



Red Hat OpenShift Data Foundation 4.9

Regional-DR 向け Advanced Cluster Management と OpenShift Data Foundation の 設定

災害復旧機能を備えたストレージインフラストラクチャーを提供するために、2つの異なる地理的ロケーション間で OpenShift Data Foundation を設定する方法

Red Hat OpenShift Data Foundation 4.9 Regional-DR 向け Advanced Cluster Management と OpenShift Data Foundation の設定

災害復旧機能を備えたストレージインフラストラクチャーを提供するために、2つの異なる地理的ロケーション間で OpenShift Data Foundation を設定する方法

Enter your first name here. Enter your surname here.

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

Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Configuring_OpenShift_Data_Foundation_for_Regional-DR_with_Advanced_Cluster_Management.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

このソリューションガイドの目的は、災害復旧向けに OpenShift Data Foundation を Advanced Cluster Management と共にデプロイするのに必要な手順の詳細を提供し、可用性の高いストレージインフラストラクチャーを実現することです。Configuring OpenShift Data Foundation for Regional-DR with Advanced Cluster Management is a developer preview feature and is subject to developer preview support limitations. Developer preview releases are not intended to be run in production environments and are not supported through the Red Hat Customer Portal case

management system. If you need assistance with developer preview features, reach out to the ocs-devpreview@redhat.com mailing list and a member of the Red Hat Development Team will assist you as quickly as possible based on their availability and work schedules.

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 REGIONAL-DR の概要	5
1.1. REGIONAL-DR ソリューションのコンポーネント	5
1.2. REGIONAL-DR デプロイメントワークフロー	6
第2章 REGIONAL-DR を有効にするための要件	8
第3章 マルチサイトストレージレプリケーションの設定	10
3.1. マネージドクラスターでの OMAP ジェネレーターおよびボリュームレプリケーションの有効化	10
3.2. OPENSIFT DATA FOUNDATION のマルチクラスターオーケストレーターのインストール	11
3.3. ハブクラスターでのミラーピアの作成	11
3.4. マネージドクラスターでのミラーリングの有効化	12
第4章 VOLUMEREPLICATIONCLASS リソースの作成	14
第5章 ミラーリング STORAGECLASS リソースの作成	15
第6章 マネージドクラスターでの OPENSIFT-DR CLUSTER OPERATOR のインストール	16
第7章 ハブクラスターへの OPENSIFT-DR HUB OPERATOR のインストール	22
第8章 ハブクラスターでの障害復旧ポリシーの作成	24
第9章 サンプルアプリケーションの作成	26
9.1. サンプルアプリケーションの削除	29
第10章 マネージドクラスター間のアプリケーションのフェイルオーバー	31
第11章 マネージドクラスター間のアプリケーションの再配置	34

多様性を受け入れるオープンソースの強化

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

RED HAT ドキュメントへのフィードバック (英語のみ)

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 REGIONAL-DR の概要

障害復旧は、自然または人が原因の障害からビジネスクリティカルなアプリケーションを復旧し、継続する機能です。これは、深刻な障害イベント中にビジネス運営の継続性を確保できるように設計されている、主要な組織の全体的なビジネス継続ストラテジーです。

Regional-DR 機能は、地理的に分散しているサイト間でボリュームの永続的なデータとメタデータのレプリケーションを提供します。パブリッククラウドでは、リージョンの失敗から保護することにあります。Regional-DR は、地理的な地域が利用できない場合でもビジネス継続性を確保し、予測可能な量のデータの損失を受け入れます。これは通常、目標復旧時点 (RPO) および目標復旧時間 (RTO) で表されます。

- RPO は、永続データのバックアップまたはスナップショットを作成する頻度の尺度です。実際には、RPO は、停止後に失われるか、再入力する必要があるデータの量を示します。
- RTO は、企業が許容できるダウンタイムの量です。RTO は、ビジネスの中断が通知されてからシステムが回復するまでにどのくらいの時間がかかりますか？という質問に答えます。

本書の目的は、障害復旧が有効になるようにインフラストラクチャーを設定するのに必要な手順およびコマンドについて詳しく説明することです。

1.1. REGIONAL-DR ソリューションのコンポーネント

Regional-DR は、Red Hat Advanced Cluster Management for Kubernetes (RHACM) と OpenShift Data Foundation コンポーネントで設定され、OpenShift Container Platform クラスター全体でアプリケーションとデータのモビリティを提供します。

Red Hat Advanced Cluster Management for Kubernetes (RHACM)

RHACM は、複数のクラスターとアプリケーションのライフサイクルを管理する機能を提供します。したがって、マルチクラスター環境でのコントロールプレーンとして機能します。

RHACM は 2 つの部分に分かれています。

- RHACM Hub: マルチクラスターコントロールプレーンで実行されるコンポーネント
- マネージドクラスター: マネージドクラスターで実行されるコンポーネント

この製品の詳細は、[RHACM のドキュメント](#) および [RHACM のアプリケーションの管理](#) を参照してください。

OpenShift Data Foundation

OpenShift Data Foundation は、OpenShift Container Platform クラスターでステートフルなアプリケーション用のストレージをプロビジョニングし、管理する機能を提供します。

OpenShift Data Foundation はストレージプロバイダーとして Ceph をベースとしていて、そのライフサイクルは OpenShift Data Foundation コンポーネントスタックの Rook によって管理されます。Ceph-CSI は、ステートフルなアプリケーション用の永続ボリュームのプロビジョニングと管理を提供します。

OpenShift Data Foundation スタックは、以下の機能で強化されています。

- ミラーリングのプールを有効にする
- RBD プール間でイメージを自動的にミラーリングする

- Persistent Volume Claim ミラーリングごとに管理する csi アドオンを提供します

OpenShift DR

OpenShift DR は、RHACM を使用してデプロイおよび管理される一連のピア OpenShift クラスター全体のステートフルアプリケーションの障害復旧オーケストレーターであり、永続ボリュームでのアプリケーションの状態のライフサイクルのオーケストレーションを行うためのクラウドネイティブインターフェイスを提供します。これらには以下が含まれます。

- OpenShift クラスター間でアプリケーションの状態の関係を保護する
- 現在デプロイされているクラスターが利用できなくなった場合に、アプリケーションの状態をピアクラスターにフェイルオーバーする
- アプリケーションの状態を以前にデプロイされたクラスターに再配置します

OpenShift DR は 3 つのコンポーネントに分類されます。

- **ODF Multicluster Orchestrator:** マルチクラスターコントロールプレーン (RHACM ハブ) にインストールされ、ブートストラップトークンを作成し、管理対象クラスター間でこのトークンを交換します。
- **OpenShift DR Hub Operator:** アプリケーションのフェイルオーバーと再配置を管理するためにハブクラスターにインストールされます。
- **OpenShift DR Cluster Operator:** 各マネージドクラスターにインストールされ、アプリケーションのすべての PVC のライフサイクルを管理します。

1.2. REGIONAL-DR デプロイメントワークフロー

このセクションでは、OpenShift Data Foundation バージョン 4.9 および RHACM バージョン 2.4 を使用して Regional-DR 機能を 2 つの異なる OpenShift Container Platform クラスターに設定およびデプロイするために必要な手順の概要を説明します。2 つのマネージドクラスターに加えて、Advanced Cluster Management ハブソリューションをデプロイするのに、3 つ目の OpenShift Container Platform クラスターが必要です。

インフラストラクチャーを設定するには、指定した順序で以下の手順を実行します。

1. Regional-DR の各要件を満たしていることを確認してください。 [Regional-DR を有効にするための要件](#) を参照してください。
2. 2 つの OpenShift Data Foundation マネージドクラスター間でミラーリング関係を作成して、マルチサイトストレージレプリケーションを設定します。 [マルチサイトストレージレプリケーションの設定](#) を参照してください。
3. 各マネージドクラスターに VolumeReplicationClass リソースを作成して、レプリケーションスケジュールを設定します (たとえば、5 分ごとにピア間でレプリケートします)。 [Creating VolumeReplicationClass resource](#) を参照してください。
4. ミラーリングが有効になっているブロックボリュームの新しい **imageFeatures** をサポートする各マネージドクラスターにミラーリング Storage Class リソースを作成します。 [ミラーリング Storage Class リソースの作成](#) を参照してください。
5. マネージドクラスターに OpenShift DR Cluster Operator をインストールし、必要なオブジェクトバケット、シークレット、および configmap を作成します。 [Installing OpenShift DR Cluster Operator on Managed clusters](#) を参照してください。

6. ハブクラスターに OpenShift DR Hub Operator をインストールし、必要なオブジェクトバケット、シークレット、および configmap を作成します。[Installing OpenShift DR Hub Operator on Hub cluster](#)を参照してください。
7. マネージドクラスター全体でワークロードをデプロイ、フェイルオーバー、および再配置するために使用されるハブクラスターに DRPolicy リソースを作成します。[Creating Disaster Recovery Policy on Hub cluster](#) を参照してください。

第2章 REGIONAL-DR を有効にするための要件

- 相互にネットワーク接続が可能な 3 つの OpenShift クラスターが必要です。
 - **ハブクラスター**: Kubernetes のための高度なクラスター管理 (RHACM 演算子)、ODF マルチクラスターの Orchestrator と OpenShift DR ハブコントローラーがインストールされています。
 - **プライマリマネージドクラスター**: OpenShift Data Foundation、OpenShift-DR クラスターコントローラー、およびアプリケーションがインストールされています。
 - **セカンダリーマネージドクラスター**: は、OpenShift Data Foundation、OpenShift-DR クラスターコントローラー、およびアプリケーションがインストールされています。
- RHACM Operator および MultiClusterHub をハブクラスターにインストールし、OpenShift 認証情報を使用して RHACM コンソールにログインしている必要があります。手順については、[RHACM インストールガイド](#) を参照してください。
Advanced Cluster Manager コンソール向けに作成されたルートを検索します。

```
$ oc get route multicloud-console -n open-cluster-management -o jsonpath --
template="https://{.spec.host}/multicloud/clusters{"n"}"
```

出力例:

```
https://multicloud-console.apps.perf3.example.com/multicloud/clusters
```

OpenShift 認証情報を使用してログインした後に、ローカルクラスターがインポートされたことが確認できるはずです。

- RHACM コンソールを使用して、**プライマリマネージドクラスター** および **セカンダリーマネージドクラスター** をインポートまたは作成していることを確認します。
Submariner アドオンを使用してマネージド OpenShift クラスターとサービスネットワークを接続するには、マネージドクラスターごとに次のコマンドを実行して、2 つのクラスターに重複しないネットワークがあることを検証する必要があります。

```
$ oc get networks.config.openshift.io cluster -o json | jq .spec
```

cluster1 の出力例 (例: **ocp4perf1**):

```
{
  "clusterNetwork": [
    {
      "cidr": "10.5.0.0/16",
      "hostPrefix": 23
    }
  ],
  "externalIP": {
    "policy": {}
  },
  "networkType": "OpenShiftSDN",
  "serviceNetwork": [
    "10.15.0.0/16"
  ]
}
```

cluster2 の出力例 (例: **ocp4perf2**):

```
{
  "clusterNetwork": [
    {
      "cidr": "10.6.0.0/16",
      "hostPrefix": 23
    }
  ],
  "externalIP": {
    "policy": {}
  },
  "networkType": "OpenShiftSDN",
  "serviceNetwork": [
    "10.16.0.0/16"
  ]
}
```

詳細は、[Submariner add-ons documentation](#) ドキュメントを参照してください。

- マネージドクラスターが **Submariner アドオン** を使用して接続できる必要があります。クラスターおよびサービスネットワークに重複しない範囲が設定されていることを確認した後に、RHACM コンソールおよび **Cluster sets** を使用して、各マネージドクラスターに **Submariner アドオン** をインストールします。手順は、[Submariner のドキュメント](#) を参照してください。
- OpenShift Data Foundation 4.9 以降が各マネージドクラスターにインストールされている必要があります。
 - OpenShift Data Foundation のデプロイメントについては、[インフラストラクチャー固有のデプロイメントガイド](#) (AWS、VMware、ベアメタル、Azure など) を参照してください。
 - 以下のコマンドを使用して、各マネージドクラスターでデプロイメントが正常であることを確認します。

```
$ oc get storagecluster -n openshift-storage ocs-storagecluster -o
jsonpath='{.status.phase}'{"\n"}
```

ステータスが **Primary** マネージドクラスター および **セカンダリー** マネージドクラスターで **Ready** である場合は、マネージドクラスターでミラーリングの有効化に進みます。

第3章 マルチサイトストレージレプリケーションの設定

ミラーリングまたはレプリケーションは、ピアマネージドクラスター内の **CephBlockPool** ごとに有効にされ、その後プール内の特定のイメージのサブセットに設定できます。**rbd-mirror** デーモンは、ローカルピアクラスターからリモートクラスターの同じイメージにイメージの更新を複製します。

この手順では、2 つの OpenShift Data Foundation マネージドクラスター間でミラーリング関係を作成する方法を詳細に説明します。

3.1. マネージドクラスターでの OMAP ジェネレーターおよびボリュームレプリケーションの有効化

プライマリマネージドクラスター および セカンダリーマネージドクラスター で以下の手順を実行し、**csi-rbdplugin-provisioner** Pod で OMAP および Volume-Replication CSI サイドカーコンテナを有効にします。

手順

1. **rook-ceph-operator-config** ConfigMap で **CSI_ENABLE_OMAP_GENERATOR** のために値を **true** に設定するには、次の **patch** コマンドを実行します。

```
$ oc patch cm rook-ceph-operator-config -n openshift-storage --type json --patch ' [{ "op": "add", "path": "/data/CSI_ENABLE_OMAP_GENERATOR", "value": "true" } ]'
```

出力例:

```
configmap/rook-ceph-operator-config patched
```

2. 以下の **patch** コマンドを実行して、**rook-ceph-operator-config** ConfigMap の **CSI_ENABLE_VOLUME_REPLICATION** の値を **true** に設定します。

```
$ oc patch cm rook-ceph-operator-config -n openshift-storage --type json --patch ' [{ "op": "add", "path": "/data/CSI_ENABLE_VOLUME_REPLICATION", "value": "true" } ]'
```

出力例:

```
configmap/rook-ceph-operator-config patched
```

3. **csi-rbdplugin-provisioner** Pod ごとに以下の 2 つの新規 CSI サイドカーコンテナがあることを確認します。

```
$ for I in $(oc get pods -n openshift-storage -l app=csi-rbdplugin-provisioner -o jsonpath={.items[*].spec.containers[*].name}); do echo $I ; done | egrep "csi-omap-generator|volume-replication"
```

出力例:

```
csi-omap-generator
volume-replication
csi-omap-generator
volume-replication
```



注記

冗長性を確保するために2つの **csi-rbdplugin-provisioner** Pod があるため、新規コンテナが繰り返されます。

3.2. OPENSIFT DATA FOUNDATION のマルチクラスターオーケストレーターのインストール

OpenShift Data Foundation のマルチクラスターオーケストレーターは、ハブクラスターの OpenShift Container Platform の OperatorHub からインストールされるコントローラーです。この Multicluster Orchestrator コントローラーと MirrorPeer カスタムリソースは、ブートストラップトークンを作成し、マネージドクラスター間でこのトークンを交換します。

手順

1. ハブクラスターで OperatorHub に移動し、キーワードフィルターを使用して **ODF Multicluster Orchestrator** を検索します。
2. **ODF Multicluster Orchestrator** タイルをクリックします。
3. すべてのデフォルト設定を維持し、Install をクリックします。
Operator リソースは **openshift-operators** にインストールされ、すべての namespace で利用可能です。
4. **ODF Multicluster Orchestrator** が、インストールが成功したことを示す緑色のチェックマークが表示されていることを確認します。

3.3. ハブクラスターでのミラーピアの作成

Mirror Peer は、ピアリング関係を持つマネージドクラスターに関する情報を保持するクラスタースコープのリソースです。

前提条件

- **ODF Multicluster Orchestrator** が ハブクラスター にインストールされていることを確認します。
- Mirror Peer の2つのクラスターのみが必要です。
- 各クラスターに、**ocp4perf1** や **ocp4perf2** などの一意に識別可能なクラスター名があることを確認してください。

手順

1. **ODF Multicluster Orchestrator** をクリックし、Operator の詳細を表示します。
Multicluster Orchestrator が正常にインストールされた後に **View Operator** をクリックできます。
2. Mirror Peer API **Create instance** をクリックしてから **YAML ビュー** を選択します。
3. YAML ビューで Mirror Peer を作成します。
 - a. <cluster1> および <cluster2> を RHACM コンソールのマネージドクラスターの正しい名前に置き換えた後に、以下の YAML をファイル名 **mirror-peer.yaml** にコピーします。

```

apiVersion: multicluster.odf.openshift.io/v1alpha1
kind: MirrorPeer
metadata:
  name: mirrorpeer-<cluster1>-<cluster2>
spec:
  items:
    - clusterName: <cluster1>
      storageClusterRef:
        name: ocs-storagecluster
        namespace: openshift-storage
    - clusterName: <cluster2>
      storageClusterRef:
        name: ocs-storagecluster
        namespace: openshift-storage

```



注記

MirrorPeer はクラスタースコープのリソースであるため、このリソースを作成するために namespace を指定する必要はありません。

- b. 一意の **mirror-peer.yaml** ファイルの内容を YAML ビューにコピーします。元のコンテンツを完全に置き換える必要があります。
 - c. YAML ビュー画面の下部にある **Create** をクリックします。
4. Phase 状態を **ExchangedSecret** として表示できることを確認します。

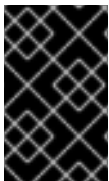


注記

一部のデプロイメントでは、検証用の出力も **ExchangingSecret** にすることができます。これは受け入れ可能な結果です。

3.4. マネージドクラスターでのミラーリングの有効化

ミラーリングを有効にするには、各マネージドクラスターのストレージクラスターのミラーリング設定を有効にする必要があります。これは、CLI および **oc patch** コマンドを使用した手動の手順です。



重要

StorageCluster のミラーリングを有効にした後に、**プライマリーマネージドクラスター** および **セカンダリーマネージドクラスター** で、**oc patch storagecluster** コマンド、それに続く検証コマンドを実行する必要があります。

手順

1. ストレージクラスター名を使用して、クラスターレベルのミラーリングフラグを有効にします。

```

$ oc patch storagecluster $(oc get storagecluster -n openshift-storage -o=jsonpath='{.items[0].metadata.name}') -n openshift-storage --type json --patch '[{"op": "replace", "path": "/spec/mirroring", "value": {"enabled": true}}]'

```

出力例:

```
storagecluster.ocs.openshift.io/ocs-storagecluster patched
```

2. デフォルトの Ceph ブロックプールでミラーリングが有効になっていることを確認します。

```
$ oc get cephblockpool -n openshift-storage -o=jsonpath='{.items[?(@.metadata.ownerReferences[*].kind=="StorageCluster")].spec.mirroring.enabled}'
```

出力例:

```
true
```

3. **rbd-mirror** Pod が稼働していることを確認します。

```
$ oc get pods -o name -l app=rook-ceph-rbd-mirror -n openshift-storage
```

出力例:

```
pod/rook-ceph-rbd-mirror-a-6486c7d875-56v2v
```

4. **daemon** の正常性の状態を検証します。

```
$ oc get cephblockpool ocs-storagecluster-cephblockpool -n openshift-storage -o jsonpath='{.status.mirroringStatus.summary}'
```

出力例:

```
{"daemon_health":"OK","health":"OK","image_health":"OK","states":{}}
```

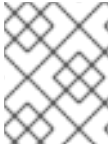


注記

daemon health および health フィールドが **Warning** から **OK** に変わるのに、最長 10 分の時間がかかる可能性があります。ステータスが約 10 分間 **OK** に変わらない場合は、RHACM コンソールを使用して Submariner アドオン接続が **Healthy** 状態のままであることを確認します。

第4章 VOLUMEREPLICATIONCLASS リソースの作成

VolumeReplicationClass を使用して、レプリケートされる各ボリュームの **mirroringMode** を指定すると共に、ローカルクラスターからリモートクラスターにボリュームまたはイメージをレプリケートする頻度 (例:5 分ごと) を指定します。



注記

このリソースは、プライマリーマネージドクラスター および セカンダリーマネージドクラスター で作成する必要があります。

手順

1. 以下の YAML を、**rbd-volumereplicationclass.yaml** のファイル名に保存します。

```
apiVersion: replication.storage.openshift.io/v1alpha1
kind: VolumeReplicationClass
metadata:
  name: odf-rbd-volumereplicationclass
spec:
  provisioner: openshift-storage.rbd.csi.ceph.com
  parameters:
    mirroringMode: snapshot
    schedulingInterval: "5m" # <-- Must be the same as scheduling interval in the DRPolicy
    replication.storage.openshift.io/replication-secret-name: rook-csi-rbd-provisioner
    replication.storage.openshift.io/replication-secret-namespace: openshift-storage
```

2. 両方のマネージドクラスターにファイルを作成します。

```
$ oc create -f rbd-volumereplicationclass.yaml
```

出力例:

```
volumereplicationclass.replication.storage.openshift.io/odf-rbd-volumereplicationclass
created
```

第5章 ミラーリング STORAGECLASS リソースの作成

マネージドクラスター間のイメージレプリケーションを高速化するために必要な追加の **imageFeatures** を備えた新しい **StorageClass** を使用して、**mirroring** を有効にしてブロックボリュームを作成する必要があります。新機能は、**exclusive-lock**、**object-map**、および **fast-diff** です。デフォルトの OpenShift Data Foundation **StorageClass** **ocs-storagecluster-ceph-rbd** には、これらの機能が含まれていません。



注記

このリソースは、プライマリーマネージドクラスター および セカンダリーマネージドクラスター で作成する必要があります。

手順

1. 次の YAML をファイル名 **ocs-storagecluster-ceph-rbdmirror.yaml** に保存します。

```
allowVolumeExpansion: true
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ocs-storagecluster-ceph-rbdmirror
parameters:
  clusterID: openshift-storage
  csi.storage.k8s.io/controller-expand-secret-name: rook-csi-rbd-provisioner
  csi.storage.k8s.io/controller-expand-secret-namespace: openshift-storage
  csi.storage.k8s.io/fstype: ext4
  csi.storage.k8s.io/node-stage-secret-name: rook-csi-rbd-node
  csi.storage.k8s.io/node-stage-secret-namespace: openshift-storage
  csi.storage.k8s.io/provisioner-secret-name: rook-csi-rbd-provisioner
  csi.storage.k8s.io/provisioner-secret-namespace: openshift-storage
  imageFeatures: layering,exclusive-lock,object-map,fast-diff
  imageFormat: "2"
  pool: ocs-storagecluster-cephblockpool
  provisioner: openshift-storage.rbd.csi.ceph.com
  reclaimPolicy: Delete
  volumeBindingMode: Immediate
```

2. 両方のマネージドクラスターにファイルを作成します。

```
$ oc create -f ocs-storagecluster-ceph-rbdmirror.yaml
```

出力例:

```
storageclass.storage.k8s.io/ocs-storagecluster-ceph-rbdmirror created
```

第6章 マネージドクラスターでの OPENSIFT-DR CLUSTER OPERATOR のインストール

手順

1. 各マネージドクラスターで OperatorHub に移動し、**OpenShift-DR Cluster Operator** を絞り込みます。
2. 画面の指示に従って、Operator をプロジェクト **openshift-dr-system** にインストールします。



注記

OpenShift DR Cluster Operator は、プライマリーマネージドクラスターとセカンダリーマネージドクラスターの両方にインストールする必要があります。

3. s3 エンドポイント間の SSL アクセスを設定して、メタデータを安全なトランスポートプロトコルを使用する MCG オブジェクトバケットの代替クラスターと、オブジェクトバケットへのアクセスを確認するための **ハブクラスター** に保存できるようにします。



注記

すべての OpenShift クラスターが環境の署名済み証明書セットを使用してデプロイされる場合、このセクションは省略できます。

- a. **プライマリーマネージドクラスター** の Ingress 証明書を展開し、出力を **primary.crt** に保存します。

```
$ oc get cm default-ingress-cert -n openshift-config-managed -o jsonpath="{['data']['ca-bundle.crt']}" > primary.crt
```

- b. **セカンダリーマネージドクラスター** の Ingress 証明書を抽出し、出力を **secondary.crt** に保存します。

```
$ oc get cm default-ingress-cert -n openshift-config-managed -o jsonpath="{['data']['ca-bundle.crt']}" > secondary.crt
```

- c. **プライマリーマネージドクラスター**、**セカンダリーマネージドクラスター**、および **ハブクラスター** 上のファイル名 **cm-clusters.crt.yaml** を使用して、リモートクラスターの証明書バンドルを保持する新しい **ConfigMap** を作成します。



注記

この例のように、クラスターごとに3つ以上の証明書が存在する可能性があります。

```
apiVersion: v1
data:
  ca-bundle.crt: |
    -----BEGIN CERTIFICATE-----
    <copy contents of cert1 from primary.crt here>
    -----END CERTIFICATE-----
```

```

-----BEGIN CERTIFICATE-----
<copy contents of cert2 from primary.crt here>
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
<copy contents of cert3 primary.crt here>
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
<copy contents of cert1 from secondary.crt here>
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
<copy contents of cert2 from secondary.crt here>
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
<copy contents of cert3 from secondary.crt here>
-----END CERTIFICATE-----
kind: ConfigMap
metadata:
  name: user-ca-bundle
  namespace: openshift-config

```

- d. プライマリーマネージドクラスター、セカンダリーマネージドクラスター、および ハブクラスター で以下のコマンドを実行してファイルを作成します。

```
$ oc create -f cm-clusters-crt.yaml
```

出力例:

```
configmap/user-ca-bundle created
```



重要

ハブクラスター が **DRPolicy** リソースを使用してオブジェクトバケットへのアクセスを確認するには、同じ **ConfigMap**、**cm-clusters-crt.yaml** を ハブクラスター に作成する必要があります。

- e. デフォルトの プロキシ クラスターリソースを変更します。

- i. 以下のコンテンツを新しい YAML ファイル **proxy-ca.yaml** にコピーし、保存します。

```

apiVersion: config.openshift.io/v1
kind: Proxy
metadata:
  name: cluster
spec:
  trustedCA:
    name: user-ca-bundle

```

- ii. この新しいファイルを、プライマリーマネージドクラスター、セカンダリーマネージドクラスター、および ハブクラスター 上のデフォルトのプロキシリソースに適用します。

```
$ oc apply -f proxy-ca.yaml
```

出力例:

```
proxy.config.openshift.io/cluster configured
```

4. マルチクラウドオブジェクトゲートウェイ (MCG) キーと外部 S3 エンドポイントを取得します。

- a. MCG がプライマリーマネージドクラスター および セカンダリーマネージドクラスター にインストールされているかどうか、および Phase が **Ready** かどうかを確認します。

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

出力例:

```
NAME      MGMT-ENDPOINTS      S3-ENDPOINTS      IMAGE
PHASE     AGE
noobaa    ["https://10.70.56.161:30145"] ["https://10.70.56.84:31721"] quay.io/rhceph-
dev/mcg-
core@sha256:c4b8857ee9832e6efc5a8597a08b81730b774b2c12a31a436e0c3fadff48e73
d Ready 27h
```

- b. 以下の YAML ファイルを、ファイル名 **odrbucket.yaml** にコピーします。

```
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: odrbucket
  namespace: openshift-dr-system
spec:
  generateBucketName: "odrbucket"
  storageClassName: openshift-storage.noobaa.io
```

- c. プライマリーマネージドクラスター および セカンダリーマネージドクラスター の両方で、MCG バケット **odrbucket** を作成します。

```
$ oc create -f odrbucket.yaml
```

出力例:

```
objectbucketclaim.objectbucket.io/odrbucket created
```

- d. 以下のコマンドを使用して、各マネージドクラスターの **odrbucket** OBC アクセスキーを **base-64** でエンコードされた値として展開します。

```
$ oc get secret odrbucket -n openshift-dr-system -o
jsonpath='{.data.AWS_ACCESS_KEY_ID}'{"\n"}
```

出力例:

```
cFpIYTZWn1NhemJjbEUyWlpwN1E=
```

- e. 以下のコマンドを使用して、各マネージドクラスターの **odrbucket OBC** シークレットキーを **base-64** でエンコードされた値として展開します。

```
$ oc get secret odrbucket -n openshift-dr-system -o  
jsonpath='{.data.AWS_SECRET_ACCESS_KEY}'{"\n"}
```

出力例:

```
V1hUSnMzZUoxMHRRTXdGMU9jQXRmUIAyMmd5bGwwYjNvMHprZVhtNw==
```

5. マネージドクラスターの S3 シークレットを作成します。
必要な MCG 情報が抽出されたので、**プライマリマネージドクラスター** および **セカンダリーマネージドクラスター** に新しいシークレットが作成されているはずです。これらの新規シークレットは、両方のマネージドクラスターの MCG アクセスとシークレットキーを保存します。



注記

OpenShift DR には、マネージドクラスターからのワークロードの関連クラスターデータを保存するため、およびフェイルオーバーまたは再配置アクション中のワークロード復旧のオーケストレーションを行うために、1つ以上の S3 ストアが必要です。これらの手順は、Multicloud Gateway (MCG) を使用して必要なオブジェクトバケットを作成するために適用できます。OpenShift Data Foundation のインストールにより、MCG がすでにインストールされているはずです。

- a. プライマリマネージドクラスター用の以下の S3 シークレット YAML 形式を、ファイル名 **odr-s3secret-primary.yaml** にコピーします。

```
apiVersion: v1  
data:  
  AWS_ACCESS_KEY_ID: <primary cluster base-64 encoded access key>  
  AWS_SECRET_ACCESS_KEY: <primary cluster base-64 encoded secret access key>  
kind: Secret  
metadata:  
  name: odr-s3secret-primary  
  namespace: openshift-dr-system
```

<primary cluster base-64 encoded access key>および <primary cluster base-64 encoded secret access key> は、先程の手順で取得した実際の値に置き換えます。

- b. **プライマリマネージドクラスター** および **セカンダリーマネージドクラスター** で、このシークレットを作成します。

```
$ oc create -f odr-s3secret-primary.yaml
```

出力例:

```
secret/odr-s3secret-primary created
```

- c. セカンダリーマネージドクラスター用の以下の S3 シークレット YAML 形式を、ファイル名 **odr-s3secret-secondary.yaml** にコピーします。

```
apiVersion: v1
data:
  AWS_ACCESS_KEY_ID: <secondary cluster base-64 encoded access key>
  AWS_SECRET_ACCESS_KEY: <secondary cluster base-64 encoded secret access
  key>
kind: Secret
metadata:
  name: odr-s3secret-secondary
  namespace: openshift-dr-system
```

<secondary cluster base-64 encoded access key>および <secondary cluster base-64 encoded secret access key> を、手順 4 で取得した実際の値に置き換えます。

- d. プライマリーマネージドクラスター および セカンダリーマネージドクラスター で、このシークレットを作成します。

```
$ oc create -f odr-s3secret-secondary.yaml
```

出力例:

```
secret/odr-s3secret-secondary created
```



重要

アクセスキーおよびシークレットキーの値は **base-64 でエンコーディングされている** 必要があります。エンコードされたキーの値は、前の手順で取得されています。

6. 各マネージドクラスターで OpenShift-DR Cluster Operator ConfigMap を設定します。

- a. 以下のコマンドを使用して、外部 S3 エンドポイント **s3CompatibleEndpoint** または各マネージドクラスターで MCG のルートを検索します。

```
$ oc get route s3 -n openshift-storage -o jsonpath --template="https://{.spec.host}"
```

出力例:

```
https://s3-openshift-storage.apps.perf1.example.com
```



重要

一意の **s3CompatibleEndpoint** ルートまたは **s3-openshift-storage.apps.<primary clusterID>.<baseDomain>** および **s3-openshift-storage.apps.<secondary clusterID>.<baseDomain>** は、プライマリーマネージドクラスターとセカンダリーマネージドクラスターの両方で取得する必要があります。

- b. **odrbucket** OBC バケット名を検索します。

```
$ oc get configmap odrbucket -n openshift-dr-system -o
jsonpath='{.data.BUCKET_NAME}'{"\n"}
```

出力例:

```
odrbucket-2f2d44e4-59cb-4577-b303-7219be809dcd
```



重要

一意の s3Bucket 名 odrbucket-**<your value1>** と odrbucket-**<your value2>** は、プライマリーマネージドクラスター と セカンダリーマネージドクラスター の両方で取得する必要があります。

- c. ConfigMap の **ramen-dr-cluster-operator-config** を変更して、新規コンテンツを追加します。

```
$ oc edit configmap ramen-dr-cluster-operator-config -n openshift-dr-system
```

- d. **s3StoreProfiles** で開始する以下の新しいコンテンツを、プライマリーマネージドクラスター および セカンダリーマネージドクラスター の ConfigMap に追加します。

```
[...]
data:
  ramen_manager_config.yaml: |
    apiVersion: ramendr.openshift.io/v1alpha1
    kind: RamenConfig
[...]
  ramenControllerType: "dr-cluster"
  ### Start of new content to be added
  s3StoreProfiles:
    - s3ProfileName: s3-primary
      s3CompatibleEndpoint: https://s3-openshift-storage.apps.<primary clusterID>.
    <baseDomain>
      s3Region: primary
      s3Bucket: odrbucket-<your value1>
      s3SecretRef:
        name: odr-s3secret-primary
        namespace: openshift-dr-system
    - s3ProfileName: s3-secondary
      s3CompatibleEndpoint: https://s3-openshift-storage.apps.<secondary clusterID>.
    <baseDomain>
      s3Region: secondary
      s3Bucket: odrbucket-<your value2>
      s3SecretRef:
        name: odr-s3secret-secondary
        namespace: openshift-dr-system
[...]
```

第7章 ハブクラスターへの OPENSIFT-DR HUB OPERATOR のインストール

前提条件

- アクセスキーおよびシークレットキーの値は **base-64** でエンコードされていることを確認します。キーのエンコードされた値は以前のセクションで取得され、生成される **Secret** はマネージドクラスターにすでに作成されたものと **同じ** です。

手順

1. ハブクラスターで OperatorHub に移動し、**OpenShift-DR Hub Operator** の検索フィルターを使用します。
2. 画面の指示に従って、Operator をプロジェクト **openshift-dr-system** にインストールします。
3. **プライマリマネージドクラスター** の以下の S3 シークレット YAML 形式を使用して、ハブクラスターの S3 シークレットを作成します。

```
apiVersion: v1
data:
  AWS_ACCESS_KEY_ID: <primary cluster base-64 encoded access key>
  AWS_SECRET_ACCESS_KEY: <primary cluster base-64 encoded secret access key>
kind: Secret
metadata:
  name: odr-s3secret-primary
  namespace: openshift-dr-system
```

以下のコマンドを実行して、ハブクラスターにこのシークレットを作成します。

```
$ oc create -f odr-s3secret-primary.yaml
```

出力例:

```
secret/odr-s3secret-primary created
```

4. **セカンダリーマネージドクラスター** の以下の S3 シークレット YAML 形式を使用して、S3 シークレットを作成します。

```
apiVersion: v1
data:
  AWS_ACCESS_KEY_ID: <secondary cluster base-64 encoded access key>
  AWS_SECRET_ACCESS_KEY: <secondary cluster base-64 encoded secret access key>
kind: Secret
metadata:
  name: odr-s3secret-secondary
  namespace: openshift-dr-system
```

以下のコマンドを実行して、ハブクラスターにこのシークレットを作成します。

```
$ oc create -f odr-s3secret-secondary.yaml
```

出力例:

```
secret/odr-s3secret-secondary created
```

5. OpenShift DR Hub Operator の ConfigMap を設定します。
Operator が正常に作成されると、**ramen-hub-operator-config** という新しい ConfigMap が作成されます。

- a. 以下のコマンドを実行してファイルを編集します。

```
$ oc edit configmap ramen-hub-operator-config -n openshift-dr-system
```

- b. **s3StoreProfiles** で開始する以下の新規コンテンツを、ハブクラスターの **ConfigMap** に追加します。

```
[...]
apiVersion: v1
data:
  ramen_manager_config.yaml: |
    apiVersion: ramendr.openshift.io/v1alpha1
    kind: RamenConfig
[...]
  ramenControllerType: "dr-hub"
  ### Start of new content to be added
  s3StoreProfiles:
    - s3ProfileName: s3-primary
      s3CompatibleEndpoint: https://s3-openshift-storage.apps.<primary clusterID>.
    <baseDomain>
      s3Region: primary
      s3Bucket: odrbucket-<your value1>
      s3SecretRef:
        name: odr-s3secret-primary
        namespace: openshift-dr-system
    - s3ProfileName: s3-secondary
      s3CompatibleEndpoint: https://s3-openshift-storage.apps.<secondary clusterID>.
    <baseDomain>
      s3Region: secondary
      s3Bucket: odrbucket-<your value2>
      s3SecretRef:
        name: odr-s3secret-secondary
        namespace: openshift-dr-system
[...]
```



注記

<primary clusterID>、<secondary clusterID>、baseDomain、odrbucket-<your value1>、および odrbucket-<your value2> 変数を、マネージドクラスターの **ramen-cluster-operator-config** ConfigMap に使用されているものと同じ値に置き換えてください。

第8章 ハブクラスターでの障害復旧ポリシーの作成

OpenShift DR は、RHACM ハブクラスターで Disaster Recovery Policy (DRPolicy) リソース (クラスタースコープ) を使用して、マネージドクラスター間でワークロードをデプロイ、フェイルオーバー、および再配置します。

前提条件

- ストレージレベルのレプリケーション用にピアリングされている 2 クラスターのセットがあり、CSI ボリュームレプリケーションが有効になっている必要があります。
- DRPolicy を使用してワークロードの粒度の粗い Recovery Point Objective (RPO) としても機能する、データレプリケーションの実行頻度を決定するスケジューリング間隔が設定されている必要があります。
- ポリシーの各クラスターに、OpenShift-DR Cluster および Hub Operator の ConfigMap で設定される S3 プロファイル名が割り当てられている必要があります。

手順

1. ハブクラスターで、**openshift-dr-system** プロジェクトで Installed Operators に移動し、**OpenShift DR Hub Operator** をクリックします。2 つの利用可能な API (DRPolicy と DRPlacementControl) が表示されるはずです。
2. DRPolicy の **Create instance** をクリックし、**YAML view** をクリックします。
3. **<cluster1>** および **<cluster2>** を RHACM のマネージドクラスターの正しい名前に置き換えてから、以下の YAML を、ファイル名 **drpolicy.yaml** に保存します。

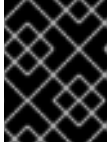
```
apiVersion: ramendr.openshift.io/v1alpha1
kind: DRPolicy
metadata:
  name: odr-policy-5m
spec:
  drClusterSet:
    - name: <cluster1>
      s3ProfileName: s3-primary
    - name: <cluster2>
      s3ProfileName: s3-secondary
  schedulingInterval: 5m
```



注記

DRPolicy はクラスタースコープのリソースであるため、このリソースを作成するために namespace を指定する必要はありません。

4. 一意の **drpolicy.yaml** ファイルの内容を YAML ビューにコピーします。元のコンテンツを完全に置き換える必要があります。
5. YAML ビュー画面の **Create** をクリックします。

**重要**

DRPolicy スケジューリングの間隔は、[VolumeReplicationClass リソースの作成セクション](#)で設定される間隔と一致する必要があります。

6. 以下のコマンドを実行して、**DRPolicy** が正常に作成されていることを確認します。

```
$ oc get drpolicy odr-policy-5m -n openshift-dr-system -o  
jsonpath='{.status.conditions[].reason}'{"\n"}
```

出力例:

```
Succeeded
```

第9章 サンプルアプリケーションの作成

プライマリマネージドクラスターからセカンダリマネージドクラスターへのフェイルオーバーをテストし、また元に戻すには、サンプルアプリケーションが必要です。**busybox** というサンプルアプリケーションを例として使用します。

手順

1. **busybox** サンプルアプリケーションのハブクラスターで **namespace** または **プロジェクト** を作成します。

```
$ oc new-project busybox-sample
```



注記

必要に応じて、**busybox-sample** 以外のプロジェクト名を使用できます。Advanced Cluster Manager コンソールでサンプルアプリケーションをデプロイする場合は、この手順で作成したものと同一プロジェクト名を使用するようにしてください。

2. **DRPlacementControl** リソースを作成します。

DRPlacementControl は、ハブクラスターに OpenShift DR Hub Operator をインストールした後に利用可能な API です。これは、概説としては、DRPolicy の一部であるクラスター間でのデータ可用性に基づいて配置の決定をオーケストレーションする Advanced Cluster Manager PlacementRule リコンサイラーです。

- a. ハブクラスターで、**busybox-sample** プロジェクトで Installed Operators に移動し、**OpenShift DR Hub Operator** をクリックします。2 つの利用可能な API (DRPolicy と DRPlacementControl) が表示されるはずです。
- b. **DRPlacementControl** のインスタンスを作成してから、YAML ビューに移動します。**busybox-sample** プロジェクトが選択されていることを確認します。
- c. **<cluster1>** を Advanced Cluster Manager のマネージドクラスターの正しい名前に置き換えたあと、以下の YAML をファイル名 **busybox-drpc.yaml** にコピーして保存します。

```
apiVersion: ramendr.openshift.io/v1alpha1
kind: DRPlacementControl
metadata:
  labels:
    app: busybox-sample
    name: busybox-drpc
spec:
  drPolicyRef:
    name: odr-policy-5m
  placementRef:
    kind: PlacementRule
    name: busybox-placement
  preferredCluster: <cluster1>
  pvcSelector:
    matchLabels:
      appname: busybox
```

- d. 一意の **busybox-drpc.yaml** ファイルの内容を YAML ビューにコピーします (元のコンテンツを完全に置き換え)。
- e. YAML ビュー画面の **Create** をクリックします。
以下の CLI コマンドを使用してこのリソースを作成することもできます。

```
$ oc create -f busybox-drpc.yaml -n busybox-sample
```

出力例:

```
drplacementcontrol.ramendr.openshift.io/busybox-drpc created
```



重要

このリソースは、**busybox-sample** namespace (または先に作成した namespace) に作成する必要があります。

3. リソーステンプレートのデプロイ先のターゲットクラスターを定義する **Placement Rule** リソースを作成します。配置ルールを使用すると、アプリケーションのマルチクラスターデプロイメントが容易になります。
 - a. 以下の YAML をファイル名 **busybox-placementrule.yaml** にコピーし、保存します。

```
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  labels:
    app: busybox-sample
    name: busybox-placement
spec:
  clusterConditions:
    - status: "True"
      type: ManagedClusterConditionAvailable
  clusterReplicas: 1
  schedulerName: ramen
```

- b. **busybox-sample** アプリケーションの PlacementRule リソースを作成します。

```
$ oc create -f busybox-placementrule.yaml -n busybox-sample
```

出力例:

```
placementrule.apps.open-cluster-management.io/busybox-placement created
```



重要

このリソースは、**busybox-sample** namespace (または先に作成した namespace) に作成する必要があります。

4. RHACM コンソールを使用したサンプルアプリケーションの作成

- a. まだログインしていない場合は、OpenShift 認証情報を使用して RHACM コンソールにログインします。

```
$ oc get route multicloud-console -n open-cluster-management -o jsonpath --
template="https://{.spec.host}/multicloud/applications{\n}"
```

出力例:

```
https://multicloud-console.apps.perf3.example.com/multicloud/applications
```

- b. **Applications** に移動し、**Create application** をクリックします。
- c. 種類は **Subscription** を選択します。
- d. アプリケーションの **Name** (**busybox** など) および **Namespace** (**busybox-sample** など) を入力します。
- e. Repository location for resources セクションで **Repository type Git** を選択します。
- f. サンプルアプリケーションの github **Branch** および **Path** で、Git リポジトリ URL を入力します。リソース **busybox** Pod および PVC が作成されます。
Branch が **main** で、**Path** は **busybox-odr** である <https://github.com/RamenDR/ocm-ramen-samples> として、サンプルアプリケーションリポジトリを使用します。

重要

続行する前に、[Create Mirroring StorageClass resource] セクションで説明されているように、新しい **StorageClass ocs-storagecluster-ceph-rbdmirror** が作成されていることを確認してください。

次のコマンドを使用して作成されていることを確認します。

```
oc get storageclass | grep rbdmirror | awk '{print $1}'
```

出力例:

```
ocs-storagecluster-ceph-rbdmirror
```

- g. **Select clusters to deploy to** セクションまでフォームを下にスクロールして、**Select an existing placement configuration** をクリックします。
- h. ドロップダウンリストから **Existing Placement Rule** (**busybox-placement** など) を選択します。
- i. **Save** をクリックします。
 アプリケーションが作成されると、アプリケーションの詳細ページが表示されます。Resource topology セクションまでスクロールダウンできます。トポロジーの要素とアプリケーションに Green チェックマークが必要です。

注記

詳細な情報を表示するには、トポロジー要素のいずれかをクリックすると、トポロジービューの右側にウィンドウが表示されます。

5. サンプルアプリケーションのデプロイメントおよびレプリケーションを確認します。

busybox アプリケーションが (DRPlacementControl で指定された) preferredCluster にデプロイされたので、デプロイメントを検証できるようになりました。

- a. **busybox** が RHACM によってデプロイされたマネージドクラスターにログオンします。

```
$ oc get pods,pvc -n busybox-sample
```

出力例:

```
NAME          READY STATUS  RESTARTS  AGE
pod/busybox    1/1   Running  0         6m

NAME          STATUS VOLUME          CAPACITY
ACCESS MODES STORAGECLASS      AGE
persistentvolumeclaim/busybox-pvc Bound    pvc-a56c138a-a1a9-4465-927f-af02afbfff37 1Gi      RWO      ocs-storagecluster-ceph-rbd 6m
```

- b. レプリケーションリソースも **busybox** PVC に作成されていることを確認します。

```
$ oc get volumereplication,volumereplicationgroup -n busybox-sample
```

出力例:

```
NAME          AGE VOLUMEREPLICATIONCLASS
PVCNAME       DESIREDSTATE CURRENTSTATE
volumereplication.replication.storage.openshift.io/busybox-pvc 6m odf-rbd-
volumereplicationclass busybox-pvc primary Primary

NAME          AGE
volumereplicationgroup.ramendr.openshift.io/busybox-drpc 6m
```

- c. **Primary managed cluster** と **Secondary マネージドクラスター** の両方で以下のコマンドを実行して、**busybox** ボリュームが代替クラスターに複製されていることを確認します。

```
$ oc get cephblockpool ocs-storagecluster-cephblockpool -n openshift-storage -o
jsonpath='{.status.mirroringStatus.summary}'
```

出力例:

```
{"daemon_health":"OK","health":"OK","image_health":"OK","states":{"replaying":2}}
```



注記

両方のマネージドクラスターの出力はまったく同じで、新しいステータスが **"states":{"replaying":2}** となっているはずです。

9.1. サンプルアプリケーションの削除

RHACM コンソールを使用してサンプルアプリケーション **busybox** を削除できます。

手順

1. RHACM コンソールで、**Applications** に移動します。

2. 削除するサンプルアプリケーションを検索します (例: **busybox**)。
3. 削除するアプリケーションの横にある Action メニュー (⋮) をクリックします。
4. **Delete application** をクリックします。
Delete application を選択すると、アプリケーション関連のリソースも削除すべきかどうかを求める新規画面が表示されます。
5. **Remove application related resources** チェックボックスを選択して、Subscription および PlacementRule を削除します。
6. **Delete** をクリックします。これにより、Primary マネージドクラスター (またはアプリケーションが実行しているクラスター) の busybox アプリケーションが削除されます。
7. RHACM コンソールを使用して削除されたリソースのほかに、**busybox** アプリケーションの削除直後に **DRPlacementControl** も削除する必要があります。
 - a. ハブクラスターの OpenShift Web コンソールにログインし、プロジェクト **busybox-sample** の Installed Operators に移動します。
 - b. **OpenShift DR Hub Operator** をクリックした後、**DRPlacementControl** タブをクリックします。
 - c. 削除する **busybox** アプリケーション DRPlacementControl の横にあるアクションメニュー (⋮) をクリックします。
 - d. **Delete DRPlacementControl** をクリックします。
 - e. **Delete** をクリックします。



注記

このプロセスを使用して、**DRPlacementControl** リソースでアプリケーションを削除できます。**DRPlacementControl** リソースは、CLI を使用してアプリケーション namespace で削除することもできます。

第10章 マネージドクラスター間のアプリケーションのフェイルオーバー

本セクションでは、busybox サンプルアプリケーションをフェイルオーバーする方法を説明します。Regional-DR のフェイルオーバー方法はアプリケーションベースです。この方法で保護される各アプリケーションには、Create Sample Application for DR testing セクションで説明されているように、対応する **DRPlacementControl** リソースとアプリケーション **namespace** で作成された **PlacementRule** リソースが必要です。

手順

1. ハブクラスターで Installed Operators に移動し、**Openshift DR Hub Operator** をクリックします。
2. **DRPlacementControl** タブをクリックします。
3. DRPC **busybox-drpc** をクリックしてから、YAML ビューをクリックします。
4. 以下のスクリーンショットのように、**action** および **failoverCluster** の詳細を追加します。**failoverCluster** はセカンダリーマネージドクラスターの ACM クラスター名である必要があります。

DRPlacementControl add action Failover

Project: busybox-sample ▼

[Installed Operators](#) > [odr-hub-operator.v4.9.0](#) > [DRPlacementControl details](#)**DRPC** busybox-drpc Relocated[Details](#) [YAML](#) [Resources](#) [Events](#)

```

1  apiVersion: ramendr.openshift.io/v1alpha1
2  kind: DRPlacementControl
3  metadata:
4    resourceVersion: '26480556'
5    name: busybox-drpc
6    uid: 1d885540-2c1b-4864-8732-a806e895ceb2
7    creationTimestamp: '2021-10-29T20:18:45Z'
8    generation: 10
9  > managedFields: ...
75 namespace: busybox-sample
76 finalizers:
77   - drpc.ramendr.openshift.io/finalizer
78 labels:
79   app: busybox-sample
80 spec:
81   action: Failover
82   failoverCluster: ocp4perf2
83   drPolicyRef:
84     name: odr-policy-ocp4perf1-ocp4perf2-5m
85   placementRef:

```

Save

Reload

Cancel

5. **Save** をクリックします。
6. アプリケーションの **busybox** がセカンダリーマネージドクラスター (YAML ファイルに指定されるフェイルオーバークラスター **ocp4perf2**) で実行されていることを確認します。

```
$ oc get pods,pvc -n busybox-sample
```

出力例:

NAME	READY	STATUS	RESTARTS	AGE
pod/busybox	1/1	Running	0	35s

NAME	STATUS	VOLUME	CAPACITY	ACCESS
MODES	STORAGECLASS	AGE		
persistentvolumeclaim/busybox-pvc	Bound	pvc-79f2a74d-6e2c-48fb-9ed9-666b74cfa1bb		
5Gi RWO		ocs-storagecluster-ceph-rbd	35s	

7. **busybox** がプライマリーマネージドクラスターで実行していないことを確認します。

```
$ oc get pods,pvc -n busybox-sample
```

出力例:

```
No resources found in busybox-sample namespace.
```

重要

リリースノートの [Known Issues](#) に記載されている既知の Regional-DR の問題に注意してください。

Developer プレビュー機能に関してサポートが必要な場合には、ocs-devpreview@redhat.com メーリングリストに連絡してください。Red Hat Development Team のメンバーが稼働状況とスケジュールに応じて可能な限り迅速に対応します。

第11章 マネージドクラスター間のアプリケーションの再配置

再配置操作はフェイルオーバーと非常に似ています。移動はアプリケーションベースで、DRPlacementControl を使用して再配置をトリガーします。再配置の主な相違点は、**resync** を発行して、セカンダリーマネージドクラスターに保存されている新規アプリケーションデータが即時にあり、ミラーリングスケジュールの間隔を待機せず、プライマリーマネージドクラスターに複製されるという点です。

手順

1. ハブクラスターで Installed Operators に移動し、**Openshift DR Hub Operator** をクリックします。
2. **DRPlacementControl** タブをクリックします。
3. DRPC **busybox-drpc** をクリックしてから、YAML ビューをクリックします。
4. **action** を **Relocate** に変更

DRPlacementControl modify action to Relocate

Project: busybox-sample ▼

[Installed Operators](#) > [odr-hub-operator.v4.9.0](#) > [DRPlacementControl details](#)**DRPC** busybox-drpc FailedOver[Details](#) [YAML](#) [Resources](#) [Events](#)

```

9 > managedFields: ...
75 namespace: busybox-sample
76 finalizers:
77   - drpc.ramendr.openshift.io/finalizer
78 labels:
79   app: busybox-sample
80 spec:
81   action: Relocate
82   drPolicyRef:
83     name: odr-policy-ocp4perf1-ocp4perf2-5m
84   failoverCluster: ocp4perf2
85   placementRef:
86     kind: PlacementRule
87     name: busybox-placement
88     namespace: busybox-sample
89   preferredCluster: ocp4perf1
90   pvcSelector:
91     matchLabels:
92       appname: busybox
93 status:

```

Save

Reload

Cancel

5. **Save** をクリックします。
6. アプリケーションの **busybox** がプライマリーマネージドクラスターで実行しているか、YAML ファイルで指定されたクラスターの **ocp4perf2** をフェイルオーバーしているかどうかを確認します。

```
$ oc get pods,pvc -n busybox-sample
```

出力例:

```
NAME          READY STATUS  RESTARTS  AGE
pod/busybox   1/1   Running  0          60s
```

```
NAME                                STATUS VOLUME          CAPACITY ACCESS
MODES STORAGECLASS                AGE
persistentvolumeclaim/busybox-pvc  Bound  pvc-79f2a74d-6e2c-48fb-9ed9-666b74cfa1bb
5Gi      RWO                ocs-storagecluster-ceph-rbd  61s
```

7. **busybox** がセカンダリーマネージドクラスターで実行しているかどうかを確認します。
busybox アプリケーションは、このマネージドクラスターでは実行しないようにしてください。

```
$ oc get pods,pvc -n busybox-sample
```

出力例:

```
No resources found in busybox-sample namespace.
```

重要

リリースノート [Known Issues](#) に記載されている既知の Regional-DR の問題に注意してください。

Developer プレビュー機能に関してサポートが必要な場合には、ocs-devpreview@redhat.com メーリングリストに連絡してください。Red Hat Development Team のメンバーが稼働状況とスケジュールに応じて可能な限り迅速に対応します。