります。また、ノードを停止してノード上の Pacemaker リソースを無効にし、特定のコンピュートノードまたはコントローラーノードでメンテナンスを実行することもできます。

3.1. 前提条件

- インスタンス HA が有効な動作中のアンダークラウドおよびオーバークラウド

3.2. アンダークラウドおよびオーバークラウドのシャットダウン順序

Red Hat OpenStack Platform 環境をシャットダウンするには、オーバークラウドおよびアンダークラウドを以下の順序でシャットダウンする必要があります。

1. オーバークラウド Compute ノード上のインスタンスをシャットダウンします。
2. Compute ノードをシャットダウンします。
3. Controller ノードの高可用性サービスおよび OpenStack Platform のサービスをすべて停止します。
4. Ceph Storage ノードをシャットダウンします。
5. Controller ノードをシャットダウンします。
6. アンダークラウドをシャットダウンします。

3.2.1. オーバークラウド Compute ノード上のインスタンスのシャットダウン

Red Hat OpenStack Platform 環境のシャットダウンのサブタスクとして、Compute ノードをシャットダウンする前に Compute ノード上のインスタンスをすべてシャットダウンします。

前提条件

- Compute サービスがアクティブなオーバークラウド

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 ~/overcloudrc
```

3. オーバークラウドで実行中のインスタンスを表示します。

```
$ openstack server list --all-projects
```

4. オーバークラウドのそれぞれのインスタンスを停止します。

```
$ openstack server stop <INSTANCE>
```

オーバークラウド内のすべてのインスタンスを停止するまで、それぞれのインスタンスでこのステップを繰り返します。

3.2.2. オーバークラウドコンピュートノードのインスタンス HA サービスの停止

Red Hat OpenStack Platform 環境のシャットダウンのサブタスクとして、インスタンスを停止してコンピュートノードをシャットダウンする前に、コンピュートノードのインスタンス HA サービスをすべてシャットダウンする必要があります。

前提条件

- Compute サービスがアクティブなオーバークラウド
- コンピュートノードで有効化されているインスタンス HA

手順

1. Pacemaker を実行するオーバ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. それぞれのコンピュートノードの Pacemaker リモートリソースを無効にします。
 - a. コンピュートノードの Pacemaker リモートリソースを特定します。

```
# pcs resource status
```

これらのリソースは **ocf::pacemaker:remote** エージェントを使用し、通常はコンピュートノードのホストの形式に基づく名前が付けられます (例: **overcloud-novacomputeiha-0**)。

- b. それぞれの Pacemaker リモートリソースを無効にします。 **overcloud-novacomputeiha-0** のリソースを無効にする方法を、以下の例に示します。

```
# pcs resource disable overcloud-novacomputeiha-0
```

3. コンピュートノードの STONITH デバイスを無効にします。
 - a. コンピュートノードの STONITH デバイスを特定します。

```
# pcs stonith status
```

- b. それぞれのコンピュートノードの STONITH デバイスを無効にします。

```
# pcs stonith disable <STONITH_DEVICE>
```

3.2.3. コンピュートノードのシャットダウン

Red Hat OpenStack Platform 環境をシャットダウンする際のサブタスクとして、それぞれの Compute ノードにログインしてシャットダウンします。

前提条件

- Compute ノード上のすべてのインスタンスがシャットダウンされている。

手順

1. Compute ノードに **root** ユーザーとしてログインします。
2. ノードをシャットダウンします。

```
# shutdown -h now
```

3. すべての Compute ノードをシャットダウンするまで、それぞれの Compute ノードでこの手順を実施します。

3.2.4. Controller ノードのサービスの停止

Red Hat OpenStack Platform 環境のシャットダウンする際のサブタスクとして、Controller ノードをシャットダウンする前にノードのサービスを停止します。これには Pacemaker サービスおよび systemd サービスが含まれます。

前提条件

- Pacemaker サービスがアクティブなオーバークラウド

手順

1. Controller ノードに **root** ユーザーとしてログインします。
2. Pacemaker クラスタを停止します。

```
# pcs cluster stop --all
```

このコマンドにより、すべてのノード上のクラスタが停止します。

3. Pacemaker サービスが停止するまで待ち、サービスが停止したことを確認します。
 - a. Pacemaker のステータスを確認します。

```
# pcs status
```

- b. Pacemaker サービスが実行されていないことを Podman で確認します。

```
# podman ps --filter "name=.*-bundle.*"
```

4. Red Hat OpenStack Platform のサービスを停止します。

```
# systemctl stop 'tripleo_*
```

5. サービスが停止するまで待ち、サービスが実行されなくなったことを Podman で確認します。

```
# podman ps
```

3.2.5. Ceph Storage ノードのシャットダウン

Red Hat OpenStack Platform 環境のシャットダウンのサブタスクとして、Ceph Storage サービスを無効にし、続いてそれぞれの Ceph Storage ノードにログインしてシャットダウンします。

前提条件

- 正常な Ceph Storage クラスタ。
- Ceph MON サービスがスタンドアロンの Ceph MON ノードまたは Controller ノードで動作している。

手順

1. Ceph MON サービスを実行するノード (例: Controller ノードまたはスタンドアロンの Ceph MON ノード) に **root** ユーザーとしてログインします。
2. クラスタが正常であることを確認します。以下の例の **podman** コマンドにより、Controller ノード上の Ceph MON コンテナ内でステータス確認が実施されます。

```
# sudo podman exec -it ceph-mon-controller-0 ceph status
```

ステータスが **HEALTH_OK** であることを確認します。

3. クラスタの **noout**、**norecover**、**norebalance**、**nobackfill**、**nodown**、および **pause** フラグを設定します。以下の例の **podman** コマンドにより、Controller ノード上の Ceph MON コンテナを通じてこれらのフラグが設定されます。

```
# sudo podman exec -it ceph-mon-controller-0 ceph osd set noout
# sudo podman exec -it ceph-mon-controller-0 ceph osd set norecover
# sudo podman exec -it ceph-mon-controller-0 ceph osd set norebalance
# sudo podman exec -it ceph-mon-controller-0 ceph osd set nobackfill
# sudo podman exec -it ceph-mon-controller-0 ceph osd set nodown
# sudo podman exec -it ceph-mon-controller-0 ceph osd set pause
```

4. それぞれの Ceph Storage ノードをシャットダウンします。
 - a. Ceph Storage ノードに **root** ユーザーとしてログインします。
 - b. ノードをシャットダウンします。

```
# shutdown -h now
```

- c. すべての Ceph Storage ノードをシャットダウンするまで、それぞれの Ceph Storage ノードでこの手順を実施します。

5. スタンドアロンの Ceph MON ノードをすべてシャットダウンします。

- a. スタンドアロンの Ceph MON ノードに **root** ユーザーとしてログインします。
- b. ノードをシャットダウンします。

```
# shutdown -h now
```

- c. スタンドアロンの Ceph MON ノードをすべてシャットダウンするまで、それぞれのスタンドアロンの Ceph MON ノードでこの手順を実施します。

関連情報

- [What is the procedure to shutdown and bring up the entire ceph cluster?](#)

3.2.6. Controller ノードのシャットダウン

Red Hat OpenStack Platform 環境のシャットダウンのサブタスクとして、それぞれの Controller ノードにログインしてシャットダウンします。

前提条件

- Pacemaker クラスタが停止している。
- Controller ノードのすべての Red Hat OpenStack Platform サービスが停止している。

手順

1. Controller ノードに **root** ユーザーとしてログインします。
2. ノードをシャットダウンします。

```
# shutdown -h now
```

3. すべての Controller ノードをシャットダウンするまで、それぞれの Controller ノードでこの手順を実施します。

3.2.7. アンダークラウドのシャットダウン

Red Hat OpenStack Platform 環境のシャットダウンのサブタスクとして、アンダークラウドノードにログインしてアンダーグラウンドをシャットダウンします。

前提条件

- 動作中のアンダークラウド

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. アンダークラウドをシャットダウンします。

```
$ sudo shutdown -h now
```

3.3. システムメンテナンスの実施

アンダークラウドおよびオーバークラウドを完全にシャットダウンしたら、環境内のシステムに対するメンテナンスを実施し、続いてアンダークラウドおよびオーバークラウドを起動します。

3.4. アンダークラウドおよびオーバークラウドの起動順序

Red Hat OpenStack Platform 環境を起動するには、アンダークラウドおよびオーバークラウドを以下の順序で起動する必要があります。

1. アンダークラウドを起動します。
2. Controller ノードを起動します。

3. Ceph Storage ノードを起動します。
4. Compute ノードを起動します。
5. オーバークラウドの Compute ノードでインスタンスを起動します。

3.4.1. アンダークラウドの起動

Red Hat OpenStack Platform 環境の起動のサブタスクとして、アンダークラウドノードの電源をオンにし、アンダークラウドにログインし、アンダークラウドのサービスを確認します。

前提条件

- アンダークラウドの電源がオフになっている。

手順

- アンダークラウドの電源をオンにし、アンダークラウドがブートするまで待ちます。

検証

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. アンダークラウドのサービスを確認します。

```
$ systemctl list-units 'tripleo_*
```

4. **tripleo-ansible-inventory.yaml** という名前の静的インベントリーファイルを検証します。

```
$ validation run --group pre-introspection -i <inventory_file>
```

- **<inventory_file>** ファイルを Ansible インベントリーファイルの名前および場所に置き換えます (例: **~/tripleo-deploy/undercloud/tripleo-ansible-inventory.yaml**)。



注記

検証を実行すると、出力の **Reasons** 列は 79 文字に制限されます。検証結果を完全に表示するには、検証ログファイルを表示します。

5. すべてのサービスとコンテナがアクティブおよび正常であることを確認します。

```
$ validation run --validation service-status --limit undercloud -i <inventory_file>
```

関連情報

- [検証フレームワークの使用](#)

3.4.2. Controller ノードの起動

Red Hat OpenStack Platform 環境の起動のサブタスクとして、それぞれの Controller ノードの電源をオンにし、そのノードの Pacemaker 以外のサービスを確認します。

前提条件

- Controller ノードの電源がオフになっている。

手順

- それぞれの Controller ノードの電源をオンにします。

検証

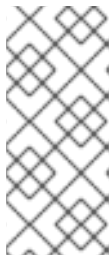
1. **root** ユーザーとして各 Controller ノードにログインします。
2. Controller ノードのサービスを確認します。

```
$ systemctl -t service
```

Pacemaker ベース以外のサービスだけが動作中です。

3. Pacemaker サービスが起動するまで待ち、サービスが起動したことを確認します。

```
$ pcs status
```



注記

環境でインスタンス HA を使用している場合、Pacemaker リソースは、Compute ノードを起動するか、**pcs stonith confirm <compute_node>** コマンドを使用して手動でフェンスを解除する操作を実行するまで起動しません。このコマンドは、インスタンス HA を使用する各 Compute ノードで実行する必要があります。

3.4.3. Ceph Storage ノードの起動

Red Hat OpenStack Platform 環境の起動のサブタスクとして、Ceph MON ノードおよび Ceph Storage ノードの電源をオンにし、Ceph Storage サービスを有効にします。

前提条件

- 電源がオフの Ceph Storage クラスタ。
- 電源がオフのスタンドアロンの Ceph MON ノードまたは電源がオンの Controller ノードで、Ceph MON サービスが有効になっている。

手順

1. 使用している環境にスタンドアロンの Ceph MON ノードがある場合、それぞれの Ceph MON ノードの電源をオンにします。
2. それぞれの Ceph Storage ノードの電源をオンにします。
3. Ceph MON サービスを実行するノード (例: Controller ノードまたはスタンドアロンの Ceph MON ノード) に **root** ユーザーとしてログインします。

4. クラスターノードのステータスを確認します。以下の例の **podman** コマンドにより、Controller ノード上の Ceph MON コンテナ内でステータス確認が実施されます。

```
# sudo podman exec -it ceph-mon-controller-0 ceph status
```

それぞれのノードの電源がオンで、接続された状態であることを確認します。

5. クラスターの **noout**、**norecover**、**norebalance**、**nobackfill**、**nodown**、および **pause** フラグの設定を解除します。以下の例の **podman** コマンドにより、Controller ノード上の Ceph MON コンテナを通じてこれらのフラグの設定が解除されます。

```
# sudo podman exec -it ceph-mon-controller-0 ceph osd unset noout
# sudo podman exec -it ceph-mon-controller-0 ceph osd unset norecover
# sudo podman exec -it ceph-mon-controller-0 ceph osd unset norebalance
# sudo podman exec -it ceph-mon-controller-0 ceph osd unset nobackfill
# sudo podman exec -it ceph-mon-controller-0 ceph osd unset nodown
# sudo podman exec -it ceph-mon-controller-0 ceph osd unset pause
```

検証

1. クラスターが正常であることを確認します。以下の例の **podman** コマンドにより、Controller ノード上の Ceph MON コンテナ内でステータス確認が実施されます。

```
# sudo podman exec -it ceph-mon-controller-0 ceph status
```

ステータスが **HEALTH_OK** であることを確認します。

関連情報

- [What is the procedure to shutdown and bring up the entire ceph cluster?](#)

3.4.4. Compute ノードの起動

Red Hat OpenStack Platform 環境の起動のサブタスクとして、それぞれの Compute ノードの電源をオンにし、そのノードのサービスを確認します。

前提条件

- 電源がオフの Compute ノード

手順

1. それぞれの Compute ノードの電源をオンにします。

検証

1. 各 Compute に **root** ユーザーとしてログインします。
2. Compute ノードのサービスを確認します。

```
$ systemctl -t service
```

3.4.5. オーバークラウドコンピュートノードのインスタンス HA サービスの起動

Red Hat OpenStack Platform 環境の起動のサブタスクとして、コンピュータノードのインスタンス HA サービスをすべて起動します。

前提条件

- コンピュートノードが動作中のオーバークラウド
- コンピュートノードで有効化されているインスタンス HA

手順

1. Pacemaker を実行するオーバ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. コンピュートノードの STONITH デバイスを有効にします。

- a. コンピュートノードの STONITH デバイスを特定します。

```
# pcs stonith status
```

- b. コンピュートノードの STONITH エラーをすべて消去します。

```
# pcs stonith confirm <COMPUTE_NODE>
```

このコマンドにより、ノードがクリーンな STONITH の状態に戻ります。

- c. コンピュートノードの STONITH デバイスを有効にします。

```
# pcs stonith enable <STONITH_DEVICE>
```

- d. STONITH を使用するそれぞれのコンピュータノードで、この手順を実施します。

3. それぞれのコンピュータノードの Pacemaker リモートリソースを有効にします。

- a. コンピュートノードの Pacemaker リモートリソースを特定します。

```
# pcs resource status
```

これらのリソースは **ocf::pacemaker:remote** エージェントを使用し、通常はコンピュータノードのホストの形式に基づく名前が付けられます (例: **overcloud-novacomputeiha-0**)。

- b. それぞれの Pacemaker リモートリソースを有効にします。 **overcloud-novacomputeiha-0** のリソースを有効にする方法を、以下の例に示します。

```
# pcs resource enable overcloud-novacomputeiha-0
```

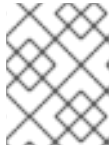
- c. Pacemaker リモート管理を行うそれぞれのコンピュータノードで、この手順を実施します。

4. Pacemaker サービスが起動するまで待ち、サービスが起動したことを確認します。

```
# pcs status
```

5. 起動プロセス中に Pacemaker リソースの起動に失敗した場合は、リソースのステータスと失敗回数をリセットします。

```
# pcs resource cleanup
```



注記

fence_compute や **fence_kdump** など、一部のサービスは開始に時間がかかる場合があります。

3.4.6. オーバークラウド Compute ノード上のインスタンスの起動

Red Hat OpenStack Platform 環境の起動のサブタスクとして、Compute ノード上のインスタンスを起動します。

前提条件

- アクティブなノードを持つアクティブなオーバークラウド

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 ~/overcloudrc
```

3. オーバークラウドで実行中のインスタンスを表示します。

```
$ openstack server list --all-projects
```

4. オーバークラウド内のインスタンスを起動します。

```
$ openstack server start <INSTANCE>
```

第4章 インスタンス HA が設定されたコンピュータードおよびコントローラーノードでのメンテナンスの実行

インスタンス HA が設定されたコンピュータードまたはコントローラーノードでメンテナンスを実行するには、そのノードを **standby** モードに設定し、ノード上の Pacemaker リソースを無効にしてノードを停止します。メンテナンス作業が完了したら、ノードを起動して、Pacemaker リソースが正常であることを確認します。

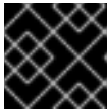
前提条件

- インスタンス HA が有効な動作中のオーバークラウド

手順

1. コントローラーノードにログインし、コンピュータードまたはコントローラーノードを停止します。

```
# pcs node standby <node UUID>
```



重要

停止するノードとは異なるノードにログインする必要があります。

2. ノード上の Pacemaker リソースを無効にします。

```
# pcs resource disable <ocf::pacemaker:remote on the node>
```

3. ノード上でメンテナンス作業を実行します。
4. IPMI の接続を復旧し、ノードを起動します。ノードの準備ができてから手順を進めます。
5. ノードで Pacemaker リソースを有効にし、ノードを起動します。

```
# pcs resource enable <ocf::pacemaker:remote on the node>  
# pcs node unstandby <node UUID>
```

6. ノードをメンテナンスモードに設定した場合は、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んで、ノードをメンテナンスモードから復帰させます。

```
# source stackrc  
# openstack baremetal node maintenance unset <baremetal node UUID>
```

検証

1. Pacemaker リソースがアクティブで正常なことを確認します。

```
# pcs status
```

2. 起動プロセス中に Pacemaker リソースが起動できない場合は、**pcs resource cleanup** コマンドを実行して、リソースのステータスと異常回数をリセットします。

3. ノードを停止する前にコンピュートノードからインスタンスを退避させた場合は、インスタンスが別のノードに移行されていることを確認します。

```
# openstack server list --long  
# nova migration-list
```