



# Red Hat OpenShift Data Foundation 4.15

## OpenShift Data Foundation のトラブルシューティング

OpenShift Data Foundation のトラブルシューティングの手順



# Red Hat OpenShift Data Foundation 4.15 OpenShift Data Foundation のトラブルシューティング

---

OpenShift Data Foundation のトラブルシューティングの手順

## 法律上の通知

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

Red Hat OpenShift Data Foundation のトラブルシューティングについては、本書をお読みください。

## 目次

|   |    |
|---|----|
| 多様性を受け入れるオープンソースの強化 .....   | 4  |
| RED HAT ドキュメントへのフィードバック (英語のみ) .....  | 5  |
| 第1章 概要 .....  | 6  |
| 第2章 MUST-GATHER を使用したログファイルおよび診断情報のダウンロード .....   | 7  |
| 2.1. MUST-GATHER コマンドのバリエーション .....   | 8  |
| 2.2. モジュールモードでの MUST-GATHER の実行 .....   | 9  |
| 第3章 トラブルシューティングに共通して必要になるログ .....   | 10 |
| 3.1. ログの詳細レベルの調整 .....  | 12 |
| 第4章 OPENSIFT DATA FOUNDATION デプロイメント後のクラスター全体のデフォルトノードセクターの上書き .....                      | 14 |
| 第5章 暗号化トークンの削除または期限切れの状態 .....  | 15 |
| 第6章 OPENSIFT DATA FOUNDATION のアラートおよびエラーのトラブルシューティング .....                                | 16 |
| 6.1. アラートとエラーの解決 .....  | 16 |
| 6.2. クラスターの健全性問題の解決 .....   | 22 |
| 6.3. クラスターアラートの解決 .....   | 23 |
| 6.4. 異常なバケットのエラーコードの検索 .....  | 50 |
| 6.5. 正常でない NAMESPACE ストアリソースのエラーコードの検索 .....  | 50 |
| 6.6. POD のリカバリー .....   | 51 |
| 6.7. EBS ボリュームの割り当て解除からのリカバリー .....   | 51 |
| 6.8. ROOK-CEPH-OPERATOR のデバッグログの有効化および無効化 .....   | 51 |
| 6.9. CEPH モニター数が少ないというアラートの解決 .....   | 52 |
| 6.10. 正常でないブロックリストノードのトラブルシューティング .....   | 52 |
| 第7章 ローカルストレージ OPERATOR デプロイメントの確認 .....   | 54 |
| 第8章 故障したまたは不要な CEPH OBJECT STORAGE デバイスの削除 .....  | 55 |
| 8.1. CEPH クラスターが正常であることの確認 .....  | 55 |
| 8.2. 動的にプロビジョニングされた RED HAT OPENSIFT DATA FOUNDATION で失敗した、または不要な CEPH OSD を削除。 .....     | 55 |
| 8.3. ローカルストレージデバイスを使用してプロビジョニングされた、失敗したまたは不要な CEPH OSD を削除 .....                          | 57 |
| 8.4. 失敗した、あるいは不要な CEPH OSD の削除中のエラー CEPHOSD:OSD.0 IS NOT OK TO DESTROY のトラブルシューティング ..... | 59 |
| 第9章 トラブルシューティングおよびアンインストール時の残りのリソースの削除 .....  | 60 |
| 第10章 外部モードでの CEPHFS PVC 作成のトラブルシューティング .....  | 62 |
| 第11章 OPENSIFT DATA FOUNDATION でのモニター POD の復元 .....  | 65 |
| 11.1. MULTICLOUD OBJECT GATEWAY の復元 .....   | 71 |
| 第12章 OPENSIFT DATA FOUNDATION での CEPH-MONITOR クォーラムの復元 .....                              | 73 |
| 第13章 RED HAT OPENSIFT DATA FOUNDATION コンソールプラグインの有効化 .....                                | 78 |
| 第14章 OPENSIFT DATA FOUNDATION コンポーネントのリソースの変更 .....                                       | 79 |
| 14.1. ROOK-CEPH POD の CPU およびメモリーリソースの変更 .....  | 79 |
| 14.2. MCG のリソースのチューニング .....  | 80 |
| 第15章 OPENSIFT DATA FOUNDATION のデプロイ後の MULTICLOUD OBJECT GATEWAY 外部サービスの無                  |    |

|  |    |
|--|----|
| 効化 .....   | 81 |
| 第16章 グローバル POD ネットワークを手動で有効化することによる OVS-MULTITENANT プラグインでの ODF-<br>CONSOLE へのアクセス ..... | 82 |
| 第17章 暗号化された RBD ストレージクラスへのアノテーション設定 .....  | 83 |



## 多様性を受け入れるオープンソースの強化

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



## RED HAT ドキュメントへのフィードバック (英語のみ)

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

フィードバックを送信するには、Bugzilla チケットを作成します。

1. [Bugzilla](#) の Web サイトに移動します。
2. **Component** セクションで、**documentation** を選択します。
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。

## 第1章 概要

OpenShift Data Foundation のトラブルシューティングは、管理者が Red Hat OpenShift Data Foundation クラスターのトラブルシューティングおよび修正方法を理解するのに役立ちます。

ほとんどのトラブルシューティングタスクは、修正または回避策のいずれかに重点を置いています。本書は、管理者が直面する可能性のあるエラーに基づいていくつかの章に分類されています。

- [2章 must-gather を使用したログファイルおよび診断情報のダウンロード](#) では、OpenShift Data Foundation で must-gather ユーティリティを使用する方法を示します。
- [3章 トラブルシューティングに共通して必要になるログ](#) では、OpenShift Data Foundation に共通して必要になるログファイルを取得する方法について説明します。
- [6章 OpenShift Data Foundation のアラートおよびエラーのトラブルシューティング](#) では、発生したエラーを特定し、必要なアクションを実行する方法を示します。



### 警告

Red Hat は、間違ったコマンドを実行するとデータ損失が発生する可能性があるため、OpenShift Data Foundation クラスターでの Ceph コマンドの実行をサポートしていません (Red Hat サポートまたは Red Hat ドキュメントで示されていない限り)。その場合、Red Hat サポートチームは商業的に合理的な努力しか提供できず、データ損失が発生した場合にすべてのデータを復元できない可能性があります。

## 第2章 MUST-GATHER を使用したログファイルおよび診断情報のダウンロード

Red Hat OpenShift Data Foundation が問題を自動的に解決できない場合、**must-gather** ツールを使用してログファイルと診断情報を収集し、お客様または Red Hat サポートが問題を確認し、解決策を判別できるようにします。



### 重要

Red Hat OpenShift Data Foundation が外部モードでデプロイされる場合、**must-gather** は OpenShift Data Foundation クラスターからのみログを収集し、外部の Red Hat Ceph Storage クラスターからデバッグデータおよびログを収集しません。外部の Red Hat Ceph Storage クラスターからデバッグログを収集するには、Red Hat Ceph Storage の [トラブルシューティングガイド](#) を参照するか、Red Hat Ceph Storage の管理者にお問い合わせください。

### 前提条件

- オプション: OpenShift Data Foundation が非接続環境にデプロイされている場合、個別の **must-gather** イメージを非接続環境で利用できるミラーレジストリーにミラーリングするようにしてください。

```
$ oc image mirror registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15 <local-registry>/odf4/odf-must-gather-rhel9:v4.15 [--registry-config=<path-to-the-registry-config>] [--insecure=true]
```

#### <local-registry>

非接続の OpenShift Container Platform クラスターで利用可能なローカルイメージのミラーレジストリーです。

#### <path-to-the-registry-config>

レジストリー認証情報へのパスで、デフォルトは `~/.docker/config.json` です。

#### --insecure

ミラーレジストリーがセキュアでない場合にのみこのフラグを追加します。

詳細は、Red Hat ナレッジベースソリューションを参照してください。

- [Redhat Openshift レジストリー間でイメージをミラーリングする方法](#)
- [プライベートレジストリーが安全でない場合の OpenShift イメージリポジトリーのミラーリングに失敗しました。](#)

### 手順

- OpenShift Data Foundation クラスターに接続されているクライアントから **must-gather** コマンドを実行します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15 --dest-dir=<directory-name>
```

#### <directory-name>

データを書き込むディレクトリーの名前です。



## 重要

非接続環境のデプロイメントの場合は、**--image** パラメーターのイメージをミラーリングされた **must-gather** イメージに置き換えます。

```
$ oc adm must-gather --image=<local-registry>/odf4/odf-must-gather-rhel9:v4.15 --dest-dir=<directory-name>
```

### <local-registry>

非接続の OpenShift Container Platform クラスタで利用可能なローカルイメージのミラーレジストリーです。

これにより、指定されたディレクトリーに以下の情報が収集されます。

- すべての Red Hat OpenShift Data Foundation クラスタ関連のカスタムリソース (CR) とそれらの namespace。
- すべての Red Hat OpenShift Data Foundation 関連の Pod の Pod ログ。
- ステータス、クラスタの正常性などの一部の標準的な Ceph コマンドの出力。

## 2.1. MUST-GATHER コマンドのバリエーション

- 状態が **Ready** ではないマスターノードが1つ以上ある場合には、**must-gather** Pod を安全にスケジュールできるように **--node-name** を使用して **Ready** のマスターノードを指定します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15 --dest-dir=_<directory-name>_ --node-name=_<node-name>_
```

- 特定の時点から情報を収集する場合は、以下を行います。
  - たとえば5秒以内または2日以内に収集されたログの相対的な期間を指定するには、**/usr/bin/gather since=<duration>** を追加します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15 --dest-dir=_<directory-name>_ /usr/bin/gather since=<duration>
```

- その後にログを収集する特定の時間を指定するには、**/usr/bin/gather since-time=<rfc3339-timestamp>** を追加します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15 --dest-dir=_<directory-name>_ /usr/bin/gather since-time=<rfc3339-timestamp>
```

以下のように、これらのコマンドのサンプルの値を置き換えます。

### <node-name>

状態が **Ready** ではないマスターノードが1つ以上ある場合には、このパラメーターを使用して、状態がまだ **Ready** のマスターノード名を指定します。これにより、**must-gather** Pod が準備状態にないマスターノードにスケジュールされないようにすることで、スケジューリングエラーを回避します。

### <directory-name>

**must-gather** によって収集される情報を保存するディレクトリー。

**<duration>**

**5h** (5 時間前から開始する) など、相対的な期間として情報を収集する期間 (の開始点) を指定します。

**<rfc3339-timestamp>**

**2020-11-10T04:00:00+00:00** (2020 年 11 月 11 日の 4am UTC から開始する) など、RFC 3339 タイムスタンプとして情報を収集する期間 (の開始点) を指定します。

## 2.2. モジュラーモードでの MUST-GATHER の実行

Red Hat OpenShift Data Foundation の **must-gather** は、環境によっては実行に長い時間がかかることがあります。これを回避するには、モジュラーモードで **must-gather** を実行し、次のコマンドを使用して必要なリソースのみを収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15 -- /usr/bin/gather <-arg>
```

**<-arg>** を次の1つ以上の引数に置き換えて、**must-gather** ログの対象とするリソースを必要に応じて指定します。

**-o, --odf**

ODF ログ (Ceph リソース、namespaced リソース、clusterscoped リソース、Ceph ログを含む)

**-d, --dr**

DR ログ

**-n, --noobaa**

Noobaa ログ

**-c, --ceph**

Ceph コマンドと Pod ログ

**-cl, --ceph-logs**

Ceph デーモン、カーネル、およびジャーナルログ、クラッシュレポート

**-ns, --namespaced**

namespaced リソース

**-cs, --clusterscoped**

clusterscoped リソース

**-h, --help**

ヘルプメッセージの印刷

## 第3章 トラブルシューティングに共通して必要になるログ

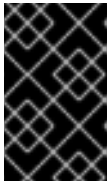
OpenShift Data Foundation のトラブルシューティングに共通して使用されるログの一部と、それらを生成するコマンドがリスト表示されます。

- 特定 Pod のログを生成します。

```
$ oc logs <pod-name> -n <namespace>
```

- Ceph または OpenShift Data Foundation クラスターのログを生成します。

```
$ oc logs rook-ceph-operator-<ID> -n openshift-storage
```



### 重要

現時点で、rook-ceph-operator ログは障害に関する情報を提供せず、問題のトラブルシューティングの制限として機能します。[Enabling and disabling debug logs for rook-ceph-operator](#)を参照してください。

- cephfs または rbd などのプラグイン Pod のログを生成し、app-pod の PVC マウントで問題を検出します。

```
$ oc logs csi-cephfsplugin-<ID> -n openshift-storage -c csi-cephfsplugin
```

```
$ oc logs csi-rbdplugin-<ID> -n openshift-storage -c csi-rbdplugin
```

- CSI Pod のすべてのコンテナのログを生成するには、以下を実行します。

```
$ oc logs csi-cephfsplugin-<ID> -n openshift-storage --all-containers
```

```
$ oc logs csi-rbdplugin-<ID> -n openshift-storage --all-containers
```

- PVC が **BOUND** 状態にない場合に問題を検出するために、cephfs または rbd プロビジョナー Pod のログを生成します。

```
$ oc logs csi-cephfsplugin-provisioner-<ID> -n openshift-storage -c csi-cephfsplugin
```

```
$ oc logs csi-rbdplugin-provisioner-<ID> -n openshift-storage -c csi-rbdplugin
```

- CSI Pod のすべてのコンテナのログを生成するには、以下を実行します。

```
$ oc logs csi-cephfsplugin-provisioner-<ID> -n openshift-storage --all-containers
```

```
$ oc logs csi-rbdplugin-provisioner-<ID> -n openshift-storage --all-containers
```

- cluster-info コマンドを使用して OpenShift Data Foundation ログを生成します。

```
$ oc cluster-info dump -n openshift-storage --output-directory=<directory-name>
```

- Local Storage Operator を使用する場合、ログの生成は cluster-info コマンドを使用して実行できます。

```
$ oc cluster-info dump -n openshift-local-storage --output-directory=<directory-name>
```

- OpenShift Data Foundation Operator ログおよびイベントを確認します。

- Operator ログを確認するには、以下を実行します。

```
# oc logs <ocs-operator> -n openshift-storage
```

```
<ocs-operator>
```

```
# oc get pods -n openshift-storage | grep -i "ocs-operator" | awk '{print $1}'
```

- Operator イベントを確認するには、以下を実行します。

```
# oc get events --sort-by=metadata.creationTimestamp -n openshift-storage
```

- OpenShift Data Foundation Operator のバージョンおよびチャネルを取得します。

```
# oc get csv -n openshift-storage
```

出力例:

| NAME                            | DISPLAY                     | VERSION | REPLACES | PHASE     |
|---------------------------------|-----------------------------|---------|----------|-----------|
| mcg-operator.v4.15.0            | NooBaa Operator             | 4.15.0  |          | Succeeded |
| ocs-operator.v4.15.0            | OpenShift Container Storage | 4.15.0  |          | Succeeded |
| odf-csi-addons-operator.v4.15.0 | CSI Addons                  | 4.15.0  |          | Succeeded |
| odf-operator.v4.15.0            | OpenShift Data Foundation   | 4.15.0  |          | Succeeded |

```
# oc get subs -n openshift-storage
```

出力例:

| NAME  | PACKAGE                 | SOURCE           |
|---|-------------------------|------------------|
| CHANNEL   |                         |                  |
| mcg-operator-stable-4.15-redhat-operators-openshift-marketplace |                         | mcg-operator     |
| redhat-operators stable-4.15                                    |                         |                  |
| ocs-operator-stable-4.15-redhat-operators-openshift-marketplace |                         | ocs-operator     |
| redhat-operators stable-4.15                                    |                         |                  |
| odf-csi-addons-operator   | odf-csi-addons-operator | redhat-operators |
| stable-4.15   |                         |                  |
| odf-operator  | odf-operator            | redhat-operators |
| 4.15  |                         | stable-          |

- installplan が作成されていることを確認します。

```
# oc get installplan -n openshift-storage
```

- OpenShift Data Foundation を事後更新するコンポーネントのイメージを確認します。

- イメージが実行中であることを確認するために使用するコンポーネントの Pod があるノードを確認します。

```
# oc get pods -o wide | grep <component-name>
```

以下に例を示します。

```
# oc get pods -o wide | grep rook-ceph-operator
```

出力例:

```
rook-ceph-operator-566cc677fd-bjqnb 1/1 Running 20 4h6m 10.128.2.5 rook-ceph-  
operator-566cc677fd-bjqnb 1/1 Running 20 4h6m 10.128.2.5 dell-r440-  
12.gsslab.pnq2.redhat.com <none> <none>  
  
<none> <none>
```

**dell-r440-12.gsslab.pnq2.redhat.com** は **node-name** です。

- イメージ ID を確認します。

```
# oc debug node/<node name>
```

**<node-name>**

イメージが実行中であることを確認するために使用するコンポーネントの Pod があるノードの名前です。

```
# chroot /host
```

```
# crictl images | grep <component>
```

以下に例を示します。

```
# crictl images | grep rook-ceph
```

**IMAGEID** を書き留め、これを [Rook Ceph Operator](#) ページの **Digest ID** にマップします。

## 関連情報

- [must-gather の使用](#)

## 3.1. ログの詳細レベルの調整

ログのデバッグによって消費されるスペースの量は、重大な問題になる可能性があります。Red Hat OpenShift Data Foundation は、ログのデバッグによって消費されるストレージの量を調整して制御する方法を提供します。

デバッグログの冗長レベルを調整するために、コンテナストレージインターフェイス (CSI) 操作を担当するコンテナのログレベルを調整できます。コンテナの yml ファイルで、次のパラメーターを調整してログレベルを設定します。

- **CSI\_LOG\_LEVEL** - デフォルトは **5**



- **CSI\_SIDECAR\_LOG\_LEVEL** - デフォルトは **1**

サポートされている値は **0** ~ **5** です。一般的な有用なログには **0** を使用し、トレースレベルの詳細度には **5** を使用します。

## 第4章 OPENSIFT DATA FOUNDATION デプロイメント後のクラスター全体のデフォルトノードセクターの上書き

クラスター全体でのデフォルトノードセクターが Openshift Data Foundation に使用される場合、container storage interface (CSI) daemonset によって生成される Pod はセクターに一致するノードでのみ起動できます。セクターに一致しないノードから OpenShift Data Foundation を使用できるようにするには、コマンドラインインターフェイスで以下の手順を実行して **cluster-wide default node selector** を上書きします。

### 手順

1. openshift-storage namespace の空のノードセクターを指定します。

```
$ oc annotate namespace openshift-storage openshift.io/node-selector=
```

2. DaemonSets によって生成される元の Pod を削除します。

```
oc delete pod -l app=csi-cephfsplugin -n openshift-storage
oc delete pod -l app=csi-rbdplugin -n openshift-storage
```

## 第5章 暗号化トークンの削除または期限切れの状態

鍵管理システムの暗号化トークンが削除されているか、有効期限が切れている場合は、以下の手順に従ってトークンを更新します。

### 前提条件

- 削除されているか、期限切れとなったトークンと同じポリシーを持つ新しいトークンがあることを確認します。

### 手順

1. OpenShift Container Platform Web コンソールにログインします。
2. **Workloads** → **Secrets** をクリックします。
3. クラスタ全体の暗号化に使用される **ocs-kms-token** を更新するには、以下を実行します。
  - a. **Project** を **openshift-storage** に設定します。
  - b. **ocs-kms-token** → **Actions** → **Edit Secret** をクリックします。
  - c. **Value** フィールドに暗号化トークンファイルをドラッグアンドドロップまたはアップロードします。トークンには、コピーおよび貼り付けが可能なファイルまたはテキストのいずれかを指定できます。
  - d. **Save** をクリックします。
4. 暗号化された永続ボリュームのある指定のプロジェクトまたは namespace の **ceph-csi-kms-token** を更新するには、以下を実行します。
  - a. 必要な **Project** を選択します。
  - b. **ceph-csi-kms-token** → **Actions** → **Edit Secret** をクリックします。
  - c. **Value** フィールドに暗号化トークンファイルをドラッグアンドドロップまたはアップロードします。トークンには、コピーおよび貼り付けが可能なファイルまたはテキストのいずれかを指定できます。
  - d. **Save** をクリックします。



### 注記

トークンは、**ceph-csi-kms-token** を使用するすべての暗号化された PVC が削除された後にのみ削除できます。

## 第6章 OPENSIFT DATA FOUNDATION のアラートおよびエラーのトラブルシューティング

### 6.1. アラートとエラーの解決

Red Hat OpenShift Data Foundation は、多くの共通する障害シナリオを検出し、これらを自動的に解決できます。ただし、一部の問題には管理者の介入が必要です。

現在発生しているエラーを確認するには、以下のいずれかの場所を確認します。

- **Observe** → **Alerting** → **Firing** オプション
- **Home** → **Overview** → **Cluster** タブ
- **Storage** → **Data Foundation** → **Storage System** → **storage system** リンクのポップアップ → **Overview** → **Block and File** タブ
- **Storage** → **Data Foundation** → **Storage System** → **storage system** リンクのポップアップ → **Overview** → **Object** タブ

表示されるエラーをコピーして、これを以下のセクションで検索し、その重大度と解決策を確認します。

**Name:** **CephMonVersionMismatch**

**Message:** **There are multiple versions of storage services running.**

**Description:** **There are {{ \$value }} different versions of Ceph Mon components running.**

**Severity:** Warning

**Resolution:** Fix

**Procedure:** Inspect the user interface and log, and verify if an update is in progress.

- If an update is in progress, this alert is temporary.
- If an update is not in progress, restart the upgrade process.

**Name:** **CephOSDVersionMismatch**

**Message:** **There are multiple versions of storage services running.**

**Description:** **There are {{ \$value }} different versions of Ceph OSD components running.**

**Severity:** Warning

**Resolution:** Fix

**Procedure:** Inspect the user interface and log, and verify if an update is in progress.

- If an update is in progress, this alert is temporary.
- If an update is not in progress, restart the upgrade process.

Name: **CephClusterCriticallyFull**

Message: **Storage cluster is critically full and needs immediate expansion**

Description: **Storage cluster utilization has crossed 85%.**

Severity: Critical

Resolution: Fix

Procedure: Remove unnecessary data or expand the cluster.

Name: **CephClusterNearFull**

Fixed: **Storage cluster is nearing full.Expansion is required.**

Description: **Storage cluster utilization has crossed 75%.**

Severity: Warning

Resolution: Fix

Procedure: Remove unnecessary data or expand the cluster.

Name: **NooBaaBucketErrorState**

Message: **A NooBaa Bucket Is In Error State**

Description: **A NooBaa bucket {{ \$labels.bucket\_name }} is in error state for more than 6m**

Severity: Warning

Resolution: Workaround

手順:[異常なバケットのエラーコードの検索](#)

Name: **NooBaaNamespaceResourceErrorState**

Message: **A NooBaa Namespace Resource Is In Error State**

Description: **A NooBaa namespace resource {{ \$labels.namespace\_resource\_name }} is in error state for more than 5m**

Severity: Warning

Resolution: Fix

手順:[正常でない namespace ストアリソースのエラーコードの検索](#)

Name: **NooBaaNamespaceBucketErrorState**

Message: **A NooBaa Namespace Bucket Is In Error State**

Description: **A NooBaa namespace bucket {{ \$labels.bucket\_name }} is in error state for more than 5m**

Severity: Warning

Resolution: Fix

手順: [異常なバケットのエラーコードの検索](#)

Name: **CephMdsMissingReplicas**

Message: **Insufficient replicas for storage metadata service.**

Description: `Minimum required replicas for storage metadata service not available.

Might affect the working of storage cluster.`

Severity: Warning

Resolution: [Contact Red Hat support](#)

Procedure:

1. Check for alerts and operator status.
2. If the issue cannot be identified, [contact Red Hat support](#).

Name: **CephMgrIsAbsent**

Message: **Storage metrics collector service not available anymore.**

Description: **Ceph Manager has disappeared from Prometheus target discovery.**

Severity: Critical

Resolution: [Contact Red Hat support](#)

Procedure:

1. ユーザーインターフェイスとログを調べて、更新が進行中であるかどうかを確認します。
  - If an update is in progress, this alert is temporary.
  - If an update is not in progress, restart the upgrade process.
2. Once the upgrade is complete, check for alerts and operator status.
3. If the issue persists or cannot be identified, [contact Red Hat support](#).

Name: **CephNodeDown**

Message: **Storage node {{ \$labels.node }} went down**

Description: **Storage node {{ \$labels.node }} went down.Check the node immediately.**

Severity: Critical

Resolution: [Contact Red Hat support](#)

Procedure:

1. Check which node stopped functioning and its cause.
2. Take appropriate actions to recover the node.If node cannot be recovered:
  - [Red Hat OpenShift Data Foundation のストレージノードの置き換え](#) を参照してください。
  - [Contact Red Hat support](#)

Name: **CephClusterErrorState**

Message: **Storage cluster is in error state**

Description: **Storage cluster is in error state for more than 10m.**

Severity: Critical

Resolution: [Contact Red Hat support](#)

Procedure:

1. Check for alerts and operator status.
2. If the issue cannot be identified, [download log files and diagnostic information using must-gather](#).
3. [Open a Support Ticket](#) with [Red Hat Support](#) with an attachment of the output of must-gather.

Name: **CephClusterWarningState**

Message: **Storage cluster is in degraded state**

Description: **Storage cluster is in warning state for more than 10m.**

Severity: Warning

Resolution: [Contact Red Hat support](#)

Procedure:

1. Check for alerts and operator status.
2. If the issue cannot be identified, [download log files and diagnostic information using must-gather](#).
3. [Open a Support Ticket](#) with [Red Hat Support](#) with an attachment of the output of must-gather.

Name: **CephDataRecoveryTakingTooLong**

Message: **Data recovery is slow**

Description: **Data recovery has been active for too long.**

Severity: Warning

Resolution: [Contact Red Hat support](#)

Name: **CephOSDDiskNotResponding**

Message: **Disk not responding**

Description: **Disk device {{ \$labels.device }} not responding, on host {{ \$labels.host }}.**

Severity: Critical

Resolution: [Contact Red Hat support](#)

Name: **CephOSDDiskUnavailable**

Message: **Disk not accessible**

Description: **Disk device {{ \$labels.device }} not accessible on host {{ \$labels.host }}.**

Severity: Critical

Resolution: [Contact Red Hat support](#)

Name: **CephPGRepairTakingTooLong**

Message: **Self heal problems detected**

Description: **Self heal operations taking too long.**

Severity: Warning

Resolution: [Contact Red Hat support](#)

Name: **CephMonHighNumberOfLeaderChanges**

Message: **Storage Cluster has seen many leader changes recently.**

Description: **'Ceph Monitor "{{ \$labels.job }}" instance {{ \$labels.instance }} has seen {{ \$value printf "%.2f" }} leader changes per minute recently.'**

Severity: Warning

Resolution: [Contact Red Hat support](#)



Name: **CephMonQuorumAtRisk**

Message: **Storage quorum at risk**

Description: **Storage cluster quorum is low.**

Severity: Critical

Resolution: [Contact Red Hat support](#)

Name: **ClusterObjectStoreState**

Message: **Cluster Object Store is in an unhealthy state.Check Ceph cluster health.**

Description: **Cluster Object Store is in an unhealthy state for more than 15s.Check Ceph cluster health.**

Severity: Critical

Resolution: [Contact Red Hat support](#)

Procedure:

- **CephObjectStore** CR インスタンスを確認します。
- [Contact Red Hat support](#)

Name: **CephOSDFlapping**

Message: **Storage daemon osd.x has restarted 5 times in the last 5 minutes.Check the pod events or Ceph status to find out the cause.**

Description: **Storage OSD restarts more than 5 times in 5 minutes.**

Severity: Critical

Resolution: [Contact Red Hat support](#)

Name: **OdfPoolMirroringImageHealth**

Message: **Mirroring image(s) (PV) in the pool <pool-name> are in Warning state for more than a 1m. Mirroring might not work as expected.**

説明: 1つまたはいくつかのアプリケーションで障害復旧が失敗しています。

Severity: Warning

Resolution: [Contact Red Hat support](#)

**Name:** OdfMirrorDaemonStatus**Message:** Mirror daemon is unhealthy.**Description:** Disaster recovery is failing for the entire cluster. Mirror daemon is in an unhealthy status for more than 1m. Mirroring on this cluster is not working as expected.**Severity:** Critical**Resolution:** [Contact Red Hat support](#)

## 6.2. クラスターの健全性問題の解決

OpenShift Data Foundation ユーザーインターフェイスに表示される Red Hat Ceph Storage クラスターが出力する可能性のある正常性メッセージには限りがあります。これらは、固有の識別子を持つヘルスチェックとして定義されています。識別子は、ツールが正常性チェックを理解し、その意味を反映する方法でそれらを提示できるようにすることを目的とした、簡潔な疑似人間可読文字列です。詳細情報およびトラブルシューティングを行うには、以下のヘルスコードをクリックします。

| 正常性コード                       | 説明                                     |
|------------------------------|--|
| <a href="#">MON_DISK_LOW</a> | 1つまたは複数の Ceph Monitor のディスク領域が不足しています。 |

### 6.2.1. MON\_DISK\_LOW

この警告は、監視データベースをパーセンテージとして格納するファイルシステムの使用可能な領域が **mon\_data\_avail\_warn** を下回る場合にトリガーされます (デフォルトは、15% です)。これは、システム上の他のプロセスまたはユーザーが、モニターで使用されているのと同じファイルシステムを満杯にしていることを示している可能性があります。また、モニターのデータベースが大きいことを示すこともできます。

#### 注記

ファイルシステムへのパスは、mon のデプロイメントによって異なります。mon が **storagecluster.yaml** でデプロイされている場所へのパスを見つけることができます。

パスの例:

- PVC パスにデプロイされる mon: **/var/lib/ceph/mon**
- ホストパス経由でデプロイされる mon: **/var/lib/rook/mon**

領域を消去するには、ファイルシステムで使用率の高いファイルを表示し、削除するファイルを選択します。ファイルを表示するには、以下のコマンドを実行します。

```
# du -a <path-in-the-mon-node> |sort -n -r |head -n10
```

**<path-in-the-mon-node>** を、mon がデプロイされているファイルシステムへのパスに置き換えます。

### 6.3. クラスタアラートの解決

OpenShift Data Foundation ユーザーインターフェイスに表示される Red Hat Ceph Storage クラスタが出力する可能性のある正常性アラートには限りがあります。これらは、固有の識別子を持つ正常性アラートとして定義されています。識別子は、ツールが正常性チェックを理解し、その意味を反映する方法でそれらを提示できるようにすることを目的とした、簡潔な疑似人間可読文字列です。詳細の確認とトラブルシューティングを行うには、正常性アラートをクリックしてください。

表6.1 クラスタの正常性アラートの種類

| 正常性アラート  | 概要  |
|--|---|
| <a href="#">CephClusterCriticallyFull</a>        | ストレージクラスタの使用率が 80% を超えました。  |
| <a href="#">CephClusterErrorState</a>            | ストレージクラスタが 10 分以上エラー状態になっています。  |
| <a href="#">CephClusterNearFull</a>              | ストレージクラスタが最大容量に近づいています。データの削除またはクラスタの拡張が必要です。   |
| <a href="#">CephClusterReadOnly</a>              | ストレージクラスタは現在読み取り専用であり、すぐにデータを削除するか、クラスタを拡張する必要があります。  |
| <a href="#">CephClusterWarningState</a>          | ストレージクラスタが 10 分以上警告状態になっています。   |
| <a href="#">CephDataRecoveryTakingTooLong</a>    | データ復旧が長期間アクティブになっています。  |
| <a href="#">CephMdsCacheUsageHigh</a>            | MDS デーモンの Ceph メタデータサービス (MDS) のキャッシュ使用量が、 <b>mds_cache_memory_limit</b> の 95% を超えました。  |
| <a href="#">CephMdsCpuUsageHigh</a>              | MDS デーモンの Ceph MDS の CPU 使用率が、適切なパフォーマンスのしきい値を超えました。  |
| <a href="#">CephMdsMissingReplicas</a>           | ストレージメタデータサービスに最低限必要なレプリカが利用できません。ストレージクラスタの動作に影響を与える可能性があります。  |
| <a href="#">CephMgrIsAbsent</a>                  | Alertmanager が Prometheus のターゲット検出に表示されません。   |
| <a href="#">CephMgrIsMissingReplicas</a>         | Ceph マネージャーにレプリカがありません。これにより、正常性ステータスのレポートが作成され、 <b>ceph status</b> コマンドによってレポートされる情報の一部が失われるか、古くなります。さらに、Ceph マネージャーは、Ceph の既存の機能を拡張することを目的としたマネージャーフレームワークを担当します。 |
| <a href="#">CephMonHighNumberOfLeaderChanges</a> | Ceph モニターのリーダーの変更回数が異常です。   |
| <a href="#">CephMonQuorumAtRisk</a>              | ストレージクラスタのクォーラムが不足しています。  |
| <a href="#">CephMonQuorumLost</a>                | ストレージクラスタ内のモニター Pod の数が十分ではありません。   |

| 正常性アラート                               | 概要   |
|---------------------------------------|--|
| CephMonVersionMismatch                | 複数の異なるバージョンの Ceph Mon コンポーネントが実行されています。  |
| CephNodeDown                          | ストレージノードがダウンしました。すぐにノードを確認してください。アラートにノード名が含まれています。  |
| CephOSDCriticallyFull                 | バックエンドオブジェクトストレージデバイス (OSD) の使用率が 80% を超えました。すぐにスペースを解放するか、ストレージクラスターを拡張するか、サポートにお問い合わせください。 |
| CephOSDDiskNotResponding              | いずれかのホストでディスクデバイスが応答していません。  |
| CephOSDDiskUnavailable                | いずれかのホストでディスクデバイスにアクセスできません。   |
| CephOSDFlapping                       | Ceph Storage OSD のフラッピング。  |
| CephOSDNearFull                       | OSD ストレージデバイスの 1 つが満杯に近づいています。   |
| CephOSDSlowOps                        | OSD リクエストの処理に時間がかかりすぎています。   |
| CephOSDVersionMismatch                | 複数の異なるバージョンの Ceph OSD コンポーネントが実行されています。  |
| CephPGRepairTakingTooLong             | 自己修復操作に時間がかかりすぎています。   |
| CephPoolQuotaBytesCriticallyExhausted | ストレージプールクォータの使用率が 90% を超えました。  |
| CephPoolQuotaBytesNearExhaustion      | ストレージプールクォータの使用率が 70% を超えました。  |
| OSDCPULoadHigh                        | 特定 Pod 上の OSD コンテナの CPU 使用率が 80% を超えており、OSD のパフォーマンスに影響する可能性があります。                           |
| PersistentVolumeUsageCritical         | 永続ボリューム要求の使用率が容量の 85% を超えました。  |
| PersistentVolumeUsageNearFull         | 永続ボリューム要求の使用量が容量の 75% を超えました。  |

### 6.3.1. CephClusterCriticallyFull

|    |   |
|----|---|
| 意味 | ストレージクラスターの使用率が 80% を超えました。85% で読み取り専用になります。使用率が 85% を超えると、Ceph クラスターは読み取り専用になります。すぐにスペースを解放するか、ストレージクラスターを拡張してください。通常、このアラートの前に、オブジェクトストレージデバイス (OSD) デバイスが満杯または満杯に近いことに関連するアラートが表示されます。 |
|----|---|

|    |      |
|----|------|
| 影響 | High |
|----|------|

## 診断

### ストレージのスケーリング

クラスターのタイプに応じて、ストレージデバイス、ノード、またはその両方を追加する必要があります。詳細は、[ストレージのスケーリングガイド](#) を参照してください。

## 軽減策

### 情報の削除

クラスターをスケールアップできない場合は、情報を削除して領域を解放する必要があります。

## 6.3.2. CephClusterErrorState

|    |  |
|----|--|
| 意味 | このアラートは、ストレージクラスターが許容できない時間にわたって <b>ERROR</b> 状態にあり、ストレージの可用性が低下していることを示しています。このアラートの前にトリガーされた他のアラートを確認し、先にそれらのアラートのトラブルシューティングを行ってください。 |
| 影響 | Critical   |

## 診断

### Pod ステータス: 保留

- リソースの問題、保留中の永続ボリューム要求 (PVC)、ノードの割り当て、および kubelet の問題を確認します。

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

```
$ oc get pod | grep rook-ceph
```

- 問題のある Pod として識別された Pod の変数として **MYPOD** を設定します。

```
# Examine the output for a rook-ceph that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

```
<pod_name>
```

問題のある Pod として識別された Pod の名前を指定します。

- リソースの制限または保留中の PVC を探します。それがない場合は、ノードの割り当てを確認します。

```
$ oc get pod/${MYPOD} -o wide
```

### Pod ステータス: 保留中や実行中ではないが、準備完了状態でもない

- readiness プロブを確認します。

```
$ oc describe pod/${MYPOD}
```

### Pod ステータス: 保留中ではないが、実行中でもない

- アプリケーションまたはイメージの問題を確認します。

```
$ oc logs pod/${MYPOD}
```



#### 重要

- ノードが割り当てられている場合は、ノードの kubelet を確認します。
- 実行中の Pod の基本的な正常性、ノードアフィニティー、およびノードでのリソースの可用性が確認されたら、Ceph ツールを実行してストレージコンポーネントのステータスを取得します。

### 軽減策

#### デバッグログの情報

- この手順はオプションです。次のコマンドを実行して、Ceph クラスターのデバッグ情報を収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
```

### 6.3.3. CephClusterNearFull

|    |   |
|----|---|
| 意味 | ストレージクラスターの使用率が 75% を超えました。85% で読み取り専用になります。スペースを解放するか、ストレージクラスターを拡張してください。 |
| 影響 | Critical  |

### 診断

#### ストレージのスケーリング

クラスターのタイプに応じて、ストレージデバイス、ノード、またはその両方を追加する必要があります。詳細は、[ストレージのスケーリングガイド](#) を参照してください。

### 軽減策

#### 情報の削除

クラスターをスケールアップできない場合は、スペースを解放するために情報を削除する必要があります。

### 6.3.4. CephClusterReadOnly

|    |   |
|----|---|
| 意味 | ストレージクラスターの使用率が 85% を超えたため、読み取り専用になります。すぐにスペースを解放するか、ストレージクラスターを拡張してください。 |
| 影響 | Critical  |

## 診断

### ストレージのスケールアップ

クラスターのタイプに応じて、ストレージデバイス、ノード、またはその両方を追加する必要があります。詳細は、[ストレージのスケールアップガイド](#) を参照してください。

## 軽減策

### 情報の削除

クラスターをスケールアップできない場合は、スペースを解放するために情報を削除する必要があります。

## 6.3.5. CephClusterWarningState

|    |  |
|----|--|
| 意味 | このアラートは、ストレージクラスターが許容できない期間にわたって警告状態にあったことを示しています。この状態でもストレージ操作は機能しますが、クラスターが操作を試みてエラー状態にならないように、エラーを修正することを推奨します。このアラートの前にトリガーされた他のアラートを確認し、先にそれらのアラートのトラブルシューティングを行ってください。 |
| 影響 | High   |

## 診断

### Pod ステータス: 保留

- リソースの問題、保留中の永続ボリューム要求 (PVC)、ノードの割り当て、および kubelet の問題を確認します。

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

```
oc get pod | grep {ceph-component}
```

- 問題のある Pod として識別された Pod の変数として **MYPOD** を設定します。

```
# Examine the output for a {ceph-component} that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

**<pod\_name>**

問題のある Pod として識別された Pod の名前を指定します。

- リソースの制限または保留中の PVC を探します。それがない場合は、ノードの割り当てを確認します。

■

```
$ oc get pod/${MYPOD} -o wide
```

Pod ステータス: 保留中や実行中ではないが、準備完了状態でもない

- readiness プロブを確認します。

```
$ oc describe pod/${MYPOD}
```

Pod ステータス: 保留中ではないが、実行中でもない

- アプリケーションまたはイメージの問題を確認します。

```
$ oc logs pod/${MYPOD}
```



### 重要

ノードが割り当てられている場合は、ノードの kubelet を確認します。

## 軽減策

### デバッグログの情報

- この手順はオプションです。次のコマンドを実行して、Ceph クラスターのデバッグ情報を収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
```

## 6.3.6. CephDataRecoveryTakingTooLong

|    |   |
|----|---|
| 意味 | データ復旧に時間がかかっています。すべてのオブジェクトストレージデバイス (OSD) が稼働しているかどうかを確認します。 |
| 影響 | High  |

## 診断

### Pod ステータス: 保留

1. リソースの問題、保留中の永続ボリューム要求 (PVC)、ノードの割り当て、および kubelet の問題を確認します。

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

```
oc get pod | grep rook-ceph-osd
```

2. 問題のある Pod として識別された Pod の変数として **MYPOD** を設定します。



```
# Examine the output for a {ceph-component} that is in the pending state, not running or
not ready
MYPOD=<pod_name>
```

<pod\_name>

問題のある Pod として識別された Pod の名前を指定します。

- リソースの制限または保留中の PVC を探します。それがない場合は、ノードの割り当てを確認します。

```
$ oc get pod/${MYPOD} -o wide
```

Pod ステータス: 保留中や実行中ではないが、準備完了状態でもない

- readiness プロブを確認します。

```
$ oc describe pod/${MYPOD}
```

Pod ステータス: 保留中ではないが、実行中でもない

- アプリケーションまたはイメージの問題を確認します。

```
$ oc logs pod/${MYPOD}
```



### 重要

ノードが割り当てられている場合は、ノードの kubelet を確認します。

## 軽減策

### デバッグログの情報

- この手順はオプションです。次のコマンドを実行して、Ceph クラスターの詳細なデバッグ情報を収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
```

## 6.3.7. CephMdsCacheUsageHigh

|    |   |
|----|---|
| 意味 | ストレージメタデータサービス (MDS) がキャッシュ使用量を <b>mds_health_cache_threshold</b> で指定されたターゲットしきい値、または <b>mds_cache_memory_limit</b> で設定されたキャッシュ制限の 150% 以下に維持できない場合、MDS は、キャッシュが大きすぎることを示す正常性アラートをモニターに送信します。その結果、MDS 関連の操作が遅くなります。 |
| 影響 | High  |

## 診断

MDS は、キャッシュ内の未使用のメタデータをトリムし、クライアントキャッシュ内のキャッシュされたアイテムを呼び出すことによって、`mds_cache_memory_limit` の予約値内に収めようとします。複数のクライアントがファイルにアクセスした結果、クライアントからの呼び出しが遅くなり、MDS がこの制限を超える可能性があります。

### 軽減策

MDS キャッシュに十分なメモリーがプロビジョニングされていることを確認します。`mds_cache_memory_limit` を増やすには、`ocs-storageCluster` で MDS Pod のメモリーリソースを更新する必要があります。次のコマンドを実行して、MDS Pod のメモリーを 16 GB などに設定します。

```
$ oc patch -n openshift-storage storagecluster ocs-storagecluster \
  --type merge \
  --patch '{"spec": {"resources": {"mds": {"limits": {"memory": "16Gi"}, "requests": {"memory": "16Gi"}}}}'
```

OpenShift Data Foundation は、`mds_cache_memory_limit` を MDS Pod のメモリー制限の半分に自動的に設定します。前のコマンドを使用してメモリーを 8 GB に設定した場合、Operator によって MDS キャッシュメモリーの制限が 4 GB に設定されます。

### 6.3.8. CephMdsCpuUsageHigh

|    |   |
|----|---|
| 意味 | ストレージメタデータサービス (MDS) は、ファイルシステムのメタデータを提供します。MDS は、ファイルの作成、名前変更、削除、および更新操作に不可欠です。MDS にはデフォルトで 2 つまたは 3 つの CPU が割り当てられます。メタデータ操作が多すぎない限り、問題は発生しません。このアラートがトリガーされるほどメタデータ操作の負荷が増加した場合、デフォルトの CPU 割り当てでは負荷に対応できません。CPU 割り当てを増やす必要があります。 |
| 影響 | High  |

### 診断

**Workloads** → **Pods** をクリックします。対応する MDS Pod を選択し、**Metrics** タブをクリックします。使用中の割り当てられている CPU が表示されます。デフォルトでは、使用中の CPU が、6 時間にわたって、割り当てられている CPU の 67% を占めている場合に、アラートが発せられます。その場合は、軽減策セクションの手順に従ってください。

### 軽減策

割り当てられた CPU を増やす必要があります。

次のコマンドを使用して、MDS に割り当てる CPU の数 (例: **8**) を設定します。

```
$ oc patch -n openshift-storage storagecluster ocs-storagecluster \
  --type merge \
  --patch '{"spec": {"resources": {"mds": {"limits": {"cpu": "8"}, "requests": {"cpu": "8"}}}}'
```

### 6.3.9. CephMdsMissingReplicas

|    |   |
|----|---|
| 意味 | ストレージメタデータサービス (MDS) に最低限必要なレプリカが利用できません。MDS は、メタデータのファイリングを担当します。MDS サービスの低下は、ストレージクラスターの動作 (CephFS ストレージクラスに関連) に影響を与える可能性があるため、できるだけ早く修正する必要があります。 |
| 影響 | High  |

## 診断

### Pod ステータス: 保留

1. リソースの問題、保留中の永続ボリューム要求 (PVC)、ノードの割り当て、および kubelet の問題を確認します。

```
$ oc project openshift-storage
oc get pod | grep rook-ceph-mds
```

2. 問題のある Pod として識別された Pod の変数として **MYPOD** を設定します。

```
# Examine the output for a {ceph-component} that is in the pending state, not running or
not ready
MYPOD=<pod_name>
<pod_name>
```

問題のある Pod として識別された Pod の名前を指定します。

3. リソースの制限または保留中の PVC を探します。それがない場合は、ノードの割り当てを確認します。

```
$ oc get pod/${MYPOD} -o wide
```

### Pod ステータス: 保留中や実行中ではないが、準備完了状態でもない

- readiness プロブを確認します。

```
$ oc describe pod/${MYPOD}
```

### Pod ステータス: 保留中ではないが、実行中でもない

- アプリケーションまたはイメージの問題を確認します。

```
$ oc logs pod/${MYPOD}
```



## 重要

ノードが割り当てられている場合は、ノードの kubelet を確認します。

## 軽減策

### デバッグログの情報

- この手順はオプションです。次のコマンドを実行して、Ceph クラスターのデバッグ情報を収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
```

### 6.3.10. CephMgrIsAbsent

|    |  |
|----|--|
| 意味 | Ceph マネージャーがクラスターの監視を実行していません。永続ボリューム要求 (PVC) の作成および削除リクエストは、できるだけ早く解決する必要があります。 |
| 影響 | High   |

### 診断

- rook-ceph-mgr** Pod に障害が発生していることを確認し、必要に応じて再起動します。Ceph mgr Pod の再起動が失敗した場合は、Pod の一般的なトラブルシューティングに従って問題を解決してください。

- Ceph mgr Pod に障害が発生していることを確認します。

```
$ oc get pods | grep mgr
```

- Ceph mgr Pod に関する情報を取得し、詳細を確認します。

```
$ oc describe pods/<pod_name>
```

**<pod\_name>**

前のステップの **rook-ceph-mgr** Pod 名を指定します。

リソースの問題に関連するエラーを分析します。

- Pod を削除し、Pod が再起動するまで待ちます。

```
$ oc get pods | grep mgr
```

Pod の一般的なトラブルシューティングでは、次の手順に従います。

#### Pod ステータス: 保留

- リソースの問題、保留中の永続ボリューム要求 (PVC)、ノードの割り当て、および kubelet の問題を確認します。

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

```
oc get pod | grep rook-ceph-mgr
```

- 問題のある Pod として識別された Pod の変数として **MYPOD** を設定します。

```
# Examine the output for a {ceph-component} that is in the pending state, not running or
not ready
MYPOD=<pod_name>
```

<pod\_name>

問題のある Pod として識別された Pod の名前を指定します。

- リソースの制限または保留中の PVC を探します。それがない場合は、ノードの割り当てを確認します。

```
$ oc get pod/${MYPOD} -o wide
```

**Pod ステータス: 保留中や実行中ではないが、準備完了状態でもない**

- readiness プロブを確認します。

```
$ oc describe pod/${MYPOD}
```

**Pod ステータス: 保留中ではないが、実行中でもない**

- アプリケーションまたはイメージの問題を確認します。

```
$ oc logs pod/${MYPOD}
```



### 重要

ノードが割り当てられている場合は、ノードの kubelet を確認します。

## 軽減策

### デバッグログの情報

- この手順はオプションです。次のコマンドを実行して、Ceph クラスターのデバッグ情報を収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
```

### 6.3.11. CephMgrIsMissingReplicas

|    |   |
|----|---|
| 意味 | このアラートを解決するには、Ceph マネージャーが消えた原因を特定し、必要に応じて再起動する必要があります。 |
| 影響 | High  |

## 診断

### Pod ステータス: 保留

- リソースの問題、保留中の永続ボリューム要求 (PVC)、ノードの割り当て、および kubelet

の問題を確認します。

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

```
oc get pod | grep rook-ceph-mgr
```

- 問題のある Pod として識別された Pod の変数として **MYPOD** を設定します。

```
# Examine the output for a {ceph-component} that is in the pending state, not running or
not ready
MYPOD=<pod_name>
```

<pod\_name>

問題のある Pod として識別された Pod の名前を指定します。

- リソースの制限または保留中の PVC を探します。それらがいない場合は、ノードの割り当てを確認します。

```
$ oc get pod/${MYPOD} -o wide
```

**Pod ステータス: 保留中や実行中ではないが、準備完了状態でもない**

- readiness プロブを確認します。

```
$ oc describe pod/${MYPOD}
```

**Pod ステータス: 保留中ではないが、実行中でもない**

- アプリケーションまたはイメージの問題を確認します。

```
$ oc logs pod/${MYPOD}
```



### 重要

ノードが割り当てられている場合は、ノードの kubelet を確認します。

軽減策

デバッグログの情報

- この手順はオプションです。次のコマンドを実行して、Ceph クラスターのデバッグ情報を収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
```

## 6.3.12. CephMonHighNumberOfLeaderChanges

|    |   |
|----|---|
| 意味 | Ceph クラスターには、ストレージクラスターに関する重要な情報を格納するモニター Pod の冗長セットがあります。モニター Pod は定期的に同期して、ストレージクラスターに関する情報を取得します。最新の情報を取得した最初のモニター Pod は、リーダーになります。その他のモニター Pod は、リーダーに問い合わせしてから同期プロセスを開始します。ネットワーク接続の問題や、1つ以上のモニター Pod で別の種類の問題が生じると、リーダーの異常な変更が発生します。この状況は、ストレージクラスターのパフォーマンスに悪影響を及ぼす可能性があります。 |
| 影響 | Medium  |



### 重要

ネットワークの問題を確認します。ネットワークに問題がある場合は、以下のトラブルシューティング手順に進む前に、OpenShift Data Foundation チームにエスカレートする必要があります。

### 診断

1. 影響を受けるモニター Pod のログを出力して、問題に関する詳細情報を収集します。

```
$ oc logs <rook-ceph-mon-X-yyyy> -n openshift-storage
```

**<rook-ceph-mon-X-yyyy>**

影響を受けるモニター Pod の名前を指定します。

2. または、Openshift Web コンソールを使用して、影響を受けるモニター Pod のログを開きます。考えられる原因に関する詳細情報がログに反映されます。
3. Pod の一般的なトラブルシューティング手順を実行します。

#### Pod ステータス: 保留

4. リソースの問題、保留中の永続ボリューム要求 (PVC)、ノードの割り当て、および kubelet の問題を確認します。

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

```
oc get pod | grep {ceph-component}
```

5. 問題のある Pod として識別された Pod の変数として **MYPOD** を設定します。

```
# Examine the output for a {ceph-component} that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

**<pod\_name>**

問題のある Pod として識別された Pod の名前を指定します。

- リソースの制限または保留中の PVC を探します。それがない場合は、ノードの割り当てを確認します。

```
$ oc get pod/${MYPOD} -o wide
```

Pod ステータス: 保留中や実行中ではないが、準備完了状態でもない

- readiness プロブを確認します。

```
$ oc describe pod/${MYPOD}
```

Pod ステータス: 保留中ではないが、実行中でもない

- アプリケーションまたはイメージの問題を確認します。

```
$ oc logs pod/${MYPOD}
```



### 重要

ノードが割り当てられている場合は、ノードの kubelet を確認します。

## 軽減策

### デバッグログの情報

- この手順はオプションです。次のコマンドを実行して、Ceph クラスターのデバッグ情報を収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
```

### 6.3.13. CephMonQuorumAtRisk

|    |  |
|----|--|
| 意味 | 複数の MON が連携して冗長性を提供します。各 MON は、メタデータのコピーを保持します。クラスターは 3 つの MON でデプロイされます。クォーラムとストレージ操作を実行するためには、2 つ以上の MON が稼働している必要があります。クォーラムが失われると、データへのアクセスが危険にさらされます。 |
| 影響 | High   |

### 診断

Ceph MON クォーラムを復元します。詳細は、[トラブルシューティングガイド](#) の **OpenShift Data Foundation** での **ceph-monitor クォーラムの復元** を参照してください。Ceph MON クォーラムの復元が失敗した場合は、Pod の一般的なトラブルシューティングに従って問題を解決してください。

Pod の一般的なトラブルシューティングでは、次の手順を実行します。

### Pod ステータス: 保留



1. リソースの問題、保留中の永続ボリューム要求 (PVC)、ノードの割り当て、および kubelet の問題を確認します。

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

```
oc get pod | grep rook-ceph-mon
```

2. 問題のある Pod として識別された Pod の変数として **MYPOD** を設定します。

```
# Examine the output for a {ceph-component} that is in the pending state, not running or
not ready
MYPOD=<pod_name>
```

<pod\_name>

問題のある Pod として識別された Pod の名前を指定します。

3. リソースの制限または保留中の PVC を探します。それらがいない場合は、ノードの割り当てを確認します。

```
$ oc get pod/${MYPOD} -o wide
```

**Pod ステータス: 保留中や実行中ではないが、準備完了状態でもない**

- readiness プロブを確認します。

```
$ oc describe pod/${MYPOD}
```

**Pod ステータス: 保留中ではないが、実行中でもない**

- アプリケーションまたはイメージの問題を確認します。

```
$ oc logs pod/${MYPOD}
```



### 重要

ノードが割り当てられている場合は、ノードの kubelet を確認します。

軽減策

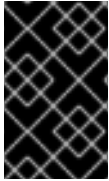
デバッグログの情報

- この手順はオプションです。次のコマンドを実行して、Ceph クラスターのデバッグ情報を収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
```

## 6.3.14. CephMonQuorumLost

|    |   |
|----|---|
| 意味 | Ceph クラスターには、ストレージクラスターに関する重要な情報を格納するモニター Pod の冗長セットがあります。モニター Pod は定期的に同期して、ストレージクラスターに関する情報を取得します。最新の情報を取得した最初のモニター Pod は、リーダーになります。その他のモニター Pod は、リーダーに問い合わせしてから同期プロセスを開始します。ネットワーク接続の問題や、1つ以上のモニター Pod で別の種類の問題が生じると、リーダーの異常な変更が発生します。この状況は、ストレージクラスターのパフォーマンスに悪影響を及ぼす可能性があります。 |
| 影響 | High  |



## 重要

ネットワークの問題を確認します。ネットワークに問題がある場合は、以下のトラブルシューティング手順に進む前に、OpenShift Data Foundation チームにエスカレートする必要があります。

## 診断

Ceph MON コーラムを復元します。詳細は、[トラブルシューティングガイドの OpenShift Data Foundation での ceph-monitor コーラムの復元](#) を参照してください。Ceph MON コーラムの復元が失敗した場合は、Pod の一般的なトラブルシューティングに従って問題を解決してください。

または、Pod の一般的なトラブルシューティングを実行します。

### Pod ステータス: 保留

- リソースの問題、保留中の永続ボリューム要求 (PVC)、ノードの割り当て、および kubelet の問題を確認します。

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

```
oc get pod | grep {ceph-component}
```

- 問題のある Pod として識別された Pod の変数として **MYPOD** を設定します。

```
# Examine the output for a {ceph-component} that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

```
<pod_name>
```

問題のある Pod として識別された Pod の名前を指定します。

- リソースの制限または保留中の PVC を探します。それらがいない場合は、ノードの割り当てを確認します。

```
$ oc get pod/${MYPOD} -o wide
```

### Pod ステータス: 保留中や実行中ではないが、準備完了状態でもない

- readiness プローブを確認します。

■

```
$ oc describe pod/${MYPOD}
```

Pod ステータス: 保留中ではないが、実行中でもない

- アプリケーションまたはイメージの問題を確認します。

```
$ oc logs pod/${MYPOD}
```



### 重要

ノードが割り当てられている場合は、ノードの kubelet を確認します。

## 軽減策

### デバッグログの情報

- この手順はオプションです。次のコマンドを実行して、Ceph クラスターの詳細なデバッグ情報を収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
```

## 6.3.15. CephMonVersionMismatch

|    |   |
|----|---|
| 意味 | 通常、このアラートは、アップグレードに長い時間がかかっているときにトリガーされません。 |
| 影響 | Medium                                      |

## 診断

**ocs-operator** サブスクリプションのステータスと Operator Pod の正常性を確認して、Operator のアップグレードが進行中かどうかを確認します。

1. **ocs-operator** サブスクリプションの正常性を確認します。

```
$ oc get sub $(oc get pods -n openshift-storage | grep -v ocs-operator) -n openshift-storage -o json | jq .status.conditions
```

ステータス条件のタイプ

は、**CatalogSourcesUnhealthy**、**InstallPlanMissing**、**InstallPlanPending**、および **InstallPlanFailed** です。各タイプのステータスが **False** である必要があります。

出力例:

```
[
  {
    "lastTransitionTime": "2021-01-26T19:21:37Z",
    "message": "all available catalogsources are healthy",
    "reason": "AllCatalogSourcesHealthy",
```

```

    "status": "False",
    "type": "CatalogSourcesUnhealthy"
  }
]

```

この出力例は、タイプ **CatalogSourcesUnHealthy** が **False** ステータスであることを示しています。これは、カタログソースが正常であることを意味します。

2. OCS Operator Pod のステータスを確認して、進行中の OCS Operator のアップグレードがあるかどうかを確認します。

```

$ oc get pod -n openshift-storage | grep ocs-operator OCSOP=$(oc get pod -n openshift-storage -o custom-columns=POD:.metadata.name --no-headers | grep ocs-operator) echo $OCSOP oc get pod/${OCSOP} -n openshift-storage oc describe pod/${OCSOP} -n openshift-storage

```

`ocs-operator` が進行中であることが確認された場合は、5 分間待てば、このアラートは自動的に解決されます。待機した場合、または別のエラーステータス条件が表示された場合は、トラブルシューティングを続けてください。

## 軽減策

### デバッグログの情報

- この手順はオプションです。次のコマンドを実行して、Ceph クラスターのデバッグ情報を収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
```

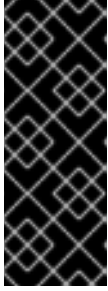
## 6.3.16. CephNodeDown

|    |   |
|----|---|
| 意味 | Ceph Pod を実行しているノードがダウンしています。Ceph はノード障害に対処するように設計されているため、ストレージ操作は引き続き機能しますが、別のノードがダウンしてストレージ機能に影響を与えるリスクを最小限に抑えるために、問題を解決することを推奨します。 |
| 影響 | Medium  |

### 診断

1. 実行中および障害が発生しているすべての Pod を一覧表示します。

```
oc -n openshift-storage get pods
```



## 重要

オブジェクトストレージデバイス (OSD) Pod が新しいノードでスケジュールされるように、OpenShift Data Foundation のリソース要件を満たしていることを確認します。Ceph クラスターが障害発生中で現在復旧中の OSD のデータを回復するため、これには数分かかる場合があります。この復旧の動作を確認するには、OSD Pod が新しいワーカーノードに正しく配置されていることを確認します。

2. 障害が発生していた OSD Pod が現在実行されているかどうかを確認します。

```
oc -n openshift-storage get pods
```

障害が発生していた OSD Pod がスケジュールされていない場合は、**describe** コマンドを使用してイベントを確認し、Pod が再スケジュールされなかった理由を特定します。

3. 障害が発生している OSD Pod のイベントに関する情報を取得します。

```
oc -n openshift-storage get pods | grep osd
```

4. 障害が発生している 1 つ以上の OSD Pod を見つけます。

```
oc -n openshift-storage describe pods/<osd_podname_from_the_previous_step>
```

イベントセクションで、リソースが満たされていないなど、障害の理由を探します。

さらに、**rook-ceph-toolbox** を使用して復旧を確認することもできます。このステップはオプションですが、大規模な Ceph クラスターの場合に役立ちます。ツールボックスにアクセスするには、次のコマンドを実行します。

```
TOOLS_POD=$(oc get pods -n openshift-storage -l app=rook-ceph-tools -o name)
oc rsh -n openshift-storage $TOOLS_POD
```

rsh コマンドプロンプトから次のコマンドを実行し、io セクションの下の "recovery" を確認します。

```
ceph status
```

5. 障害が発生したノードがあるかどうかを確認します。

- a. ワーカーノードのリストを取得し、ノードのステータスを確認します。

```
oc get nodes --selector='node-role.kubernetes.io/worker','!node-role.kubernetes.io/infra'
```

- b. **NotReady** ステータスのノードに対して describe を使用し、障害に関する詳細情報を取得します。

```
oc describe node <node_name>
```

## 軽減策

### デバッグログの情報

- この手順はオプションです。次のコマンドを実行して、Ceph クラスターのデバッグ情報を収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
```

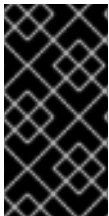
### 6.3.17. CephOSDCriticallyFull

|    |   |
|----|---|
| 意味 | オブジェクトストレージデバイス (OSD) の1つがほぼ満杯です。すぐにクラスターを拡張してください。 |
| 影響 | High  |

#### 診断

##### データの削除によるストレージスペースの解放

データを削除すると、クラスターは自己修復プロセスを通じてアラートを解決します。



#### 重要

これは、読み取り専用モードではないものの、ほぼ満杯の OpenShift Data Foundation クラスターにのみ適用されます。読み取り専用モードでは、データの削除を含む変更、つまり永続ボリューム要求 (PVC)、永続ボリューム (PV)、またはその両方の削除を含む変更が防止されます。

#### ストレージ容量の拡張

現在のストレージサイズは 1TB 未満です

まず拡張能力を評価する必要があります。1TB のストレージを追加するごとに、クラスターには最低限利用可能な 2 つの vCPU と 8 GiB メモリーを持つノードがそれぞれ 3 つ必要です。

アドオンを使用してストレージ容量を 4 TB に増やすことができます。クラスターは自己修復プロセスによってアラートを解決します。vCPU とメモリーリソースの最小要件が満たされていない場合は、クラスターにさらに 3 つのワーカーノードを追加する必要があります。

#### 軽減策

- 現在のストレージサイズが 4 TB の場合は、Red Hat サポートにお問い合わせください。
- オプション: 次のコマンドを実行して、Ceph クラスターのデバッグ情報を収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
```

### 6.3.18. CephOSDDiskNotResponding

|    |  |
|----|--|
| 意味 | ディスクデバイスが応答していません。すべてのオブジェクトストレージデバイス (OSD) が稼働しているかどうかを確認します。 |
| 影響 | Medium   |

## 診断

### Pod ステータス: 保留

1. リソースの問題、保留中の永続ボリューム要求 (PVC)、ノードの割り当て、および kubelet の問題を確認します。

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

```
$ oc get pod | grep rook-ceph
```

2. 問題のある Pod として識別された Pod の変数として **MYPOD** を設定します。

```
# Examine the output for a rook-ceph that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

<pod\_name>

問題のある Pod として識別された Pod の名前を指定します。

3. リソースの制限または保留中の PVC を探します。それがない場合は、ノードの割り当てを確認します。

```
$ oc get pod/${MYPOD} -o wide
```

### Pod ステータス: 保留中や実行中ではないが、準備完了状態でもない

- readiness プロブを確認します。

```
$ oc describe pod/${MYPOD}
```

### Pod ステータス: 保留中ではないが、実行中でもない

- アプリケーションまたはイメージの問題を確認します。

```
$ oc logs pod/${MYPOD}
```



#### 重要

- ノードが割り当てられている場合は、ノードの kubelet を確認します。
- 実行中の Pod の基本的な正常性、ノードアフィニティー、およびノードでのリソースの可用性が確認されたら、Ceph ツールを実行してストレージコンポーネントのステータスを取得します。

## 軽減策

### デバッグログの情報

- この手順はオプションです。次のコマンドを実行して、Ceph クラスターのデバッグ情報を収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
```

■

### 6.3.19. CephOSDiskUnavailable

|    |   |
|----|---|
| 意味 | いずれかのホストでディスクデバイスにアクセスできず、対応するオブジェクトストレージデバイス (OSD) が Ceph クラスターによって out とマークされています。このアラートは、Ceph ノードが 10 分以内に回復に失敗した場合に発生します。 |
| 影響 | High  |

#### 診断

##### 障害が発生したノードの特定

1. ワーカーノードのリストを取得し、ノードのステータスを確認します。

```
oc get nodes --selector='node-role.kubernetes.io/worker','!node-role.kubernetes.io/infra'
```

1. **NotReady** ステータスのノードに対して describe を使用し、障害に関する詳細情報を取得します。

```
oc describe node <node_name>
```

### 6.3.20. CephOSDFlapping

|    |   |
|----|---|
| 意味 | 過去 5 分間にストレージデーモンが 5 回再起動しました。Pod イベントまたは Ceph のステータスを確認し、原因を突き止めてください。 |
| 影響 | High  |

#### 診断

Red Hat Ceph Storage トラブルシューティングガイドの [OSD のフラップ](#) セクションの手順に従います。

または、Pod の一般的なトラブルシューティング手順に従います。

##### Pod ステータス: 保留

1. リソースの問題、保留中の永続ボリューム要求 (PVC)、ノードの割り当て、および kubelet の問題を確認します。

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

```
$ oc get pod | grep rook-ceph
```

2. 問題のある Pod として識別された Pod の変数として **MYPOD** を設定します。



```
# Examine the output for a rook-ceph that is in the pending state, not running or not ready
MYPOD=<pod_name>
```

<pod\_name>

問題のある Pod として識別された Pod の名前を指定します。

- リソースの制限または保留中の PVC を探します。それがない場合は、ノードの割り当てを確認します。

```
$ oc get pod/${MYPOD} -o wide
```

**Pod ステータス: 保留中や実行中ではないが、準備完了状態でもない**

- readiness プロブを確認します。

```
$ oc describe pod/${MYPOD}
```

**Pod ステータス: 保留中ではないが、実行中でもない**

- アプリケーションまたはイメージの問題を確認します。

```
$ oc logs pod/${MYPOD}
```



### 重要

- ノードが割り当てられている場合は、ノードの kubelet を確認します。
- 実行中の Pod の基本的な正常性、ノードアフィニティー、およびノードでのリソースの可用性が確認されたら、Ceph ツールを実行してストレージコンポーネントのステータスを取得します。

## 軽減策

### デバッグログの情報

- この手順はオプションです。次のコマンドを実行して、Ceph クラスターのデバッグ情報を収集します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
```

### 6.3.21. CephOSDNearFull

|    |  |
|----|--|
| 意味 | バックエンドストレージデバイスのオブジェクトストレージデバイス (OSD) の使用率が、ホストで 75% を超えました。 |
| 影響 | High   |

## 軽減策

クラスター内のスペースを解放するか、ストレージクラスターを拡張するか、Red Hat サポートにお問い合わせください。ストレージのスケーリングの詳細は、ストレージのスケーリング [ガイド](#) を参照してください。

### 6.3.22. CephOSDSlowOps

|    |   |
|----|---|
| 意味 | リクエストが遅いオブジェクトストレージデバイス (OSD) とは、 <b>osd_op_complaint_time</b> パラメーターで定義される時間内にキュー内の1秒あたりの I/O 操作 (IOPS) を処理しないすべての OSD です。デフォルトでは、このパラメーターは 30 秒に設定されています。 |
| 影響 | Medium  |

#### 診断

遅いリクエストの詳細は、Openshift コンソールを使用して取得できます。

1. OSD Pod ターミナルにアクセスし、次のコマンドを実行します。

```
$ ceph daemon osd.<id> ops
```

```
$ ceph daemon osd.<id> dump_historic_ops
```



#### 注記

OSD の番号は Pod 名に表示されます。たとえば、**rook-ceph-osd-0-5d86d4d8d4-zlqkx** では、**<0>** が OSD です。

#### 軽減策

OSD のリクエストが遅い主な原因は次のとおりです。

- ディスクドライブ、ホスト、ラック、ネットワークスイッチなどの基礎となるハードウェアまたはインフラストラクチャーに関する問題 Openshift 監視コンソールを使用して、クラスターリソースに関するアラートまたはエラーを見つけます。これにより、OSD の操作が遅くなる根本原因を把握できます。
- ネットワークの問題。これらの問題は、通常、OSD のフラップに関連しています。Red Hat Ceph Storage トラブルシューティングガイドの [OSD のフラップ](#) セクションを参照してください。
- ネットワークに問題がある場合は、OpenShift Data Foundation チームにエスカレートされません。
- システムの負荷。Openshift コンソールを使用して、OSD Pod と OSD を実行しているノードのメトリクスを確認します。より多くのリソースを追加または割り当てるのが、解決策になる可能性があります。

### 6.3.23. CephOSDVersionMismatch

|    |  |
|----|--|
| 意味 | 通常、このアラートは、アップグレードに長い時間がかかっているときにトリガーされま<br>す。 |
| 影響 | Medium   |

## 診断

**ocs-operator** サブスクリプションのステータスと Operator Pod の正常性を確認して、Operator のアップグレードが進行中かどうかを確認します。

1. **ocs-operator** サブスクリプションの正常性を確認します。

```
$ oc get sub $(oc get pods -n openshift-storage | grep -v ocs-operator) -n openshift-storage -o json | jq .status.conditions
```

ステータス条件のタイプ

は、**CatalogSourcesUnhealthy**、**InstallPlanMissing**、**InstallPlanPending**、および **InstallPlanFailed** です。各タイプのステータスが **False** である必要があります。

出力例:

```
[
  {
    "lastTransitionTime": "2021-01-26T19:21:37Z",
    "message": "all available catalogsources are healthy",
    "reason": "AllCatalogSourcesHealthy",
    "status": "False",
    "type": "CatalogSourcesUnhealthy"
  }
]
```

この出力例は、タイプ **CatalogSourcesUnHealthy** が **False** ステータスであることを示しています。これは、カタログソースが正常であることを意味します。

2. OCS Operator Pod のステータスを確認して、進行中の OCS Operator のアップグレードがあるかどうかを確認します。

```
$ oc get pod -n openshift-storage | grep ocs-operator OCSOP=$(oc get pod -n openshift-storage -o custom-columns=POD:.metadata.name --no-headers | grep ocs-operator) echo $OCSOP oc get pod/${OCSOP} -n openshift-storage oc describe pod/${OCSOP} -n openshift-storage
```

`ocs-operator` が進行中であることが確認された場合は、5分間待てば、このアラートは自動的に解決されます。待機した場合、または別のエラーステータス条件が表示された場合は、トラブルシューティングを続けてください。

### 6.3.24. CephPGRepairTakingTooLong

|    |                      |
|----|----------------------|
| 意味 | 自己修復操作に時間がかかりすぎています。 |
| 影響 | High                 |

## 診断

一貫性のない配置グループ (PG) を確認し、修正します。詳細は、Red Hat ナレッジベースソリューション [Ceph の一貫性のない配置グループの処理](#) を参照してください。

### 6.3.25. CephPoolQuotaBytesCriticallyExhausted

|    |   |
|----|---|
| 意味 | 1つ以上のプールがクォータに達したか、ほぼ達しています。このエラー状態を引き起こすための閾値は、 <b>mon_pool_quota_crit_threshold</b> 設定オプションで制御されます。 |
| 影響 | High  |

## 軽減策

プールクォータを調整します。次のコマンドを実行して、プールクォータを完全に削除するか、上下に調整します。

```
ceph osd pool set-quota <pool> max_bytes <bytes>
```

```
ceph osd pool set-quota <pool> max_objects <objects>
```

クォータ値を **0** に設定すると、クォータが無効になります。

### 6.3.26. CephPoolQuotaBytesNearExhaustion

|    |  |
|----|--|
| 意味 | 1つまたは複数のプールが、設定された満杯のしきい値に近づいています。この警告状態を引き起こす可能性のあるしきい値としては、 <b>mon_pool_quota_warn_threshold</b> 設定オプションがあります。 |
| 影響 | High   |

## 軽減策

プールクォータを調整します。次のコマンドを実行して、プールクォータを完全に削除するか、上下に調整します。

```
ceph osd pool set-quota <pool> max_bytes <bytes>
```

```
ceph osd pool set-quota <pool> max_objects <objects>
```

クォータ値を **0** に設定すると、クォータが無効になります。

### 6.3.27. OSDCPULoadHigh

|    |   |
|----|---|
| 意味 | OSD は Ceph Storage の重要なコンポーネントであり、データの配置と回復の管理を担当します。OSD コンテナ内の CPU 使用率が高い場合、処理要求が増加していることを示しています。その結果、ストレージパフォーマンスが低下する可能性があります。 |
| 影響 | High  |

## 診断

1. Kubernetes ダッシュボードまたは同等のダッシュボードに移動します。
2. **Workloads** セクションにアクセスし、OSD のアラートに関連する適切な Pod を選択します。
3. **Metrics** タブをクリックし、OSD コンテナの CPU メトリクスを表示します。
4. CPU 使用率が (アラート設定の指定のとおり) 一定期間にわたって 80% を超えていることを確認します。

## 軽減策

OSD の CPU 使用率が常に高い場合は、次の手順の実施を検討してください。

1. ストレージクラスターの全体的なパフォーマンスを評価し、CPU 使用率が高くなる原因となっている OSD を特定します。
2. 既存のノードに新しいストレージデバイスを追加するか、新しいストレージデバイスを備えた新しいノードを追加して、クラスター内の OSD の数を増やします。負荷の分散やシステム全体のパフォーマンス向上に役立つ方法については、[ストレージのスケーリングガイド](#)を確認してください。

### 6.3.28. PersistentVolumeUsageCritical

|    |  |
|----|--|
| 意味 | 永続ボリューム要求 (PVC) が最大容量に近づいており、タイムリーに対処しないとデータが失われる可能性があります。 |
| 影響 | High   |

## 軽減策

PVC サイズを拡張して容量を増やします。

1. OpenShift Web コンソールにログインします。
2. **Storage** → **PersistentVolumeClaim** をクリックします。
3. **Project** ドロップダウンリストから **openshift-storage** を選択します。
4. 拡張したい PVC で、**Action menu (⋮)** → **Expand PVC** をクリックします。
5. **Total size** を目的のサイズに更新します。
6. **Expand** をクリックします。

または、スペースを占有している可能性のある不要なデータを削除することもできます。

### 6.3.29. PersistentVolumeUsageNearFull

|    |  |
|----|--|
| 意味 | 永続ボリューム要求 (PVC) が最大容量に近づいており、タイムリーに対処しないとデータが失われる可能性があります。 |
| 影響 | High   |

## 軽減策

PVC サイズを拡張して容量を増やします。

1. OpenShift Web コンソールにログインします。
2. **Storage** → **PersistentVolumeClaim** をクリックします。
3. **Project** ドロップダウンリストから **openshift-storage** を選択します。
4. 拡張したい PVC で、**Action menu (⋮)** → **Expand PVC** をクリックします。
5. **Total size** を目的のサイズに更新します。
6. **Expand** をクリックします。

または、スペースを占有している可能性のある不要なデータを削除することもできます。

## 6.4. 異常なバケットのエラーコードの検索

### 手順

1. OpenShift Web コンソールで、**Storage** → **Object Storage** をクリックします。
2. **Object Bucket Claims** タブをクリックします。
3. **Bound** 状態ではないオブジェクトバケット要求(OBC)を検索し、クリックします。
4. **Events** タブをクリックし、次のいずれかを実行します。
  - バケットの現在の状態についてヒントの可能性があるイベントを見つけます。
  - **YAML** タブをクリックし、YAML の status セクションおよび mode セクションの関連エラーを探します。

OBC が **Pending** 状態にある場合、そのエラーが製品ログに表示される可能性があります。ただし、この場合は、指定したすべての変数が正確であることを確認することが推奨されます。

## 6.5. 正常でない NAMESPACE ストアリソースのエラーコードの検索

### 手順

1. OpenShift Web コンソールで、**Storage** → **Object Storage** をクリックします。
2. **Namespace Store** タブをクリックします。
3. **Bound** 状態ではない namespace ストアリソースを探し、これをクリックします。
4. **Events** タブをクリックし、次のいずれかを実行します。
  - リソースの現在の状態についてヒントの可能性があるイベントを見つけます。
  - **YAML** タブをクリックし、YAML の status セクションおよび mode セクションの関連エラーを探します。

## 6.6. POD のリカバリー

一部の問題により最初のノード (例: **NODE1**) が NotReady 状態になると、ReadWriteOnce (RWO) アクセスモードで PVC を使用するホストされた Pod は、2 つ目のノード (例: **NODE2**) に移行しようとしませんが、multi-attach エラーにより停止します。このような場合には、以下の手順に従って MON、OSD、およびアプリケーション Pod を回復できます。

### 手順

1. (AWS または vSphere 側から) **NODE1** の電源をオフにし、**NODE1** が完全に停止していることを確認します。
2. 以下のコマンドを使用して **NODE1** で Pod を強制的に削除します。

```
$ oc delete pod <pod-name> --grace-period=0 --force
```

## 6.7. EBS ボリュームの割り当て解除からのリカバリー

OSD ディスクがある OSD または MON Elastic Block Storage (EBS) ボリュームがワーカー Amazon EC2 インスタンスからアタッチ解除すると、ボリュームは1分または2分以内に自動的に再度アタッチされます。ただし、OSD Pod は **CrashLoopBackOff** 状態になります。Pod を回復して **Running** 状態に戻すには、EC2 インスタンスを再起動する必要があります。

## 6.8. ROOK-CEPH-OPERATOR のデバッグログの有効化および無効化

rook-ceph-operator のデバッグログを有効にし、問題のトラブルシューティングに役立つ障害情報を取得します。

### 手順

#### デバッグログの有効化

1. rook-ceph-operator の configmap を編集します。

```
$ oc edit configmap rook-ceph-operator-config
```

2. **ROOK\_LOG\_LEVEL: DEBUG** パラメーターを **rook-ceph-operator-config** yaml ファイルに追加して、rook-ceph-operator のデバッグログを有効にします。

```
...
data:
  # The logging level for the operator: INFO | DEBUG
  ROOK_LOG_LEVEL: DEBUG
```

rook-ceph-operator ログはデバッグ情報で設定されます。

#### デバッグログの無効化

1. rook-ceph-operator の configmap を編集します。

```
$ oc edit configmap rook-ceph-operator-config
```

2. **ROOK\_LOG\_LEVEL: INFO** パラメーターを **rook-ceph-operator-config** yaml ファイルに追加して、rook-ceph-operator のデバッグログを無効にします。

```
...
data:
  # The logging level for the operator: INFO | DEBUG
  ROOK_LOG_LEVEL: INFO
```

## 6.9. CEPH モニター数が少ないというアラートの解決

**CephMonLowNumber** アラートは、Ceph モニターの数が少ないことを示すために、OpenShift Web コンソールの通知パネルまたは Alert Center に表示されます。表示されるのは、内部モードのデプロイメントに5つ以上のノード、ラック、またはルームがあり、デプロイメント内に5つ以上の障害ドメインがある場合です。Ceph モニターの数を増やすと、クラスターの可用性が向上します。

### 手順

1. OpenShift Web コンソールの通知パネルまたは Alert Center の **CephMonLowNumber** アラートで、**Configure** をクリックします。
2. **Configure Ceph Monitor** ポップアップで、**Update count** をクリックします。ポップアップには、障害ゾーンの数に応じた推奨モニター数が表示されます。
3. **Configure CephMon** ポップアップで、推奨値に基づいてモニター数の値を更新し、**Save changes** をクリックします。

## 6.10. 正常でないブロックリストノードのトラブルシューティング

### 6.10.1. ODFRBDClientBlocked

|    |  |
|----|--|
| 意味 | このアラートは、Kubernetes クラスター内の特定のノード上で RADOS Block Device (RBD) クライアントが Ceph によってブロックされる可能性があることを示します。ブロックリストは、 <b>ocs_rbd_client_blocklisted metric</b> がノードの値1を報告すると発生します。さらに、同じノードに <b>CreateContainerError</b> 状態の Pod があります。ブロックリストに登録すると、RBD を使用する Persistent Volume Claims (PVC) のファイルシステムが読み取り専用になる可能性があります。このアラートを調査して、ストレージクラスターの中断を防ぐことが重要です。 |
| 影響 | High   |

### 診断

RBD クライアントのブロックリストへの登録は、ネットワークやクラスターの速度など、いくつかの要因によって発生する可能性があります。場合によっては、競合する3つのクライアント(ワークロード、ミラーデーモン、マネージャー/スケジューラー)間の排他的ロックの競合により、ブロックリストが作成される可能性があります。

### 軽減策

1. ブロックリストに登録されたノードをテイントする: Kubernetes では、別のノードへの Pod のエビクションをトリガーするために、ブロックリストに登録されたノードをテイントすること



を検討します。このアプローチは、アンマウント/アンマッピングプロセスが正常に進行するという前提に基づいています。Pod が正常に削除されると、ブロックリストに登録されたノードのテイントが解除され、ブロックリストをクリアできるようになります。その後、Pod をテイントされていないノードに戻すことができます。

2. ブロックリストに登録されたノードを再起動する: ノードをテイントし、Pod を削除してもブロックリストに登録される問題が解決しない場合は、ブロックリストに登録されたノードの再起動を試みることができます。この手順は、ブロックリストの原因となっている根本的な問題を軽減し、通常の機能を復元するのに役立つ場合があります。



### 重要

ストレージクラスターへのさらなる影響を回避するには、ブロックリストの問題を迅速に調査して解決することが不可欠です。

## 第7章 ローカルストレージ OPERATOR デプロイメントの確認

ローカルストレージ Operator を使用する Red Hat OpenShift Data Foundation クラスターは、ローカルストレージデバイスを使用してデプロイされます。ローカルストレージデバイスを使用して既存のクラスターが OpenShift Data Foundation でデプロイされているかどうかを確認するには、以下の手順に従います。

### 前提条件

- OpenShift Data Foundation が **openshift-storage** namespace にインストールされ、実行されている。

### 手順

OpenShift Data Foundation クラスターの Persistent Volume Claim(永続ボリューム要求、PVC)に関連付けられたストレージクラスをチェックすることにより、ローカルストレージデバイスを使用してクラスターがデプロイされているかどうかを確認できます。

1. 以下のコマンドを使用して、OpenShift Data Foundation クラスターの PVC に関連付けられたストレージクラスを確認します。

```
$ oc get pvc -n openshift-storage
```

2. 出力を確認します。ローカルストレージ Operator を含むクラスターの場合、**ocs-deviceset** に関連付けられた PVC はストレージクラス **localblock** を使用します。出力は以下の例のようになります。

| NAME                    | STATUS                      | VOLUME                                   | CAPACITY | ACCESS |
|-------------------------|-----------------------------|--|----------|--------|
| MODES                   | STORAGECLASS                | AGE                                      |          |        |
| db-noobaa-db-0          | Bound                       | pvc-d96c747b-2ab5-47e2-b07e-1079623748d8 | 50Gi     |        |
| RWO                     | ocs-storagecluster-ceph-rbd | 114s                                     |          |        |
| ocs-deviceset-0-0-lzfrd | Bound                       | local-pv-7e70c77c                        | 1769Gi   | RWO    |
| localblock              | 2m10s                       |  |          |        |
| ocs-deviceset-1-0-7rggl | Bound                       | local-pv-b19b3d48                        | 1769Gi   | RWO    |
| localblock              | 2m10s                       |  |          |        |
| ocs-deviceset-2-0-znhk8 | Bound                       | local-pv-e9f22cdc                        | 1769Gi   | RWO    |
| localblock              | 2m10s                       |  |          |        |

### 関連情報

- [Deploying OpenShift Data Foundation using local storage devices on VMware](#)
- [Deploying OpenShift Data Foundation using local storage devices on Red Hat Virtualization](#)
- [Deploying OpenShift Data Foundation using local storage devices on bare metal](#)
- [Deploying OpenShift Data Foundation using local storage devices on IBM Power](#)

## 第8章 故障したまたは不要な CEPH OBJECT STORAGE デバイスの削除

障害が発生した、または不要な Ceph OSD (オブジェクトストレージデバイス) は、ストレージインフラストラクチャーのパフォーマンスに影響を与えます。したがって、ストレージクラスターの信頼性と回復力を向上させるには、障害が発生した、または不要な Ceph OSD を削除する必要があります。

失敗した Ceph OSD または不要な Ceph OSD を削除する場合は、次の手順を実行します。

1. Ceph の健全性ステータスを確認します。  
詳細は、[Ceph クラスターが正常であることの確認](#) を参照してください。
2. OSD のプロビジョニングに基づいて、失敗した、または不要な Ceph OSD を削除します。  
参照:
  - [動的にプロビジョニングされた Red Hat OpenShift Data Foundation で失敗した、または不要な Ceph OSD を削除。](#)
  - [ローカルストレージデバイスを使用してプロビジョニングされた、失敗したまたは不要な Ceph OSD を削除](#)

ローカルディスクを使用している場合は、古い OSD を削除した後、これらのディスクを再利用できません。

### 8.1. CEPH クラスターが正常であることの確認

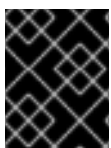
ストレージの健全性は、**Block**、**File**、**Object** のダッシュボードに表示されます。

#### 手順

1. OpenShift Web コンソールで、**Storage → Data Foundation** をクリックします。
2. **Overview** タブの **Status** カードで **Storage System** をクリックし、表示されたポップアップからストレージシステムリンクをクリックします。
3. **Block and File** タブの **Status** カードで、**Storage Cluster** に緑色のチェックマークが表示されていることを確認します。
4. **Details** カードで、クラスター情報が表示されていることを確認します。

### 8.2. 動的にプロビジョニングされた RED HAT OPENSIFT DATA FOUNDATION で失敗した、または不要な CEPH OSD を削除。

動的にプロビジョニングされた Red Hat OpenShift Data Foundation で障害が発生した、または不要な Ceph オブジェクトストレージデバイス (OSD) を削除するには、次の手順に従います。



#### 重要

クラスターのスケールダウンは、Red Hat サポートチームの支援がある場合にのみサポートされます。



## 警告

- Ceph コンポーネントが正常な状態ではないときに OSD を削除すると、データが失われる可能性があります。
- 2 つ以上の OSD を同時に削除すると、データが失われます。

## 前提条件

- Ceph が正常かどうかを確認します。詳細は、[Ceph クラスタが正常であることの確認](#) を参照してください。
- アラートが発生していないか、再構築プロセスが進行中ではないことを確認してください。

## 手順

1. OSD デプロイメントをスケールダウンします。

```
# oc scale deployment rook-ceph-osd-<osd-id> --replicas=0
```

2. Ceph OSD を削除するための **osd-prepare** Pod を取得します。

```
# oc get deployment rook-ceph-osd-<osd-id> -oyaml | grep ceph.rook.io/pvc
```

3. **osd-prepare** Pod を削除します。

```
# oc delete -n openshift-storage pod rook-ceph-osd-prepare-<pvc-from-above-command>-<pod-suffix>
```

4. 失敗した OSD をクラスターから削除します。

```
# failed_osd_id=<osd-id>

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

ここで、**FAILED\_OSD\_ID** は、**rook-ceph-osd** 接頭辞の直後の Pod 名の整数です。

5. ログを確認して、OSD が正常に削除されたことを確認します。

```
# oc logs -n openshift-storage ocs-osd-removal-${failed_osd_id}-<pod-suffix>
```

6. オプション : OpenShift Container Platform の **ocs-osd-removal-job** Pod から **cephosd:osd.0 is NOT ok to destroy** としてエラーが発生した場合は、[エラー cephosd:osd.0 is NOT ok to destroy のトラブルシューティング](#) を参照してください。

7. OSD デプロイメントを削除します。

```
# oc delete deployment rook-ceph-osd-<osd-id>
```

## 検証手順

- OSD が正常に削除されたかどうかを確認するには、次のコマンドを実行します。

```
# oc get pod -n openshift-storage ocs-osd-removal-$(failed_osd_id)-<pod-suffix>
```

このコマンドはステータスを **Completed** として返す必要があります。

## 8.3. ローカルストレージデバイスを使用してプロビジョニングされた、失敗したまたは不要な CEPH OSD を削除

以下の手順に従って、ローカルストレージデバイスを使用して、Ceph がプロビジョニングした障害のあるオブジェクトまたは不要なオブジェクトストレージデバイス (OSD) を削除できます。



### 重要

クラスターのスケールダウンは、Red Hat サポートチームの支援がある場合にのみサポートされます。



### 警告

- Ceph コンポーネントが正常な状態ではないときに OSD を削除すると、データが失われる可能性があります。
- 2 つ以上の OSD を同時に削除すると、データが失われます。

## 前提条件

- Ceph が正常かどうかを確認します。詳細は、[Ceph クラスターが正常であることの確認](#) を参照してください。
- アラートが発生していないか、再構築プロセスが進行中ではないことを確認してください。

## 手順

1. OSD デプロイメント上のレプリカを 0 にスケールして、OSD を強制的にマークダウンします。障害により OSD がすでにダウンしている場合は、この手順をスキップできます。

```
# oc scale deployment rook-ceph-osd-<osd-id> --replicas=0
```

2. 失敗した OSD をクラスターから削除します。

```
# failed_osd_id=<osd_id>
```

```
# oc process -n openshift-storage ocs-osd-removal -p FAILED_OSD_IDS=$(failed_osd_id) |
oc create -f -
```

ここで、**FAILED\_OSD\_ID** は、**rook-ceph-osd** 接頭辞の直後の Pod 名の整数です。

3. ログを確認して、OSD が正常に削除されたことを確認します。

```
# oc logs -n openshift-storage ocs-osd-removal-$(failed_osd_id)-<pod-suffix>
```

4. オプション : OpenShift Container Platform の **ocs-osd-removal-job** Pod から **cephosd:osd.0 is NOT ok to destroy** としてエラーが発生した場合は、エラー **cephosd:osd.0 is NOT ok to destroy** のトラブルシューティングを参照してください。
5. 障害のある OSD に関連付けられた Persistent Volume Claim (永続ボリューム要求、PVC) リソースを削除します。
  - a. 失敗した OSD に関連付けられた **PVC** を取得します。

```
# oc get -n openshift-storage -o yaml deployment rook-ceph-osd-<osd-id> | grep ceph.rook.io/pvc
```

- b. PVC に関連付けられた **persistent volume (PV)** を取得します。

```
# oc get -n openshift-storage pvc <pvc-name>
```

- c. 障害が発生したデバイス名を取得します。

```
# oc get pv <pv-name-from-above-command> -oyaml | grep path
```

- d. 失敗した OSD に関連付けられた **prepare-pod** を取得します。

```
# oc describe -n openshift-storage pvc ocs-deviceset-0-0-nvs68 | grep Mounted
```

- e. 関連付けられた PVC を削除する前に **osd-prepare pod** を削除します。

```
# oc delete -n openshift-storage pod <osd-prepare-pod-from-above-command>
```

- f. 失敗した OSD に関連付けられた **PVC** を削除します。

```
# oc delete -n openshift-storage pvc <pvc-name-from-step-a>
```

6. 障害が発生したデバイスエントリーを **LocalVolume custom resource (CR)** から削除します。

- a. 障害が発生したデバイスを使用してノードにログインします。

```
# oc debug node/<node_with_failed_osd>
```

- b. 障害が発生したデバイス名の `/dev/disk/by-id/<id>` を記録します。

```
# ls -alh /mnt/local-storage/localblock/
```

7. オプション: OSD のプロビジョニングにローカルストレージオペレーターが使用されている場合は、`{osd-id}` を使用してマシンにログインし、デバイスのシンボリックリンクを削除します。

```
# oc debug node/<node_with_failed_osd>
```

- a. 障害が発生したデバイス名の OSD シンボリックリンクを取得します。

```
# ls -alh /mnt/local-storage/localblock
```

- b. symlink を削除します。

```
# rm /mnt/local-storage/localblock/<failed-device-name>
```

8. OSD に関連付けられた PV を削除します。

```
# oc delete pv <pv-name>
```

### 検証手順

- OSD が正常に削除されたかどうかを確認するには、次のコマンドを実行します。

```
#oc get pod -n openshift-storage ocs-osd-removal-$$<failed_osd_id>-<pod-suffix>
```

このコマンドはステータスを **Completed** として返す必要があります。

## 8.4. 失敗した、あるいは不要な CEPH OSD の削除中のエラー **CEPH OSD: OSD.0 IS NOT OK TO DESTROY** のトラブルシューティング

このエラーが、OpenShift Container Platform の **ocs-osd-removal-job** から **cephosd:osd.0 is NOT ok to destroy** として発生した場合は、**FORCE\_OSD\_REMOVAL** オプションを指定して Object Storage Device (OSD) 削除を実行し、OSD を破棄状態に移行します。

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



### 注記

**FORCE\_OSD\_REMOVAL** オプションは、すべての PG がアクティブな状態にある場合にのみ使用する必要があります。そうでない場合、PG はバックフィルを完了するか、PG がアクティブであることを確認するためにさらに調査する必要があります。

## 第9章 トラブルシューティングおよびアンインストール時の残りのリソースの削除

Operator によって管理されるカスタムリソースの一部は、必要なすべてのクリーンアップタスクを実行しても、ファイナライザーで **Terminating** ステータスのままになり、完了まで待機する必要がある場合があります。このような場合には、このようなリソースを強制的に削除する必要があります。これを実行しないと、すべてのアンインストール手順を実行しても、リソースは **Terminating** 状態のままになります。

1. openshift-storage namespace が削除時に **Terminating** 状態のままかどうかを確認します。

```
$ oc get project -n <namespace>
```

出力:

```
NAME          DISPLAY NAME  STATUS
openshift-storage  Terminating
```

2. コマンド出力の **STATUS** セクションで **NamespaceFinalizersRemaining** および **NamespaceContentRemaining** メッセージの有無を確認し、リスト表示される各リソースについて以下の手順を実行します。

```
$ oc get project openshift-storage -o yaml
```

出力例:

```
status:
  conditions:
  - lastTransitionTime: "2020-07-26T12:32:56Z"
    message: All resources successfully discovered
    reason: ResourcesDiscovered
    status: "False"
    type: NamespaceDeletionDiscoveryFailure
  - lastTransitionTime: "2020-07-26T12:32:56Z"
    message: All legacy kube types successfully parsed
    reason: ParsedGroupVersions
    status: "False"
    type: NamespaceDeletionGroupVersionParsingFailure
  - lastTransitionTime: "2020-07-26T12:32:56Z"
    message: All content successfully deleted, may be waiting on finalization
    reason: ContentDeleted
    status: "False"
    type: NamespaceDeletionContentFailure
  - lastTransitionTime: "2020-07-26T12:32:56Z"
    message: 'Some resources are remaining: cephobjectstoreusers.ceph.rook.io has
      1 resource instances'
    reason: SomeResourcesRemain
    status: "True"
    type: NamespaceContentRemaining
  - lastTransitionTime: "2020-07-26T12:32:56Z"
    message: 'Some content in the namespace has finalizers remaining:
      cephobjectstoreuser.ceph.rook.io
      in 1 resource instances'
```



```
reason: SomeFinalizersRemain
status: "True"
type: NamespaceFinalizersRemaining
```

3. 先の手順に記載されている残りのすべてのリソースを削除します。  
削除する各リソースについて、以下を実行します。

- a. 削除する必要があるリソースの種類を取得します。上記の出力のメッセージを確認します。  
例:

```
message: Some content in the namespace has finalizers remaining:
cephobjectstoreuser.ceph.rook.io
```

ここで、`cephobjectstoreuser.ceph.rook.io` はオブジェクトの種類です。

- b. オブジェクトの種類に対応するオブジェクト名を取得します。

```
$ oc get <Object-kind> -n <project-name>
```

例:

```
$ oc get cephobjectstoreusers.ceph.rook.io -n openshift-storage
```

出力例:

```
NAME                                AGE
noobaa-ceph-objectstore-user      26h
```

- c. リソースにパッチを適用します。

```
$ oc patch -n <project-name> <object-kind>/<object-name> --type=merge -p
'{"metadata":{"finalizers":null}}'
```

以下に例を示します。

```
$ oc patch -n openshift-storage cephobjectstoreusers.ceph.rook.io/noobaa-ceph-
objectstore-user \
--type=merge -p '{"metadata":{"finalizers":null}}'
```

出力:

```
cephobjectstoreuser.ceph.rook.io/noobaa-ceph-objectstore-user patched
```

4. `openshift-storage` プロジェクトが削除されていることを確認します。

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

出力:

```
Error from server (NotFound): namespaces "openshift-storage" not found
```

問題が解決しない場合は、[Red Hat サポート](#) にご連絡ください。

## 第10章 外部モードでの CEPHFS PVC 作成のトラブルシューティング

Red Hat Ceph Storage クラスターを 4.1.1 以前のバージョンから最新リリースに更新し、これが新規にデプロイされたクラスターではない場合は、Red Hat Ceph Storage クラスターで CephFS プールのアプリケーションタイプを手動で設定し、外部モードで CephFS 永続ボリューム要求 (PVC) の作成を有効にする。

1. CephFS pvc が **Pending** ステータスで停止しているかどうかを確認します。

```
# oc get pvc -n <namespace>
```

出力例:

```
NAME                STATUS  VOLUME
CAPACITY ACCESS MODES  STORAGECLASS          AGE
ngx-fs-pxkncix20-pod  Pending
                                ocs-external-storagecluster-cephfs 28h
[...]
```

2. **oc description** コマンドの出力を確認して、各 PVC のイベントを確認します。予想されるエラーメッセージは **cephfs\_metadata/csi.volumes.default/csi.volume.pvc-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx: (1) Operation not permitted** です。

```
# oc describe pvc ngx-fs-pxkncix20-pod -n nginx-file
```

出力例:

```
Name:          ngx-fs-pxkncix20-pod
Namespace:     nginx-file
StorageClass:  ocs-external-storagecluster-cephfs
Status:        Pending
Volume:
Labels:        <none>
Annotations:   volume.beta.kubernetes.io/storage-provisioner: openshift-storage.cephfs.csi.ceph.com
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:
Access Modes:
VolumeMode:   Filesystem
Mounted By:    ngx-fs-oyoe047v2bn2ka42jfgg-pod-hqzfh
Events:
  Type    Reason          Age          From
  Message
  ----    -
  -----
Warning ProvisioningFailed 107m (x245 over 22h) openshift-storage.cephfs.csi.ceph.com_csi-cephfspugin-provisioner-5f8b66cc96-hvcqp_6b7044afc904-4795-9ce5-bf0cf63cc4a4
(combined from similar events): failed to provision volume with StorageClass "ocs-external-storagecluster-cephfs": rpc error: code = Internal desc = error (an error (exit status 1) occurred while
running rados args: [-m 192.168.13.212:6789,192.168.13.211:6789,192.168.13.213:6789 --id csi-cephfs-provisioner --keyfile=stripped -c /etc/ceph/ceph.conf -p cephfs_metadata
```

```
getomapval
csi.volumes.default csi.volume.pvc-1ac0c6e6-9428-445d-bbd6-1284d54ddb47 /tmp/omap-
get-186436239 --namespace=csi]) occurred, command output streams is ( error getting
omap value
cephfs_metadata/csi.volumes.default/csi.volume.pvc-1ac0c6e6-9428-445d-bbd6-
1284d54ddb47: (1) Operation not permitted)
```

3. **<cephfs metadata pool name>** (ここでは **cephfs\_metadata**) および **<cephfs data pool name>** (ここでは **cephfs\_data**) の設定を確認します。コマンドを実行するには、**jq** を Red Hat Ceph Storage クライアントノードに事前にインストールする必要があります。

```
# ceph osd pool ls detail --format=json | jq '[] | select(.pool_name| startswith("cephfs")) |
.pool_name, .application_metadata' "cephfs_data"
{
  "cephfs": {}
}
"cephfs_metadata"
{
  "cephfs": {}
}
```

4. CephFS プールのアプリケーションタイプを設定します。

- Red Hat Ceph Storage クライアントノードで以下のコマンドを実行します。

```
# ceph osd pool application set <cephfs metadata pool name> cephfs metadata cephfs
```

```
# ceph osd pool application set <cephfs data pool name> cephfs data cephfs
```

5. 設定が適用されているかどうかを確認します。

```
# ceph osd pool ls detail --format=json | jq '[] | select(.pool_name| startswith("cephfs")) |
.pool_name, .application_metadata' "cephfs_data"
{
  "cephfs": {
    "data": "cephfs"
  }
}
"cephfs_metadata"
{
  "cephfs": {
    "metadata": "cephfs"
  }
}
```

6. CephFS PVC のステータスを再度確認します。PVC が **Bound** 状態になるはずです。

```
# oc get pvc -n <namespace>
```

出力例:

```
NAME                STATUS  VOLUME
CAPACITY ACCESS MODES  STORAGECLASS          AGE
ngx-fs-pxknkcix20-pod  Bound  pvc-1ac0c6e6-9428-445d-bbd6-1284d54ddb47
```

|       |     |                                    |     |
|-------|-----|------------------------------------|-----|
| 1Mi   | RWO | ocs-external-storagecluster-cephfs | 29h |
| [...] |     |                                    |     |

## 第11章 OPENSIFT DATA FOUNDATION でのモニター POD の復元

3 つすべてが停止している場合はモニター Pod を復元し、OpenShift Data Foundation がモニター Pod を自動的に復元できない場合は、モニター Pod を復元します。



### 注記

これは障害復旧手順であり、Red Hat サポートチームの指導の下で実行する必要があります。Red Hat サポートチーム ([Red Hat サポート](#)) にお問い合わせください。

### 手順

1. **rook-ceph-operator** および **ocs Operator** デプロイメントをスケールダウンします。

```
# oc scale deployment rook-ceph-operator --replicas=0 -n openshift-storage
```

```
# oc scale deployment ocs-operator --replicas=0 -n openshift-storage
```

2. **openshift-storage** namespace ですべてのデプロイメントのバックアップを作成します。

```
# mkdir backup
```

```
# cd backup
```

```
# oc project openshift-storage
```

```
# for d in $(oc get deployment|awk -F' ' '{print $1}'|grep -v NAME); do echo $d;oc get deployment $d -o yaml > oc_get_deployment.${d}.yaml; done
```

3. オブジェクトストレージデバイス (OSD) デプロイメントにパッチを適用して **livenessProbe** パラメーターを削除し、コマンドパラメーターを **sleep** として実行します。

```
# for i in $(oc get deployment -l app=rook-ceph-osd -oname);do oc patch ${i} -n openshift-storage --type=json' -p [{"op":"remove", "path":"/spec/template/spec/containers/0/livenessProbe"}] ; oc patch ${i} -n openshift-storage -p '{"spec": {"template": {"spec": {"containers": [{"name": "osd", "command": ["sleep", "infinity"], "args": []}]}}}' ; done
```

4. すべての OSD から **monstore** クラスタマップを取得します。

- a. **recover\_mon.sh** スクリプトを作成します。

```
#!/bin/bash
ms=/tmp/monstore
```

```
rm -rf $ms
mkdir $ms
```

```
for osd_pod in $(oc get po -l app=rook-ceph-osd -oname -n openshift-storage); do
```

```

echo "Starting with pod: $osd_pod"

podname=$(echo $osd_pod|sed 's/pod\\//g')
oc exec $osd_pod -- rm -rf $ms
oc cp $ms $podname:$ms

rm -rf $ms
mkdir $ms

echo "pod in loop: $osd_pod ; done deleting local dirs"

oc exec $osd_pod -- ceph-objectstore-tool --type bluestore --data-path
/var/lib/ceph/osd/ceph-$(oc get $osd_pod -ojsonpath='{
.metadata.labels.ceph_daemon_id }') --op update-mon-db --no-mon-config --mon-store-
path $ms
echo "Done with COT on pod: $osd_pod"

oc cp $podname:$ms $ms

echo "Finished pulling COT data from pod: $osd_pod"
done

```

- b. **recover\_mon.sh** スクリプトを実行します。

```
# chmod +x recover_mon.sh
```

```
# ./recover_mon.sh
```

5. MON デプロイメントにパッチを適用し、コマンドパラメーターを **sleep** として実行します。

- a. MON デプロイメントを編集します。

```
# for i in $(oc get deployment -l app=rook-ceph-mon -oname);do oc patch ${i} -n
openshift-storage -p '{"spec": {"template": {"spec": {"containers": [{"name": "mon",
"command": ["sleep", "infinity"], "args": []}]}}}}'; done
```

- b. MON デプロイメントにパッチを適用して、**initialDelaySeconds** を増やします。

```
# oc get deployment rook-ceph-mon-a -o yaml | sed "s/initialDelaySeconds:
10/initialDelaySeconds: 2000/g" | oc replace -f -
```

```
# oc get deployment rook-ceph-mon-b -o yaml | sed "s/initialDelaySeconds:
10/initialDelaySeconds: 2000/g" | oc replace -f -
```

```
# oc get deployment rook-ceph-mon-c -o yaml | sed "s/initialDelaySeconds:
10/initialDelaySeconds: 2000/g" | oc replace -f -
```

6. 以前に取得した **monstore** を **mon-a** Pod にコピーします。

```
# oc cp /tmp/monstore/ $(oc get po -l app=rook-ceph-mon,mon=a -oname |sed
's/pod\\//g'):tmp/
```

7. MON Pod に移動し、取得した **monstore** の所有権を変更します。

```
# oc rsh $(oc get po -l app=rook-ceph-mon,mon=a -oname)
```

```
# chown -R ceph:ceph /tmp/monstore
```

8. **mon db** を再構築する前に、キーリングテンプレートファイルをコピーします。

```
# oc rsh $(oc get po -l app=rook-ceph-mon,mon=a -oname)
```

```
# cp /etc/ceph/keyring-store/keyring /tmp/keyring
```

```
# cat /tmp/keyring
[mon.]
key = AQCleqldWqm5lhAAgZQbEzoShkZV42RiQVffnA==
caps mon = "allow *"
[client.admin]
key = AQCmAKld8J05KxAArOWeRAw63gAwwZO5o75ZNQ==
auid = 0
caps mds = "allow *"
caps mgr = "allow *"
caps mon = "allow *"
caps osd = "allow *"
```

9. それぞれのシークレットから、他のすべての Ceph デーモン (MGR、MDS、RGW、Crash、CSI、および CSI プロビジョナー) のキーリングを特定します。

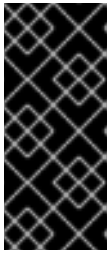
```
# oc get secret rook-ceph-mds-ocs-storagecluster-cephfilesystem-a-keyring -ojson | jq
.data.keyring | xargs echo | base64 -d
```

```
[mds.ocs-storagecluster-cephfilesystem-a]
key = AQB3r8VgAtr6OhAAVhhXpNKqRTuEVdRoxG4uRA==
caps mon = "allow profile mds"
caps osd = "allow *"
caps mds = "allow"
```

キーリングファイルのサンプル **/etc/ceph/ceph.client.admin.keyring**:

```
[mon.]
key = AQDxTF1hNgLTNxAAi51cCojs01b4I5E6v2H8Uw==
caps mon = "allow "
[client.admin]
key = AQDxTF1hpzguOxAA0sS8nN4udoO35OEbt3bqMQ==
caps mds = "allow " caps mgr = "allow *" caps mon = "allow *" caps osd = "allow *"
[mds.ocs-storagecluster-cephfilesystem-a] key =
AQCKTV1horgjARAA8aF/BDh/4+eG4RCNBCI+aw== caps mds = "allow" caps mon = "allow
profile mds" caps osd = "allow *" [mds.ocs-storagecluster-cephfilesystem-b] key =
AQCKTV1hN4gKLBAA5emIVq3ncV7AMEM1c1RmGA== caps mds = "allow" caps mon =
"allow profile mds" caps osd = "allow *" [client.rgw.ocs.storagecluster.cephobjectstore.a] key
= AQCOkdBixmpiAxAA4X7zjn6SGTI9c1MBflszYA== caps mon = "allow rw" caps osd =
"allow rwx" [mgr.a] key = AQBOTV1hGYOEORAA87471+eIZLZtptfkcHvTRg== caps mds =
"allow *" caps mon = "allow profile mgr" caps osd = "allow *" [client.crash] key =
AQBOTV1htO1aGRAAE2MPYcGdiAT+Oo4CNPSF1g== caps mgr = "allow rw" caps mon =
"allow profile crash" [client.csi-cephfs-node] key =
AQBOTV1hiAtuBBAAaPPBVgh1AqZJIDeHWdoFLw== caps mds = "allow rw" caps mgr =
```

```
"allow rw" caps mon = "allow r" caps osd = "allow rw tag cephfs *=" [client.csi-cephfs-
provisioner] key = AQBNTV1hHu6wMBAAzNXZv36aZJuE1iz7S7GfeQ== caps mgr = "allow
rw" caps mon = "allow r" caps osd = "allow rw tag cephfs metadata="
[client.csi-rbd-node]
key = AQBNTV1h+LnkIRAAWnpIN9bUAmSHOVJ0EJXHRw==
caps mgr = "allow rw"
caps mon = "profile rbd"
caps osd = "profile rbd"
[client.csi-rbd-provisioner]
key = AQBNTV1hMNcsExAAvA3gHB2qaY33LOdWCvHG/A==
caps mgr = "allow rw"
caps mon = "profile rbd"
caps osd = "profile rbd"
```



## 重要

- **client.csi** 関連のキーリングについては、前のキーリングファイルの出力を参照し、それぞれの OpenShift Data Foundation シークレットからキーをフェッチした後、デフォルトの **caps** を追加します。
- OSD キーリングは、復元後に自動的に追加されます。

10. **mon-a** Pod に移動し、**monstore** に **monmap** があることを確認します。

a. **mon-a** Pod に移動します。

```
# oc rsh $(oc get po -l app=rook-ceph-mon,mon=a -oname)
```

b. **monstore** に **monmap** があることを確認します。

```
# ceph-monstore-tool /tmp/monstore get monmap -- --out /tmp/monmap
```

```
# monmaptool /tmp/monmap --print
```

11. オプション: **monmap** がない場合は、新しい **monmap** を作成します。

```
# monmaptool --create --add <mon-a-id> <mon-a-ip> --add <mon-b-id> <mon-b-ip> --add
<mon-c-id> <mon-c-ip> --enable-all-features --clobber /root/monmap --fsid <fsid>
```

**<mon-a-id>**

**mon-a** Pod の ID です。

**<mon-a-ip>**

**mon-a** Pod の IP アドレスです。

**<mon-b-id>**

**mon-b** Pod の ID です。

**<mon-b-ip>**

**mon-b** Pod の IP アドレスです。

**<mon-c-id>**

**mon-c** Pod の ID です。

**<mon-c-ip>**



`mon-c` Pod の IP アドレスです。

**<fsid>**

ファイルシステム ID です。

12. **monmap** を確認します。

```
# monmaptool /root/monmap --print
```

13. **monmap** をインポートします。



### 重要

以前に作成した **キーリング** ファイルを使用します。

```
# ceph-monstore-tool /tmp/monstore rebuild -- --keyring /tmp/keyring --monmap /root/monmap
```

```
# chown -R ceph:ceph /tmp/monstore
```

14. 以前の **store.db** ファイルのバックアップを作成します。

```
# mv /var/lib/ceph/mon/ceph-a/store.db /var/lib/ceph/mon/ceph-a/store.db.corrupted
```

```
# mv /var/lib/ceph/mon/ceph-b/store.db /var/lib/ceph/mon/ceph-b/store.db.corrupted
```

```
# mv /var/lib/ceph/mon/ceph-c/store.db /var/lib/ceph/mon/ceph-c/store.db.corrupted
```

15. 再構築した **store.db** ファイルを **monstore** ディレクトリーにコピーします。

```
# mv /tmp/monstore/store.db /var/lib/ceph/mon/ceph-a/store.db
```

```
# chown -R ceph:ceph /var/lib/ceph/mon/ceph-a/store.db
```

16. **monstore** ディレクトリーを再構築したら、**store.db** ファイルをローカルから残りの MON Pod にコピーします。

```
# oc cp $(oc get po -l app=rook-ceph-mon,mon=a -oname | sed 's/pod\\//g'):/var/lib/ceph/mon/ceph-a/store.db /tmp/store.db
```

```
# oc cp /tmp/store.db $(oc get po -l app=rook-ceph-mon,mon=<id> -oname | sed 's/pod\\//g'):/var/lib/ceph/mon/ceph-<id>
```

**<id>**

MON Pod の ID です。

17. 残りの MON Pod に移動し、コピーした **monstore** の所有権を変更します。

```
# oc rsh $(oc get po -l app=rook-ceph-mon,mon=<id> -oname)
```

```
# chown -R ceph:ceph /var/lib/ceph/mon/ceph-<id>/store.db
```

<id>

MON Pod の ID です。

18. パッチが適用された変更を元に戻します。

- MON デプロイメントの場合:

```
# oc replace --force -f <mon-deployment.yaml>
```

<mon-deployment.yaml>

MON デプロイメントの yaml ファイルです。

- OSD デプロイメントの場合:

```
# oc replace --force -f <osd-deployment.yaml>
```

<osd-deployment.yaml>

OSD デプロイメントの yaml ファイルです。

- MGR デプロイメントの場合:

```
# oc replace --force -f <mgr-deployment.yaml>
```

<mgr-deployment.yaml>

MGR デプロイメントの yaml ファイルです。



### 重要

MON、Milla、および OSD Pod が稼働していることを確認します。

19. **rook-ceph-operator** および **ocs-operator** デプロイメントをスケールアップします。

```
# oc -n openshift-storage scale deployment ocs-operator --replicas=1
```

### 検証手順

1. Ceph のステータスをチェックして、CephFS が実行していることを確認します。

```
# ceph -s
```

出力例:

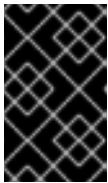
```
cluster:
  id: f111402f-84d1-4e06-9fdb-c27607676e55
  health: HEALTH_ERR
    1 filesystem is offline
    1 filesystem is online with fewer MDS than max_mds
    3 daemons have recently crashed
```

```
services:
  mon: 3 daemons, quorum b,c,a (age 15m)
  mgr: a(active, since 14m)
  mds: ocs-storagecluster-cephfilesystem:0
  osd: 3 osds: 3 up (since 15m), 3 in (since 2h)
```

```
data:
  pools: 3 pools, 96 pgs
  objects: 500 objects, 1.1 GiB
  usage: 5.5 GiB used, 295 GiB / 300 GiB avail
  pgs: 96 active+clean
```

- マルチクラウドオブジェクトゲートウェイ (MCG) のステータスを確認します。アクティブで、backingstore と bucketclass が **Ready** 状態になっている必要があります。

```
noobaa status -n openshift-storage
```



### 重要

MCG がアクティブ状態でなく、backingstore と bucketclass が **Ready** 状態でない場合は、すべての MCG 関連 Pod を再起動する必要があります。詳細は、「[Multicloud Object Gateway の復元](#)」を参照してください。

## 11.1. MULTICLOUD OBJECT GATEWAY の復元

Multicloud Object Gateway (MCG) がアクティブ状態でなく、backingstore および bucketclass が **Ready** 状態でない場合は、MCG 関連のすべての Pod を再起動し、MCG ステータスをチェックして、MCG がバックアップされ、および実行していることを確認する必要があります。

### 手順

- MCG に関連するすべての Pod を再起動します。

```
# oc delete pods <noobaa-operator> -n openshift-storage
```

```
# oc delete pods <noobaa-core> -n openshift-storage
```

```
# oc delete pods <noobaa-endpoint> -n openshift-storage
```

```
# oc delete pods <noobaa-db> -n openshift-storage
```

#### <noobaa-operator>

MCG Operator の名前です。

#### <noobaa-core>

MCG コア Pod の名前です。

#### <noobaa-endpoint>

MCG エンドポイントの名前です。

#### <noobaa-db>

MCG db Pod の名前です。

2. RADOS Object Gateway (RGW) が設定されている場合は、Pod を再起動します。

```
# oc delete pods <rgw-pod> -n openshift-storage
```

<rgw-pod>

RGW Pod の名前です。



### 注記

OpenShift Container Platform 4.11 では、リカバリー後、RBD PVC がアプリケーション Pod にマウントされません。したがって、アプリケーション Pod をホストしているノードを再起動する必要があります。アプリケーション Pod をホストしているノード名を取得するには、次のコマンドを実行します。

```
# oc get pods <application-pod> -n <namespace> -o yaml | grep nodeName  
nodeName: node_name
```

## 第12章 OPENSIFT DATA FOUNDATION での CEPH-MONITOR クォーラムの復元

状況によっては、**ceph-mons** がクォーラムを失う可能性があります。**mons** が再びクォーラムを形成できない場合は、クォーラムを再度取得する手動の手順があります。唯一の要件は、1つ以上の **mon** が正常である必要があることです。以下の手順は、正常でない **mon** をクォーラムから削除し、単一の **mon** でクォーラムを再度作成してから、クォーラムを元のサイズに戻すことができます。

たとえば、3つの **mons** があり、クォーラムが失われる場合は、クォーラムから2つの悪い **mons** を削除して、適切な **mon** がクォーラムの唯一の **mon** であることを通知してから、適切な **mon** を再起動する必要があります。

### 手順

1. **monmap** を変更する場合に **mons** がフェイルオーバーしないように、**rook-ceph-operator** を停止します。

```
# oc -n openshift-storage scale deployment rook-ceph-operator --replicas=0
```

2. 新しい **monmap** を注入します。



### 警告

**monmap** は非常に慎重に注入する必要があります。誤って実行すると、クラスターは永続的に破棄される可能性があります。Ceph **monmap** は、**mon** クォーラムを追跡します。**monmap** は、正常な **mon** のみが含まれるように更新されます。この例では、正常な **mon** は **rook-ceph-mon-b** ですが、正常でない **mon** は **rook-ceph-mon-a** および **rook-ceph-mon-c** になります。

- a. 現在の **rook-ceph-mon-b** デプロイメントのバックアップを作成します。

```
# oc -n openshift-storage get deployment rook-ceph-mon-b -o yaml > rook-ceph-mon-b-deployment.yaml
```

- b. YAML ファイルを開き、**コマンド** および **引数** を **mon** コンテナからコピーします (以下の例の **containers** リストを参照)。これは、**monmap** の変更が必要です。

```
[...]
containers:
- args:
  - --fsid=41a537f2-f282-428e-989f-a9e07be32e47
  - --keyring=/etc/ceph/keyring-store/keyring
  - --log-to-stderr=true
  - --err-to-stderr=true
  - --mon-cluster-log-to-stderr=true
  - '--log-stderr-prefix=debug '
  - --default-log-to-file=false
  - --default-mon-cluster-log-to-file=false
```

```

--mon-host=$(ROOK_CEPH_MON_HOST)
--mon-initial-members=$(ROOK_CEPH_MON_INITIAL_MEMBERS)
--id=b
--setuser=ceph
--setgroup=ceph
--foreground
--public-addr=10.100.13.242
--setuser-match-path=/var/lib/ceph/mon/ceph-b/store.db
--public-bind-addr=$(ROOK_POD_IP)
command:
- ceph-mon
[...]

```

- c. コピーした **command** および **args** フィールドを、以下のように貼り付け可能なコマンドを形成するためにクリーンアップします。

```

# ceph-mon \
--fsid=41a537f2-f282-428e-989f-a9e07be32e47 \
--keyring=/etc/ceph/keyring-store/keyring \
--log-to-stderr=true \
--err-to-stderr=true \
--mon-cluster-log-to-stderr=true \
--log-stderr-prefix=debug \
--default-log-to-file=false \
--default-mon-cluster-log-to-file=false \
--mon-host=$ROOK_CEPH_MON_HOST \
--mon-initial-members=$ROOK_CEPH_MON_INITIAL_MEMBERS \
--id=b \
--setuser=ceph \
--setgroup=ceph \
--foreground \
--public-addr=10.100.13.242 \
--setuser-match-path=/var/lib/ceph/mon/ceph-b/store.db \
--public-bind-addr=$ROOK_POD_IP

```



### 注記

**--log-stderr-prefix** フラグおよび括弧の周りの一重引用符を必ず削除し、**ROOK\_CEPH\_MON\_HOST**、**ROOK\_CEPH\_MON\_INITIAL\_MEMBER S**、および **ROOK\_POD\_IP** に渡されます。

- d. **rook-ceph-mon-b** デプロイメントにパッチを適用し、**mon** Pod を削除せずにこの **mon** の作業を停止します。

```

# oc -n openshift-storage patch deployment rook-ceph-mon-b --type='json' -p
'[{ "op": "remove", "path": "/spec/template/spec/containers/0/livenessProbe" }]'

# oc -n openshift-storage patch deployment rook-ceph-mon-b -p '{"spec": {"template":
{"spec": {"containers": [{"name": "mon", "command": ["sleep", "infinity"], "args": []}]}}}'

```

- e. **mon-b** Pod で以下の手順を実行します。

- i. 正常な **mon** の Pod に接続し、以下のコマンドを実行します。

```
# oc -n openshift-storage exec -it <mon-pod> bash
```

- ii. 変数を設定します。

```
# monmap_path=/tmp/monmap
```

- iii. ceph **mon** を適切な **mon** デプロイメントから貼り付け、**--extract-monmap=\${monmap\_path}** フラグを追加して、**monmap** をファイルにデプロイメントします。

```
# ceph-mon \
  --fsid=41a537f2-f282-428e-989f-a9e07be32e47 \
  --keyring=/etc/ceph/keyring-store/keyring \
  --log-to-stderr=true \
  --err-to-stderr=true \
  --mon-cluster-log-to-stderr=true \
  --log-stderr-prefix=debug \
  --default-log-to-file=false \
  --default-mon-cluster-log-to-file=false \
  --mon-host=$ROOK_CEPH_MON_HOST \
  --mon-initial-members=$ROOK_CEPH_MON_INITIAL_MEMBERS \
  --id=b \
  --setuser=ceph \
  --setgroup=ceph \
  --foreground \
  --public-addr=10.100.13.242 \
  --setuser-match-path=/var/lib/ceph/mon/ceph-b/store.db \
  --public-bind-addr=$ROOK_POD_IP \
  --extract-monmap=${monmap_path}
```

- iv. **monmap** の内容を確認します。

```
# monmaptool --print /tmp/monmap
```

- v. **monmap** から不正な **mons** を削除します。

```
# monmaptool ${monmap_path} --rm <bad_mon>
```

この例では、**mon0** および **mon2** を削除します。

```
# monmaptool ${monmap_path} --rm a
# monmaptool ${monmap_path} --rm c
```

- vi. ceph **mon** コマンドを貼り付け、**--inject-monmap=\${monmap\_path}** フラグを以下のように追加することで、変更した **monmap** を適切な **mon** に挿入します。

```
# ceph-mon \
  --fsid=41a537f2-f282-428e-989f-a9e07be32e47 \
  --keyring=/etc/ceph/keyring-store/keyring \
  --log-to-stderr=true \
  --err-to-stderr=true \
  --mon-cluster-log-to-stderr=true \
  --log-stderr-prefix=debug \
```

```
--default-log-to-file=false \
--default-mon-cluster-log-to-file=false \
--mon-host=$ROOK_CEPH_MON_HOST \
--mon-initial-members=$ROOK_CEPH_MON_INITIAL_MEMBERS \
--id=b \
--setuser=ceph \
--setgroup=ceph \
--foreground \
--public-addr=10.100.13.242 \
--setuser-match-path=/var/lib/ceph/mon/ceph-b/store.db \
--public-bind-addr=$ROOK_POD_IP \
--inject-monmap=${monmap_path}
```

vii. シェルを終了して続行します。

### 3. Rook **configmaps** を編集します。

a. Operator が **mon** を追跡するのに使用する **configmap** を編集します。

```
# oc -n openshift-storage edit configmap rook-ceph-mon-endpoints
```

b. data 要素で、以下のような 3 つの **mon**(または **moncount** に応じて) が表示されることを確認します。

```
data: a=10.100.35.200:6789;b=10.100.13.242:6789;c=10.100.35.12:6789
```

c. リストから問題の **mon** を削除し、末尾に適切な **mon** を 1 つ削除します。以下に例を示します。

```
data: b=10.100.13.242:6789
```

d. ファイルを保存して終了します。

e. ここで、**mons** およびその他のコンポーネントに使用される **Secret** を調整する必要があります。

i. 変数 **good\_mon\_id** の値を設定します。  
以下に例を示します。

```
# good_mon_id=b
```

ii. **oc patch** コマンドを使用して、**rook-ceph-config** シークレットにパッチを適用し、**mon\_host** および **mon\_initial\_members** の 2 つのキー/値のペアを更新できます。

```
# mon_host=$(oc -n openshift-storage get svc rook-ceph-mon-b -o
jsonpath='{.spec.clusterIP}')

# oc -n openshift-storage patch secret rook-ceph-config -p '{"stringData":
{"mon_host": "[v2:""${mon_host}":3300,v1:""${mon_host}":6789]",
"mon_initial_members": ""${good_mon_id}""}'
```



**注記**

**hostNetwork: true** を使用している場合は、**mon\_host** 変数を **mon** がピンングされるノード IP (**nodeSelector**) に置き換える必要があります。これは、mode で作成された **rook-ceph-mon-\*** サービスがないためです。

4. **mon** を再起動します。

変更を取得するには、元の **ceph-mon** コマンドで適切な **mon** Pod を再起動する必要があります。

- a. **mon** デプロイメント YAML ファイルのバックアップで **oc replace** コマンドを使用します。

```
# oc replace --force -f rook-ceph-mon-b-deployment.yaml
```

**注記**

**--force** オプションはデプロイメントを削除し、新たに作成します。

- b. クラスターのステータスを確認します。  
ステータスは、クォーラムの **mon** が1つ表示されるはずですが、ステータスが適切であれば、クラスターは再度正常であるはずですが。

5. クォーラムにある2つの **mon** デプロイメントを削除します。  
以下に例を示します。

```
# oc delete deploy <rook-ceph-mon-1>
# oc delete deploy <rook-ceph-mon-2>
```

この例では、削除するデプロイメントは **rook-ceph-mon-a** および **rook-ceph-mon-c** です。

## 6. Operator を再起動します。

- a. rook Operator を再び起動し、クラスターの健全性の監視を再開します。

**注記**

多数のリソースがすでに存在するエラーを無視するのは安全です。

```
# oc -n openshift-storage scale deployment rook-ceph-operator --replicas=1
```

Operator は **mons** をさらに追加し、**mon** 数に応じて再びクォーラムサイズを増やします。

## 第13章 RED HAT OPENSIFT DATA FOUNDATION コンソールプラグインの有効化

Data Foundation コンソールプラグインはデフォルトで有効になっています。OpenShift Data Foundation Operator のインストール時にこのオプションの選択を解除された場合は、以下の手順を使用して、グラフィカルユーザーインターフェイス(GUI)またはコマンドラインインターフェイスのいずれかからコンソールプラグインをデプロイ後に有効にします。

### 前提条件

- OpenShift Web コンソールへの管理者アクセスがある。
- OpenShift Data Foundation Operator が **openshift-storage** namespace にインストールされ、実行されている。

### 手順

#### ユーザーインターフェイスを使用する場合

1. OpenShift Web コンソールで、**Operators → Installed Operators** をクリックし、インストールされた Operator を表示します。
2. 選択された **Project** が **openshift-storage** であることを確認します。
3. **OpenShift Data Foundation Operator** をクリックします。
4. console プラグインオプションを有効にします。
  - a. **Details** タブで、**Console plugin** の下にある **鉛筆** アイコンをクリックします。
  - b. **Enable** を選択し、**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 console プラグインオプションを有効にします。

```
$ oc patch console.operator cluster -n openshift-storage --type json -p [{"op": "add", "path": "/spec/plugins", "value": ["odf-console"]}]
```

### 検証手順

- console プラグインオプションが有効になると、ポップアップメッセージが表示され、**Web console update is available** が GUI に表示されます。このポップアップから **Refresh web console** をクリックして、反映するコンソールを変更します。
  - Web コンソールで、**Storage** に移動し、**Data Foundation** が使用可能かどうかを確認します。

## 第14章 OPENSIFT DATA FOUNDATION コンポーネントのリソースの変更

OpenShift Data Foundation をインストールすると、OpenShift Data Foundation Pod が消費できる事前に定義されたリソースが提供されます。I/O 負荷が高い状況では、これらの制限を引き上げる必要がある場合があります。

- rook-ceph Pod の CPU およびメモリーリソースを変更するには、[「rook-ceph Pod の CPU およびメモリーリソースの変更」](#) を参照してください。
- Multicloud Object Gateway(MCG) のリソースを調整するには、[「MCG のリソースのチューニング」](#) を参照してください。

### 14.1. ROOK-CEPH POD の CPU およびメモリーリソースの変更

OpenShift Data Foundation のインストール時に、rook-ceph Pod の事前に定義された CPU およびメモリーリソースが提供されます。要件に応じてこれらの値を手動で増やすことができます。

以下の Pod で CPU およびメモリーリソースを変更できます。

- **mgr**
- **mds**
- **rgw**

以下の例は、rook-ceph Pod の CPU およびメモリーリソースを変更する方法を示しています。この例では、既存の MDS Pod 値である **cpu** および **memory** がそれぞれ **1** および **4Gi** から **2** および **8Gi** に増えています。

1. ストレージクラスターを編集します。

```
# oc edit storagecluster -n openshift-storage <storagecluster_name>
```

**<storagecluster\_name>**

ストレージクラスターの名前を指定します。  
以下に例を示します。

```
# oc edit storagecluster -n openshift-storage ocs-storagecluster
```

2. 次の行をストレージクラスターのカスタムリソース (CR) に追加します。

```
spec:
  resources:
    mds:
      limits:
        cpu: 2
        memory: 8Gi
      requests:
        cpu: 2
        memory: 8Gi
```

3. 変更を保存し、エディターを終了します。

4. または、**oc patch** コマンドを実行して、**mds** Pod の CPU およびメモリーの値を変更します。

```
# oc patch -n openshift-storage storagecluster <storagecluster_name>
--type merge \
--patch '{"spec": {"resources": {"mds": {"limits": {"cpu": "2","memory": "8Gi"},"requests":
{"cpu": "2","memory": "8Gi"}}}}'
```

#### <storagecluster\_name>

ストレージクラスターの名前を指定します。  
以下に例を示します。

```
# oc patch -n openshift-storage storagecluster ocs-storagecluster \
--type merge \
--patch '{"spec": {"resources": {"mds": {"limits": {"cpu": "2","memory": "8Gi"},"requests":
{"cpu": "2","memory": "8Gi"}}}}'
```

## 14.2. MCG のリソースのチューニング

Multicloud Object Gateway (MCG) のデフォルト設定は、パフォーマンスではなくリソース消費量が少なくなよように最適化されています。MCG のリソースを調整する方法の詳細については、Red Hat ナレッジベースソリューションの[マルチクラウドオブジェクトゲートウェイ \(NooBaa\) のパフォーマンス調整ガイド](#)を参照してください。

## 第15章 OPENSIFT DATA FOUNDATION のデプロイ後の MULTICLOUD OBJECT GATEWAY 外部サービスの無効化

OpenShift Data Foundation をデプロイすると、OpenShift がプライベートクラスターとしてインストールされている場合でも、パブリック IP が作成されます。ただし、storagecluster CRD の **disableLoadBalancerService** 変数を使用すると、マルチクラウドオブジェクトゲートウェイ (MCG) ロードバランサーの使用を無効にすることができます。これにより、MCG がプライベートクラスター用のパブリックリソースを作成できなくなり、NooBaa サービス **EXTERNAL-IP** を無効にできます。

### 手順

- 次のコマンドを実行し、storagecluster YAML に **disableLoadBalancerService** 変数を追加して、サービスを ClusterIP に設定します。

```
$ oc edit storagecluster -n openshift-storage <storagecluster_name>
[...]
spec:
  arbiter: {}
  encryption:
    kms: {}
  externalStorage: {}
  managedResources:
    cephBlockPools: {}
    cephCluster: {}
    cephConfig: {}
    cephDashboard: {}
    cephFilesystems: {}
    cephNonResilientPools: {}
    cephObjectStoreUsers: {}
    cephObjectStores: {}
    cephRBDMirror: {}
    cephToolbox: {}
  mirroring: {}
  multiCloudGateway:
    disableLoadBalancerService: true    <----- Add this
  endpoints:
  [...]

```



### 注記

変更を元に戻してサービスを LoadBalancer に設定するには、**disableLoadBalancerService** 変数を **false** に設定するか、その行を完全に削除します。

## 第16章 グローバル POD ネットワークを手動で有効化することによる OVS-MULTITENANT プラグインでの ODF-CONSOLE へのアクセス

OpenShift Container Platform では、**ovs-multitenant** プラグインが Software-Defined Networking (SDN) に使用されている場合、異なるプロジェクトの Pod は、異なるプロジェクトの Pod およびサービスとの間でパケットを送受信できません。プロジェクトの Pod ネットワークはグローバルではないため、デフォルトでは、Pod は namespace またはプロジェクト間で通信できません。

odf-console にアクセスするには、**openshift-console** namespace の OpenShift コンソール Pod が、**openshift-storage** namespace の OpenShift Data Foundation odf-console に接続する必要があります。これは、グローバル Pod ネットワークを手動で有効にした場合にのみ可能です。

### 問題

- OpenShift Container Platform で ovs-multitenant プラグインが使用されている場合、odf-console プラグインが失敗し、次のメッセージが表示されます。

```
GET request for "odf-console" plugin failed: Get "https://odf-console-service.openshift-storage.svc.cluster.local:9001/locales/en/plugin__odf-console.json": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
```

### 解決方法

- OpenShift Data Foundation プロジェクトの Pod ネットワーキングをグローバルにします。

```
$ oc adm pod-network make-projects-global openshift-storage
```

## 第17章 暗号化された RBD ストレージクラスへのアノテーション設定

OpenShift Data Foundation 4.14 以降では、OpenShift コンソールが暗号化を有効にして RADOS ブロックデバイス (RBD) ストレージクラスを作成すると、アノテーションが自動的に設定されます。ただし、OpenShift Data Foundation バージョン 4.14 に更新する前に作成された暗号化された RBD ストレージクラスのいずれかに対して、アノテーション `cdi.kubevirt.io/clone-strategy=copy` を追加する必要があります。これにより、顧客データ統合 (CDI) で、デフォルトのスマートクローン作成の代わりにホスト支援型のクローン作成を使用できるようになります。

暗号化されたボリュームへのアクセスに使用されるキーは、ボリュームが作成された namespace に関連付けられます。新しい OpenShift Virtualization 仮想マシンのプロビジョニングなど、暗号化されたボリュームを新しい namespace にクローン作成する場合は、新しいボリュームを作成し、ソースボリュームのコンテンツを新しいボリュームにコピーする必要があります。ストレージクラスに適切にアノテーションが付けられている場合、この動作は自動的にトリガーされます。