



Red Hat OpenShift Container Storage 4.7

ノードの置き換え

置き換え用のノードを準備し、障害が発生したノードを置き換える方法

Red Hat OpenShift Container Storage 4.7 ノードの置き換え

置き換え用のノードを準備し、障害が発生したノードを置き換える方法

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/Replacing_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

本書では、Red Hat OpenShift Container Storage クラスターのノードを安全に置き換える方法について説明します。

目次

多様性を受け入れるオープンソースの強化	4
RED HAT ドキュメントへのフィードバックの提供	5
はじめに	6
第1章 動的デバイスを使用してデプロイされた OPENSIFT CONTAINER STORAGE	7
1.1. AWS にデプロイされる OPENSIFT CONTAINER STORAGE	7
1.1.1. ユーザーによってプロビジョニングされるインフラストラクチャーで動作する AWS ノードの置き換え	7
1.1.2. インストーラーでプロビジョニングされるインフラストラクチャーで動作する AWS ノードの置き換え	9
1.1.3. ユーザーによってプロビジョニングされるインフラストラクチャーでの失敗した AWS ノードの置き換え	10
1.1.4. インストーラーでプロビジョニングされるインフラストラクチャーでの失敗した AWS ノードの置き換え	12
1.2. VMWARE にデプロイされる OPENSIFT CONTAINER STORAGE	13
1.2.1. ユーザーによってプロビジョニングされるインフラストラクチャーで動作する VMware ノードの置き換え	14
1.2.2. インストーラーでプロビジョニングされるインフラストラクチャーで動作する VMware ノードの置き換え	16
1.2.3. ユーザーによってプロビジョニングされるインフラストラクチャーでの失敗した VMware ノードの置き換え	17
1.2.4. インストーラーでプロビジョニングされるインフラストラクチャーでの失敗した VMware ノードの置き換え	19
1.3. RED HAT VIRTUALIZATION にデプロイされた OPENSIFT CONTAINER STORAGE	21
1.3.1. インストーラーでプロビジョニングされるインフラストラクチャーで動作する Red Hat Virtualization ノードの置き換え	21
1.3.2. インストーラーでプロビジョニングされるインフラストラクチャーで障害のある Red Hat Virtualization ノードの置き換え	23
1.4. MICROSOFT AZURE でデプロイされた OPENSIFT CONTAINER STORAGE	24
1.4.1. Azure のインストーラーでプロビジョニングされるインフラストラクチャーで動作するノードの置き換え	24
1.4.2. Azure のインストーラーでプロビジョニングされるインフラストラクチャーでの失敗したノードの置き換え	26
第2章 ローカルストレージデバイスを使用した OPENSIFT CONTAINER STORAGE のデプロイ	28
2.1. ベアメタルインフラストラクチャーでのストレージノードの置き換え	28
2.1.1. ユーザーによってプロビジョニングされるインフラストラクチャーで動作するノードの置き換え	28
2.1.2. ユーザーによってプロビジョニングされるインフラストラクチャーでの失敗したノードの置き換え	32
2.2. IBM Z または LINUXONE インフラストラクチャーでのストレージノードの置き換え	37
2.2.1. IBM Z または LinuxONE インフラストラクチャーでの動作するノードの置き換え	37
2.2.2. IBM Z または LinuxONE インフラストラクチャーでの障害のあるノードの置き換え	39
2.3. AMAZON EC2 インフラストラクチャーでのストレージノードの置き換え	41
2.3.1. ユーザーによってプロビジョニングされるインフラストラクチャーで動作する Amazon EC2 ノードの置き換え	41
2.3.2. インストーラーでプロビジョニングされるインフラストラクチャーで動作する Amazon EC2 ノードの置き換え	47
2.3.3. ユーザーによってプロビジョニングされるインフラストラクチャーでの障害のある Amazon EC2 ノードの置き換え	52
2.3.4. インストーラーでプロビジョニングされるインフラストラクチャーでの失敗した Amazon EC2 ノードの置き換え	58
2.4. VMWARE インフラストラクチャーでのストレージノードの置き換え	64
2.4.1. VMware のユーザーによってプロビジョニングされるインフラストラクチャーで動作するノードの置き換え	64
2.4.2. VMware のインストーラーでプロビジョニングされるインフラストラクチャーで動作するノードの置き換え	

え	69
2.4.3. VMware ユーザーによってプロビジョニングされるインフラストラクチャーでの障害のあるノードの置き換え	74
2.4.4. VMware のインストーラーでプロビジョニングされるインフラストラクチャーで障害のあるノードの置き換え	78
2.5. RED HAT VIRTUALIZATION インフラストラクチャーでのストレージノードの置き換え	83
2.5.1. Red Hat Virtualization インストーラーでプロビジョニングされるインフラストラクチャーで動作するノードの置き換え	83
2.5.2. Red Hat Virtualization インストーラーでプロビジョニングされるインフラストラクチャーで障害のあるノードの置き換え	88
2.6. IBM POWER SYSTEMS インフラストラクチャーでのストレージノードの置き換え	94
2.6.1. IBM Power Systems で動作するストレージまたは障害のあるストレージノードの置き換え	94

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

弊社のドキュメントについての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、ぜひお知らせください。フィードバックをお寄せいただくには、以下をご確認ください。

- 特定の部分についての簡単なコメントをお寄せいただく場合は、以下をご確認ください。
 1. ドキュメントの表示が **Multi-page HTML** 形式になっていることを確認してください。ドキュメントの右上隅に **Feedback** ボタンがあることを確認してください。
 2. マウスカーソルを使用して、コメントを追加するテキストの部分を強調表示します。
 3. 強調表示されたテキストの下に表示される **Add Feedback** ポップアップをクリックします。
 4. 表示される指示に従ってください。
- より詳細なフィードバックをお寄せいただく場合は、Bugzilla のチケットを作成してください。
 1. [Bugzilla](#) の Web サイトに移動します。
 2. Component (コンポーネント) として **Documentation** を使用します。
 3. **Description** フィールドに、ドキュメントの改善に向けたご提案を記入してください。ドキュメントの該当部分へのリンクも追加してください。
 4. **Submit Bug** をクリックします。

はじめに

OpenShift Container Storage では、動作ノードに対しては事前対応として、以下のデプロイメントで障害のあるノードに対しては事後対応として、ノードを交換できます。

- Amazon Web Services (AWS)
 - ユーザーによってプロビジョニングされるインフラストラクチャー
 - インストーラーでプロビジョニングされるインフラストラクチャー
- VMware
 - ユーザーによってプロビジョニングされるインフラストラクチャー
- Red Hat Virtualization の場合:
 - インストーラーでプロビジョニングされるインフラストラクチャー
- Microsoft Azure
 - インストーラーでプロビジョニングされるインフラストラクチャー
- ローカルストレージデバイスの場合
 - ベアメタル
 - Amazon EC2 I3
 - VMware
 - Red Hat Virtualization
 - IBM Power Systems
- 外部モードでストレージノードを置き換える場合は、[Red Hat Ceph Storage のドキュメント](#)を参照してください。

第1章 動的デバイスを使用してデプロイされた OPENSIFT CONTAINER STORAGE

1.1. AWS にデプロイされる OPENSIFT CONTAINER STORAGE

1.1.1. ユーザーによってプロビジョニングされるインフラストラクチャーで動作する AWS ノードの置き換え

以下の手順に従って、AWS のユーザーによってプロビジョニングされるインフラストラクチャーで動作するノードを置き換えます。

前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャーおよびリソースで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスターにログインしている必要があります。

手順

- 置き換える必要のあるノードを特定します。
- 以下のコマンドを実行して、ノードにスケジューリング対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

- 以下のコマンドを使用してノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```



重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。この期間に生成される Ceph のエラーは一時的なもので、新規ノードにラベルが付けられ、これが機能すると自動的に解決されます。

- 以下のコマンドを使用してノードを削除します。

```
$ oc delete nodes <node_name>
```

- 必要なインフラストラクチャーで新規 AWS マシンインスタンスを作成します。 [プラットフォーム要件](#) を参照してください。
- 新規 AWS マシンインスタンスを使用して新規 OpenShift Container Platform ノードを作成します。
- Pending** 状態の OpenShift Container Platform に関連する証明書署名要求 (CSR) の有無を確認します。

```
$ oc get csr
```

- 新規ノードに必要なすべての OpenShift Container Platform CSR を承認します。

```
$ oc adm certificate approve <Certificate_Name>
```

- Compute** → **Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。
- OpenShift Container Storage ラベルを新規ノードに適用します。

Web ユーザーインターフェイスの使用

- 新規ノードについて、**Action Menu (⋮)** → **Edit Labels** をクリックします。
- cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

検証手順

- 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

- Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- csi-cephfsplugin-***
- csi-rbdplugin-***

- 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。
- 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

- (オプション) クラスターでクラスター全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- 直前の手順で特定された新規ノードごとに、以下を実行します。
 - デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- lsblk を実行し、**ocs-deviceset** 名の横にある crypt キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

1.1.2. インストーラーでプロビジョニングされるインフラストラクチャーで動作する AWS ノードの置き換え

以下の手順を使用して、AWS のインストーラーでプロビジョニングされるインフラストラクチャー (IPI) で動作するノードを置き換えます。

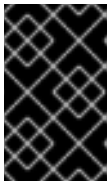
手順

1. OpenShift Web コンソールにログインし、**Compute → Nodes** をクリックします。
2. 置き換える必要のあるノードを特定します。その **マシン名** をメモします。
3. 以下のコマンドを実行して、ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

4. 以下のコマンドを使用してノードをドレイン (解放) します。

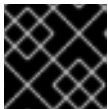
```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```



重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。この期間に生成される Ceph のエラーは一時的なもので、新規ノードにラベルが付けられ、これが機能すると自動的に解決されます。

5. **Compute → Machines** をクリックします。必要なマシンを検索します。
6. 必要なマシンの横にある **Action menu (⋮) → Delete Machine** をクリックします。
7. **Delete** をクリックしてマシンの削除を確認します。新しいマシンが自動的に作成されます。
8. 新規マシンが起動し、**Running** 状態に移行するまで待機します。



重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。

9. **Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。
10. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu (⋮) → Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-***
- **csi-rbdplugin-***

3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。
4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスターでクラスター全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。
 - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. `lsblk` を実行し、**ocs-deviceset** 名の横にある `crypt` キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

1.1.3. ユーザーによってプロビジョニングされるインフラストラクチャーでの失敗した AWS ノードの置き換え

以下の手順に従って、OpenShift Container Storage の AWS のユーザーによってプロビジョニングされるインフラストラクチャー (UPI) で動作しない障害のあるノードを置き換えます。

前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャーおよびリソースで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスターにログインしている必要があります。

手順

1. 置き換える必要のあるノードの AWS マシンインスタンスを特定します。
2. AWS にログインし、特定された AWS マシンインスタンスを終了します。
3. 必要なインフラストラクチャーで新規 AWS マシンインスタンスを作成します。[プラットフォーム要件](#)を参照してください。
4. 新規 AWS マシンインスタンスを使用して新規 OpenShift Container Platform ノードを作成します。
5. **Pending** 状態の OpenShift Container Platform に関連する証明書署名要求 (CSR) の有無を確認します。

```
$ oc get csr
```

6. 新規ノードに必要なすべての OpenShift Container Platform CSR を承認します。

```
$ oc adm certificate approve <Certificate_Name>
```

7. **Compute** → **Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。
8. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu (⋮)** → **Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。
- ```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```
2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。
    - **csi-cephfsplugin-\***
    - **csi-rbdplugin-\***
  3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。
  4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスターでクラスター全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。

- i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. `lsblk` を実行し、**ocs-deviceset** 名の横にある `crypt` キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

### 1.1.4. インストーラーでプロビジョニングされるインフラストラクチャーでの失敗した AWS ノードの置き換え

以下の手順に従って、OpenShift Container Storage の AWS のインストーラーでプロビジョニングされるインフラストラクチャー (IPI) で動作しない障害のあるノードを置き換えます。

#### 手順

1. OpenShift Web コンソールにログインし、**Compute → Nodes** をクリックします。
2. 障害のあるノードを特定し、その **Machine Name** をクリックします。
3. **Actions → Edit Annotations** をクリックし、**Add More** をクリックします。
4. **machine.openshift.io/exclude-node-draining** を追加し、**Save** をクリックします。
5. **Actions → Delete Machine** をクリックしてから、**Delete** をクリックします。
6. 新しいマシンが自動的に作成されます。新規マシンが起動するのを待機します。



#### 重要

このアクティビティには少なくとも 5-10 分以上かかる場合があります。この期間に生成される Ceph のエラーは一時的なもので、新規ノードにラベルが付けられ、これが機能すると自動的に解決されます。

7. **Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。
8. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu (⋮) → Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。



## コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

9. [オプション]: 失敗した AWS インスタンスが自動的に削除されない場合、インスタンスを AWS コンソールで終了します。

## 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。
4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。
  - a. 直前の手順で特定された新規ノードごとに、以下を実行します。
    - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. lsblk を実行し、**ocs-deviceset** 名の横にある crypt キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポート](#)にお問い合わせください。

## 1.2. VMWARE にデプロイされる OPENSIFT CONTAINER STORAGE

- 動作するノードを置き換えるには、以下を参照してください。
  - [「ユーザーによってプロビジョニングされるインフラストラクチャーで動作する VMware ノードの置き換え」](#)

- 「インストーラーでプロビジョニングされるインフラストラクチャーで動作する VMware ノードの置き換え」
- 障害のあるノードを置き換えるには、以下を参照してください。
  - 「ユーザーによってプロビジョニングされるインフラストラクチャーでの失敗した VMware ノードの置き換え」
  - 「インストーラーでプロビジョニングされるインフラストラクチャーでの失敗した VMware ノードの置き換え」

### 1.2.1. ユーザーによってプロビジョニングされるインフラストラクチャーで動作する VMware ノードの置き換え

以下の手順に従って、VMware のユーザーによってプロビジョニングされるインフラストラクチャー (UPI) で動作するノードを置き換えます。

#### 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャー、リソースおよびディスクで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスタにログインしている必要があります。

#### 手順

- 置き換える必要があるノードとその仮想マシンを特定します。
- 以下のコマンドを実行して、ノードにスケジューリング対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

- 以下のコマンドを使用してノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```



#### 重要

このアクティビティには少なくとも 5-10 分以上かかる場合があります。この期間に生成される Ceph のエラーは一時的なもので、新規ノードにラベルが付けられ、これが機能すると自動的に解決されます。

- 以下のコマンドを使用してノードを削除します。

```
$ oc delete nodes <node_name>
```

- VSphere にログインし、特定された仮想マシンを終了します。



#### 重要

仮想マシンはインベントリからのみ削除し、ディスクから削除しないでください。

6. 必要なインフラストラクチャーで vSphere に新規の仮想マシンを作成します。[プラットフォーム要件](#) を参照してください。
7. 新規の仮想マシンを使用して新規 OpenShift Container Platform ワーカーノードを作成します。
8. **Pending** 状態の OpenShift Container Platform に関連する証明書署名要求 (CSR) の有無を確認します。

```
$ oc get csr
```

9. 新規ノードに必要なすべての OpenShift Container Platform CSR を承認します。

```
$ oc adm certificate approve <Certificate_Name>
```

10. **Compute** → **Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。
11. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu ( ⋮ )** → **Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

#### 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。
  - **csi-cephfsplugin-\***
  - **csi-rbdplugin-\***
3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。
4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスターでクラスター全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。
  - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. lsblk を実行し、**ocs-deviceset** 名の横にある crypt キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

### 1.2.2. インストーラーでプロビジョニングされるインフラストラクチャーで動作する VMware ノードの置き換え

以下の手順を使用して、VMware のインストーラーでプロビジョニングされるインフラストラクチャー (IPI) で動作するノードを置き換えます。

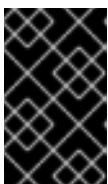
#### 手順

1. OpenShift Web コンソールにログインし、**Compute → Nodes** をクリックします。
2. 置き換える必要のあるノードを特定します。その **マシン名** をメモします。
3. 以下のコマンドを実行して、ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

4. 以下のコマンドを使用してノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```



#### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。この期間に生成される Ceph のエラーは一時的なもので、新規ノードにラベルが付けられ、これが機能すると自動的に解決されます。

5. **Compute → Machines** をクリックします。必要なマシンを検索します。
6. 必要なマシンの横にある **Action menu (⋮) → Delete Machine** をクリックします。
7. **Delete** をクリックしてマシンの削除を確認します。新しいマシンが自動的に作成されます。
8. 新規マシンが起動し、**Running** 状態に移行するまで待機します。



#### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。

9. **Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。

10. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu (⋮) → Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

#### 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads → Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。
4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。
  - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. lsblk を実行し、**ocs-deviceset** 名の横にある crypt キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

#### 1.2.3. ユーザーによってプロビジョニングされるインフラストラクチャーでの失敗した VMware ノードの置き換え

以下の手順に従って、VMware のユーザーによってプロビジョニングされるインフラストラクチャー (UPI) で失敗したノードを置き換えます。

## 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャー、リソースおよびディスクで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスターにログインしている必要があります。

## 手順

- 置き換える必要があるノードとその仮想マシンを特定します。
- 以下のコマンドを使用してノードを削除します。

```
$ oc delete nodes <node_name>
```

- VSphere にログインし、特定された仮想マシンを終了します。



### 重要

仮想マシンはインベントリーからのみ削除し、ディスクから削除しないでください。

- 必要なインフラストラクチャーで vSphere に新規の仮想マシンを作成します。[プラットフォーム要件](#) を参照してください。
- 新規の仮想マシンを使用して新規 OpenShift Container Platform ワーカーノードを作成します。
- Pending** 状態の OpenShift Container Platform に関連する証明書署名要求 (CSR) の有無を確認します。

```
$ oc get csr
```

- 新規ノードに必要なすべての OpenShift Container Platform CSR を承認します。

```
$ oc adm certificate approve <Certificate_Name>
```

- Compute** → **Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。
- 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

### ユーザーインターフェイスを使用する場合

- 新規ノードについて、**Action Menu (⋮)** → **Edit Labels** をクリックします。
- cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

—

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

## 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。
4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。
  - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. `lsblk` を実行し、**ocs-deviceset** 名の横にある `crypt` キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

## 1.2.4. インストーラーでプロビジョニングされるインフラストラクチャーでの失敗した VMware ノードの置き換え

以下の手順に従って、OpenShift Container Storage の VMware のインストーラーでプロビジョニングされるインフラストラクチャー (IPI) で動作しない障害のあるノードを置き換えます。

## 手順

1. OpenShift Web コンソールにログインし、**Compute** → **Nodes** をクリックします。
2. 障害のあるノードを特定し、その **Machine Name** をクリックします。
3. **Actions** → **Edit Annotations** をクリックし、**Add More** をクリックします。
4. **machine.openshift.io/exclude-node-draining** を追加し、**Save** をクリックします。

5. **Actions** → **Delete Machine** をクリックしてから、**Delete** をクリックします。
6. 新しいマシンが自動的に作成されます。新規マシンが起動するのを待機します。



### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。この期間に生成される Ceph のエラーは一時的なもので、新規ノードにラベルが付けられ、これが機能すると自動的に解決されます。

7. **Compute** → **Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。
8. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu (⋮)** → **Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

9. [オプション]: 失敗した VM インスタンスが自動的に削除されない場合、仮想マシンを vSphere で終了します。

### 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。
  - **csi-cephfsplugin-\***
  - **csi-rbdplugin-\***
3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。
4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。
  - a. 直前の手順で特定された新規ノードごとに、以下を実行します。



- i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. lsblk を実行し、**ocs-deviceset** 名の横にある crypt キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

## 1.3. RED HAT VIRTUALIZATION にデプロイされた OPENSIFT CONTAINER STORAGE

### 1.3.1. インストーラーでプロビジョニングされるインフラストラクチャーで動作する Red Hat Virtualization ノードの置き換え

以下の手順を使用して、Red Hat Virtualization のインストーラーでプロビジョニングされるインフラストラクチャー (IPI) で動作するノードを置き換えます。

#### 手順

1. OpenShift Web コンソールにログインし、**Compute → Nodes** をクリックします。
2. 置き換える必要のあるノードを特定します。その **マシン名** をメモします。
3. 以下のコマンドを実行して、ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

4. 以下のコマンドを使用してノードをドレイン (解放) します。

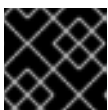
```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```



#### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。この期間に生成される Ceph のエラーは一時的なもので、新規ノードにラベルが付けられ、これが機能すると自動的に解決されます。

5. **Compute → Machines** をクリックします。必要なマシンを検索します。
6. 必要なマシンの横にある **Action menu (⋮) → Delete Machine** をクリックします。
7. **Delete** をクリックしてマシンの削除を確認します。新しいマシンが自動的に作成されます。新規マシンが起動し、**Running** 状態に移行するまで待機します。



#### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。

8. **Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。
9. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu ( ⋮ ) → Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

#### 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads → Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。
4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。
  - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. `lsblk` を実行し、**ocs-deviceset** 名の横にある `crypt` キーワードを確認します。

```
$ lsblk
```

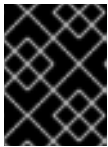
6. 検証手順が失敗した場合は、[Red Hat サポート](#)にお問い合わせください。

### 1.3.2. インストーラーでプロビジョニングされるインフラストラクチャーで障害のある Red Hat Virtualization ノードの置き換え

以下の手順に従って、OpenShift Container Storage の Red Hat Virtualization のインストーラーでプロビジョニングされるインフラストラクチャー (IPI) で動作しない障害のあるノードを置き換えます。

#### 手順

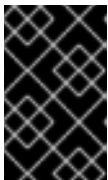
1. OpenShift Web コンソールにログインし、**Compute → Nodes** をクリックします。
2. 障害のあるノードを特定します。その **マシン名** をメモします。
3. **Red Hat Virtualization 管理ポータル** にログインし、**mon** および **OSD** に関連付けられた仮想ディスクを障害の発生した仮想マシンから削除します。  
この手順は、仮想マシンインスタンスがマシンの削除ステップの一部として削除される際にディスクが削除されないようにするために必要です。



#### 重要

ディスクの削除時に、**Remove Permanently** オプションを選択しないでください。

4. OpenShift Web コンソールで、**Compute → Machines** をクリックします。必要なマシンを検索します。
5. **Actions → Edit Annotations** をクリックし、**Add More** をクリックします。
6. **machine.openshift.io/exclude-node-draining** を追加し、**Save** をクリックします。
7. **Actions → Delete Machine** をクリックしてから、**Delete** をクリックします。  
新しいマシンが自動的に作成されます。新規マシンが起動するのを待機します。



#### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。この期間に生成される Ceph のエラーは一時的なもので、新規ノードにラベルが付けられ、これが機能すると自動的に解決されます。

8. **Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。
9. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu (⋮) → Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

10. (オプション) 失敗した仮想マシンが自動的に削除されない場合は、Red Hat Virtualization 管理ポータルから仮想マシンを削除します。

## 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。
4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。
  - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. `lsblk` を実行し、**ocs-deviceset** 名の横にある `crypt` キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポート](#)にお問い合わせください。

## 1.4. MICROSOFT AZURE でデプロイされた OPENSIFT CONTAINER STORAGE

### 1.4.1. Azure のインストーラーでプロビジョニングされるインフラストラクチャーで動作するノードの置き換え

以下の手順を使用して、Azure のインストーラーでプロビジョニングされるインフラストラクチャー (IPI) で動作するノードを置き換えます。

## 手順

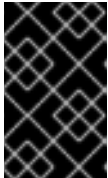
1. OpenShift Web コンソールにログインし、**Compute** → **Nodes** をクリックします。
2. 置き換える必要のあるノードを特定します。その **マシン名** をメモします。

- 以下のコマンドを実行して、ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

- 以下のコマンドを使用してノードをドレイン (解放) します。

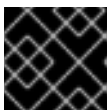
```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```



### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。この期間に生成される Ceph のエラーは一時的なもので、新規ノードにラベルが付けられ、これが機能すると自動的に解決されます。

- Compute → Machines** をクリックします。必要なマシンを検索します。
- 必要なマシンの横にある **Action menu (⋮) → Delete Machine** をクリックします。
- Delete** をクリックしてマシンの削除を確認します。新しいマシンが自動的に作成されます。
- 新規マシンが起動し、**Running** 状態に移行するまで待機します。



### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。

- Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。
- 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- 新規ノードについて、**Action Menu (⋮) → Edit Labels** をクリックします。
- cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

### 検証手順

- 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

- Workloads → Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
  - **csi-rbdplugin-\***
3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。
  4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスターでクラスター全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。
  - a. 直前の手順で特定された新規ノードごとに、以下を実行します。
    - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. lsblk を実行し、**ocs-deviceset** 名の横にある crypt キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

### 1.4.2. Azure のインストーラーでプロビジョニングされるインフラストラクチャーでの失敗したノードの置き換え

以下の手順に従って、OpenShift Container Storage の Azure のインストーラーでプロビジョニングされるインフラストラクチャー (IPI) で動作しない障害のあるノードを置き換えます。

#### 手順

1. OpenShift Web コンソールにログインし、**Compute → Nodes** をクリックします。
2. 障害のあるノードを特定し、その **Machine Name** をクリックします。
3. **Actions → Edit Annotations** をクリックし、**Add More** をクリックします。
4. **machine.openshift.io/exclude-node-draining** を追加し、**Save** をクリックします。
5. **Actions → Delete Machine** をクリックしてから、**Delete** をクリックします。
6. 新しいマシンが自動的に作成されます。新規マシンが起動するのを待機します。



#### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。この期間に生成される Ceph のエラーは一時的なもので、新規ノードにラベルが付けられ、これが機能すると自動的に解決されます。

7. **Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。

8. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- 新規ノードについて、**Action Menu (⋮)** → **Edit Labels** をクリックします。
- cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

9. [オプション]: 失敗した Azure インスタンスが自動的に削除されない場合、インスタンスを Azure コンソールで終了します。

#### 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- csi-cephfsplugin-\***
- csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。
4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- 直前の手順で特定された新規ノードごとに、以下を実行します。
  - デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- lsblk を実行し、**ocs-deviceset** 名の横にある crypt キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポート](#)にお問い合わせください。

## 第2章 ローカルストレージデバイスを使用した OPENSIFT CONTAINER STORAGE のデプロイ

### 2.1. ベアメタルインフラストラクチャーでのストレージノードの置き換え

- 動作するノードを置き換えるには、「[ユーザーによってプロビジョニングされるインフラストラクチャーで動作するノードの置き換え](#)」を参照してください。
- 障害のあるノードを置き換えるには、「[ユーザーによってプロビジョニングされるインフラストラクチャーでの失敗したノードの置き換え](#)」を参照してください。

#### 2.1.1. ユーザーによってプロビジョニングされるインフラストラクチャーで動作するノードの置き換え

##### 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャー、リソースおよびディスクで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスターにログインしている必要があります。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、デバイスの自動プロビジョニングを有効にするために **LocalVolumeSet** オブジェクトを作成していない場合は、[ローカルストレージでサポートされるクラスターの更新後の設定の変更](#) についての以下の手順に従って、これを実行します。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、**LocalVolumeDiscovery** オブジェクトを作成していない場合は、[ローカルストレージでサポートされるクラスターの更新後の設定の変更](#) についての以下の手順に従って、これを実行します。

##### 手順

1. NODE を特定し、置き換えるノードのラベルを取得します。

```
$ oc get nodes --show-labels | grep <node_name>
```

2. 置き換えるノードで実行されている **mon** (ある場合) および OSD を特定します。

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. 先の手順で特定された Pod のデプロイメントをスケールダウンします。  
以下に例を示します。

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

4. ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```



5. ノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

6. ノードを削除します。

```
$ oc delete node <node_name>
```

7. 必要なインフラストラクチャーで新規のベアメタルマシンを取得します。[クラスターのベアメタルへのインストール](#) について参照してください。

8. 新規ベアメタルマシンを使用して新規 OpenShift Container Platform ノードを作成します。

9. Pending 状態の OpenShift Container Platform に関連する証明書署名要求 (CSR) の有無を確認します。

```
$ oc get csr
```

10. 新規ノードに必要なすべての OpenShift Container Platform CSR を承認します。

```
$ oc adm certificate approve <Certificate_Name>
```

11. OpenShift Web コンソールで **Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあるかどうかを確認します。

12. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu ( ⋮ ) → Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

13. OpenShift ローカルストレージ Operator がインストールされている namespace を特定し、これを **local\_storage\_project** 変数に割り当てます。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

以下に例を示します。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

14. 新規ワーカーノードを **localVolumeDiscovery** および **localVolumeSet** に追加します。

- a. **localVolumeDiscovery** 定義を更新し、新規ノードを追加して失敗したノードを削除します。

```
oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
 nodeSelectorTerms:
 - matchExpressions:
 - key: kubernetes.io/hostname
 operator: In
 values:
 - server1.example.com
 - server2.example.com
 #- server3.example.com
 - newnode.example.com
 [...]
[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

- b. 編集する **localVolumeSet** を決定します。

```
oc get -n $local_storage_project localvolumeset
NAME AGE
localblock 25h
```

- c. **localVolumeSet** 定義を更新して、新規ノードを追加し、障害が発生したノードを削除します。

```
oc edit -n $local_storage_project localvolumeset localblock
[...]
nodeSelector:
 nodeSelectorTerms:
 - matchExpressions:
 - key: kubernetes.io/hostname
 operator: In
 values:
 - server1.example.com
 - server2.example.com
 #- server3.example.com
 - newnode.example.com
 [...]
[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

15. 新規 **localblock** PV が利用可能であることを確認します。

```
$oc get pv | grep localblock | grep Available
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

16. **openshift-storage** プロジェクトを変更します。

```
$ oc project openshift-storage
```

17. 失敗した OSD をクラスターから削除します。必要に応じて、複数の障害のある OSD を指定することができます。

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> FORCE_OSD_REMOVAL=false | oc create -n
openshift-storage -f -
```

<failed\_osd\_id>

**rook-ceph-osd** 接頭辞の直後の Pod 名の整数です。コマンドにコンマ区切りの OSD ID を追加して、複数の OSD を削除できます (例: **FAILED\_OSD\_IDS=0,1,2**)

OSD が 3 つしかないクラスター、または OSD が削除された後にデータの 3 つのレプリカすべてを復元するにはスペースが不十分なクラスターでは、**FORCE\_OSD\_REMOVAL** 値を **true** に変更する必要があります。

18. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されたことを確認します。

**Completed** のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```



#### 注記

**ocs-osd-removal-job** が失敗し、Pod が予想される **Completed** の状態にない場合、追加のデバッグのために Pod ログを確認します。以下に例を示します。

```
oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```

19. **ocs-osd-removal-job** を削除します。

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

#### 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が Running 状態にあることを確認します。  
また、増分の **mon** が新規に作成されており、Running 状態にあることを確認します。

```
$ oc get pod -n openshift-storage | grep mon
```

出力例:

```
rook-ceph-mon-a-cd575c89b-b6k66 2/2 Running
0 38m
rook-ceph-mon-b-6776bc469b-tzxt8 2/2 Running
0 38m
rook-ceph-mon-d-5ff5d488b5-7v8xh 2/2 Running
0 4m8s
```

OSD と Mon が **Running** 状態になるまで数分かかる場合があります。

4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。
  - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. `lsblk` を実行し、**ocs-deviceset** 名の横にある `crypt` キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

## 2.1.2. ユーザーによってプロビジョニングされるインフラストラクチャーでの失敗したノードの置き換え

### 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャー、リソースおよびディスクで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスタにログインしている必要があります。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、デバイスの自動プロビジョニングを有効にするために **LocalVolumeSet** オブジェクトを作成していない場合は、[ローカルストレージでサポートされるクラスタの更新後の設定の変更](#) についての以下の手順に従って、これを実行します。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、**LocalVolumeDiscovery** オブジェクトを作成していない場合は、[ローカルストレージでサ](#)

[ポートされるクラスターの更新後の設定の変更](#) についての以下の手順に従って、これを実行します。

## 手順

1. NODE を特定し、置き換えるノードのラベルを取得します。

```
$ oc get nodes --show-labels | grep <node_name>
```

2. 置き換えるノードで実行されている **mon** (ある場合) および OSD を特定します。

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. 先の手順で特定された Pod のデプロイメントをスケールダウンします。  
以下に例を示します。

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

4. ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

5. Terminating 状態の Pod を削除します。

```
$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system ("oc -
n " $1 " delete pods " $2 " --grace-period=0 " " --force ")}'
```

6. ノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

7. ノードを削除します。

```
$ oc delete node <node_name>
```

8. 必要なインフラストラクチャーで新規のベアメタルマシンを取得します。[クラスターのベアメタルへのインストール](#) について参照してください。

9. 新規ベアメタルマシンを使用して新規 OpenShift Container Platform ノードを作成します。

10. Pending 状態の OpenShift Container Platform に関連する証明書署名要求 (CSR) の有無を確認します。

```
$ oc get csr
```

11. 新規ノードに必要なすべての OpenShift Container Platform CSR を承認します。

```
$ oc adm certificate approve <Certificate_Name>
```

12. OpenShift Web コンソールで **Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあるかどうかを確認します。
13. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu ( ⋮ ) → Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

14. OpenShift ローカルストレージ Operator がインストールされている namespace を特定し、これを **local\_storage\_project** 変数に割り当てます。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

以下に例を示します。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

15. 新規ワーカーノードを **localVolumeDiscovery** および **localVolumeSet** に追加します。
  - a. **localVolumeDiscovery** 定義を更新し、新規ノードを追加して失敗したノードを削除します。

```
oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

- b. 編集する **localVolumeSet** を決定します。

■

```
oc get -n $local_storage_project localvolumeset
NAME AGE
localblock 25h
```

- c. **localVolumeSet** 定義を更新して、新規ノードを追加し、障害が発生したノードを削除します。

```
oc edit -n $local_storage_project localvolumeset localblock
[...]
nodeSelector:
 nodeSelectorTerms:
 - matchExpressions:
 - key: kubernetes.io/hostname
 operator: In
 values:
 - server1.example.com
 - server2.example.com
 #- server3.example.com
 - newnode.example.com
[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

16. 新規 **localblock** PV が利用可能であることを確認します。

```
$oc get pv | grep localblock | grep Available
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

17. **openshift-storage** プロジェクトを変更します。

```
$ oc project openshift-storage
```

18. 失敗した OSD をクラスターから削除します。必要に応じて、複数の障害のある OSD を指定することができます。

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> FORCE_OSD_REMOVAL=false | oc create -n
openshift-storage -f -
```

<failed\_osd\_id>

**rook-ceph-osd** 接頭辞の直後の Pod 名の整数です。コマンドにコンマ区切りの OSD ID を追加して、複数の OSD を削除できます (例: **FAILED\_OSD\_IDS=0,1,2**)

OSD が 3 つしかないクラスター、または OSD が削除された後にデータの 3 つのレプリカすべてを復元するにはスペースが不十分なクラスターでは、**FORCE\_OSD\_REMOVAL** 値を **true** に変更する必要があります。

19. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されたことを確認します。

**Completed** のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```



### 注記

**ocs-osd-removal-job** が失敗し、Pod が予想される **Completed** の状態にない場合、追加のデバッグのために Pod ログを確認します。以下に例を示します。

```
oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```

## 20. **ocs-osd-removal-job** を削除します。

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

### 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が Running 状態にあることを確認します。  
また、増分の **mon** が新規に作成されており、Running 状態にあることを確認します。

```
$ oc get pod -n openshift-storage | grep mon
```

出力例:

```
rook-ceph-mon-a-cd575c89b-b6k66 2/2 Running
0 38m
rook-ceph-mon-b-6776bc469b-tzzt8 2/2 Running
0 38m
rook-ceph-mon-d-5ff5d488b5-7v8xh 2/2 Running
0 4m8s
```

OSD と Mon が **Running** 状態になるまで数分かかる場合があります。

4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```



5. (オプション) クラスターでクラスター全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。

- i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. lsblk を実行し、**ocs-deviceset** 名の横にある crypt キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

## 2.2. IBM Z または LINUXONE インフラストラクチャーでのストレージノードの置き換え

以下のいずれかの手順を選択して、ストレージノードを置き換えることができます。

- [「IBM Z または LinuxONE インフラストラクチャーでの動作するノードの置き換え」](#)
- [「IBM Z または LinuxONE インフラストラクチャーでの障害のあるノードの置き換え」](#)

### 2.2.1. IBM Z または LinuxONE インフラストラクチャーでの動作するノードの置き換え

以下の手順に従って、IBM Z または LinuxONE インフラストラクチャーで動作するノードを置き換えます。

#### 手順

1. OpenShift Web コンソールにログインします。
2. **Compute** → **Nodes** をクリックします。
3. 置き換える必要のあるノードを特定します。その **マシン名** をメモします。
4. 以下のコマンドを実行して、ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

5. 以下のコマンドを使用してノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```



#### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。この期間に生成される Ceph のエラーは一時的なもので、新規ノードにラベルが付けられ、これが機能すると自動的に解決されます。

6. **Compute** → **Machines** をクリックします。必要なマシンを検索します。
7. 必要なマシンの横にある **Action menu ( ⋮ )** → **Delete Machine** をクリックします。
8. **Delete** をクリックしてマシンの削除を確認します。新しいマシンが自動的に作成されます。
9. 新規マシンが起動し、**Running** 状態に移行するまで待機します。



### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。

10. **Compute** → **Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。
11. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu ( ⋮ )** → **Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

### 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。
4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスターでデータの暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。
  - a. 指定の OSD にバインドされる Persistent Volume Claim(永続ボリューム要求、PVC) を特定します。

```
$ oc describe pod/rook-ceph-osd-0-544db49d7f-qrgqm|grep pvc
ceph.rook.io/pvc=ocs-devicset-thin-0-data-0lg6zp
```

- b. OSD Pod が実行される場所を特定します。

```
$ oc get -o=custom-columns=NODE:.spec.nodeName pod/rook-ceph-osd-0-544db49d7f-qrgqm
```

- c. デバッグ Pod を作成し、ホストの **chroot** 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- d. デバイスが暗号化されていることを確認します。

```
$ dmsetup ls | grep ocs-devicset
ocs-devicset-0-data-0-57snx-block-dmccrypt (253:1)
```

```
$ lsblk | grep ocs-devicset
`-ocs-devicset-0-data-0-57snx-block-dmccrypt 253:1 0 512G 0 crypt
```

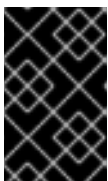
6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

### 2.2.2. IBM Z または LinuxONE インフラストラクチャーでの障害のあるノードの置き換え

以下の手順に従って、OpenShift Container Storage の IBM Z または LinuxONE インフラストラクチャーで動作しない障害のあるノードを置き換えます。

#### 手順

1. OpenShift Web コンソールにログインし、**Compute → Nodes** をクリックします。
2. 障害のあるノードを特定し、その **Machine Name** をクリックします。
3. **Actions → Edit Annotations** をクリックし、**Add More** をクリックします。
4. **machine.openshift.io/exclude-node-draining** を追加し、**Save** をクリックします。
5. **Actions → Delete Machine** をクリックしてから、**Delete** をクリックします。
6. 新しいマシンが自動的に作成されます。新規マシンが起動するのを待機します。



#### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。この期間に生成される Ceph のエラーは一時的なもので、新規ノードにラベルが付けられ、これが機能すると自動的に解決されます。

7. **Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。
8. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

## Web ユーザーインターフェイスの使用

- a. 新規ノードについて、**Action Menu ( ⋮ )** → **Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

## コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

9. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

10. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

11. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。
12. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

13. (オプション) クラスタでデータの暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 指定の OSD にバインドされる Persistent Volume Claim(永続ボリューム要求、PVC) を特定します。

```
$ oc describe pod/rook-ceph-osd-0-544db49d7f-qrgqm | grep pvc
ceph.rook.io/pvc=ocs-deviceset-thin-0-data-0lg6zp
```

- b. OSD Pod が実行される場所を特定します。

```
$ oc get -o=custom-columns=NODE:.spec.nodeName pod/rook-ceph-osd-0-544db49d7f-qrgqm
```

- c. デバッグ Pod を作成し、ホストの **chroot** 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- d. デバイスが暗号化されていることを確認します。

```
$ dmsetup ls | grep ocs-deviceset
ocs-deviceset-0-data-0-57snx-block-dmccrypt (253:1)
```

```
$ lsblk | grep ocs-deviceset
`-ocs-deviceset-0-data-0-57snx-block-dmccrypt 253:1 0 512G 0 crypt
```

14. 検証手順が失敗した場合は、[Red Hat サポート](#)にお問い合わせください。

## 2.3. AMAZON EC2 インフラストラクチャーでのストレージノードの置き換え

- ユーザーによってプロビジョニングされるインフラストラクチャーおよびインストーラーでプロビジョニングされるインフラストラクチャーで動作する Amazon EC2 ノードを置き換えるには、以下を参照してください。
  - 「ユーザーによってプロビジョニングされるインフラストラクチャーで動作する Amazon EC2 ノードの置き換え」
  - 「インストーラーでプロビジョニングされるインフラストラクチャーで動作する Amazon EC2 ノードの置き換え」
- ユーザーによってプロビジョニングされるインフラストラクチャーおよびインストーラーでプロビジョニングされるインフラストラクチャーで障害のある Amazon EC2 ノードを置き換えるには、以下を参照してください。
  - 「ユーザーによってプロビジョニングされるインフラストラクチャーでの障害のある Amazon EC2 ノードの置き換え」
  - 「インストーラーでプロビジョニングされるインフラストラクチャーでの失敗した Amazon EC2 ノードの置き換え」

### 2.3.1. ユーザーによってプロビジョニングされるインフラストラクチャーで動作する Amazon EC2 ノードの置き換え

以下の手順に従って、Amazon EC2 I3 のユーザーによってプロビジョニングされるインフラストラクチャー (UPI) で動作するノードを置き換えます。



#### 重要

Amazon EC2 I3 インフラストラクチャーのストレージノードの置き換えはテクノロジープレビュー機能です。テクノロジープレビュー機能は、Red Hat の実稼働環境のサービスレベルアグリーメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。Red Hat は実稼働環境でこれらを使用することを推奨していません。これらの機能は、近々発表予定の製品機能をリリースに先駆けてご提供することにより、お客様は機能性をテストし、開発プロセス中にフィードバックをお寄せいただくことができます。

#### 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャーおよびリソースで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスターにログインしている必要があります。

## 手順

1. ノードを特定し、置き換えるノードのラベルを取得します。

```
$ oc get nodes --show-labels | grep <node_name>
```

2. 置き換えるノードで実行されている mon (ある場合) および OSD を特定します。

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. 先の手順で特定された Pod のデプロイメントをスケールダウンします。  
以下に例を示します。

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

4. ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

5. ノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

6. ノードを削除します。

```
$ oc delete node <node_name>
```

7. 必要なインフラストラクチャーで新規 Amazon EC2 I3 マシンインスタンスを作成します。[サポートされるインフラストラクチャーおよびプラットフォーム](#) について参照してください。

8. 新規 Amazon EC2 I3 マシンインスタンスを使用して新規 OpenShift Container Platform ノードを作成します。

9. Pending 状態の OpenShift Container Platform に関連する証明書署名要求 (CSR) の有無を確認します。

```
$ oc get csr
```

10. 新規ノードに必要なすべての OpenShift Container Platform CSR を承認します。

```
$ oc adm certificate approve <Certificate_Name>
```

11. OpenShift Web コンソールで **Compute** → **Nodes** をクリックします。新規ノードが **Ready** 状態にあるかどうかを確認します。

12. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu (⋮)** → **Edit Labels** をクリックします。

- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

13. OpenShift ローカルストレージ Operator がインストールされている namespace を特定し、これを **local\_storage\_project** 変数に割り当てます。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

以下に例を示します。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

14. 新規ワーカーノードで利用可能なローカルストレージデバイスを OpenShift Container Storage StorageCluster に追加します。

- a. 新規ディスクエントリーを LocalVolume CR に追加します。

**LocalVolume** CR を編集します。障害のあるデバイス **/dev/disk/by-id/{id}** を削除またはコメントアウトし、新規の **/dev/disk/by-id/{id}** を追加します。

```
$ oc get -n $local_storage_project localvolume
```

出力例:

```
NAME AGE
local-block 25h
```

```
$ oc edit -n $local_storage_project localvolume local-block
```

出力例:

```
[...]
storageClassDevices:
- devicePaths:
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441494EC
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84FE3E9
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE4
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441464EP
 # - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84F43E7
 # - /dev/disk/by-id/nvme-
```

```

Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE8
- /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS6F45C01D7E84FE3E9
- /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS636BC945B4ECB9AE4
 storageClassName: localblock
 volumeMode: Block
[...]

```

CR の編集後に変更を保存するようにしてください。

この CR に by-id を使用する 2 つの新規デバイスが追加されていることを確認できます。

- **nvme-Amazon\_EC2\_NVMe\_Instance\_Storage\_AWS6F45C01D7E84FE3E9**
- **nvme-Amazon\_EC2\_NVMe\_Instance\_Storage\_AWS636BC945B4ECB9AE4**

b. **localblock** と共に PV を表示します。

```
$ oc get pv | grep localblock
```

出力例:

```

local-pv-3646185e 2328Gi RWO Delete Available
localblock 9s
local-pv-3933e86 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-2-1-v9jp4 localblock 5h1m
local-pv-8176b2bf 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-0-0-nvs68 localblock 5h1m
local-pv-ab7cabb3 2328Gi RWO Delete Available
localblock 9s
local-pv-ac52e8a 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-1-0-knrgr localblock 5h1m
local-pv-b7e6fd37 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-2-0-rdm7m localblock 5h1m
local-pv-cb454338 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-0-1-h9hfm localblock 5h1m
local-pv-da5e3175 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-1-1-g97lq localblock 5h
...

```

15. 障害のあるノードに関連付けられたストレージリソースを削除します。

a. 置き換える OSD に関連付けられた DeviceSet を特定します。

```

$ osd_id_to_remove=0
$ oc get -n openshift-storage -o yaml deployment rook-ceph-osd-${osd_id_to_remove} |
grep ceph.rook.io/pvc

```

ここで、**osd\_id\_to\_remove** は **rook-ceph-osd** 接頭辞の直後にくる Pod 名の整数です。  
この例では、デプロイメント名は **rook-ceph-osd-0** です。

出力例:



```
ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68
ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68
```

- b. PVC に関連付けられた PV を特定します。

```
$ oc get -n openshift-storage pvc ocs-deviceset-<x>-<y>-<pvc-suffix>
```

ここで、**x**、**y**、および **pvc-suffix** は、前の手順で識別された DeviceSet の値です。

出力例:

| NAME                    | STATUS | VOLUME            | CAPACITY | ACCESS MODES   |
|-------------------------|--------|-------------------|----------|----------------|
| STORAGECLASS            | AGE    |                   |          |                |
| ocs-deviceset-0-0-nvs68 | Bound  | local-pv-8176b2bf | 2328Gi   | RWO localblock |
| 4h49m                   |        |                   |          |                |

この例では、関連付けられた PV は **local-pv-8176b2bf** です。

- c. **openshift-storage** プロジェクトを変更します。

```
$ oc project openshift-storage
```

- d. 失敗した OSD をクラスターから削除します。必要に応じて、複数の障害のある OSD を指定することができます。

```
$ oc process -n openshift-storage ocs-osd-removal -p
FAILED_OSD_IDS=${osd_id_to_remove} | oc create -f -
```

- e. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されることを確認します。**Completed** のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```



### 注記

**ocs-osd-removal-job** が失敗し、Pod が予想される **Completed** の状態にならない場合、追加のデバッグのために Pod ログを確認します。以下に例を示します。

```
oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```

- f. 先のステップで特定された PV を削除します。この例では、物理ボリューム名は **local-pv-8176b2bf** です。

```
$ oc delete pv local-pv-8176b2bf
```

出力例:

```
persistentvolume "local-pv-8176b2bf" deleted
```

16. 先の手順で特定された **crashcollector** Pod デプロイメントを削除します。

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=
<old_node_name> -n openshift-storage
```

17. **ocs-osd-removal-job** を削除します。

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

## 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。  
また、増分の **mon** が新規に作成されており、**Running** 状態にあることを確認します。

```
$ oc get pod -n openshift-storage | grep mon
```

出力例:

```
rook-ceph-mon-a-64556f7659-c2ngc 1/1 Running 0 5h1m
rook-ceph-mon-b-7c8b74dc4d-tt6hd 1/1 Running 0 5h1m
rook-ceph-mon-d-57fb8c657-wg5f2 1/1 Running 0 27m
```

OSD と mon が **Running** 状態になるまで数分かかる場合があります。

4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- 直前の手順で特定された新規ノードごとに、以下を実行します。
  - デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

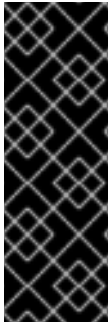
- ii. lsblk を実行し、**ocs-deviceset** 名の横にある crypt キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

### 2.3.2. インストーラーでプロビジョニングされるインフラストラクチャーで動作する Amazon EC2 ノードの置き換え

以下の手順を使用して、Amazon EC2 I3 のインストーラーでプロビジョニングされるインフラストラクチャー (IPI) で動作するノードを置き換えます。



#### 重要

Amazon EC2 I3 インフラストラクチャーのストレージノードの置き換えはテクノロジープレビュー機能です。テクノロジープレビュー機能は、Red Hat の実稼働環境のサービスレベルアグリーメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。Red Hat は実稼働環境でこれらを使用することを推奨していません。これらの機能は、近々発表予定の製品機能をリリースに先駆けてご提供することにより、お客様は機能性をテストし、開発プロセス中にフィードバックをお寄せいただくことができます。

#### 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャーおよびリソースで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOCP) クラスタにログインしている必要があります。

#### 手順

1. OpenShift Web コンソールにログインし、**Compute → Nodes** をクリックします。
2. 置き換える必要のあるノードを特定します。そのマシン名をメモします。
3. 置き換えるノードのラベルを取得します。

```
$ oc get nodes --show-labels | grep <node_name>
```

4. 置き換えるノードで実行されている mon (ある場合) および OSD を特定します。

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

5. 先の手順で特定された Pod のデプロイメントをスケールダウンします。  
以下に例を示します。

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

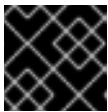
6. ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

7. ノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

8. **Compute → Machines** をクリックします。必要なマシンを検索します。
9. 必要なマシンの横にある **Action menu (⋮) → Delete Machine** をクリックします。
10. **Delete** をクリックしてマシンの削除を確認します。新しいマシンが自動的に作成されます。
11. 新規マシンが起動し、Running 状態に移行するまで待機します。



### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。

12. OpenShift Web コンソールで **Compute → Nodes** をクリックします。新規ノードが **Ready** 状態にあるかどうかを確認します。
13. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu (⋮) → Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

14. OpenShift ローカルストレージ Operator がインストールされている namespace を特定し、これを **local\_storage\_project** 変数に割り当てます。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

以下に例を示します。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

15. 新規ワーカーノードで利用可能なローカルストレージデバイスを OpenShift Container Storage StorageCluster に追加します。
  - a. 新規ディスクエントリーを LocalVolume CR に追加します。  
**LocalVolume** CR を編集します。障害のあるデバイス **/dev/disk/by-id/{id}** を削除またはコメントアウトし、新規の **/dev/disk/by-id/{id}** を追加します。

```
$ oc get -n $local_storage_project localvolume
```

出力例:

```
NAME AGE
local-block 25h
```

```
$ oc edit -n $local_storage_project localvolume local-block
```

出力例:

```
[...]
 storageClassDevices:
 - devicePaths:
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441494EC
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84FE3E9
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE4
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441464EP
 # - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84F43E7
 # - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE8
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS6F45C01D7E84FE3E9
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS636BC945B4ECB9AE4
 storageClassName: localblock
 volumeMode: Block
[...]
```

CR の編集後に変更を保存するようにしてください。

この CR に by-id を使用する 2 つの新規デバイスが追加されていることを確認できます。

- **nvme-Amazon\_EC2\_NVMe\_Instance\_Storage\_AWS6F45C01D7E84FE3E9**
- **nvme-Amazon\_EC2\_NVMe\_Instance\_Storage\_AWS636BC945B4ECB9AE4**

b. **localblock** と共に PV を表示します。

```
$ oc get pv | grep localblock
```

出力例:

```
local-pv-3646185e 2328Gi RWO Delete Available
localblock 9s
local-pv-3933e86 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-2-1-v9jp4 localblock 5h1m
local-pv-8176b2bf 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-0-0-nvs68 localblock 5h1m
```

```

local-pv-ab7cabb3 2328Gi RWO Delete Available
localblock 9s
local-pv-ac52e8a 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-1-0-knrgr localblock 5h1m
local-pv-b7e6fd37 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-2-0-rdm7m localblock 5h1m
local-pv-cb454338 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-0-1-h9hfm localblock 5h1m
local-pv-da5e3175 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-1-1-g97lq localblock 5h
...

```

16. 障害のあるノードに関連付けられたストレージリソースを削除します。

- a. 置き換える OSD に関連付けられた DeviceSet を特定します。

```

$ osd_id_to_remove=0
$ oc get -n openshift-storage -o yaml deployment rook-ceph-osd-${osd_id_to_remove} |
grep ceph.rook.io/pvc

```

ここで、**osd\_id\_to\_remove** は **rook-ceph-osd** 接頭辞の直後にくる Pod 名の整数です。  
この例では、デプロイメント名は **rook-ceph-osd-0** です。

出力例:

```

ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68
ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68

```

- b. PVC に関連付けられた PV を特定します。

```
$ oc get -n openshift-storage pvc ocs-deviceset-<x>-<y>-<pvc-suffix>
```

ここで、**x**、**y**、および **pvc-suffix** は、前の手順で識別された DeviceSet の値です。

出力例:

```

NAME STATUS VOLUME CAPACITY ACCESS MODES
STORAGECLASS AGE
ocs-deviceset-0-0-nvs68 Bound local-pv-8176b2bf 2328Gi RWO localblock
4h49m

```

この例では、関連付けられた PV は **local-pv-8176b2bf** です。

- c. **openshift-storage** プロジェクトを変更します。

```
$ oc project openshift-storage
```

- d. 失敗した OSD をクラスターから削除します。必要に応じて、複数の障害のある OSD を指定することができます。

```

$ oc process -n openshift-storage ocs-osd-removal -p
FAILED_OSD_IDS=${osd_id_to_remove} | oc create -f -

```

- e. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されることを確認します。**Completed** のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```



#### 注記

**ocs-osd-removal-job** が失敗し、Pod が予想される **Completed** の状態にならない場合、追加のデバッグのために Pod ログを確認します。以下に例を示します。

```
oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```

- f. 先のステップで特定された PV を削除します。この例では、物理ボリューム名は **local-pv-8176b2bf** です。

```
$ oc delete pv local-pv-8176b2bf
```

出力例:

```
persistentvolume "local-pv-8176b2bf" deleted
```

17. 先の手順で特定された **crashcollector** Pod デプロイメントを削除します。

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<old_node_name> -n openshift-storage
```

- a. **rook-ceph-operator** を削除します。

```
$ oc delete -n openshift-storage pod rook-ceph-operator-6f74fb5bff-2d982
```

出力例:

```
pod "rook-ceph-operator-6f74fb5bff-2d982" deleted
```

- b. **rook-ceph-operator** Pod が再起動していることを確認します。

```
$ oc get -n openshift-storage pod -l app=rook-ceph-operator
```

出力例:

| NAME                                | READY | STATUS  | RESTARTS | AGE |
|-------------------------------------|-------|---------|----------|-----|
| rook-ceph-operator-6f74fb5bff-7mvrq | 1/1   | Running | 0        | 66s |

新規 OSD の作成には、Operator が起動するまでに数分かかる場合があります。

18. **ocs-osd-removal-job** を削除します。

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

## 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。  
また、増分の **mon** が新規に作成されており、**Running** 状態にあることを確認します。

```
$ oc get pod -n openshift-storage | grep mon
```

出力例:

```
rook-ceph-mon-a-64556f7659-c2ngc 1/1 Running 0 5h1m
rook-ceph-mon-b-7c8b74dc4d-tt6hd 1/1 Running 0 5h1m
rook-ceph-mon-d-57fb8c657-wg5f2 1/1 Running 0 27m
```

OSD と mon が **Running** 状態になるまで数分かかる場合があります。

4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。
  - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. **lsblk** を実行し、**ocs-deviceset** 名の横にある **crypt** キーワードを確認します。

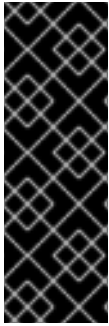
```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポート](#)にお問い合わせください。

### 2.3.3. ユーザーによってプロビジョニングされるインフラストラクチャーでの障害のある Amazon EC2 ノードの置き換え



OpenShift Container Storage の Amazon EC2 I3 の一時ストレージにより、インスタンスの電源がオフにされる場合にデータが失われる可能性があります。以下の手順を使用して、Amazon EC2 インフラストラクチャーでのインスタンスの電源オフからのリカバリーを行います。



## 重要

Amazon EC2 I3 インフラストラクチャーのストレージノードの置き換えはテクノロジープレビュー機能です。テクノロジープレビュー機能は、Red Hat の実稼働環境のサービスレベルアグリーメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。Red Hat は実稼働環境でこれらを使用することを推奨していません。これらの機能は、近々発表予定の製品機能をリリースに先駆けてご提供することにより、お客様は機能性をテストし、開発プロセス中にフィードバックをお寄せいただくことができます。

## 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャーおよびリソースで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスターにログインしている必要があります。

## 手順

1. ノードを特定し、置き換えるノードのラベルを取得します。

```
$ oc get nodes --show-labels | grep <node_name>
```

2. 置き換えるノードで実行されている mon (ある場合) および OSD を特定します。

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. 先の手順で特定された Pod のデプロイメントをスケールダウンします。  
以下に例を示します。

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

4. ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

5. Terminating 状態の Pod を削除します。

```
$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system ("oc -n " $1 " delete pods " $2 " --grace-period=0 " " --force ")}'
```

6. ノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

7. ノードを削除します。

```
$ oc delete node <node_name>
```

8. 必要なインフラストラクチャーで新規 Amazon EC2 I3 マシンインスタンスを作成します。[サポートされるインフラストラクチャーおよびプラットフォーム](#) について参照してください。
9. 新規 Amazon EC2 I3 マシンインスタンスを使用して新規 OpenShift Container Platform ノードを作成します。
10. Pending 状態の OpenShift Container Platform に関連する証明書署名要求 (CSR) の有無を確認します。

```
$ oc get csr
```

11. 新規ノードに必要なすべての OpenShift Container Platform CSR を承認します。

```
$ oc adm certificate approve <Certificate_Name>
```

12. OpenShift Web コンソールで **Compute → Nodes** をクリックします。新規ノードが **Ready** 状態にあるかどうかを確認します。
13. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu ( ⋮ ) → Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

14. OpenShift ローカルストレージ Operator がインストールされている namespace を特定し、これを **local\_storage\_project** 変数に割り当てます。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

以下に例を示します。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

15. 新規ワーカーノードで利用可能なローカルストレージデバイスを OpenShift Container Storage StorageCluster に追加します。
  - a. 新規ディスクエントリを LocalVolume CR に追加します。  
**LocalVolume** CR を編集します。障害のあるデバイス **/dev/disk/by-id/{id}** を削除またはコメントアウトし、新規の **/dev/disk/by-id/{id}** を追加します。

```
$ oc get -n $local_storage_project localvolume
```

出力例:

```
NAME AGE
local-block 25h
```

```
$ oc edit -n $local_storage_project localvolume local-block
```

出力例:

```
[...]
 storageClassDevices:
 - devicePaths:
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441494EC
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84FE3E9
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE4
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441464EP
 # - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84F43E7
 # - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE8
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS6F45C01D7E84FE3E9
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS636BC945B4ECB9AE4
 storageClassName: localblock
 volumeMode: Block
[...]
```

CR の編集後に変更を保存するようにしてください。

この CR に by-id を使用する 2 つの新規デバイスが追加されていることを確認できます。

- **nvme-Amazon\_EC2\_NVMe\_Instance\_Storage\_AWS6F45C01D7E84FE3E9**
- **nvme-Amazon\_EC2\_NVMe\_Instance\_Storage\_AWS636BC945B4ECB9AE4**

b. **localblock** と共に PV を表示します。

```
$ oc get pv | grep localblock
```

出力例:

```
local-pv-3646185e 2328Gi RWO Delete Available
localblock 9s
local-pv-3933e86 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-2-1-v9jp4 localblock 5h1m
local-pv-8176b2bf 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-0-0-nvs68 localblock 5h1m
```

```

local-pv-ab7cabb3 2328Gi RWO Delete Available
localblock 9s
local-pv-ac52e8a 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-1-0-knrgr localblock 5h1m
local-pv-b7e6fd37 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-2-0-rdm7m localblock 5h1m
local-pv-cb454338 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-0-1-h9hfm localblock 5h1m
local-pv-da5e3175 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-1-1-g97lq localblock 5h
...

```

16. 障害のあるノードに関連付けられたストレージリソースを削除します。

- a. 置き換える OSD に関連付けられた DeviceSet を特定します。

```

$ osd_id_to_remove=0
$ oc get -n openshift-storage -o yaml deployment rook-ceph-osd-${osd_id_to_remove} |
grep ceph.rook.io/pvc

```

ここで、**osd\_id\_to\_remove** は **rook-ceph-osd** 接頭辞の直後にくる Pod 名の整数です。  
この例では、デプロイメント名は **rook-ceph-osd-0** です。

出力例:

```

ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68
ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68

```

- b. PVC に関連付けられた PV を特定します。

```
$ oc get -n openshift-storage pvc ocs-deviceset-<x>-<y>-<pvc-suffix>
```

ここで、**x**、**y**、および **pvc-suffix** は、前の手順で識別された DeviceSet の値です。

出力例:

```

NAME STATUS VOLUME CAPACITY ACCESS MODES
STORAGECLASS AGE
ocs-deviceset-0-0-nvs68 Bound local-pv-8176b2bf 2328Gi RWO localblock
4h49m

```

この例では、関連付けられた PV は **local-pv-8176b2bf** です。

- c. **openshift-storage** プロジェクトに変更します。

```
$ oc project openshift-storage
```

- d. 失敗した OSD をクラスターから削除します。必要に応じて、複数の障害のある OSD を指定することができます。

```

$ oc process -n openshift-storage ocs-osd-removal -p
FAILED_OSD_IDS=${osd_ids_to_remove} | oc create -f -

```

... `oc process` は、`ocs-osd-removal` のコマンドを実行して、OSD が正常に削除されるのを

- e. **ocs-osd-removal-job** Pod のステータスをナエックして、OSD が正常に削除されることを確認します。**Completed** のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```



### 注記

**ocs-osd-removal-job** が失敗し、Pod が予想される **Completed** の状態にならない場合、追加のデバッグのために Pod ログを確認します。以下に例を示します。

```
oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```

- f. 先のステップで特定された PV を削除します。この例では、物理ボリューム名は **local-pv-8176b2bf** です。

```
$ oc delete pv local-pv-8176b2bf
```

出力例:

```
persistentvolume "local-pv-8176b2bf" deleted
```

17. 先の手順で特定された **crashcollector** Pod デプロイメントを削除します。

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<old_node_name> -n openshift-storage
```

18. **ocs-osd-removal-job** を削除します。

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

## 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads → Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。  
また、増分の **mon** が新規に作成されており、**Running** 状態にあることを確認します。

```
$ oc get pod -n openshift-storage | grep mon
```

出力例:

```
rook-ceph-mon-a-64556f7659-c2ngc 1/1 Running 0 5h1m
rook-ceph-mon-b-7c8b74dc4d-tt6hd 1/1 Running 0 5h1m
rook-ceph-mon-d-57fb8c657-wg5f2 1/1 Running 0 27m
```

OSD と mon が **Running** 状態になるまで数分かかる場合があります。

4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。

- i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

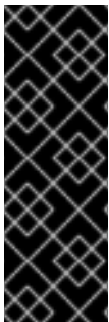
- ii. lsblk を実行し、**ocs-deviceset** 名の横にある crypt キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

### 2.3.4. インストーラーでプロビジョニングされるインフラストラクチャーでの失敗した Amazon EC2 ノードの置き換え

OpenShift Container Storage の Amazon EC2 I3 の一時ストレージにより、インスタンスの電源がオフにされる場合にデータが失われる可能性があります。以下の手順を使用して、Amazon EC2 インフラストラクチャーでのインスタンスの電源オフからのリカバリーを行います。



#### 重要

Amazon EC2 I3 インフラストラクチャーのストレージノードの置き換えはテクノロジープレビュー機能です。テクノロジープレビュー機能は、Red Hat の実稼働環境のサービスレベルアグリーメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。Red Hat は実稼働環境でこれらを使用することを推奨していません。これらの機能は、近々発表予定の製品機能をリリースに先駆けてご提供することにより、お客様は機能性をテストし、開発プロセス中にフィードバックをお寄せいただくことができます。

#### 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャーおよびリソースで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスタにログインしている必要があります。

## 手順

1. OpenShift Web コンソールにログインし、 **Compute → Nodes** をクリックします。
2. 置き換える必要のあるノードを特定します。そのマシン名をメモします。
3. 置き換えるノードのラベルを取得します。

```
$ oc get nodes --show-labels | grep <node_name>
```

4. 置き換えるノードで実行されている mon (ある場合) および OSD を特定します。

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

5. 先の手順で特定された Pod のデプロイメントをスケールダウンします。  
以下に例を示します。

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

6. ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

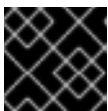
7. Terminating 状態の Pod を削除します。

```
$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system ("oc -
n " $1 " delete pods " $2 " --grace-period=0 " " --force ")}'
```

8. ノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

9. **Compute → Machines** をクリックします。必要なマシンを検索します。
10. 必要なマシンの横にある **Action menu (⋮) → Delete Machine** をクリックします。
11. **Delete** をクリックしてマシンの削除を確認します。新しいマシンが自動的に作成されます。
12. 新規マシンが起動し、Running 状態に移行するまで待機します。



### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。

13. OpenShift Web コンソールで **Compute → Nodes** をクリックします。新規ノードが **Ready** 状態にあるかどうかを確認します。
14. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

**ユーザーインターフェイスを使用する場合**

- a. 新規ノードについて、**Action Menu ( ⋮ )** → **Edit Labels**をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

## コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

- OpenShift ローカルストレージ Operator がインストールされている namespace を特定し、これを **local\_storage\_project** 変数に割り当てます。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

以下に例を示します。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

- 新規ワーカーノードで利用可能なローカルストレージデバイスを OpenShift Container Storage StorageCluster に追加します。

- a. 新規ディスクエントリーを LocalVolume CR に追加します。

**LocalVolume** CR を編集します。障害のあるデバイス **/dev/disk/by-id/{id}** を削除またはコメントアウトし、新規の **/dev/disk/by-id/{id}** を追加します。

```
$ oc get -n $local_storage_project localvolume
```

出力例:

```
NAME AGE
local-block 25h
```

```
$ oc edit -n $local_storage_project localvolume local-block
```

出力例:

```
[...]
storageClassDevices:
- devicePaths:
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441494EC
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84FE3E9
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE4
 - /dev/disk/by-id/nvme-
 Amazon_EC2_NVMe_Instance_Storage_AWS10382E5D7441464EP
- /dev/disk/by-id/nvme-
```



```

Amazon_EC2_NVMe_Instance_Storage_AWS1F45C01D7E84F43E7
- /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS136BC945B4ECB9AE8
- /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS6F45C01D7E84FE3E9
- /dev/disk/by-id/nvme-
Amazon_EC2_NVMe_Instance_Storage_AWS636BC945B4ECB9AE4
storageClassName: localblock
volumeMode: Block
[...]

```

CR の編集後に変更を保存するようにしてください。

この CR に by-id を使用する 2 つの新規デバイスが追加されていることを確認できます。

- **nvme-Amazon\_EC2\_NVMe\_Instance\_Storage\_AWS6F45C01D7E84FE3E9**
- **nvme-Amazon\_EC2\_NVMe\_Instance\_Storage\_AWS636BC945B4ECB9AE4**

b. **localblock** と共に PV を表示します。

```
$ oc get pv | grep localblock
```

出力例:

```

local-pv-3646185e 2328Gi RWO Delete Available
localblock 9s
local-pv-3933e86 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-2-1-v9jp4 localblock 5h1m
local-pv-8176b2bf 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-0-0-nvs68 localblock 5h1m
local-pv-ab7cabb3 2328Gi RWO Delete Available
localblock 9s
local-pv-ac52e8a 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-1-0-knrgr localblock 5h1m
local-pv-b7e6fd37 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-2-0-rdm7m localblock 5h1m
local-pv-cb454338 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-0-1-h9hfm localblock 5h1m
local-pv-da5e3175 2328Gi RWO Delete Bound openshift-storage/ocs-
deviceset-1-1-g97lq localblock 5h
...

```

17. 障害のあるノードに関連付けられたストレージリソースを削除します。

a. 置き換える OSD に関連付けられた DeviceSet を特定します。

```

$ osd_id_to_remove=0
$ oc get -n openshift-storage -o yaml deployment rook-ceph-osd-${osd_id_to_remove} |
grep ceph.rook.io/pvc

```

ここで、**osd\_id\_to\_remove** は **rook-ceph-osd** 接頭辞の直後にくる Pod 名の整数です。  
この例では、デプロイメント名は **rook-ceph-osd-0** です。

出力例:

-

```
ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68
ceph.rook.io/pvc: ocs-deviceset-0-0-nvs68
```

- b. PVC に関連付けられた PV を特定します。

```
$ oc get -n openshift-storage pvc ocs-deviceset-<x>-<y>-<pvc-suffix>
```

ここで、**x**、**y**、および **pvc-suffix** は、前の手順で識別された DeviceSet の値です。

出力例:

| NAME                    | STATUS | VOLUME            | CAPACITY | ACCESS MODES   |
|-------------------------|--------|-------------------|----------|----------------|
| STORAGECLASS            | AGE    |                   |          |                |
| ocs-deviceset-0-0-nvs68 | Bound  | local-pv-8176b2bf | 2328Gi   | RWO localblock |
| 4h49m                   |        |                   |          |                |

この例では、関連付けられた PV は **local-pv-8176b2bf** です。

- c. **openshift-storage** プロジェクトに変更します。

```
$ oc project openshift-storage
```

- d. 失敗した OSD をクラスターから削除します。必要に応じて、複数の障害のある OSD を指定することができます。

```
$ oc process -n openshift-storage ocs-osd-removal -p
FAILED_OSD_IDS=${osd_ids_to_remove} | oc create -f -
```

- e. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されることを確認します。**Completed** のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```



### 注記

**ocs-osd-removal-job** が失敗し、Pod が予想される **Completed** の状態にならない場合、追加のデバッグのために Pod ログを確認します。以下に例を示します。

```
oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```

- f. 先のステップで特定された PV を削除します。この例では、物理ボリューム名は **local-pv-8176b2bf** です。

```
$ oc delete pv local-pv-8176b2bf
```

出力例:

```
persistentvolume "local-pv-8176b2bf" deleted
```

18. 先の手順で特定された **crashcollector** Pod デプロイメントを削除します。

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=
<old_node_name> -n openshift-storage
```

19. **ocs-osd-removal-job** を削除します。

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

## 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads → Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。  
また、増分の **mon** が新規に作成されており、**Running** 状態にあることを確認します。

```
$ oc get pod -n openshift-storage | grep mon
```

出力例:

```
rook-ceph-mon-a-64556f7659-c2ngc 1/1 Running 0 5h1m
rook-ceph-mon-b-7c8b74dc4d-tt6hd 1/1 Running 0 5h1m
rook-ceph-mon-d-57fb8c657-wg5f2 1/1 Running 0 27m
```

OSD と mon が **Running** 状態になるまで数分かかる場合があります。

4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- 直前の手順で特定された新規ノードごとに、以下を実行します。
  - デバッグ Pod を作成し、選択したホストの **chroot** 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. `lsblk` を実行し、**ocs-deviceset** 名の横にある `crypt` キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポート](#)にお問い合わせください。

## 2.4. VMWARE インフラストラクチャーでのストレージノードの置き換え

- 動作するノードを置き換えるには、以下を参照してください。
  - [「VMware のユーザーによってプロビジョニングされるインフラストラクチャーで動作するノードの置き換え」](#)
  - [「VMware のインストーラーでプロビジョニングされるインフラストラクチャーで動作するノードの置き換え」](#)
- 障害のあるノードを置き換えるには、以下を参照してください。
  - [「VMware ユーザーによってプロビジョニングされるインフラストラクチャーでの障害のあるノードの置き換え」](#)
  - [「VMware のインストーラーでプロビジョニングされるインフラストラクチャーで障害のあるノードの置き換え」](#)

### 2.4.1. VMware のユーザーによってプロビジョニングされるインフラストラクチャーで動作するノードの置き換え

#### 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャー、リソースおよびディスクで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスターにログインしている必要があります。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、デバイスの自動プロビジョニングを有効にするために **LocalVolumeSet** オブジェクトを作成していない場合は、[ローカルストレージでサポートされるクラスターの更新後の設定の変更](#) についての以下の手順に従って、これを実行します。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、**LocalVolumeDiscovery** オブジェクトを作成していない場合は、[ローカルストレージでサポートされるクラスターの更新後の設定の変更](#) についての以下の手順に従って、これを実行します。

#### 手順

1. NODE を特定し、置き換えるノードのラベルを取得します。

```
$ oc get nodes --show-labels | grep <node_name>
```

2. 置き換えるノードで実行されている **mon** (ある場合) および OSD を特定します。

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. 先の手順で特定された Pod のデプロイメントをスケールダウンします。  
以下に例を示します。

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

4. ノードにスケジューリング対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

5. ノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

6. ノードを削除します。

```
$ oc delete node <node_name>
```

7. VSphere にログインし、特定された仮想マシンを終了します。

8. 必要なインフラストラクチャーで VMware に新規の仮想マシンを作成します。[サポートされるインフラストラクチャーおよびプラットフォーム](#) について参照してください。

9. 新規の仮想マシンを使用して新規 OpenShift Container Platform ワーカーノードを作成します。

10. Pending 状態の OpenShift Container Platform に関連する証明書署名要求 (CSR) の有無を確認します。

```
$ oc get csr
```

11. 新規ノードに必要なすべての OpenShift Container Platform CSR を承認します。

```
$ oc adm certificate approve <Certificate_Name>
```

12. OpenShift Web コンソールで **Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあるかどうかを確認します。

13. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu (⋮) → Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

14. OpenShift ローカルストレージ Operator がインストールされている namespace を特定し、これを **local\_storage\_project** 変数に割り当てます。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

以下に例を示します。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

15. 新規ワーカーノードを **localVolumeDiscovery** および **localVolumeSet** に追加します。

- a. **localVolumeDiscovery** 定義を更新し、新規ノードを追加して失敗したノードを削除します。

```
oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

- b. 編集する **localVolumeSet** を決定します。

```
oc get -n $local_storage_project localvolumeset
NAME AGE
localblock 25h
```

- c. **localVolumeSet** 定義を更新して、新規ノードを追加し、障害が発生したノードを削除します。

```
oc edit -n $local_storage_project localvolumeset localblock
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
```

```
#- server3.example.com
- newnode.example.com

[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

16. 新規 **localblock** PV が利用可能であることを確認します。

```
$oc get pv | grep localblock | grep Available
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

17. **openshift-storage** プロジェクトを変更します。

```
$ oc project openshift-storage
```

18. 失敗した OSD をクラスターから削除します。必要に応じて、複数の障害のある OSD を指定することができます。

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> FORCE_OSD_REMOVAL=false | oc create -n
openshift-storage -f -
```

**<failed\_osd\_id>**

**rook-ceph-osd** 接頭辞の直後の Pod 名の整数です。コマンドにコンマ区切りの OSD ID を追加して、複数の OSD を削除できます (例: **FAILED\_OSD\_IDS=0,1,2**)

OSD が 3 つしかないクラスター、または OSD が削除された後にデータの 3 つのレプリカすべてを復元するにはスペースが不十分なクラスターでは、**FORCE\_OSD\_REMOVAL** 値を **true** に変更する必要があります。

19. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されたことを確認します。

**Completed** のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```



### 注記

**ocs-osd-removal-job** が失敗し、Pod が予想される **Completed** の状態にない場合、追加のデバッグのために Pod ログを確認します。以下に例を示します。

```
oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```

20. **ocs-osd-removal-job** を削除します。

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

## 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d ' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が Running 状態にあることを確認します。  
また、増分の **mon** が新規に作成されており、Running 状態にあることを確認します。

```
$ oc get pod -n openshift-storage | grep mon
```

出力例:

```
rook-ceph-mon-a-cd575c89b-b6k66 2/2 Running
0 38m
rook-ceph-mon-b-6776bc469b-tzzt8 2/2 Running
0 38m
rook-ceph-mon-d-5ff5d488b5-7v8xh 2/2 Running
0 4m8s
```

OSD と Mon が **Running** 状態になるまで数分かかる場合があります。

4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。
  - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. `lsblk` を実行し、**ocs-deviceset** 名の横にある `crypt` キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポート](#)にお問い合わせください。



## 2.4.2. VMware のインストーラーでプロビジョニングされるインフラストラクチャーで動作するノードの置き換え

### 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャー、リソースおよびディスクで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスターにログインしている必要があります。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、デバイスの自動プロビジョニングを有効にするために **LocalVolumeSet** オブジェクトを作成していない場合は、[ローカルストレージでサポートされるクラスターの更新後の設定の変更](#) についての以下の手順に従って、これを実行します。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、**LocalVolumeDiscovery** オブジェクトを作成していない場合は、[ローカルストレージでサポートされるクラスターの更新後の設定の変更](#) についての以下の手順に従って、これを実行します。

### 手順

- OpenShift Web コンソールにログインし、**Compute → Nodes** をクリックします。
- 置き換える必要のあるノードを特定します。その **マシン名** をメモします。
- 置き換えるノードのラベルを取得します。

```
$ oc get nodes --show-labels | grep <node_name>
```

- 置き換えるノードで実行されている **mon** (ある場合) および **OSD** を特定します。

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

- 先の手順で特定された Pod のデプロイメントをスケールダウンします。  
以下に例を示します。

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

- ノードにスケジュール対象外 (unschedulable) のマークを付けます。

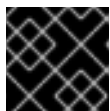
```
$ oc adm cordon <node_name>
```

- ノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

- Compute → Machines** をクリックします。必要なマシンを検索します。
- 必要なマシンの横にある **Action menu (⋮) → Delete Machine** をクリックします。

10. **Delete** をクリックしてマシンの削除を確認します。新しいマシンが自動的に作成されます。
11. 新規マシンが起動し、**Running** 状態に移行するまで待機します。



### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。

12. OpenShift Web コンソールで **Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあるかどうかを確認します。
13. 物理的に新規デバイスをノードに追加します。
14. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu ( ⋮ ) → Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

15. OpenShift ローカルストレージ Operator がインストールされている namespace を特定し、これを **local\_storage\_project** 変数に割り当てます。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

以下に例を示します。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

16. 新規ワーカーノードを **localVolumeDiscovery** および **localVolumeSet** に追加します。
  - a. **localVolumeDiscovery** 定義を更新し、新規ノードを追加して失敗したノードを削除します。

```
oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
```

```
#- server3.example.com
- newnode.example.com

[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

- b. 編集する **localVolumeSet** を決定します。

```
oc get -n $local_storage_project localvolumeset
NAME AGE
localblock 25h
```

- c. **localVolumeSet** 定義を更新して、新規ノードを追加し、障害が発生したノードを削除します。

```
oc edit -n $local_storage_project localvolumeset localblock
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
 operator: In
 values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com

[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

17. 新規 **localblock** PV が利用可能であることを確認します。

```
$oc get pv | grep localblock | grep Available
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

18. **openshift-storage** プロジェクトを変更します。

```
$ oc project openshift-storage
```

19. 失敗した OSD をクラスターから削除します。必要に応じて、複数の障害のある OSD を指定することができます。

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> FORCE_OSD_REMOVAL=false | oc create -n
openshift-storage -f -
```

## &lt;failed\_osd\_id&gt;

**rook-ceph-osd** 接頭辞の直後の Pod 名の整数です。コマンドにコンマ区切りの OSD ID を追加して、複数の OSD を削除できます (例: **FAILED\_OSD\_IDS=0,1,2**)

OSD が 3 つしかないクラスター、または OSD が削除された後にデータの 3 つのレプリカすべてを復元するにはスペースが不十分なクラスターでは、**FORCE\_OSD\_REMOVAL** 値を **true** に変更する必要があります。

20. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されたことを確認します。

**Completed** のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```



## 注記

**ocs-osd-removal-job** が失敗し、Pod が予想される **Completed** の状態にない場合、追加のデバッグのために Pod ログを確認します。以下に例を示します。

```
oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```

21. PVC に関連付けられた PV を特定します。

```
#oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h compute-1
```

**Released** 状態の PV がある場合は、これを削除します。

```
oc delete pv <persistent-volume>
```

以下に例を示します。

```
#oc delete pv local-pv-d6bf175b
persistentvolume "local-pv-d9c5cbd6" deleted
```

22. **crashcollector** Pod デプロイメントを特定します。

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage
```

既存の **crashcollector** Pod デプロイメントがある場合は、これを削除します。

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage
```

23. **ocs-osd-removal-job** を削除します。

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

## 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d ' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が Running 状態にあることを確認します。  
また、増分の **mon** が新規に作成されており、Running 状態にあることを確認します。

```
$ oc get pod -n openshift-storage | grep mon
```

出力例:

```
rook-ceph-mon-a-cd575c89b-b6k66 2/2 Running
0 38m
rook-ceph-mon-b-6776bc469b-tzzt8 2/2 Running
0 38m
rook-ceph-mon-d-5ff5d488b5-7v8xh 2/2 Running
0 4m8s
```

OSD と Mon が **Running** 状態になるまで数分かかる場合があります。

4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。
  - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. `lsblk` を実行し、**ocs-deviceset** 名の横にある `crypt` キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポート](#)にお問い合わせください。

### 2.4.3. VMware ユーザーによってプロビジョニングされるインフラストラクチャーでの障害のあるノードの置き換え

#### 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャー、リソースおよびディスクで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスターにログインしている必要があります。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、デバイスの自動プロビジョニングを有効にするために **LocalVolumeSet** オブジェクトを作成していない場合は、[ローカルストレージでサポートされるクラスターの更新後の設定の変更](#) についての以下の手順に従って、これを実行します。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、**LocalVolumeDiscovery** オブジェクトを作成していない場合は、[ローカルストレージでサポートされるクラスターの更新後の設定の変更](#) についての以下の手順に従って、これを実行します。

#### 手順

1. NODE を特定し、置き換えるノードのラベルを取得します。

```
$ oc get nodes --show-labels | grep <node_name>
```

2. 置き換えるノードで実行されている **mon** (ある場合) および OSD を特定します。

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. 先の手順で特定された Pod のデプロイメントをスケールダウンします。  
以下に例を示します。

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

4. ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

5. Terminating 状態の Pod を削除します。

```
$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system ("oc -n "$1" delete pods "$2 --grace-period=0 " " --force")}'
```

6. ノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

7. ノードを削除します。

```
$ oc delete node <node_name>
```

8. VSphere にログインし、特定された仮想マシンを終了します。
9. 必要なインフラストラクチャーで VMware に新規の仮想マシンを作成します。[サポートされるインフラストラクチャーおよびプラットフォーム](#) について参照してください。
10. 新規の仮想マシンを使用して新規 OpenShift Container Platform ワーカーノードを作成します。
11. Pending 状態の OpenShift Container Platform に関連する証明書署名要求 (CSR) の有無を確認します。

```
$ oc get csr
```

12. 新規ノードに必要なすべての OpenShift Container Platform CSR を承認します。

```
$ oc adm certificate approve <Certificate_Name>
```

13. OpenShift Web コンソールで **Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあるかどうかを確認します。
14. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu (⋮) → Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

15. OpenShift ローカルストレージ Operator がインストールされている namespace を特定し、これを **local\_storage\_project** 変数に割り当てます。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

以下に例を示します。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

16. 新規ワーカーノードを **localVolumeDiscovery** および **localVolumeSet** に追加します。
  - a. **localVolumeDiscovery** 定義を更新し、新規ノードを追加して失敗したノードを削除します。

```
oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

- b. 編集する **localVolumeSet** を決定します。

```
oc get -n $local_storage_project localvolumeset
NAME AGE
localblock 25h
```

- c. **localVolumeSet** 定義を更新して、新規ノードを追加し、障害が発生したノードを削除します。

```
oc edit -n $local_storage_project localvolumeset localblock
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

17. 新規 **localblock** PV が利用可能であることを確認します。

```
$oc get pv | grep localblock | grep Available
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

18. **openshift-storage** プロジェクトを変更します。



```
$ oc project openshift-storage
```

19. 失敗した OSD をクラスターから削除します。必要に応じて、複数の障害のある OSD を指定することができます。

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> FORCE_OSD_REMOVAL=false | oc create -n
openshift-storage -f -
```

<failed\_osd\_id>

**rook-ceph-osd** 接頭辞の直後の Pod 名の整数です。コマンドにコンマ区切りの OSD ID を追加して、複数の OSD を削除できます (例: **FAILED\_OSD\_IDS=0,1,2**)

OSD が 3 つしかないクラスター、または OSD が削除された後にデータの 3 つのレプリカすべてを復元するにはスペースが不十分なクラスターでは、**FORCE\_OSD\_REMOVAL** 値を **true** に変更する必要があります。

20. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されたことを確認します。

**Completed** のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```



#### 注記

**ocs-osd-removal-job** が失敗し、Pod が予想される **Completed** の状態にない場合、追加のデバッグのために Pod ログを確認します。以下に例を示します。

```
oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```

21. **ocs-osd-removal-job** を削除します。

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

#### 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が Running 状態にあることを確認します。  
また、増分の **mon** が新規に作成されており、Running 状態にあることを確認します。

```
$ oc get pod -n openshift-storage | grep mon
```

出力例:

```
rook-ceph-mon-a-cd575c89b-b6k66 2/2 Running
0 38m
rook-ceph-mon-b-6776bc469b-tzzt8 2/2 Running
0 38m
rook-ceph-mon-d-5ff5d488b5-7v8xh 2/2 Running
0 4m8s
```

OSD と Mon が **Running** 状態になるまで数分かかる場合があります。

4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。
  - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. `lsblk` を実行し、**ocs-deviceset** 名の横にある `crypt` キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

#### 2.4.4. VMware のインストーラーでプロビジョニングされるインフラストラクチャーで障害のあるノードの置き換え

##### 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャー、リソースおよびディスクで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスタにログインしている必要があります。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、デバイスの自動プロビジョニングを有効にするために **LocalVolumeSet** オブジェクトを作成していない場合は、[ローカルストレージでサポートされるクラスタの更新後の設定の変更](#) についての以下の手順に従って、これを実行します。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、**LocalVolumeDiscovery** オブジェクトを作成していない場合は、[ローカルストレージでサ](#)

ポートされるクラスターの更新後の設定の変更 についての以下の手順に従って、これを実行します。

## 手順

1. OpenShift Web コンソールにログインし、 **Compute → Nodes** をクリックします。
2. 置き換える必要のあるノードを特定します。その **マシン名** をメモします。
3. 置き換えるノードのラベルを取得します。

```
$ oc get nodes --show-labels | grep <node_name>
```

4. 置き換えるノードで実行されている **mon** (ある場合) および **OSD** を特定します。

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

5. 先の手順で特定された Pod のデプロイメントをスケールダウンします。  
以下に例を示します。

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

6. ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

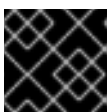
7. Terminating 状態の Pod を削除します。

```
$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system ("oc -
n " $1 " delete pods " $2 " --grace-period=0 " " --force ")}'
```

8. ノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

9. **Compute → Machines** をクリックします。必要なマシンを検索します。
10. 必要なマシンの横にある **Action menu (⋮) → Delete Machine** をクリックします。
11. **Delete** をクリックしてマシンの削除を確認します。新しいマシンが自動的に作成されます。
12. 新規マシンが起動し、**Running** 状態に移行するまで待機します。



### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。

13. OpenShift Web コンソールで **Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあるかどうかを確認します。

14. 物理的に新規デバイスをノードに追加します。
15. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu ( ⋮ )** → **Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

16. OpenShift ローカルストレージ Operator がインストールされている namespace を特定し、これを **local\_storage\_project** 変数に割り当てます。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

以下に例を示します。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

17. 新規ワーカーノードを **localVolumeDiscovery** および **localVolumeSet** に追加します。
  - a. **localVolumeDiscovery** 定義を更新し、新規ノードを追加して失敗したノードを削除します。

```
oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
 nodeSelectorTerms:
 - matchExpressions:
 - key: kubernetes.io/hostname
 operator: In
 values:
 - server1.example.com
 - server2.example.com
 #- server3.example.com
 - newnode.example.com
[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

- b. 編集する **localVolumeSet** を決定します。

```
oc get -n $local_storage_project localvolumeset
NAME AGE
localblock 25h
```

- c. **localVolumeSet** 定義を更新して、新規ノードを追加し、障害が発生したノードを削除します。

```
oc edit -n $local_storage_project localvolumeset localblock
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

18. 新規 **localblock** PV が利用可能であることを確認します。

```
$oc get pv | grep localblock | grep Available
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

19. **openshift-storage** プロジェクトを変更します。

```
$ oc project openshift-storage
```

20. 失敗した OSD をクラスターから削除します。必要に応じて、複数の障害のある OSD を指定することができます。

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> FORCE_OSD_REMOVAL=false | oc create -n
openshift-storage -f -
```

<failed\_osd\_id>

**rook-ceph-osd** 接頭辞の直後の Pod 名の整数です。コマンドにコンマ区切りの OSD ID を追加して、複数の OSD を削除できます (例: **FAILED\_OSD\_IDS=0,1,2**)

OSD が 3 つしかないクラスター、または OSD が削除された後にデータの 3 つのレプリカすべてを復元するにはスペースが不十分なクラスターでは、**FORCE\_OSD\_REMOVAL** 値を **true** に変更する必要があります。

21. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されたことを確認します。

**Completed** のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```



### 注記

**ocs-osd-removal-job** が失敗し、Pod が予想される **Completed** の状態にない場合、追加のデバッグのために Pod ログを確認します。以下に例を示します。

```
oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```

22. PVC に関連付けられた PV を特定します。

```
#oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h compute-1
```

**Released** 状態の PV がある場合は、これを削除します。

```
oc delete pv <persistent-volume>
```

以下に例を示します。

```
#oc delete pv local-pv-d6bf175b
persistentvolume "local-pv-d9c5cbd6" deleted
```

23. **crashcollector** Pod デプロイメントを特定します。

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage
```

既存の **crashcollector** Pod デプロイメントがある場合は、これを削除します。

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage
```

24. **ocs-osd-removal-job** を削除します。

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

## 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
  - **csi-rbdplugin-\***
3. 他の必要なすべての OpenShift Container Storage Pod が Running 状態にあることを確認します。  
また、増分の **mon** が新規に作成されており、Running 状態にあることを確認します。

```
$ oc get pod -n openshift-storage | grep mon
```

出力例:

```
rook-ceph-mon-a-cd575c89b-b6k66 2/2 Running
0 38m
rook-ceph-mon-b-6776bc469b-tzzt8 2/2 Running
0 38m
rook-ceph-mon-d-5ff5d488b5-7v8xh 2/2 Running
0 4m8s
```

OSD と Mon が **Running** 状態になるまで数分かかる場合があります。

4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。
  - a. 直前の手順で特定された新規ノードごとに、以下を実行します。
    - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. lsblk を実行し、**ocs-deviceset** 名の横にある crypt キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

## 2.5. RED HAT VIRTUALIZATION インフラストラクチャーでのストレージノードの置き換え

- 動作するノードを置き換えるには、「[Red Hat Virtualization インストーラーでプロビジョニングされるインフラストラクチャーで動作するノードの置き換え](#)」を参照してください。
- 障害のあるノードを置き換えるには、「[Red Hat Virtualization インストーラーでプロビジョニングされるインフラストラクチャーで障害のあるノードの置き換え](#)」を参照してください。

### 2.5.1. Red Hat Virtualization インストーラーでプロビジョニングされるインフラストラクチャーで動作するノードの置き換え

以下の手順を使用して、Red Hat Virtualization のインストーラーでプロビジョニングされるインフラストラクチャー (IPI) で動作するノードを置き換えます。

## 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャー、リソース、およびディスクで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスターにログインしている必要があります。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、デバイスの自動プロビジョニングを有効にするために **LocalVolumeSet** オブジェクトを作成していない場合は、[ローカルストレージでサポートされるクラスターの更新後の設定の変更](#) の手順に従って、いますぐそれを行うことができます。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、**LocalVolumeDiscovery** オブジェクトを作成していない場合は、[ローカルストレージでサポートされるクラスターの更新後の設定の変更](#) の手順に従って、いますぐそれを行うことができます。

## 手順

- OpenShift Web コンソールにログインし、**Compute → Nodes** をクリックします。
- 置き換える必要のあるノードを特定します。その **マシン名** をメモします。
- 置き換えるノードのラベルを取得します。

```
$ oc get nodes --show-labels | grep <node_name>
```

- 置き換えるノードで実行されている mon (ある場合) および OSD を特定します。

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

- 先の手順で特定された Pod のデプロイメントをスケールダウンします。以下に例を示します。

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

- ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

- ノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

- Compute → Machines** をクリックします。必要なマシンを検索します。
- 必要なマシンの横にある **Action menu (⋮) → Delete Machine** をクリックします。



10. **Delete** をクリックしてマシンの削除を確認します。新しいマシンが自動的に作成されます。新規マシンが起動し、Running 状態に移行するまで待機します。



### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。

11. OpenShift Web コンソールで **Compute → Nodes** をクリックします。新規ノードが **Ready** 状態にあるかどうかを確認します。
12. 物理的に新しいデバイスをノードに追加します。
13. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu ( ⋮ ) → Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

14. OpenShift ローカルストレージ Operator がインストールされている namespace を特定し、これを **local\_storage\_project** 変数に割り当てます。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

以下に例を示します。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

15. 新規ワーカーノードを **localVolumeDiscovery** および **localVolumeSet** に追加します。
  - a. **localVolumeDiscovery** 定義を更新し、新規ノードを追加して失敗したノードを削除します。

```
oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
```

```
#- server3.example.com
- newnode.example.com

[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

- b. 編集する **localVolumeSet** を決定します。

```
oc get -n $local_storage_project localvolumeset
NAME AGE
localblock 25h
```

- c. **localVolumeSet** 定義を更新して、新規ノードを追加し、障害が発生したノードを削除します。

```
oc edit -n $local_storage_project localvolumeset localblock
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
 operator: In
 values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com

[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

16. 新規 **localblock** PV が利用可能であることを確認します。

```
$oc get pv | grep localblock | grep Available
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

17. **openshift-storage** プロジェクトを変更します。

```
$ oc project openshift-storage
```

18. 失敗した OSD をクラスターから削除します。必要に応じて、複数の障害のある OSD を指定することができます。

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=_<failed_osd_id>_ FORCE_OSD_REMOVAL=false | oc create -n
openshift-storage -f -
```

## &lt;failed\_osd\_id&gt;

**rook-ceph-osd** 接頭辞の直後の Pod 名の整数です。コマンドにコンマ区切りの OSD ID を追加して、複数の OSD を削除できます (例: **FAILED\_OSD\_IDS=0,1,2**)

OSD が 3 つしかないクラスター、または OSD が削除された後にデータの 3 つのレプリカすべてを復元するにはスペースが不十分なクラスターでは、**FORCE\_OSD\_REMOVAL** 値を **true** に変更する必要があります。

19. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されたことを確認します。

**Completed** のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```



## 注記

**ocs-osd-removal-job** が失敗し、Pod が予想される **Completed** の状態にない場合、追加のデバッグのために Pod ログを確認します。以下に例を示します。

```
oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```

20. PVC に関連付けられた PV を特定します。

```
oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
local-pv-d6bf175b 512Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h server3.example.com
```

**Released** 状態の PV がある場合は、これを削除します。

```
oc delete pv <persistent-volume>
```

以下に例を示します。

```
oc delete pv local-pv-d6bf175b
persistentvolume "local-pv-d6bf175b" deleted
```

21. **crashcollector** Pod デプロイメントを特定します。

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage
```

既存の **crashcollector** Pod がある場合は、これを削除します。

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name -n openshift-storage
```

22. **ocs-osd-removal** ジョブを削除します。

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

## 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d ' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が Running 状態にあることを確認します。  
また、増分の **mon** が新規に作成されており、Running 状態にあることを確認します。

```
$ oc get pod -n openshift-storage | grep mon
```

出力例:

```
rook-ceph-mon-a-cd575c89b-b6k66 2/2 Running 0 38m
rook-ceph-mon-b-6776bc469b-tzxt8 2/2 Running 0 38m
rook-ceph-mon-d-5ff5d488b5-7v8xh 2/2 Running 0 4m8s
```

OSD と Mon が **Running** 状態になるまで数分かかる場合があります。

4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。
  - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. `lsblk` を実行し、**ocs-deviceset** 名の横にある `crypt` キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

## 2.5.2. Red Hat Virtualization インストーラーでプロビジョニングされるインフラストラクチャーで障害のあるノードの置き換え

以下の手順に従って、OpenShift Container Storage の Red Hat Virtualization のインストーラーでプロビジョニングされるインフラストラクチャー (IPI) で動作しない障害のあるノードを置き換えます。

## 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャー、リソース、およびディスクで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスタにログインしている必要があります。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、デバイスの自動プロビジョニングを有効にするために **LocalVolumeSet** オブジェクトを作成していない場合は、[ローカルストレージでサポートされるクラスタの更新後の設定の変更](#) の手順に従って、いますぐそれを行うことができます。
- 以前のバージョンから OpenShift Container Storage 4.7 にアップグレードし、**LocalVolumeDiscovery** オブジェクトを作成していない場合は、[ローカルストレージでサポートされるクラスタの更新後の設定の変更](#) の手順に従って、いますぐそれを行うことができます。

## 手順

- OpenShift Web コンソールにログインし、**Compute → Nodes** をクリックします。
- 置き換える必要のあるノードを特定します。その **マシン名** をメモします。
- 置き換えるノードのラベルを取得します。

```
$ oc get nodes --show-labels | grep <node_name>
```

- 置き換えるノードで実行されている mon (ある場合) および OSD を特定します。

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

- 先の手順で特定された Pod のデプロイメントをスケールダウンします。  
以下に例を示します。

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

- ノードにスケジュール対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

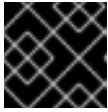
- Terminating** 状態の Pod を削除します。

```
$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system ("oc -n " $1 " delete pods " $2 " --grace-period=0 " " --force ")}'
```

- ノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

9. **Compute → Machines** をクリックします。必要なマシンを検索します。
10. 必要なマシンの横にある **Action menu ( ⋮ ) → Delete Machine** をクリックします。
11. **Delete** をクリックしてマシンの削除を確認します。新しいマシンが自動的に作成されます。新規マシンが起動し、Running 状態に移行するまで待機します。



### 重要

このアクティビティーには少なくとも 5-10 分以上かかる場合があります。

12. OpenShift Web コンソールで **Compute → Nodes** をクリックします。新規ノードが Ready 状態にあるかどうかを確認します。
13. 物理的に新しいデバイスをノードに追加します。
14. 以下のいずれかを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu ( ⋮ ) → Edit Labels** をクリックします。
- b. `cluster.ocs.openshift.io/openshift-storage` を追加し、**Save** をクリックします。

### コマンドラインインターフェイスの使用

- 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

15. OpenShift ローカルストレージ Operator がインストールされている namespace を特定し、これを **local\_storage\_project** 変数に割り当てます。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

以下に例を示します。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

16. 新規ワーカーノードを **localVolumeDiscovery** および **localVolumeSet** に追加します。
  - a. **localVolumeDiscovery** 定義を更新し、新規ノードを追加して失敗したノードを削除します。

```
oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
```

```

values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com

[...]

```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

- b. 編集する **localVolumeSet** を決定します。

```

oc get -n $local_storage_project localvolumeset
NAME AGE
localblock 25h

```

- c. **localVolumeSet** 定義を更新して、新規ノードを追加し、障害が発生したノードを削除します。

```

oc edit -n $local_storage_project localvolumeset localblock
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
 operator: In
 values:
 - server1.example.com
 - server2.example.com
 #- server3.example.com
 - newnode.example.com

[...]

```

エディターを終了する前に必ず保存します。

上記の例では、**server3.example.com** が削除され、**newnode.example.com** が新規ノードになります。

17. 新規 **localblock** PV が利用可能であることを確認します。

```

$oc get pv | grep localblock | grep Available
local-pv-551d950 512Gi RWO Delete Available
localblock 26s

```

18. **openshift-storage** プロジェクトを変更します。

```

$ oc project openshift-storage

```

19. 失敗した OSD をクラスターから削除します。必要に応じて、複数の障害のある OSD を指定することができます。

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=_<failed_osd_id>_ FORCE_OSD_REMOVAL=false | oc create -n
openshift-storage -f -
```

#### <failed\_osd\_id>

**rook-ceph-osd** 接頭辞の直後の Pod 名の整数です。コマンドにコンマ区切りの OSD ID を追加して、複数の OSD を削除できます (例: **FAILED\_OSD\_IDS=0,1,2**)

OSD が 3 つしかないクラスター、または OSD が削除された後にデータの 3 つのレプリカすべてを復元するにはスペースが不十分なクラスターでは、**FORCE\_OSD\_REMOVAL** 値を **true** に変更する必要があります。

20. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されたことを確認します。

**Completed** のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```



#### 注記

**ocs-osd-removal-job** が失敗し、Pod が予想される Completed の状態にない場合、追加のデバッグのために Pod ログを確認します。以下に例を示します。

```
oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```

21. PVC に関連付けられた PV を特定します。

```
oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
local-pv-d6bf175b 512Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-
0-6c5pw localblock 2d22h server3.example.com
```

Released 状態の PV がある場合は、これを削除します。

```
oc delete pv <persistent-volume>
```

以下に例を示します。

```
oc delete pv local-pv-d6bf175b
persistentvolume "local-pv-d6bf175b" deleted
```

22. **crashcollector** Pod デプロイメントを特定します。

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-name
-n openshift-storage
```

既存の crashcollector Pod デプロイメントがある場合は、これを削除します。

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=failed-node-
name -n openshift-storage
```

23. **ocs-osd-removal** ジョブを削除します。



```
oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

## 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads → Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が Running 状態にあることを確認します。  
また、増分の **mon** が新規に作成されており、Running 状態にあることを確認します。

```
$ oc get pod -n openshift-storage | grep mon
```

出力例:

```
rook-ceph-mon-a-cd575c89b-b6k66 2/2 Running 0 38m
rook-ceph-mon-b-6776bc469b-tzzt8 2/2 Running 0 38m
rook-ceph-mon-d-5ff5d488b5-7v8xh 2/2 Running 0 4m8s
```

OSD と Mon が **Running** 状態になるまで数分かかる場合があります。

4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. (オプション) クラスタでクラスタ全体の暗号化が有効な場合には、新規 OSD デバイスが暗号化されていることを確認します。

- a. 直前の手順で特定された新規ノードごとに、以下を実行します。
  - i. デバッグ Pod を作成し、選択したホストの chroot 環境を開きます。

```
$ oc debug node/<node name>
$ chroot /host
```

- ii. lsblk を実行し、**ocs-deviceset** 名の横にある crypt キーワードを確認します。

```
$ lsblk
```

6. 検証手順が失敗した場合は、[Red Hat サポートにお問い合わせください](#)。

## 2.6. IBM POWER SYSTEMS インフラストラクチャーでのストレージノードの置き換え

OpenShift Container Storage 4.3 では、ノード置き換えを、IBM Power Systems 関連のデプロイメントで動作するノードについてプロアクティブに実行し、失敗したノードのそれぞれについてリアクティブに実行することができます。

### 2.6.1. IBM Power Systems で動作するストレージまたは障害のあるストレージノードの置き換え

#### 前提条件

- Red Hat では、交換前のノードと同様のインフラストラクチャーおよびリソースで、交換後のノードを設定することを推奨します。
- OpenShift Container Platform (RHOC) クラスターにログインしている必要があります。

#### 手順

1. ノードを特定し、置き換えるノードのラベルを取得します。

```
$ oc get nodes --show-labels | grep <node_name>
```

2. 置き換えるノードで実行されている **mon** (ある場合) および OSD を特定します。

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. 先の手順で特定された Pod のデプロイメントをスケールダウンします。  
以下に例を示します。

```
$ oc scale deployment rook-ceph-mon-a --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-1 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

4. ノードにスケジューラ対象外 (unschedulable) のマークを付けます。

```
$ oc adm cordon <node_name>
```

5. Terminating 状態の Pod の削除

```
$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system ("oc -n " $1 " delete pods " $2 " --grace-period=0 " " --force ")}'
```

6. ノードをドレイン (解放) します。

```
$ oc adm drain <node_name> --force --delete-local-data --ignore-daemonsets
```

7. ノードを削除します。

-

```
$ oc delete node <node_name>
```

8. 必要なインフラストラクチャーで新規の IBM Power マシンを取得します。 [クラスターの IBM Power Systems へのインストール](#) について参照してください。
9. 新規 IBM Power Systems マシンを使用して新規 OpenShift Container Platform Systems ノードを作成します。
10. **Pending** 状態の OpenShift Container Storage に関連する証明書署名要求 (CSR) の有無を確認します。

```
$ oc get csr
```

11. 新規ノードに必要なすべての OpenShift Container Storage CSR を承認します。

```
$ oc adm certificate approve <Certificate_Name>
```

12. OpenShift Web コンソールで **Compute → Nodes** をクリックし、新規ノードが **Ready** 状態にあることを確認します。
13. 優先するインターフェイスを使用して、OpenShift Container Storage ラベルを新規ノードに適用します。

#### ユーザーインターフェイスを使用する場合

- a. 新規ノードについて、**Action Menu (⋮) → Edit Labels** をクリックします。
- b. **cluster.ocs.openshift.io/openshift-storage** を追加し、**Save** をクリックします。

#### コマンドラインインターフェイスの使用

- a. 以下のコマンドを実行して、OpenShift Container Storage ラベルを新規ノードに適用します。

```
$ oc label node <new_node_name> cluster.ocs.openshift.io/openshift-storage=""
```

14. OpenShift ローカルストレージ Operator がインストールされている namespace を特定し、これを **local\_storage\_project** 変数に割り当てます。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

以下に例を示します。

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
echo $local_storage_project
openshift-local-storage
```

15. 新規に追加されたワーカーノードを localVolumeSet に追加します。
  - a. 編集する **localVolumeSet** を決定します。

```
oc get -n $local_storage_project localvolumeset
NAME AGE
localblock 25h
```

- b. **localVolumeSet** 定義を更新して、新規ノードを追加し、障害が発生したノードを削除します。

```
oc edit -n $local_storage_project localvolumeset localblock
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
 operator: In
 values:
 #- worker-0
 - worker-1
 - worker-2
 - worker-3
[...]
```

エディターを終了する前に必ず保存します。

上記の例では、**worker-0** が削除されて **worker-3** が新規ノードになります。

16. 新規 **localblock** PV が利用可能であることを確認します。

```
$ oc get pv | grep localblock
NAME CAPACITY ACCESSMODES RECLAIMPOLICY STATUS CLAIM
STORAGECLASS AGE
local-pv-3e8964d3 500Gi RWO Delete Bound ocs-deviceset-localblock-2-
data-0-mdbg9 localblock 25h
local-pv-414755e0 500Gi RWO Delete Bound ocs-deviceset-localblock-1-
data-0-4cs1f localblock 25h
local-pv-b481410 500Gi RWO Delete Available
localblock 3m24s
local-pv-5c9b8982 500Gi RWO Delete Bound ocs-deviceset-localblock-0-
data-0-g2mmc localblock 25h
```

17. **openshift-storage** プロジェクトを変更します。

```
$ oc project openshift-storage
```

18. 失敗した OSD をクラスターから削除します。必要に応じて、複数の障害のある OSD を指定することができます。

- a. PVC を特定します。後に、その特定の PVC に関連付けられた PV を削除する必要があるためです。

```
$ osd_id_to_remove=1
$ oc get -n openshift-storage -o yaml deployment rook-ceph-osd-${osd_id_to_remove} |
grep ceph.rook.io/pvc
```

ここで、**osd\_id\_to\_remove** は **rook-ceph-osd** 接頭辞の直後にくる Pod 名の整数です。  
この例では、デプロイメント名は **rook-ceph-osd-1** です。

出力例:

```
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc
```

この例では、PVC 名は **ocs-deviceset-localblock-0-data-0-g2mmc** です。

- b. 失敗した OSD をクラスターから削除します。

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> FORCE_OSD_REMOVAL=false | oc create -n
openshift-storage -f -
```

コマンドにコンマ区切りの OSD ID を追加して、複数の OSD を削除できます。(例:  
FAILED\_OSD\_IDS=0,1,2)

OSD が 3 つしかないクラスター、または OSD が削除された後にデータの 3 つのレプリカすべてを復元するにはスペースが不十分なクラスターでは、**FORCE\_OSD\_REMOVAL** 値を **true** に変更する必要があります。



#### 警告

この手順により、OSD はクラスターから完全に削除されます。**osd\_id\_to\_remove** の正しい値が指定されていることを確認します。

19. **ocs-osd-removal-job** Pod のステータスをチェックして、OSD が正常に削除されることを確認します。

**Completed** のステータスで、OSD の削除ジョブが正常に完了したことを確認します。

```
oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```



#### 注記

**ocs-osd-removal-job** が失敗し、Pod が予想される **Completed** の状態にない場合、追加のデバッグのために Pod ログを確認します。以下に例を示します。

```
oc logs -l job-name=ocs-osd-removal-job -n openshift-storage
```

20. 障害のあるノードに関連付けられた PV を削除します。

- a. PVC に関連付けられた PV を特定します。PVC 名は、手順 16(a) で取得した内容と同じでなければなりません。

```
oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
local-pv-5c9b8982 500Gi RWO Delete Released openshift-storage/ocs-deviceset-
localblock-0-data-0-g2mmc localblock 24h worker-0
```

b. PV を削除します。

```
oc delete pv <persistent-volume>
```

以下に例を示します。

```
oc delete pv local-pv-5c9b8982
persistentvolume "local-pv-5c9b8982" deleted
```

21. **crashcollector** Pod デプロイメントを削除します。

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
-n openshift-storage
```

22. **ocs-osd-removal-job** を削除します。

```
oc delete -n openshift-storage job ocs-osd-removal-job
```

出力例:

```
job.batch "ocs-osd-removal-job" deleted
```

## 検証手順

1. 以下のコマンドを実行して、出力で新規ノードが表示されていることを確認します。

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. **Workloads** → **Pods** をクリックし、新規ノード上の少なくとも以下の Pod が **Running** 状態にあることを確認します。

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 他の必要なすべての OpenShift Container Storage Pod が **Running** 状態にあることを確認します。  
また、増分の **mon** が新規に作成されており、**Running** 状態にあることを確認します。

```
$ oc get pod -n openshift-storage | grep mon
```

出力例:

|                                  |     |         |   |       |
|----------------------------------|-----|---------|---|-------|
| rook-ceph-mon-b-74f6dc9dd6-4llzq | 1/1 | Running | 0 | 6h14m |
| rook-ceph-mon-c-74948755c-h7wtx  | 1/1 | Running | 0 | 4h24m |
| rook-ceph-mon-d-598f69869b-4bv49 | 1/1 | Running | 0 | 162m  |

OSD と Mon が **Running** 状態になるまで数分かかる場合があります。

4. 新規 OSD Pod が交換後のノードで実行されていることを確認します。

```
$ oc get pods -o wide -n openshift-storage | egrep -i new-node-name | egrep osd
```

5. 検証手順が失敗した場合は、[Red Hat サポート](#)にお問い合わせください。