



Red Hat OpenShift Container Storage 4.6

ストレージリソースの管理および割り当て

クラスターおよびストレージ管理者の管理タスク

Red Hat OpenShift Container Storage 4.6 ストレージリソースの管理および割り当て

クラスターおよびストレージ管理者の管理タスク

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/Managing_and_allocating_storage_resources.ent file |.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本書では、Red Hat OpenShift Container Storage でコアサービスおよびホスト型アプリケーションにストレージを割り当てる方法について説明します。

目次

第1章 概要	3
第2章 ストレージクラスおよびストレージプール	4
2.1. ストレージクラスおよびプールの作成	4
第3章 OPENSIFT CONTAINER PLATFORM サービスのストレージの設定	6
3.1. OPENSIFT CONTAINER STORAGE を使用するためのイメージレジストリーの設定	6
3.2. OPENSIFT CONTAINER STORAGE を使用するためのモニタリングの設定	8
3.3. OPENSIFT CONTAINER STORAGE のクラスターロギング	11
3.3.1. 永続ストレージの設定	11
3.3.2. OpenShift Container Storage を使用するためのクラスターロギングの設定	12
第4章 OPENSIFT CONTAINER STORAGE を使用した OPENSIFT CONTAINER PLATFORM アプリケーションのサポート	16
第5章 既存の外部 OPENSIFT CONTAINER STORAGE クラスターへのファイルおよびオブジェクトストレージの追加	18
第6章 RED HAT OPENSIFT CONTAINER STORAGE に専用のワーカーノードを使用する方法	22
6.1. インフラストラクチャーノードの仕組み	22
6.2. インフラストラクチャーノードを作成するためのマシンセット	22
6.3. インフラストラクチャーノードの手動作成	23
第7章 PERSISTENT VOLUME CLAIM (永続ボリューム要求、PVC) の管理	25
7.1. OPENSIFT CONTAINER PLATFORM を使用するためのアプリケーション POD の設定	25
7.2. PERSISTENT VOLUME CLAIM (永続ボリューム要求、PVC) 要求ステータスの表示	26
7.3. PERSISTENT VOLUME CLAIM (永続ボリューム要求、PVC) 要求イベントの確認	27
7.4. PERSISTENT VOLUME CLAIM (永続ボリューム要求、PVC) の拡張	27
7.5. 動的プロビジョニング	29
7.5.1. 動的プロビジョニングについて	29
7.5.2. OpenShift Container Storage の動的プロビジョニング	29
7.5.3. 利用可能な動的プロビジョニングプラグイン	30
第8章 ボリュームスナップショット	32
8.1. ボリュームスナップショットの作成	32
8.2. ボリュームスナップショットの復元	33
8.3. ボリュームスナップショットの削除	34
第9章 ボリュームのクローン作成	36
9.1. クローンの作成	36
第10章 CONTAINER STORAGE INTERFACE (CSI) コンポーネントの配置の管理	37

第1章 概要

本書では、ストレージを作成し、設定し、Red Hat OpenShift Container Storage のコアサービスまたはホスト型アプリケーションに割り当てる方法について説明します。

- [2章ストレージクラスおよびストレージプール](#) カスタムのストレージクラスおよびストレージプールを作成する方法を説明します。
- [3章OpenShift Container Platform サービスのストレージの設定](#) コアとなる OpenShift Container Platform サービスに OpenShift Container Storage を使用する方法を説明します。
- [4章OpenShift Container Storage を使用した OpenShift Container Platform アプリケーションのサポート](#) OpenShift Container Storage を使用するように OpenShift Container Platform アプリケーションを設定する方法についての情報を提供します。
- [既存の外部 OpenShift Container Storage クラスタへのファイルおよびオブジェクトストレージの追加](#)
- [6章Red Hat OpenShift Container Storage に専用のワーカーノードを使用する方法](#) Red Hat OpenShift Container Storage に専用のワーカーノードを使用する方法についての情報を提供します。
- [7章Persistent Volume Claim \(永続ボリューム要求、PVC\) の管理](#) Persistent Volume Claim (永続ボリューム要求、PVC) の要求の管理とそれらの要求への対応の自動化に関する情報を提供します。
- [8章ボリュームスナップショット](#) ボリュームスナップショットを作成し、復元し、削除する方法を説明します。
- [9章ボリュームのクローン作成](#) ボリュームのクローンを作成する方法を説明します。
- [10章Container Storage Interface \(CSI\) コンポーネントの配置の管理](#) 容認を設定してノードでコンテナストレージのインターフェースコンポーネントを起動する方法についての情報を提供します。

第2章 ストレージクラスおよびストレージプール

OpenShift Container Storage Operator は、使用されるプラットフォームに応じてデフォルトのストレージクラスをインストールします。このデフォルトストレージクラスは Operator によって所有され、制御されるため、削除したり変更したりすることはできません。ただし、ストレージクラスの異なる動作が必要な場合は、カスタムストレージクラスを作成できます。

以下の機能を提供するストレージクラスにマップする複数のストレージプールを作成できます。

- それぞれに高可用性のあるアプリケーションを有効にして、2つのレプリカを持つ永続ボリュームを使用できるようにします。これにより、アプリケーションのパフォーマンスが向上する可能性があります。
- 圧縮が有効にされているストレージクラスを使用して永続ボリューム要求の領域を節約します。



注記

カスタムストレージクラスは、**外部モード**の OpenShift Container Storage クラスタではサポートされません。



注記

単一デバイスセットの最小クラスタで新規作成できるストレージクラスは、2つだけです。ストレージクラスを拡張するたびに、新規ストレージクラスを2つ追加できます。

2.1. ストレージクラスおよびプールの作成

既存のプールを使用してストレージクラスを作成するか、またはストレージクラスの作成中にストレージクラスの新規プールを作成できます。

前提条件

OpenShift Container Storage クラスタが **Ready** 状態にあることを確認します。

手順

1. OpenShift Web コンソールにログインします。
2. **Storage** → **Storage Classes** をクリックします。
3. **Create Storage Class** をクリックします。
4. ストレージクラスの **Name** および **Description** を入力します。
5. Reclaim Policy について **Delete** または **Retain** のいずれかを選択します。デフォルトでは、**Delete** が選択されます。
6. 永続ボリュームのプロビジョニングに使用されるプラグインである RBD Provisioner を選択します。
7. 新規プールを作成するか、または既存プールを使用できます。

新規プールを作成します。

- a. プールの名前を入力します。
- b. Data Protection Policy として **2-way-Replication** または **3-way-Replication** を選択します。
- c. データを圧縮する必要がある場合は、**Enable compression** を選択します。
圧縮を有効にするとアプリケーションのパフォーマンスに影響がある可能性があり、書き込まれるデータがすでに圧縮または暗号化されている場合は効果的ではない可能性があります。圧縮を有効にする前に書き込まれたデータは圧縮されません。
- d. **Create** をクリックしてストレージクラスを作成します。
- e. プールの作成後に **Finish** をクリックします。
- f. **Create** をクリックしてストレージクラスを作成します。

既存プールを使用します。

- a. 一覧からプールを選択します。
- b. **Create** をクリックしてプールが選択されたストレージクラスを作成します。

第3章 OPENSIFT CONTAINER PLATFORM サービスのストレージの設定

OpenShift Container Storage を使用して、イメージレジストリー、モニタリング、およびロギングなどの OpenShift Container Platform サービスのストレージを提供できます。

これらのサービスのストレージを設定するプロセスは、OpenShift Container Storage デプロイメントで使用されるインフラストラクチャーによって異なります。



警告

これらのサービスに十分なストレージ容量があることを常に確認します。これらの重要なサービスのストレージの容量が不足すると、クラスターは動作しなくなり、これを回復することは非常に困難になります。

Red Hat は、これらのサービスのキューレーションおよび保持期間を短く設定することを推奨します。詳細は、OpenShift Container Platform ドキュメントの『[モニタリングガイド](#)』の「[Curator スケジュールの設定](#)」と「[Prometheus メトリクスデータの保持期間の編集](#)」を参照してください。

これらのサービスのストレージ領域が不足する場合は、Red Hat カスタマーサポートにお問い合わせください。

3.1. OPENSIFT CONTAINER STORAGE を使用するためのイメージレジストリーの設定

OpenShift Container Platform は、クラスター上の標準ワークロードとして実行されるコンテナイメージレジストリーでビルドを提供します。通常、レジストリーはクラスター上にビルドされたイメージの公開ターゲットとして、またクラスター上で実行されるワークロードのイメージのソースとして使用されます。

このセクションの手順に従って、OpenShift Container Storage をコンテナイメージレジストリーのストレージとして設定します。AWS では、レジストリーのストレージを変更する必要はありません。ただし vSphere およびベアメタルプラットフォームの場合は、OpenShift Container Storage 永続ボリュームに対してストレージを変更することが推奨されます。



警告

このプロセスでは、データを既存イメージレジストリーから新規イメージレジストリーに移行しません。コンテナイメージが既存レジストリーにすでにある場合、このプロセスを完了する前にレジストリーのバックアップを作成し、このプロセスが完了したらイメージを再登録します。

前提条件

- OpenShift Web コンソールへの管理者アクセスがある。
- OpenShift Container Storage Operator が **openshift-storage** namespace にインストールされ、実行されている。OpenShift Web Console で、**Operators** → **Installed Operators** をクリックしてインストールされた Operator を表示します。
- イメージレジストリー Operator が **openshift-image-registry** namespace にインストールされ、実行されている。OpenShift Web コンソールで、**Administration** → **Cluster Settings** → **Cluster Operators** をクリックしてクラスター Operator を表示します。
- プロビジョナー **openshift-storage.cephfs.csi.ceph.com** を持つストレージクラスが利用可能である。OpenShift Web コンソールで、**Storage** → **Storage Classes** をクリックし、利用可能なストレージクラスを表示します。

手順

1. 使用するイメージレジストリーの Persistent Volume Claim (永続ボリューム要求、PVC) を作成します。
 - a. OpenShift Web コンソールで、**Storage** → **Persistent Volume Claims** をクリックします。
 - b. **Project** を **openshift-image-registry** に設定します。
 - c. **Create Persistent Volume Claim** をクリックします。
 - i. 上記で取得した利用可能なストレージクラス一覧から、プロビジョナー **openshift-storage.cephfs.csi.ceph.com** で **Storage Class** を指定します。
 - ii. Persistent Volume Claim (永続ボリューム要求、PVC) の **Name** (名前) を指定します (例: **ocs4registry**)。
 - iii. **Shared Access (RWX)** の **Access Mode** (アクセスモード) を指定します。
 - iv. 100 GB 以上の **Size** (サイズ) を指定します。
 - v. **Create** をクリックします。
新規 Persistent Volume Claim (永続ボリューム要求、PVC) のステータスが **Bound** と表示されるまで待機します。
2. クラスターのイメージレジストリーを、新規の Persistent Volume Claim (永続ボリューム要求、PVC) を使用するように設定します。
 - a. **Administration** → **Custom Resource Definitions** をクリックします。
 - b. **imageregistry.operator.openshift.io** グループと関連付けられた **Config** カスタムリソース定義をクリックします。
 - c. **Instances** タブをクリックします。
 - d. クラスターインスタンスの横にある **Action メニュー (⋮)** → **Edit Config** をクリックします。
 - e. 新規 Persistent Volume Claim (永続ボリューム要求、PVC) をイメージレジストリーの永続ストレージとして追加します。
 - i. 以下を **spec:** の下に追加し、必要な場合は既存の **storage:** セクションを置き換えます。

```
storage:
  pvc:
    claim: <new-pvc-name>
```

以下は例になります。

```
storage:
  pvc:
    claim: ocs4registry
```

ii. **Save** をクリックします。

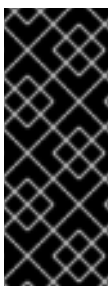
3. 新規設定が使用されていることを確認します。

- a. **Workloads** → **Pods** をクリックします。
- b. **Project** を **openshift-image-registry** に設定します。
- c. 新規の **image-registry-*** Pod が **Running** のステータスと共に表示され、以前の **image-registry-*** Pod が終了していることを確認します。
- d. 新規の **image-registry-*** Pod をクリックし、Pod の詳細を表示します。
- e. **Volumes** までスクロールダウンし、**registry-storage** ボリュームに新規 Persistent Volume Claim (永続ボリューム要求、PVC) に一致する **Type** があることを確認します (例: **ocs4registry**)。

3.2. OPENSIFT CONTAINER STORAGE を使用するためのモニタリングの設定

OpenShift Container Storage は、Prometheus および AlertManager で構成されるモニタリングスタックを提供します。

このセクションの手順に従って、OpenShift Container Storage をモニタリングスタックのストレージとして設定します。



重要

ストレージ領域が不足すると、モニタリングは機能しません。モニタリング用に十分なストレージ容量があることを常に確認します。

Red Hat は、このサービスの保持期間を短く設定することを推奨します。詳細は、OpenShift Container Platform ドキュメントの『モニタリングガイド』の「[Prometheus メトリクスデータの保持期間の編集](#)」を参照してください。

前提条件

- OpenShift Web コンソールへの管理者アクセスがある。
- OpenShift Container Storage Operator が **openshift-storage** namespace にインストールされ、実行されている。OpenShift Web Console で、**Operators** → **Installed Operators** をクリックしてインストールされた Operator を表示します。

- モニタリング Operator は **openshift-monitoring** namespace にインストールされ、実行されている。OpenShift Web コンソールで、**Administration** → **Cluster Settings** → **Cluster Operators** をクリックしてクラスター Operator を表示します。
- プロビジョナー **openshift-storage.rbd.csi.ceph.com** を持つストレージクラスが利用可能である。OpenShift Web コンソールで、**Storage** → **Storage Classes** をクリックし、利用可能なストレージクラスを表示します。

手順

1. OpenShift Web コンソールで、**Workloads** → **Config Maps** に移動します。
2. **Project** ドロップダウンを **openshift-monitoring** に設定します。
3. **Create Config Map** をクリックします。
4. 以下の例を使用して新規の **cluster-monitoring-config** Config Map を定義します。
山括弧 (<, >) 内の内容を独自の値に置き換えます (例: **retention: 24h** または **storage: 40Gi**)。

storageClassName、をプロビジョナー **openshift-storage.rbd.csi.ceph.com** を使用する **storageclass** に置き換えます。以下の例では、**storageclass** の名前は **ocs-storagecluster-ceph-rbd** です。

cluster-monitoring-config Config Map の例

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cluster-monitoring-config
  namespace: openshift-monitoring
data:
  config.yaml: |
    prometheusK8s:
      retention: <time to retain monitoring files, e.g. 24h>
      volumeClaimTemplate:
        metadata:
          name: ocs-prometheus-claim
        spec:
          storageClassName: ocs-storagecluster-ceph-rbd
          resources:
            requests:
              storage: <size of claim, e.g. 40Gi>
    alertmanagerMain:
      volumeClaimTemplate:
        metadata:
          name: ocs-alertmanager-claim
        spec:
          storageClassName: ocs-storagecluster-ceph-rbd
          resources:
            requests:
              storage: <size of claim, e.g. 40Gi>
```

5. **Create** をクリックして Config Map を保存し、作成します。

検証手順

1. Persistent Volume Claim (永続ボリューム要求、PVC) が Pod にバインドされていることを確認します。
 - a. **Storage** → **Persistent Volume Claims**に移動します。
 - b. **Project** ドロップダウンを **openshift-monitoring** に設定します。
 - c. 5つの Persistent Volume Claim (永続ボリューム要求、PVC) が **Bound** (バインド) の状態で表示され、3つの **alertmanager-main-*** Pod および2つの **prometheus-k8s-*** Pod に割り当てられていることを確認します。

作成済みのバインドされているストレージのモニタリング

Project: openshift-monitoring ▼

Persistent Volume Claims

Create Persistent Volume Claim Filter by name...

0 Pending 5 Bound 0 Lost Select All Filters 5 Items

Name ↑	Namespace ↓	Status ↓	Persistent Volume ↓	Requested ↓
my-alertmanager-claim-alertmanager-main-0	openshift-monitoring	Bound	pvc-d00428a5-0ce6-11ea-8fe8-023bdfa29edc	40Gi
my-alertmanager-claim-alertmanager-main-1	openshift-monitoring	Bound	pvc-d00be111-0ce6-11ea-8fe8-023bdfa29edc	40Gi
my-alertmanager-claim-alertmanager-main-2	openshift-monitoring	Bound	pvc-d01ac717-0ce6-11ea-8fe8-023bdfa29edc	40Gi
my-prometheus-claim-prometheus-k8s-0	openshift-monitoring	Bound	pvc-ce290f1b-0ce6-11ea-8fe8-023bdfa29edc	40Gi
my-prometheus-claim-prometheus-k8s-1	openshift-monitoring	Bound	pvc-ce361010-0ce6-11ea-8fe8-023bdfa29edc	40Gi

2. 新規の **alertmanager-main-*** Pod が **Running** 状態が表示されることを確認します。
 - a. **Workloads** → **Pods**に移動します。
 - b. 新規の **alertmanager-main-*** Pod をクリックし、Pod の詳細を表示します。
 - c. **Volumes** にスクロールダウンし、ボリュームに新規 Persistent Volume Claim (永続ボリューム要求、PVC) のいずれかに一致する **Type** の **ocs-alertmanager-claim** があることを確認します (例: **ocs-alertmanager-claim-alertmanager-main-0**)。

alertmanager-main-* Pod に割り当てられた Persistent Volume Claim (永続ボリューム要求、PVC)

Volumes

Name ↓	Mount Path ↓	SubPath ↓	Type	Permissions ↓	Utilized By ↓
config-volume	/etc/alertmanager/config		alertmanager-main	Read/Write	alertmanager
ocs-alertmanager-claim	/alertmanager	alertmanager.db	ocs-alertmanager-claim-alertmanager-main-0	Read/Write	alertmanager

3. 新規 **prometheus-k8s-*** Pod が **Running** 状態が表示されることを確認します。
 - a. 新規 **prometheus-k8s-*** Pod をクリックし、Pod の詳細を表示します。

- b. **Volumes** までスクロールダウンし、ボリュームに新規の Persistent Volume Claim (永続ボリューム要求、PVC) のいずれかに一致する **Type** の **ocs-prometheus-claim** があることを確認します (例: **ocs-prometheus-claim-prometheus-k8s-0**)。

prometheus-k8s-* Pod に割り当てられた Persistent Volume Claim (永続ボリューム要求、PVC)

Name	Mount Path	SubPath	Type	Permissions	Utilized By
config-out	/etc/prometheus/config_out		Container Volume	Read-only	prometheus
ocs-prometheus-claim	/prometheus	prometheus-db	PVC ocs-prometheus-claim-prometheus-k8s-0	Read/Write	prometheus

3.3. OPENSIFT CONTAINER STORAGE のクラスターロギング

クラスターロギングをデプロイして、各種の OpenShift Container Platform サービスについてのログを集計できます。クラスターロギングのデプロイ方法については、「[クラスターロギングのデプロイ](#)」を参照してください。

OpenShift Container Platform の初回のデプロイメントでは、OpenShift Container Storage はデフォルトで設定されず、OpenShift Container Platform クラスターはノードから利用可能なデフォルトストレージのみに依存します。OpenShift ロギング (ElasticSearch) のデフォルト設定を OpenShift Container Storage で対応されるように編集し、OpenShift Container Storage でサポートされるロギング (Elasticsearch) を設定できます。

重要

これらのサービスに十分なストレージ容量があることを常に確認します。これらの重要なサービスのストレージ領域が不足すると、ロギングアプリケーションは動作しなくなり、これを回復することは非常に困難になります。

Red Hat は、これらのサービスのキューレーションおよび保持期間を短く設定することを推奨します。詳細は、OpenShift Container Platform ドキュメントで[クラスターロギング Curator](#)について参照してください。

これらのサービスのストレージ領域が不足している場合は、Red Hat カスタマーポータルにお問い合わせください。

3.3.1. 永続ストレージの設定

ストレージクラス名およびサイズパラメーターを使用して、Elasticsearch クラスターの永続ストレージクラスおよびサイズを設定できます。Cluster Logging Operator は、これらのパラメーターに基づいて、Elasticsearch クラスターの各データノードについて Persistent Volume Claim (永続ボリューム要求、PVC) を作成します。以下は例になります。

```
spec:
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      storage:
        storageClassName: "ocs-storagecluster-ceph-rbd"
        size: "200G"
```

この例では、クラスター内の各データノードが、**200GiB** の **ocs-storagecluster-ceph-rbd** ストレージを要求する Persistent Volume Claim (永続ボリューム要求、PVC) にバインドされるように指定しま

す。それぞれのプライマリーシャードは単一のレプリカによってサポートされます。シャードのコピーはすべてのノードにレプリケートされ、常に利用可能となり、冗長性ポリシーにより2つ以上のノードが存在する場合にコピーを復元できます。Elasticsearch レプリケーションポリシーについての詳細は、「[クラスターロギングのデプロイおよび設定について](#)」の **Elasticsearch レプリケーションポリシー** について参照してください。



注記

ストレージブロックを省略すると、デプロイメントはデフォルトのストレージでサポートされます。以下は例になります。

```
spec:
  logStore:
    type: "elasticsearch"
  elasticsearch:
    nodeCount: 3
    storage: {}
```

詳細は、「[クラスターロギングの設定](#)」を参照してください。

3.3.2. OpenShift Container Storage を使用するためのクラスターロギングの設定

このセクションの手順に従って、OpenShift Container Storage を OpenShift クラスターロギングのストレージとして設定します。



注記

OpenShift Container Storage でロギングを初めて設定する際に、すべてのログを取得できます。ただし、ロギングをアンインストールしてから再インストールすると、古いログが削除され、新しいログのみが処理されます。

前提条件

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

手順

1. OpenShift Web コンソールの左側のペインから **Administration** → **Custom Resource Definitions** をクリックします。
2. Custom Resource Definitions ページで、**ClusterLogging** をクリックします。
3. Custom Resource Definition Overview ページで、Actions メニューから **View Instances** を選択するか、または **Instances** タブをクリックします。
4. Cluster Logging ページで、**Create Cluster Logging** をクリックします。データを読み込むためにページを更新する必要がある場合があります。

5. YAML において、`storageClassName`、をプロビジョナー **openshift-storage.rbd.csi.ceph.com** を使用する **storageclass** に置き換えます。以下の例では、`storageclass` の名前は **ocs-storagecluster-ceph-rbd** です。

```

apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: "openshift-logging"
spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    elasticsearch:
      nodeCount: 3
      storage:
        storageClassName: ocs-storagecluster-ceph-rbd
        size: 200G # Change as per your requirement
        redundancyPolicy: "SingleRedundancy"
  visualization:
    type: "kibana"
    kibana:
      replicas: 1
  curation:
    type: "curator"
    curator:
      schedule: "30 3 * * *"
  collection:
    logs:
      type: "fluentd"
      fluentd: {}

```

OpenShift Container Storage ノードにテイントのマークが付けられている場合、ロギング用に daemonset Pod のスケジューリングを有効にするために容認を追加する必要があります。

```

spec:
  [...]
  collection:
    logs:
      fluentd:
        tolerations:
          - effect: NoSchedule
            key: node.ocs.openshift.io/storage
            value: 'true'
        type: fluentd

```

6. **Save** をクリックします。

検証手順

1. Persistent Volume Claim (永続ボリューム要求、PVC) が **elasticsearch** Pod にバインドされていることを確認します。
 - a. **Storage** → **Persistent Volume Claims** に移動します。

- b. **Project** ドロップダウンを **openshift-logging** に設定します。
- c. Persistent Volume Claim (永続ボリューム要求、PVC) が **elasticsearch-*** Pod に割り当てられ、**Bound** (バインド) の状態で表示されることを確認します。

図3.1 作成済みのバインドされたクラスターロギング

Name	Namespace	Status	Persistent Volume	Requested
elasticsearch-elasticsearch-cdm-9r6z4biv-1	openshift-logging	Bound	pvc-8993013d-1a6e-11ea-8d2f-027bataf61a	200G
elasticsearch-elasticsearch-cdm-9r6z4biv-2	openshift-logging	Bound	pvc-89947c90-1a6e-11ea-8d2f-027bataf61a	200G
elasticsearch-elasticsearch-cdm-9r6z4biv-3	openshift-logging	Bound	pvc-8995f557-1a6e-11ea-8d2f-027bataf61a	200G

2. 新規クラスターロギングが使用されていることを確認します。
 - a. **Workload** → **Pods** をクリックします。
 - b. プロジェクトを **openshift-logging** に設定します。
 - c. 新規の **elasticsearch-*** Pod が **Running** 状態で表示されることを確認します。
 - d. 新規の **elasticsearch-*** Pod をクリックし、Pod の詳細を表示します。
 - e. **Volumes** までスクロールダウンし、elasticsearch ボリュームに新規 Persistent Volume Claim (永続ボリューム要求、PVC) に一致する **Type** があることを確認します (例: **elasticsearch-elasticsearch-cdm-9r6z4biv-3**)。
 - f. Persistent Volume Claim (永続ボリューム要求、PVC) の名前をクリックし、PersistenVolumeClaim Overview ページでストレージクラス名を確認します。

注記

Elasticsearch Pod に割り当てられる PV の詳細シナリオを回避するために、キュレーターの時間を短く設定して使用するようしてください。

Curator を、保持設定に基づいて Elasticsearch データを削除するように設定できます。以下の 5 日間のインデックスデータの保持期間をデフォルトとして設定することが推奨されます。

```
config.yaml: |
  openshift-storage:
    delete:
      days: 5
```

詳細は、「[Elasticsearch データのキュレーション](#)」を参照してください。



注記

Persistent Volume Claim (永続ボリューム要求、PVC) がサポートするクラスターロギングをアンインストールするには、それぞれのデプロイメントガイドのアンインストールについての章に記載されている、クラスターロギング Operator の OpenShift Container Storage からの削除についての手順を使用します。

第4章 OPENSIFT CONTAINER STORAGE を使用した OPENSIFT CONTAINER PLATFORM アプリケーションのサ ポート

OpenShift Container Platform のインストール時に OpenShift Container Storage を直接インストールすることはできません。ただし、Operator Hub を使用して OpenShift Container Platform を既存の OpenShift Container Platform にインストールし、OpenShift Container Platform アプリケーションを OpenShift Container Storage でサポートされるように設定することができます。

前提条件

- OpenShift Container Platform がインストールされ、OpenShift Web コンソールへの管理者アクセスがある。
- 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

1. OpenShift Web コンソールで、以下のいずれかを実行します。

- **Workloads → Deployments** をクリックします。
Deployments ページで、以下のいずれかを実行できます。
 - 既存のデプロイメントを選択し、**Action** メニュー(:) から **Add Storage** オプションをクリックします。
 - 新規デプロイメントを作成してからストレージを追加します。
 - i. **Create Deployment** をクリックして新規デプロイメントを作成します。
 - ii. 要件に応じて **YAML** を編集し、デプロイメントを作成します。
 - iii. **Create** をクリックします。
 - iv. ページ右上の **Actions** ドロップダウンメニューから **Add Storage** を選択します。
- **Workloads → Deployment Configs** をクリックします。
Deployment Configs ページで、以下のいずれかを実行できます。
 - 既存のデプロイメントを選択し、**Action** メニュー(:) から **Add Storage** オプションをクリックします。
 - 新規デプロイメントを作成してからストレージを追加します。
 - i. **Create Deployment Config** をクリックし、新規デプロイメントを作成します。
 - ii. 要件に応じて **YAML** を編集し、デプロイメントを作成します。
 - iii. **Create** をクリックします。
 - iv. ページ右上の **Actions** ドロップダウンメニューから **Add Storage** を選択します。

2. Add Storage ページで、以下のオプションのいずれかを選択できます。

- **Use existing claim** オプションをクリックし、ドロップダウンリストから適切な PVC を選

択します。

- **Create new claim** オプションをクリックします。
 - a. **Storage Class** ドロップダウンリストから適切な **CephFS** または **RBD** ストレージクラスを選択します。
 - b. Persistent Volume Claim (永続ボリューム要求、PVC) の名前を指定します。
 - c. ReadWriteOnce (RWO) または ReadWriteMany (RWX) アクセスモードを選択します。



注記

ReadOnlyMany (ROX) はサポートされないため、非アクティブになります。

- d. 必要なストレージ容量のサイズを選択します。



注記

ブロック PV を拡張することはできますが、Persistent Volume Claim (永続ボリューム要求: PVC) の作成後にストレージ容量のサイズを縮小することはできません。

3. コンテナ内のマウントパスボリュームのマウントパスとサブパス（必要な場合）を指定します。
4. **Save** をクリックします。

検証手順

1. 設定に応じて、以下のいずれかを実行します。
 - **Workloads** → **Deployments** をクリックします。
 - **Workloads** → **Deployment Configs** をクリックします。
2. 必要に応じてプロジェクトを設定します。
3. ストレージを追加したデプロイメントをクリックして、デプロイメントの詳細を表示します。
4. **Volumes** までスクロールダウンし、デプロイメントに、割り当てた Persistent Volume Claim (永続ボリューム要求、PVC) に一致する **Type** があることを確認します。
5. Persistent Volume Claim (永続ボリューム要求、PVC) の名前をクリックし、Persistent Volume Claim Overview ページでストレージクラス名を確認します。

第5章 既存の外部 OPENSIFT CONTAINER STORAGE クラスターへのファイルおよびオブジェクトストレージの追加

OpenShift Container Storage が外部モードで設定されている場合に、Persistent Volume Claim (永続ボリューム要求、PVC) および Object Bucket Claim (オブジェクトバケット要求) 向けにストレージを提供する方法は複数あります。

- ブロックストレージ用の Persistent Volume Claim (永続ボリューム要求、PVC) は、外部の Red Hat Ceph Storage クラスターから直接提供されます。
- ファイルストレージ用の Persistent Volume Claim (永続ボリューム要求、PVC) は、メタデータサーバー (MDS) を外部の Red Hat Ceph Storage クラスターに追加して提供できます。
- オブジェクトストレージのオブジェクトバケット要求は、Multicloud Object Gateway を使用するか、または Ceph Object Gateway を外部の Red Hat Ceph Storage クラスターに追加して提供できます。

以下のプロセスを使用して、ブロックストレージだけを提供するために最初にデプロイされていたファイルストレージ (メタデータサーバー 使用)、オブジェクトストレージ (Ceph Object Gateway 使用) または両方を外部の OpenShift Container Storage クラスターに追加します。

前提条件

- バージョン 4.6 の OpenShift Container Platform クラスターをインストールしておく。
- OpenShift Container Storage ノードを OpenShift Container Storage Operator のバージョン 4.6 と共にインストールしておく。
- 以前のバージョンからバージョン 4.6 にアップグレードした場合には、モニタリングを有効にする ([Object Service ダッシュボードのモニタリングの有効化](#))。
- 外部 Red Hat Ceph Storage クラスターが以下のいずれかまたは両方で設定されている。
 - オブジェクトストレージ用に OpenShift Container Platform クラスターがアクセスできる Ceph Object Gateway (RGW) エンドポイント
 - ファイルストレージ用のメタデータサーバー (MDS) プール
- 外部の OpenShift Container Storage クラスターのデプロイメント時に **ceph-external-cluster-details-exporter.py** スクリプトで使用されるパラメーターを把握している。

手順

1. 以下のコマンドを使用して **ceph-external-cluster-details-exporter.py** Python スクリプトの OpenShift Container Storage 4.6 バージョンをダウンロードします。

```
oc get csv $(oc get csv -n openshift-storage | grep ocs-operator | awk '{print $1}') -n
openshift-storage -o
jsonpath='{.metadata.annotations.external\.features\.ocs\.openshift\.io/export-script}' | base64
--decode > ceph-external-cluster-details-exporter.py
```

2. 更新パーミッションは、外部の Red Hat Ceph Storage クラスターのクライアントノードで **ceph-external-cluster-details-exporter.py** を実行して、外部の Red Hat Ceph Storage クラスターに制限を課します。これには、Red Hat Ceph Storage の管理者に問い合わせる必要がある場合があります。

```
# python3 ceph-external-cluster-details-exporter.py --upgrade \
--run-as-user=ocs-client-name \
--rgw-pool-prefix rgw-pool-prefix
```

--run-as-user

OpenShift Container Storage 4.5 のデプロイメント時に使用されるクライアント名。別のクライアント名が設定されていない場合は、デフォルトのクライアント名 **client.healthchecker** を使用します。

--rgw-pool-prefix

Ceph Object Gateway プールに使用する接頭辞。デフォルトの接頭辞を使用している場合は、省略できます。

3. 外部の Red Hat Ceph Storage クラスターから設定詳細を生成して保存します。

- a. 外部の Red Hat Ceph Storage クラスターのクライアントノードで **ceph-external-cluster-details-exporter.py** を実行して、設定の詳細を生成します。

```
# python3 ceph-external-cluster-details-exporter.py --rbd-data-pool-name rbd-block-pool-name --monitoring-endpoint ceph-mgr-prometheus-exporter-endpoint --
monitoring-endpoint-port ceph-mgr-prometheus-exporter-port --run-as-user ocs-client-name --rgw-endpoint rgw-endpoint --rgw-pool-prefix rgw-pool-prefix
```

--monitoring-endpoint

OpenShift Container Storage クラスターからアクセスできるアクティブな Ceph Manager の IP アドレス。

--monitoring-endpoint-port

Ceph Manager Prometheus Exporter エンドポイントのポート。

--run-as-user

OpenShift Container Storage 4.5 のデプロイメント時に使用されるクライアント名。別のクライアント名が設定されていない場合は、デフォルトのクライアント名 **client.healthchecker** を使用します。

--rgw-endpoint

このパラメーターを指定して OpenShift Container Storage の Ceph Object Gateway でオブジェクトストレージをプロビジョニングします (任意のパラメーター)。

--rgw-pool-prefix

Ceph Object Gateway プールに使用する接頭辞。デフォルトの接頭辞を使用している場合は、省略できます。

ユーザーパーミッションは、以下のように更新されます。

```
caps: [mgr] allow command config
caps: [mon] allow r, allow command quorum_status, allow command version
caps: [osd] allow rwx pool=default.rgw.meta, allow r pool=.rgw.root, allow rw
pool=default.rgw.control, allow rx pool=default.rgw.log, allow x
pool=default.rgw.buckets.index
```



注記

Ceph Object Gateway の詳細 (指定されている場合) 以外の全パラメーター (任意の引数を含む) は、OpenShift Container Storage を外部モードでデプロイした時に使用したものと同じです。

- b. スクリプトの出力を **external-cluster-config.json** ファイルに保存します。
以下の出力例では、生成された設定変更を太字で示しています。

```

[{"name": "rook-ceph-mon-endpoints", "kind": "ConfigMap", "data": {"data":
"xxx.xxx.xxx.xxx:xxxx", "maxMonId": "0", "mapping": "{}"}, {"name": "rook-ceph-mon",
"kind": "Secret", "data": {"admin-secret": "admin-secret", "fsid": "<fs-id>", "mon-secret":
"mon-secret"}}, {"name": "rook-ceph-operator-creds", "kind": "Secret", "data": {"userID":
"client.healthchecker", "userKey": "<user-key>"}, {"name": "rook-csi-rbd-node", "kind":
"Secret", "data": {"userID": "csi-rbd-node", "userKey": "<user-key>"}, {"name": "ceph-
rbd", "kind": "StorageClass", "data": {"pool": "ceph-rbd"}}, {"name": "monitoring-endpoint",
"kind": "CephCluster", "data": {"MonitoringEndpoint": "xxx.xxx.xxx.xxx", "MonitoringPort":
"xxxx"}}, {"name": "rook-csi-rbd-provisioner", "kind": "Secret", "data": {"userID": "csi-rbd-
provisioner", "userKey": "<user-key>"}, {"name": "rook-csi-cephfs-provisioner", "kind":
"Secret", "data": {"adminID": "csi-cephfs-provisioner", "adminKey": "<admin-key>"},
{"name": "rook-csi-cephfs-node", "kind": "Secret", "data": {"adminID": "csi-cephfs-node",
"adminKey": "<admin-key>"}, {"name": "cephfs", "kind": "StorageClass", "data":
{"fsName": "cephfs", "pool": "cephfs_data"}}, {"name": "ceph-rgw", "kind": "StorageClass",
"data": {"endpoint": "xxx.xxx.xxx.xxx:xxxx", "poolPrefix": "default"}}]

```

4. 生成された JSON ファイルをアップロードします。
 - a. OpenShift Web コンソールにログインします。
 - b. **Workloads** → **Secrets** をクリックします。
 - c. **プロジェクト** を **openshift-storage** に設定します。
 - d. **rook-ceph-external-cluster-details** をクリックします。
 - e. **Actions (:)** → **Edit Secret** をクリックします。
 - f. **Browse** をクリックして **external-cluster-config.json** ファイルをアップロードします。
 - g. **Save** をクリックします。

検証手順

- **Overview** → **Home** → **Persistent Storage** をクリックして OpenShift Container Storage クラスターの正常性を確認します。
- ファイルストレージ用のメタデータサーバー (MDS) を追加した場合:
 - a. **Workloads** → **Pods** をクリックして、**csi-cephfsplugin-*** Pod が新規作成され、状態が **Running** であることを確認します。
 - b. **Storage** → **Storage Classes** をクリックして **ocs-external-storagecluster-cephfs** ストレージクラスが作成されていることを確認します。
- オブジェクトストレージ用に Ceph Object Gateway を追加した場合:

- a. **Storage** → **Storage Classes** をクリックして **ocs-external-storagecluster-ceph-rgw** ストレージクラスが作成されていることを確認します。
- b. OpenShift Web コンソールから **Home** → **Overview** をクリックし、**Object Service** タブをクリックします。**Status** カードで Object Service に緑色のチェックアイコンが表示されていることを確認します。

第6章 RED HAT OPENSIFT CONTAINER STORAGE に専用のワーカーノードを使用する方法

インフラストラクチャーノードを使用して Red Hat OpenShift Container Storage リソースをスケジュールすると、Red Hat OpenShift Container Platform サブスクリプションコストを節約できます。**infra** ノードロールのラベルのある Red Hat OpenShift Container Platform (RHOCP) ノードには OpenShift Container Storage サブスクリプションが必要ですが、RHOCP サブスクリプションは必要ありません。

マシン API サポートの有無にかかわらず複数の環境全体で一貫性を維持することが重要です。そのため、いずれの場合でも、worker または infra のいずれかのラベルが付けられたノードの特別なカテゴリーや、両方のロールを使用できるようにすることが強く推奨されます。詳細は、「[インフラストラクチャーノードの手動作成](#)」セクションを参照してください。

6.1. インフラストラクチャーノードの仕組み

OpenShift Container Storage で使用するインフラストラクチャーノードにはいくつかの属性があります。ノードが RHOCP エンタイトルメントを使用しないようにするには、**infra** ノードロールのラベルが必要です。**infra** ノードロールラベルは、OpenShift Container Storage を実行するノードには OpenShift Container Storage エンタイトルメントのみが必要となるようにします。

- **node-role.kubernetes.io/infra** のラベル

infra ノードが OpenShift Container Storage リソースのみをスケジュールできるようにするには、**NoSchedule** effect のある OpenShift Container Storage テイントを追加する必要があります。

- **node.ocs.openshift.io/storage="true"** のテイント

RHOCP サブスクリプションコストが適用されないように、ラベルは RHOCP ノードを **infra** ノードとして識別します。テイントは、OpenShift Container Storage 以外のリソースがテイントのマークが付けられたノードでスケジュールされないようにします。

OpenShift Container Storage サービスの実行に使用されるインフラストラクチャーノードに必要なテイントおよびラベルの例:

```
spec:
  taints:
  - effect: NoSchedule
    key: node.ocs.openshift.io/storage
    value: "true"
  metadata:
    creationTimestamp: null
  labels:
    node-role.kubernetes.io/worker: ""
    node-role.kubernetes.io/infra: ""
    cluster.ocs.openshift.io/openshift-storage: ""
```

6.2. インフラストラクチャーノードを作成するためのマシンセット

マシン API が環境でサポートされている場合には、インフラストラクチャーノードのプロビジョニングを行うマシンセットのテンプレートにラベルを追加する必要があります。ラベルをマシン API によって作成されるノードに手動で追加するアンチパターンを回避します。これを実行することは、デプロイメントで作成される Pod にラベルを追加することに似ています。いずれの場合も、Pod/ノードが失敗する場合、置き換え用の Pod/ノードには適切なラベルがありません。



注記

EC2 環境では、3つのマシンセットが必要です。それぞれは、異なるアベイラビリティゾーン (us-east-2a、us-east-2b、us-east-2c など) でインフラストラクチャーノードをプロビジョニングするように設定されます。現時点で、OpenShift Container Storage は4つ以上のアベイラビリティゾーンへのデプロイをサポートしていません。

以下の Machine Set テンプレートのサンプルは、インフラストラクチャーノードに必要な適切なテイントおよびラベルを持つノードを作成します。これは OpenShift Container Storage サービスを実行するために使用されます。

```
template:
  metadata:
    creationTimestamp: null
  labels:
    machine.openshift.io/cluster-api-cluster: kb-s25vf
    machine.openshift.io/cluster-api-machine-role: worker
    machine.openshift.io/cluster-api-machine-type: worker
    machine.openshift.io/cluster-api-machineset: kb-s25vf-infra-us-west-2a
  spec:
    taints:
    - effect: NoSchedule
      key: node.ocs.openshift.io/storage
      value: "true"
  metadata:
    creationTimestamp: null
    labels:
      node-role.kubernetes.io/infra: ""
      cluster.ocs.openshift.io/openshift-storage: ""
```

6.3. インフラストラクチャーノードの手動作成

マシン API が環境内でサポートされない場合にのみ、ラベルはノードに直接適用される必要があります。手動作成では、OpenShift Container Storage サービスをスケジュールするために少なくとも3つの RHOCP ワーカーノードが利用可能であり、これらのノードに CPU およびメモリーリソースが十分にある必要があります。RHOCP サブスクリプションコストの発生を防ぐには、以下が必要です。

```
oc label node <node> node-role.kubernetes.io/infra=""
oc label node <node> cluster.ocs.openshift.io/openshift-storage=""
```

また、**NoSchedule** OpenShift Container Storage テイントを追加することも、**infra** ノードが OpenShift Container Storage リソースのみをスケジュールし、その他の OpenShift Container Storage ワークロードを拒否できるようにするために必要です。

```
oc adm taint node <node> node.ocs.openshift.io/storage="true":NoSchedule
```



警告

ノードロール `node-role.kubernetes.io/worker=""` は削除しないでください。

`node-role.kubernetes.io/worker=""` ノードロールを削除すると、OpenShift スケジューラーおよび MachineConfig リソースの両方に変更が加えられない場合に問題が発生する可能性があります。

すでに削除されている場合は、各 **infra** ノードに再度追加する必要があります。`node-role.kubernetes.io/infra=""` ノードロールおよび OpenShift Container Storage ティントを追加するだけで、エンタイトルメント免除要件を満たすことができます。

第7章 PERSISTENT VOLUME CLAIM (永続ボリューム要求、PVC) の管理

7.1. OPENSIFT CONTAINER PLATFORM を使用するためのアプリケーション POD の設定

このセクションの手順に従って、OpenShift Container Storage をアプリケーション Pod のストレージとして設定します。

前提条件

- OpenShift Web コンソールへの管理者アクセスがある。
- OpenShift Container Storage Operator が **openshift-storage** namespace にインストールされ、実行されている。OpenShift Web Console で、**Operators** → **Installed Operators** をクリックしてインストールされた Operator を表示します。
- OpenShift Container Storage が提供するデフォルトのストレージクラスが利用可能である。OpenShift Web コンソールで **Storage** → **Storage Class** をクリックし、デフォルトのストレージクラスを表示します。

手順

1. 使用するアプリケーションの Persistent Volume Claim (永続ボリューム要求、PVC) を作成します。
 - a. OpenShift Web コンソールで、**Storage** → **Persistent Volume Claims** をクリックします。
 - b. アプリケーション Pod の **Project** を設定します。
 - c. **Create Persistent Volume Claim** をクリックします。
 - i. OpenShift Container Storage によって提供される **Storage Class** を指定します。
 - ii. PVC **Name** (例: **myclaim**) を指定します。
 - iii. 必要な **Access Mode** を選択します。
 - iv. アプリケーション要件に応じて **Size** を指定します。
 - v. **Create** をクリックし、PVC のステータスが **Bound** になるまで待機します。
2. 新規または既存のアプリケーション Pod を新規 PVC を使用するように設定します。
 - 新規アプリケーション Pod の場合、以下の手順を実行します。
 - i. **Workloads** → **Pods** をクリックします。
 - ii. 新規アプリケーション Pod を作成します。
 - iii. **spec:** セクションで、**volume:** セクションを追加し、新規 PVC をアプリケーション Pod のボリュームとして追加します。

```
volumes:  
- name: <volume_name>
```

```

persistentVolumeClaim:
  claimName: <pvc_name>

```

以下は例になります。

```

volumes:
- name: mypd
  persistentVolumeClaim:
    claimName: myclaim

```

- 既存のアプリケーション Pod の場合、以下の手順を実行します。
 - i. **Workloads** → **Deployment Configs** をクリックします。
 - ii. アプリケーション Pod に関連付けられた必要なデプロイメント設定を検索します。
 - iii. **Action menu (⋮)** → **Edit Deployment Config** をクリックします。
 - iv. **spec:** セクションで、**volume:** セクションを追加し、新規 PVC をアプリケーション Pod のボリュームとして追加し、**Save** をクリックします。

```

volumes:
- name: <volume_name>
  persistentVolumeClaim:
    claimName: <pvc_name>

```

以下は例になります。

```

volumes:
- name: mypd
  persistentVolumeClaim:
    claimName: myclaim

```

3. **新規設定が使用されていることを確認します。**
 - a. **Workloads** → **Pods** をクリックします。
 - b. アプリケーション Pod の **Project** を設定します。
 - c. アプリケーション Pod が **Running** ステータスで表示されていることを確認します。
 - d. アプリケーション Pod 名をクリックし、Pod の詳細を表示します。
 - e. **Volumes** セクションまでスクロールダウンし、ボリュームに新規 Persistent Volume Claim (永続ボリューム要求、PVC) に一致する **Type** があることを確認します (例: **myclaim**)。

7.2. PERSISTENT VOLUME CLAIM (永続ボリューム要求、PVC) 要求ステータスの表示

以下の手順を使用して、PVC 要求のステータスを表示します。

前提条件

- OpenShift Container Storage への管理者アクセス。

手順

1. OpenShift Web コンソールにログインします。
2. **Storage** → **Persistent Volume Claims** をクリックします。
3. **Filter** テキストボックスを使用して、必要な PVC 名を検索します。また、一覧を絞り込むために Name または Label で PVC の一覧をフィルターすることもできます。
4. 必要な PVC に対応する **Status** 列を確認します。
5. 必要な **Name** をクリックして PVC の詳細を表示します。

7.3. PERSISTENT VOLUME CLAIM (永続ボリューム要求、PVC) 要求イベントの確認

以下の手順を使用して、Persistent Volume Claim (永続ボリューム要求、PVC) 要求イベントを確認し、これに対応します。

前提条件

- OpenShift Web コンソールへの管理者アクセス。

手順

1. OpenShift Web コンソールにログインします。
2. **Home** → **Overview** → **Persistent Storage** をクリックします。
3. **Inventory** カードを見つけ、エラーのある PVC の数を確認します。
4. **Storage** → **Persistent Volume Claims** をクリックします。
5. **Filter** テキストボックスを使用して、必要な PVC を検索します。
6. PVC 名をクリックし、**Events** に移動します。
7. 必要に応じて、または指示に応じてイベントに対応します。

7.4. PERSISTENT VOLUME CLAIM (永続ボリューム要求、PVC) の拡張

OpenShift Container Storage 4.6 では、Persistent Volume Claim (永続ボリューム要求、PVC) を拡張する機能が導入され、永続ストレージリソース管理の柔軟性が向上します。

拡張は、以下の永続ボリュームでサポートされます。

- ボリュームモードが **Filesystem** の Ceph File System (CephFS) をベースとする PVC (ReadWriteOnce (RWO) および ReadWriteMany (RWX) アクセス)。
- ボリュームモードが **Filesystem** の Ceph RADOS Block Device (Ceph RBD) をベースとする PVC (ReadWriteOnce (RWO) アクセス)。
- ボリュームモードが **Block** の Ceph RADOS Block Device (Ceph RBD) をベースとする PVC (ReadWriteOnce (RWO) アクセス)。



警告

OSD および MON PVC の拡張機能は Red Hat によってサポートされていません。

前提条件

- OpenShift Web コンソールへの管理者アクセス。

手順

1. OpenShift Web コンソールで、**Storage** → **Persistent Volume Claims** に移動します。
2. 拡張する Persistent Volume Claim (永続ボリューム要求、PVC) の横にある Action メニュー (⋮) をクリックします。
3. **Expand PVC** をクリックします。

The screenshot shows the OpenShift web console interface. The left sidebar is open to 'Storage' > 'Persistent Volume Claims'. The main area displays a table of PVCs. The first row is selected, and its action menu is open, showing the 'Expand PVC' option.

Name	Namespace	Status	Persistent Volume	Capacity	Used	Storage Class
pvc-db-noobaa-db-0	openshift-storage	Bound	pvc-81a35e5a-357f-45f4-93c4-8e679f58f661	50 GiB	3551 MiB	ocs-storagecluster-ceph-
pvc-ocs-deviceset-0-data-0-r6sw5	openshift-storage	Bound	pvc-71185f4d-caaa-4a00-8a94-da49517c5b06	512 GiB	-	ocs-storagecluster-ceph-
pvc-ocs-deviceset-1-data-0-5wh9l	openshift-storage	Bound	pvc-547f6053-cd21-404d-a77f-d861336937f8	512 GiB	-	ocs-storagecluster-ceph-

4. Persistent Volume Claim (永続ボリューム要求、PVC) の新しいサイズを選択してから、**Expand** をクリックします。

Expand Persistent Volume Claim

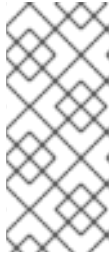
Increase the capacity of claim `db-noobaa-db-0`. This can be a time-consuming process.

Size *

Cancel

Expand

5. 拡張を確認するには、PVC の詳細ページに移動し、**Capacity** フィールドでサイズが正しく要求されていることを確認します。



注記

Ceph RADOS Block Device (RBD) に基づいて PVC を拡張する場合、PVC がまだ Pod に割り当てられていない場合は、PVC の詳細ページで **Condition type** は **FileSystemResizePending** になります。ボリュームをマウントすると、ファイルシステムのサイズ変更が正常に実行され、新しいサイズが **Capacity** フィールドに反映されます。

7.5. 動的プロビジョニング

7.5.1. 動的プロビジョニングについて

StorageClass リソースオブジェクトは、要求可能なストレージを記述し、分類するほか、要求に応じて動的にプロビジョニングされるストレージのパラメーターを渡すための手段を提供します。

StorageClass オブジェクトは、さまざまなレベルのストレージおよびストレージへのアクセスを制御するための管理メカニズムとしても機能します。クラスター管理者 (**cluster-admin**) またはストレージ管理者 (**storage-admin**) は、ユーザーが基礎となるストレージボリュームソースに関する詳しい知識なしに要求できる StorageClass オブジェクトを定義し、作成します。

OpenShift Container Platform の永続ボリュームフレームワークはこの機能を有効にし、管理者がクラスターに永続ストレージをプロビジョニングできるようにします。フレームワークにより、ユーザーは基礎となるインフラストラクチャーの知識がなくてもこれらのリソースを要求できるようになります。

OpenShift Container Platform では、数多くのストレージタイプを永続ボリュームとして使用することができます。これらはすべて管理者によって静的にプロビジョニングされますが、一部のストレージタイプは組み込みプロバイダーとプラグイン API を使用して動的に作成できます。

7.5.2. OpenShift Container Storage の動的プロビジョニング

Red Hat OpenShift Container Storage は、コンテナ環境向けに最適化されたソフトウェアで定義されるストレージです。これは OpenShift Container Platform の Operator として実行され、コンテナの統合され、単純化された永続ストレージの管理を可能にします。

OpenShift Container Storage は、以下を含む各種のストレージタイプをサポートします。

- データベースのブロックストレージ
- 継続的な統合、メッセージングおよびデータ集約のための共有ファイルストレージ
- アーカイブ、バックアップおよびメディアストレージのオブジェクトストレージ

バージョン 4 では、Red Hat Ceph Storage を使用して永続ボリュームをサポートするファイル、ブロック、およびオブジェクトストレージを提供し、Rook.io を使用して永続ボリュームおよび要求のプロビジョニングを管理し、オーケストレーションします。NooBaa はオブジェクトストレージを提供し、その Multicloud Gateway は複数のクラウド環境でのオブジェクトのフェデレーションを可能にします (テクノロジープレビューとしてご利用いただけます)。

OpenShift Container Storage 4 では、RADOS Block Device (RBD) および Ceph File System (CephFS) の Red Hat Ceph Storage Container Storage Interface (CSI) ドライバーが動的プロビジョニング要求を処理します。PVC 要求が動的に送信される場合、CSI ドライバーでは以下のオプションを使用できません。

- ボリュームモードが **Block** の Ceph RBD をベースとする PVC (ReadWriteOnce (RWO) および ReadWriteMany (RWX) アクセス) を作成します。

- ボリュームモードが **Filesystem** の Ceph RBD をベースとする PVC (ReadWriteOnce (RWO) アクセス) を作成します。
- ボリュームモードが **Filesystem** の CephFS をベースとする PVC (ReadWriteOnce (RWO) および ReadWriteMany (RWX) アクセス) を作成します。

使用するドライバー (RBD または CephFS) の判断は、**storageclass.yaml** ファイルのエントリーに基づいて行われます。

7.5.3. 利用可能な動的プロビジョニングプラグイン

OpenShift Container Platform は、以下のプロビジョナープラグインを提供します。これらには、クラスターの設定済みプロバイダーの API を使用して新規ストレージリソースを作成する動的プロビジョニング用の一般的な実装が含まれます。

ストレージタイプ	プロビジョナープラグインの名前	注記
OpenStack Cinder	kubernetes.io/cinder	
AWS Elastic Block Store (EBS)	kubernetes.io/aws-ebs	異なるゾーンで複数のクラスターを使用する場合の動的プロビジョニングの場合、各ノードに Key=kubernetes.io/cluster/<cluster_name>,Value=<cluster_id> のタグを付けます。ここで、 <cluster_name> と <cluster_id> はクラスターごとに一意になります。
AWS Elastic File System (EFS)		動的プロビジョニングは、EFS プロビジョナー Pod で実行され、プロビジョナープラグインでは実行されません。
Azure Disk	kubernetes.io/azure-disk	
Azure File	kubernetes.io/azure-file	persistent-volume-binder ServiceAccount では、Azure ストレージアカウントおよびキーを保存するためにシークレットを作成し、取得するためのパーミッションが必要です。
GCE Persistent Disk (gcePD)	kubernetes.io/gce-pd	マルチゾーン設定では、GCE プロジェクトごとに OpenShift Container Platform クラスターを実行し、現行クラスターのノードが存在しないゾーンで PV が作成されないようにすることが推奨されます。

ストレージタイプ	プロビジョナープラグインの名前	注記
VMware vSphere	kubernetes.io/vsphere-volume	



重要

選択したプロビジョナープラグインでは、関連するクラウド、ホスト、またはサードパーティープロバイダーを、関連するドキュメントに従って設定する必要があります。

第8章 ボリュームスナップショット

ボリュームスナップショットは、特定の時点におけるクラスター内のストレージボリュームの状態を表します。これらのスナップショットは、毎回フルコピーを作成する必要がないので、より効率的にストレージを使用するのに役立ち、アプリケーション開発のビルディングブロックとして使用できます。

ボリュームスナップショットクラスを使用すると、管理者はボリュームスナップショットオブジェクトに属する異なる属性を指定できます。OpenShift Container Storage Operator は、使用されるプラットフォームに応じてデフォルトのボリュームスナップショットクラスをインストールします。これらのデフォルトボリュームスナップショットクラスは Operator によって所有され、制御されるため、削除したり変更したりすることはできません。

同じ永続ボリューム要求 (PVC) の複数のスナップショットを作成できます。CephFS の場合、PVC ごとに最大 100 スナップショットを作成できます。RADOS Block Device (RBD) の場合、PVC ごとに最大 512 スナップショットを作成できます。



注記

スナップショットの定期的な作成をスケジュールすることはできません。

8.1. ボリュームスナップショットの作成

Persistent Volume Claim (永続ボリューム要求、PVC) ページまたは Volume Snapshots ページのいずれかからボリュームスナップショットを作成できます。

前提条件

- PVC は **Bound** 状態にある必要があり、使用中の状態にすることはできません。



注記

Pod が使用している場合、OpenShift Container Storage は PVC のボリュームスナップショットのクラッシュの一貫性だけを提供します。アプリケーションの一貫性を保つために、まず実行中の Pod を破棄してスナップショットの一貫性を確保するか、またはアプリケーションが提供する静止メカニズムを使用してこれを確保します。

手順

Persistent Volume Claims ページの使用

1. OpenShift Web コンソールで、**Storage** → **Persistent Volume Claims** をクリックします。
2. ボリュームのスナップショットを作成するには、以下のいずれかを実行します。
 - 必要な PVC の横にある Action メニュー (!) → **Create Snapshot** をクリックします。
 - スナップショットを作成する PVC をクリックし、**Actions** → **Create Snapshot** をクリックします。
3. ボリュームスナップショットの **Name** を入力します。
4. ドロップダウンリストから **Snapshot Class** を選択します。
5. **Create** をクリックします。作成されるボリュームスナップショットの Details ページにリダイレクトされます。

Volume Snapshots ページの使用

1. OpenShift Web コンソールで **Storage** → **Volume Snapshots** をクリックします。
2. **Volume Snapshots** ページで、**Create Volume Snapshot** をクリックします。
3. ドロップダウンリストから必要な **Project** を選択します。
4. ドロップダウンリストから **Persistent Volume Claim** を選択します。
5. スナップショットの **Name** を入力します。
6. ドロップダウンリストから **Snapshot Class** を選択します。
7. **Create** をクリックします。作成されるボリュームスナップショットの Details ページにリダイレクトされます。

検証手順

- PVC の **Details** ページに移動し、**Volume Snapshots** タブをクリックしてボリュームスナップショットの一覧を表示します。新規スナップショットが一覧表示されていることを確認します。
- OpenShift Web コンソールで **Storage** → **Volume Snapshots** をクリックします。新規スナップショットが一覧表示されていることを確認します。
- ボリュームスナップショットが **Ready** 状態になるまで待機します。

8.2. ボリュームスナップショットの復元

ボリュームスナップショットを復元する際に、新規の Persistent Volume Claim (永続ボリューム要求、PVC) が作成されます。復元される PVC はボリュームスナップショットおよび親 PVC とは切り離されています。

Persistent Volume Claim ページまたは Volume Snapshots ページのいずれかからボリュームスナップショットを復元できます。

手順

Persistent Volume Claims ページの使用

親 PVC が存在する場合に限り、Persistent Volume Claims ページからボリュームスナップショットを復元できます。

1. OpenShift Web コンソールで、**Storage** → **Persistent Volume Claims** をクリックします。
2. 新規 PVC として復元する必要があるボリュームスナップショットがある PVC 名をクリックします。
3. **Volume Snapshots** タブで、必要なボリュームスナップショットの横にある Action メニュー (⋮) → **Restore as new PVC** をクリックします。
4. 新規 PVC の名前を入力します。
5. **Storage Class** 名を選択します。



注記

(Rados Block Device (RBD) の場合) 親 PVC と同じプールが指定されるストレージクラスを選択する必要があります。

6. **Restore** をクリックします。新規 PVC の詳細ページにリダイレクトされます。

Volume Snapshots ページの使用

1. OpenShift Web コンソールで **Storage** → **Volume Snapshots** をクリックします。
2. 必要なボリュームスナップショットの横にある Action Menu (⋮) → **Restore as new PVC** をクリックします。
3. 新規 PVC の名前を入力します。
4. **Storage Class** 名を選択します。



注記

(Rados Block Device (RBD) の場合) 親 PVC と同じプールが指定されるストレージクラスを選択する必要があります。

5. **Restore** をクリックします。新規 PVC の詳細ページにリダイレクトされます。



注記

ボリュームスナップショットの復元時に、PVC は親 PVC が存在する場合にのみ、親 PVC のアクセスモードで作成されます。それ以外の場合は、PVC は ReadWriteOnce (RWO) アクセスモードでのみ作成されます。現時点で、OpenShift Web コンソールを使用してアクセスモードを指定することはできません。ただし、YAML を使用して CLI からアクセスモードを指定できます。詳細は、[ボリュームスナップショットの復元](#)について参照してください。

検証手順

- OpenShift Web コンソールから **Storage** → **Persistent Volume Claims** をクリックし、新規 PVC が **Persistent Volume Claims** ページに一覧表示されていることを確認します。
- 新規 PVC が **Bound** の状態になるまで待機します。

8.3. ボリュームスナップショットの削除

前提条件

- ボリュームスナップショットを削除する場合は、その特定のボリュームスナップショットで使用されるボリュームスナップショットクラスが存在している必要があります。

手順

Persistent Volume Claims ページの使用

1. OpenShift Web コンソールで、**Storage** → **Persistent Volume Claims** をクリックします。

2. 削除する必要があるボリュームスナップショットがある PVC 名をクリックします。
3. **Volume Snapshots** タブで、必要なボリュームスナップショットの横にある Action メニュー (⋮) → **Delete Volume Snapshot** をクリックします。

Volume Snapshots ページの使用

1. OpenShift Web コンソールで **Storage** → **Volume Snapshots** をクリックします。
2. **Volume Snapshots** ページで、必要なスナップショットの横にある Action メニュー (⋮) → **Delete Volume Snapshot** をクリックします。

検証手順

- 削除されたボリュームスナップショットが PVC の詳細ページの **Volume Snapshots** タブにな
いことを確認します。
- **Storage** → **Volume Snapshots** をクリックし、削除されたボリュームスナップショットが一覧
表示されていないことを確認します。

第9章 ボリュームのクローン作成

クローンは、標準のボリュームとして使用される既存のストレージボリュームの複製です。ボリュームのクローンを作成し、データの特定の時点のコピーを作成します。永続ボリューム要求 (PVC) は別のサイズでクローンできません。CephFS および RADOS Block Device (RBD) の両方で、PVC ごとに最大 512 のクローンを作成できます。

9.1. クローンの作成

前提条件

- ソース PVC は **Bound** 状態にある必要があり、使用中の状態にすることはできません。



注記

Pod が PVC を使用している場合は、PVC のクローンを作成しません。これを実行すると、PVC が一時停止 (停止) されないため、データが破損する可能性があります。

手順

1. OpenShift Web コンソールで、**Storage** → **Persistent Volume Claims** をクリックします。
2. クローンを作成するには、以下のいずれかを実行します。
 - 必要な PVC の横にある Action メニュー (⋮) → **Clone PVC** をクリックします。
 - クローンを作成する必要がある PVC をクリックし、**Actions** → **Clone PVC** をクリックします。
3. クローンの **Name** を入力します。
4. **Clone** をクリックします。新規 PVC の詳細ページにリダイレクトされます。



注記

クローンは、親 PVC のアクセスモードで作成されます。現時点で、OpenShift Web コンソール UI を使用してアクセスモードを指定することはできません。ただし、YAML を使用して CLI からアクセスモードを指定できます。詳細は、[CSI ボリュームクローンのプロビジョニング](#)について参照してください。

5. クローン作成された PVC のステータスが **Bound** になるまで待機します。クローン作成された PVC が Pod で使用できるようになります。このクローン作成された PVC は dataSource PVC とは切り離されています。

第10章 CONTAINER STORAGE INTERFACE (CSI) コンポーネントの配置の管理

各クラスターは、**infra** や **storage** ノードなどの数多くの専用ノードで構成されます。ただし、カスタムテイントを持つ **infra** ノードは、ノードで OpenShift Container Storage Persistent Volume Claims (永続ボリューム要求、PVC) を使用することができません。そのため、このようなノードを使用する必要がある場合は、容認を設定してノードで **csi-plugins** を起動することができます。詳細は、<https://access.redhat.com/solutions/4827161> を参照してください。

手順

1. configmap を編集して、カスタムテイントの容認を追加します。エディターを終了する前に必ず保存します。

```
$ oc edit configmap rook-ceph-operator-config -n openshift-storage
```

2. **configmap** を表示して、追加された容認を確認します。

```
$ oc get configmap rook-ceph-operator-config -n openshift-storage -o yaml
```

テイント **nodetype=infra:NoSchedule** の追加された容認の出力例

```
apiVersion: v1
data:
[...]
```

CSI_PLUGIN_TOLERATIONS: |

```
- key: nodetype
  operator: Equal
  value: infra
  effect: NoSchedule
- key: node.ocs.openshift.io/storage
  operator: Equal
  value: "true"
  effect: NoSchedule
[...]
```

```
kind: ConfigMap
metadata:
[...]
```

3. 独自の **infra** ノードで **csi-cephfsplugin-*** および **csi-rbdplugin-*** Pod の起動に失敗した場合、**rook-ceph-operator** を再起動します。

```
$ oc delete -n openshift-storage pod <name of the rook_ceph_operator pod>
```

例:

```
$ oc delete -n openshift-storage pod rook-ceph-operator-5446f9b95b-jrn2j
pod "rook-ceph-operator-5446f9b95b-jrn2j" deleted
```

検証手順

csi-cephfsplugin-* および **csi-rbdplugin-*** Pod が **infra** ノードで実行されていることを確認します。

