



Red Hat OpenShift Container Storage 4.8

Red Hat OpenShift Container Storage のトラブルシューティング

OpenShift Container Storage のエラーおよび問題のトラブルシューティング方法

Red Hat OpenShift Container Storage 4.8 Red Hat OpenShift Container Storage のトラブルシューティング

OpenShift Container Storage のエラーおよび問題のトラブルシューティング方法

Enter your first name here. Enter your surname here.

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

Enter your email address here.

法律上の通知

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

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

Red Hat OpenShift Container Storage のトラブルシューティングについては、本書を参照してください。

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	8
第4章 OPENSIFT CONTAINER STORAGE 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	11
第5章 暗号化トークンの削除または期限切れの状態	12
第6章 OPENSIFT CONTAINER STORAGE のアラートおよびエラーのトラブルシューティング	13
6.1. アラートとエラーの解決	13
6.2. NOOBAA BUCKET エラー状態の解決	20
6.3. クォータを超過した状態の NOOBAA BUCKET の解決	21
6.4. NOOBAA バケット容量またはクォータの状態の解決	21
6.5. POD のリカバリー	21
6.6. EBS ボリュームの割り当て解除からのリカバリー	22
6.7. ROOK-CEPH-OPERATOR のデバッグログの有効化および無効化	22
第7章 ローカルストレージ OPERATOR デプロイメントの確認	24
第8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	25
第9章 外部モードでの CEPHFS PVC 作成のトラブルシューティング	27
第10章 OPENSIFT CONTAINER STORAGE でのモニター POD の復元	30
10.1. CEPHFS の復元	36
10.2. MULTICLOUD OBJECT GATEWAY の復元	37
第11章 OPENSIFT CONTAINER STORAGE コンポーネントのリソースの変更	39
11.1. ROOK-CEPH POD の CPU およびメモリーリソースの変更	39
11.2. MCG のリソースのチューニング	40

多様性を受け入れるオープンソースの強化

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

RED HAT ドキュメントへのフィードバック (英語のみ)

弊社のドキュメントについてのご意見をお聞かせください。ドキュメントの改善点があれば、ぜひお知らせください。フィードバックをお寄せいただくには、以下をご確認ください。

- 特定の部分についての簡単なコメントをお寄せいただく場合は、以下をご確認ください。
 1. ドキュメントの表示が **Multi-page HTML** 形式になっていることを確認してください。ドキュメントの右上隅に **Feedback** ボタンがあることを確認してください。
 2. マウスカーソルを使用して、コメントを追加するテキストの部分を強調表示します。
 3. 強調表示されたテキストの下に表示される **Add Feedback** ポップアップをクリックします。
 4. 表示される指示に従ってください。
- より詳細なフィードバックをお寄せいただく場合は、Bugzilla のチケットを作成してください。
 1. [Bugzilla](#) の Web サイトに移動します。
 2. **Component** 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 Container Storage のトラブルシューティングは、管理者が Red Hat OpenShift Container Storage クラスターのトラブルシューティングおよび修正を行う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 Container Storage 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 Container Storage 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 Container Storage 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。

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 Container Storage 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

OpenShift Container Storage が外部モードでデプロイされる場合には、`must-gather` は Red Hat OpenShift Container Storage クラスターからのみログを収集し、外部 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、または Red Hat Ceph Storage の管理者にお問い合わせください。

手順

- OpenShift Container Storage 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/ocs4/ocs-must-gather-rhel8:v4.8 --dest-dir=<directory-name>
```

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 OpenShift Container Storage クラスター関連のカスタムリソース (CR) とそれらの namespace。
- すべての OpenShift Container Storage 関連の Pod の Pod ログ。
- ステータス、クラスター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/ocs4/ocs-must-gather-rhel8:v4.8 --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/ocs4/ocs-must-gather-rhel8:v4.8 --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/ocs4/ocs-must-gather-rhel8:v4.8 --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 Container Storage 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 Container Storage 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 Container Storage ログを生成します。

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

- OpenShift Container Storage の 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 Container Storage の 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
ocs-operator.v4.7.2  OpenShift Container Storage 4.7.2
Succeeded
```

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

出力例:

```
NAME      PACKAGE      SOURCE
CHANNEL
ocs-operator ocs-operator redhat-operators
stable-4.8
```

- installplan が作成されていることを確認します。

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

- OpenShift Container Storage を更新後の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
```

出力例:

```
IMAGE                               TAG
  IMAGEID      SIZE
registry.redhat.io/ocs4/rook-ceph-rhel8-operator@sha256  <none>
5600a36370df4  1.55GB
```

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 CONTAINER STORAGE 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 Container Storage 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 Container Storage 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、または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 CONTAINER STORAGE 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 Container Storage 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。

- **Monitoring** → **Alerting** → **Firing** オプション
- **Home** → **Overview** → **Cluster** タブ
- **Storage** → **Overview** → **Block and File** タブ
- **Storage** → **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 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 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

Procedure: [Resolving NooBaa Bucket Error State](#)

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

Procedure: [Resolving NooBaa Bucket Error State](#)

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

Procedure: [Resolving NooBaa Bucket Error State](#)

Name: **NooBaaBucketExceedingQuotaState**

Message: **A NooBaa Bucket Is In Exceeding Quota State**

Description: **A NooBaa bucket {{ \$labels.bucket_name }} is exceeding its quota - {{ printf "%.0f" \$value }}% used message: A NooBaa Bucket Is In Exceeding Quota State**

Severity: Warning

Resolution: Fix

Procedure: [Resolving NooBaa Bucket Exceeding Quota State](#)

Name: **NooBaaBucketLowCapacityState**

Message: **A NooBaa Bucket Is In Low Capacity State**

Description: **A NooBaa bucket {{ \$labels.bucket_name }} is using {{ printf "%.0f" \$value }}% of its capacity**

Severity: Warning

Resolution: Fix

Procedure: [Resolving NooBaa Bucket Capacity or Quota State](#)

Name: **NooBaaBucketNoCapacityState**

Message: **A NooBaa Bucket Is In No Capacity State**

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

Severity: Warning

Resolution: Fix

Procedure: [Resolving NooBaa Bucket Capacity or Quota State](#)

Name: **NooBaaBucketReachingQuotaState**

Message: **A NooBaa Bucket Is In Reaching Quota State**

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

Severity: Warning

Resolution: Fix

Procedure: [Resolving NooBaa Bucket Capacity or Quota State](#)

Name: **NooBaaResourceErrorState**

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

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

Severity: Warning

Resolution: Workaround

Procedure: [Resolving NooBaa Bucket Error State](#)

Name: **NooBaaSystemCapacityWarning100**

Message: **A NooBaa System Approached Its Capacity**

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

Severity: Warning

Resolution: Fix

Procedure: [Resolving NooBaa Bucket Capacity or Quota State](#)

Name: **NooBaaSystemCapacityWarning85**

Message: **A NooBaa System Is Approaching Its Capacity**

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

Severity: Warning

Resolution: Fix

Procedure: [Resolving NooBaa Bucket Capacity or Quota State](#)

Name: **NooBaaSystemCapacityWarning95**

Message: **A NooBaa System Is Approaching Its Capacity**

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

Severity: Warning

Resolution: Fix

Procedure: [Resolving NooBaa Bucket Capacity or Quota State](#)

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 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.Please 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:
 - [OpenShift Container Storage 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](#)

<p>Name: CephMonQuorumAtRisk</p> <p>Message: Storage quorum at risk</p> <p>Description: Storage cluster quorum is low.</p> <p>Severity: Critical</p> <p>Resolution: Contact Red Hat support</p>
<p>Name: ClusterObjectStoreState</p> <p>Message: Cluster Object Store is in unhealthy state.Please check Ceph cluster health.</p> <p>Description: Cluster Object Store is in unhealthy state for more than 15s.Please check Ceph cluster health.</p> <p>Severity: Critical</p> <p>Resolution: Contact Red Hat support</p> <p>Procedure:</p> <ul style="list-style-type: none">● CephObjectStore CR インスタンスを確認します。● Contact Red Hat support
<p>Name: CephOSDFlapping</p> <p>Message: 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.</p> <p>Description: Storage OSD restarts more than 5 times in 5 minutes.</p> <p>Severity: Critical</p> <p>Resolution: Contact Red Hat support</p>

6.2. NOOBAA BUCKET エラー状態の解決

手順

1. OpenShift Web コンソールにログインし、**Storage → Overview → Object**をクリックします。
2. **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。
3. 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。
4. 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。
5. 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ルリングします。
- 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、バックアップ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。

6.3. 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 コンソールにログインし、**Storage** → **Overview** → **Object**をクリックします。
 2. **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。
 3. 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。
 4. 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。
 5. **Bucket Policies** → **Edit Quota** をクリックし、クォータを増やします。

6.4. 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 コンソールにログインし、**Storage** → **Overview** → **Object**をクリックします。
2. **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。
3. 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。
4. 容量が低いステータスの PV プールリソースについては、その **Resource Name** をクリックします。
5. プール設定を編集し、エージェントの数を増やします。

6.5. 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
```

6.6. 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。

6.7. 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
```

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 OpenShift Container Storage 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 Container Storage と共に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 Container Storage が **openshift-storage** namespace にインストールされ、実行されている。

手順

OpenShift Container Storage 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 Container Storage 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

- [ローカルストレージデバイスを使用した OpenShift Container Storage の VMWare へのデプロイ](#)
- [ローカルストレージデバイスを使用した OpenShift Container Storage の Red Hat Virtualization へのデプロイ](#)
- [ローカルストレージデバイスを使用した OpenShift Container Storage のベアメタルへのデプロイ](#)

第8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9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 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-hqhzf
Events:
  Type    Reason          Age          From
  Message
  ----    -
  -----
Warning ProvisioningFailed 107m (x245 over 22h) openshift-
storage.cephfs.csi.ceph.com_csi-cephfspugin-provisioner-5f8b66cc96-hvcqp_6b7044af-
c904-4795-9ce5-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10章 OPENSIFT CONTAINER STORAGE 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 Container Storage 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/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/podV//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 Container Storage 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。

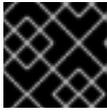
<fsid>

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
```



重要

ファイルシステムがオフラインであるか、MDS サービスが見つからない場合は、CephFS を復元する必要があります。詳細は、「[CephFS の復元](#)」を参照してください。

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。

10.1. CEPHFS の復元

ファイルシステムがオフラインであるか、MDS サービスが見つからない場合は、CephFS を復元する必要があります。

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. MDS 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 **sleep** を使用して実行します。

```
# for i in $(oc get deployment -l app=rook-ceph-mds -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": "mds", "command": ["sleep", "infinity"], "args": []}]}}}' ; done
```

3. CephFS を復元します。

```
# ceph fs reset ocs-storagecluster-cephfilesystem --yes-i-really-mean-it
```

reset コマンドが失敗した場合は、データおよびメタデータプールを使用してデフォルトのファイルシステムを強制的に作成し、リセットします。



注記

cephfilesystem がない場合は、**reset** コマンドが失敗する場合があります。

```
# ceph fs new ocs-storagecluster-cephfilesystem ocs-storagecluster-cephfilesystem-  
metadata ocs-storagecluster-cephfilesystem-data0 --force
```

```
# ceph fs reset ocs-storagecluster-cephfilesystem --yes-i-really-mean-it
```

4. MDS デプロイメントを置き換えます。

```
# oc replace --force -f oc_get_deployment.rook-ceph-mds-ocs-storagecluster-cephfilesystem-  
a.yaml
```

```
# oc replace --force -f oc_get_deployment.rook-ceph-mds-ocs-storagecluster-cephfilesystem-  
b.yaml
```

5. **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 scale deployment ocs-operator --replicas=1 -n openshift-storage
```

6. CephFS のステータスを確認します。

```
# ceph fs status
```

ステータスは active である必要があります。

重要

- CephFS Persistent Volume Claim(永続ボリューム要求、PVC) を使用していたデプロイメントに割り当てられたアプリケーション Pod が、CephFS の復元後に **CreateContainerError** 状態のままになる場合は、アプリケーション Pod を再起動します。

```
# oc -n <namespace> delete pods <cephfs-app-pod>
```

<namespace>

プロジェクトの namespace です。

<cephfs-app-pod>

CephFS アプリケーション Pod の名前です。

- 新規 CephFS または RBD PVC がバインドされない場合は、Ceph CSI に関連するすべての Pod を再起動します。

10.2. 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11章 OPENSIFT CONTAINER STORAGE コンポーネントのリソースの変更

OpenShift Container Storage をインストールすると、OpenShift Container Storage Pod が消費できる事前定義されたリソースが付属しています。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。

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

OpenShift Container Storage をインストールすると、rook-ceph Pod 用に事前定義された CPU およびメモリーリソースが付属しています。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11.1 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
```

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。

例11.2 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"}}}}'
```

11.2. 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f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。