



Red Hat OpenShift Data Foundation 4.10

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 4.10 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 © 2023 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

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	3
RED HAT ドキュメントへのフィードバック (英語のみ)	4
第1章 概要	5
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	6
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	9
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レクターの上書き	12
第5章 暗号化トークンの削除または期限切れの状態	13
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	14
6.1. アラートとエラーの解決	14
6.2. クラスターの健全性問題の解決	22
6.3. NOOBAA BUCKET エラー状態の解決	23
6.4. クォータを超過した状態の NOOBAA BUCKET の解決	23
6.5. NOOBAA バケット容量またはクォータの状態の解決	24
6.6. POD のリカバリー	24
6.7. EBS ボリュームの割り当て解除からのリカバリー	24
6.8. ROOK-CEPH-OPERATOR のデバッグログの有効化および無効化	24
第7章 ローカルストレージ OPERATOR デプロイメントの確認	26
第8章 故障したまたは不要な CEPH OBJECT STORAGE デバイスの削除	27
8.1. CEPH クラスターが正常であることの確認	27
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。	27
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	29
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	31
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	32
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	34
第11章 OPENSIFT DATA FOUNDATION でのモニター POD の復元	37
11.1. MULTICLOUD OBJECT GATEWAY の復元	43
第12章 OPENSIFT DATA FOUNDATION での CEPH-MONITOR クォーラムの復元	45
第13章 RED HAT OPENSIFT DATA FOUNDATION コンソールプラグインの有効化	50
第14章 OPENSIFT DATA FOUNDATION コンポーネントのリソースの変更	51
14.1. ROOK-CEPH POD の CPU およびメモリーリソースの変更	51
14.2. MCG のリソースのチューニング	52

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。フィードバックをお寄せいただくには、以下をご確認ください。

- 特定の部分についての簡単なコメントをお寄せいただく場合は、以下をご確認ください。
 1. ドキュメントの表示が **Multi-page HTML** 形式になっていることを確認してください。ドキュメントの右上隅に **Feedback** ボタンがあることを確認してください。
 2. マウスカーソルを使用して、コメントを追加するテキストの部分を強調表示します。
 3. 強調表示されたテキストの下に表示される **Add Feedback** ポップアップをクリックします。
 4. 表示される指示に従ってください。
- より詳細なフィードバックをお寄せいただく場合は、Bugzilla のチケットを作成してください。
 1. [Bugzilla](#) の Web サイトに移動します。
 2. **Component** 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/ocs-must-gather-rhel8:v4.10 <local-registry>/odf4/ocs-must-gather-rhel8:v4.10 [--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/ocs-must-gather-rhel8:v4.10 --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/ocs-must-gather-rhel8:v4.10 --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。

コマンドの差異

- 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/ocs-must-gather-rhel8:v4.10 --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/ocs-must-gather-rhel8:v4.10 --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/ocs-must-gather-rhel8:v4.10 --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。

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一覧表示されます。

- 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:

NNAME	DISPLAY	VERSION	REPLACES	PHASE
mcg-operator.v4.10.0	NooBaa Operator	4.10.0		Succeeded
ocs-operator.v4.10.0	OpenShift Container Storage	4.10.0		Succeeded
odf-csi-addons-operator.v4.10.0	CSI Addons	4.10.0		Succeeded
odf-operator.v4.10.0	OpenShift Data Foundation	4.10.0		Succeeded

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

出力例:

NAME	PACKAGE	SOURCE
CHANNEL		
mcg-operator-stable-4.10-redhat-operators-openshift-marketplace		mcg-operator
redhat-operators stable-4.10		
ocs-operator-stable-4.10-redhat-operators-openshift-marketplace		ocs-operator
redhat-operators stable-4.10		
odf-csi-addons-operator	odf-csi-addons-operator	redhat-operators
stable-4.10		
odf-operator	odf-operator	redhat-operators
4.10		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

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、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。

名前: CephMonVersionMismatch

メッセージ: There are multiple versions of storage services running.

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

重大度: 警告

解決策: 修正

手順: 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 in progress, this alert is temporary.
- 更新が進行中でない場合は、アップグレードプロセスを再開します。

名前: CephOSDVersionMismatch

メッセージ: There are multiple versions of storage services running.

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

重大度: 警告

解決策: 修正

手順: 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 in progress, this alert is temporary.
- 更新が進行中でない場合は、アップグレードプロセスを再開します。

名前: **CephClusterCriticallyFull**

メッセージ: **Storage cluster is critically full and needs immediate expansion**

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

重大度: 重大

解決策: 修正

手順: 不要なデータを削除するか、クラスターを拡張します。

名前: **CephClusterNearFull**

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

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

重大度: 警告

解決策: 修正

手順: 不要なデータを削除するか、クラスターを拡張します。

名前: **NooBaaBucketErrorState**

メッセージ: **A NooBaa Bucket Is In Error State**

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

重大度: 警告

解決策: 回避策

手順: [NooBaa Bucket エラー状態の解決](#)

名前: **NooBaaNamespaceResourceErrorState**

メッセージ: **A NooBaa Namespace Resource Is In Error State**

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

重大度: 警告

解決策: 修正

手順: [NooBaa Bucket エラー状態の解決](#)

名前: NooBaaNamespaceBucketErrorState

メッセージ: A NooBaa Namespace Bucket Is In Error State

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

重大度: 警告

解決策: 修正

手順: [NooBaa Bucket エラー状態の解決](#)

名前: NooBaaBucketExceedingQuotaState

メッセージ: A NooBaa Bucket Is In Exceeding Quota State

説明: A NooBaa bucket {{ \$labels.bucket_name }} is exceeding its quota - {{ printf "%0.0f" \$value }}% used message:A NooBaa Bucket Is In Exceeding Quota State

重大度: 警告

解決策: 修正

手順: [クォータを超過した状態の NooBaa Bucket の解決](#)

名前: NooBaaBucketLowCapacityState

メッセージ: A NooBaa Bucket Is In Low Capacity State

説明: A NooBaa bucket {{ \$labels.bucket_name }} is using {{ printf "%0.0f" \$value }}% of its capacity

重大度: 警告

解決策: 修正

手順: [NooBaa バケット容量またはクォータの状態の解決](#)

名前: NooBaaBucketNoCapacityState

メッセージ: A NooBaa Bucket Is In No Capacity State

説明: A NooBaa bucket {{ \$labels.bucket_name }} is using all of its capacity

重大度: 警告

解決策: 修正

手順: [NooBaa バケット容量またはクォータの状態の解決](#)

名前: NooBaaBucketReachingQuotaState

メッセージ: A NooBaa Bucket Is In Reaching Quota State

説明: A NooBaa bucket {{ \$labels.bucket_name }} is using {{ printf "%0.0f" \$value }}% of its quota

重大度: 警告

解決策: 修正

手順: [NooBaa バケット容量またはクォータの状態の解決](#)

名前: NooBaaResourceErrorState

メッセージ: A NooBaa Resource Is In Error State

説明: A NooBaa resource {{ \$labels.resource_name }} is in error state for more than 6m

重大度: 警告

解決策: 回避策

手順: [NooBaa Bucket エラー状態の解決](#)

名前: NooBaaSystemCapacityWarning100

メッセージ: A NooBaa System Approached Its Capacity

説明: A NooBaa system approached its capacity, usage is at 100%

重大度: 警告

解決策: 修正

手順: [NooBaa バケット容量またはクォータの状態の解決](#)

名前: NooBaaSystemCapacityWarning85

メッセージ: A NooBaa System Is Approaching Its Capacity

説明: A NooBaa system is approaching its capacity, usage is more than 85%

重大度: 警告

解決策: 修正

手順: [NooBaa バケット容量またはクォータの状態の解決](#)

名前: NooBaaSystemCapacityWarning95

メッセージ: A NooBaa System Is Approaching Its Capacity

説明: A NooBaa system is approaching its capacity, usage is more than 95%

重大度: 警告

解決策: 修正

手順: [NooBaa バケット容量またはクォータの状態の解決](#)

名前: CephMdsMissingReplicas

メッセージ: Insufficient replicas for storage metadata service.

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

Might affect the working of storage cluster.`

重大度: 警告

解決策: [Red Hat サポートへのお問い合わせ](#)

手順:

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

名前: CephMgrIsAbsent

メッセージ: Storage metrics collector service not available anymore.

説明: Ceph Manager has disappeared from Prometheus target discovery.

重大度: Critical

解決策: [Red Hat サポートへのお問い合わせ](#)

手順:

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 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](#).

名前: CephNodeDown

メッセージ: Storage node {{ \$labels.node }} went down

説明: Storage node {{ \$labels.node }} went down. Please check the node immediately.

重大度: Critical

解決策: [Red Hat サポートへのお問い合わせ](#)

手順:

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交換](#)を参照してください
 - [Contact Red Hat support](#)

名前: CephClusterErrorState

メッセージ: Storage cluster is in error state

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

重大度: Critical

解決策: [Red Hat サポートへのお問い合わせ](#)

手順:

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.

名前: CephClusterWarningState

メッセージ: Storage cluster is in degraded state

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

重大度: 警告

解決策: [Red Hat サポートへのお問い合わせ](#)

手順:

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.

名前: **CephDataRecoveryTakingTooLong**

メッセージ: **Data recovery is slow**

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

重大度: 警告

解決策: [Red Hat サポートへのお問い合わせ](#)

名前: **CephOSDDiskNotResponding**

メッセージ: **Disk not responding**

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

重大度: Critical

解決策: [Red Hat サポートへのお問い合わせ](#)

名前: **CephOSDDiskUnavailable**

メッセージ: **Disk not accessible**

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

重大度: Critical

解決策: [Red Hat サポートへのお問い合わせ](#)

名前: **CephPGRepairTakingTooLong**

メッセージ: **Self heal problems detected**

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

重大度: 警告

解決策: [Red Hat サポートへのお問い合わせ](#)

名前: **CephMonHighNumberOfLeaderChanges**

メッセージ: **Storage Cluster has seen many leader changes recently.**

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

重大度: 警告

解決策: [Red Hat サポートへのお問い合わせ](#)

名前: **CephMonQuorumAtRisk**

メッセージ: **Storage quorum at risk**

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

重大度: Critical

解決策: [Red Hat サポートへのお問い合わせ](#)

名前: **ClusterObjectStoreState**

メッセージ: **Cluster Object Store is in unhealthy state.Please check Ceph cluster health.**

説明: **Cluster Object Store is in unhealthy state for more than 15s.Please check Ceph cluster health.**

重大度: Critical

解決策: [Red Hat サポートへのお問い合わせ](#)

手順:

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

名前: **CephOSDFlapping**

メッセージ: **Storage daemon osd.x has restarted 5 times in the last 5 minutes.Please check the pod events or Ceph status to find out the cause.**

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

重大度: Critical

解決策: [Red Hat サポートへのお問い合わせ](#)

名前: **OdfPoolMirroringImageHealth**

メッセージ: **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つまたは少数のアプリケーションでは障害復旧に失敗します。

重大度: 警告

解決策: [Red Hat サポートへのお問い合わせ](#)

名前: **OdfMirrorDaemonStatus**

メッセージ: **Mirror daemon is unhealthy.**

説明: クラスター全体で障害復旧に失敗します。mirror デーモンが1分以上異常状態になっています。このクラスターのミラーリングは予想通りに機能しません。

重大度: Critical

解決策: [Red Hat サポートへのお問い合わせ](#)

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
MON_DISK_LOW	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. NOOBAA BUCKET エラー状態の解決

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. **Object** タブをクリックします。
4. **Details** カードの **System Name** フィールドにあるリンクをクリックします。
5. 左側のペインで、**Buckets** オプションをクリックし、エラー状態のバケットを検索します。エラー状態のバケットが namespace バケットである場合は、必ず **Namespace Buckets** ペインをクリックします。
6. その **Bucket Name** をクリックします。バケットで発生しているエラーが表示されます。
7. バケットの特定のエラーに応じて、以下のいずれかまたは両方を実行します。
 - a. 領域に関連するエラーの場合:
 - i. 左側のペインで **Resources** オプションをクリックします。
 - ii. エラー状態のリソースをクリックします。
 - iii. エージェントを追加してリソースをスケーリングします。
 - b. リソースの正常性エラーの場合:
 - i. 左側のペインで **Resources** オプションをクリックします。
 - ii. エラー状態のリソースをクリックします。
 - iii. 接続エラーは、バックギングサービスが利用できないため、復元する必要があることを示します。
 - iv. アクセス/パーミッションのエラーについては、接続の **Access Key** および **Secret Key** を更新します。

6.4. クォータを超過した状態の NOOBAA BUCKET の解決

A NooBaa Bucket Is In Exceeding Quota Stateエラーを解決するには、以下のいずれかを実行します。

- バケットの一部のデータをクリーンアップします。
- 以下の手順に従って、バケットクォータを増やします。
 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. **Object** タブをクリックします。

4. **Details** カードの **System Name** フィールドにあるリンクをクリックします。
5. 左側のペインで、**Buckets** オプションをクリックし、エラー状態のバケットを検索します。
6. **Bucket Name** をクリックします。バケットで発生しているエラーが表示されます。
7. **Bucket Policies** → **Edit Quota** をクリックし、クォータを増やします。

6.5. NOOBAA バケット容量またはクォータの状態の解決

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. **Object** タブをクリックします。
4. **Details** カードの **System Name** フィールドにあるリンクをクリックします。
5. 左側のペインで **Resources** オプションをクリックし、PV プールリソースを検索します。
6. 容量が低いステータスの PV プールリソースの場合は、その **Resource Name** をクリックします。
7. プール設定を編集し、エージェントの数を増やします。

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**) に移行しようとしていますが、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-ceph-operator-config yaml ファイルに **ROOK_LOG_LEVEL: DEBUG** パラメーターを追加して、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-ceph-operator-config yaml ファイルに **ROOK_LOG_LEVEL: INFO** パラメーターを追加して、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: INFO
```

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
db-noobaa-db-0	Bound	pvc-d96c747b-2ab5-47e2-b07e-1079623748d8	50Gi	
ocs-deviceset-0-0-lzfrd	Bound	local-pv-7e70c77c	1769Gi	RWO
ocs-deviceset-1-0-7rggl	Bound	local-pv-b19b3d48	1769Gi	RWO
ocs-deviceset-2-0-znhk8	Bound	local-pv-e9f22cdc	1769Gi	RWO

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 で失敗した、または不要な Ceph OSD を削除するには、手順のステップに従ってください。



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** 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

次の手順に従って、ローカルストレージデバイスを使用してプロビジョニングされた失敗した Ceph または不要な Ceph 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** 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 に関連付けられている 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 **CEPHOSD: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 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、一覧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ではない場合は、Red Hat Ceph Storage クラスタで 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 PVC の作成を有効にする必要があります。

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-pxknkcix20-pod  Pending
                                ocs-external-storagecluster-cephfs 28h
[...]
```

2. **describe** 出力を確認し、それぞれの pvc のイベントを表示します。
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-pxknkcix20-pod -n nginx-file
```

出力例:

```
Name:          ngx-fs-pxknkcix20-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
  ----    -
  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。

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. **livenessProbe** パラメーターを削除するように OSD 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 **sleep** を指定して実行します。

```
# 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/podV//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。

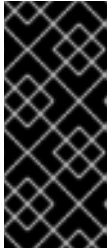
```
# oc get secret rook-ceph-mds-ocs-storagecluster-cephfilesystem-a-keyring -ojson | jq
.data.keyring | xargs echo | base64 -d

[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、新規の **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。

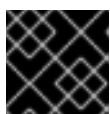
<fsid>

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:

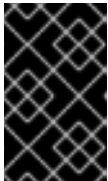
```
cluster:
  id: f111402f-84d1-4e06-9fdb-c27607676e55
  health: HEALTH_ERR
    1 filesystem is offline
    1 filesystem is online with fewer MDS than max_mds
    3 daemons have recently crashed

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
```

2. 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

1. 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。

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** 一覧を参照)。これは、**monmap** の変更に必要なです。

```
[...]
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** をファイルに展開します。

```
# 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

多数のリソースが既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

OpenShift Data Foundation Operator のインストール後に自動的に有効にされていない場合は、console プラグインオプションを有効にします。console プラグインは、Web コンソールに含まれるカスタムインターフェイスを提供します。console プラグインオプションは、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から有効にできます。

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。

例14.1 例

```
# 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
```

- 変更を保存し、エディターを終了します。
- 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。

例14.2 例

```
# 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い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。