



Red Hat OpenShift Data Foundation 4.10

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

開発者プレビュー:Regional-DR 機能を使用して OpenShift Data Foundation をセットアップする手順。このソリューションは開発者向けのプレビュー機能であり、実稼働環境で実行することを目的としたものではありません。

Red Hat OpenShift Data Foundation 4.10 Regional-DR 向け Advanced Cluster Management と OpenShift Data Foundation の設定

開発者プレビュー:Regional-DR 機能を使用して OpenShift Data Foundation をセットアップする手順。このソリューションは開発者向けのプレビュー機能であり、実稼働環境で実行することを目的としたものではありません。

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 マネージドクラスターへの OPENSIFT DATA FOUNDATION のインストール	10
第4章 ハブクラスターへの OPENSIFT-DR HUB OPERATOR のインストール	11
第5章 マルチサイトストレージレプリケーションの設定	12
5.1. OPENSIFT DATA FOUNDATION のマルチクラスターオーケストレーターの実インストール	12
5.2. ハブクラスターでのミラーピアの作成	12
5.3. マネージドクラスターでの CEPH ミラーリングの検証	13
5.4. オブジェクトバケットと S3STOREPROFILES の検証	14
第6章 ミラーリング STORAGECLASS リソースの作成	17
第7章 S3 エンドポイント間の SSL アクセスの設定	18
第8章 ハブクラスターでの障害復旧ポリシーの作成	20
第9章 OPENSIFT DR クラスターオペレーターの自動インストールを有効にする	22
第10章 マネージドクラスターへの S3SECRETS の自動転送の有効化	23
第11章 サンプルアプリケーションの作成	24
11.1. サンプルアプリケーションの削除	27
第12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	29
第13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	32

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 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、これらはリージョン障害からの保護に似ています。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

Red Hat Advanced Cluster Management 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が含まれます。
- マネージドクラスター: マネージドクラスターには、マネージドクラスターで実行されるコンポーネントが含まれます。

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 ブロックプール間でイメージを自動的にミラーリング
- Persistent Volume Claim (PVC) 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
- アプリケーションの状態をピアクラスターに移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する

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、次のアクションも実行します。
 - 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。
 - マネージドクラスターでデフォルトの **CephBlockPool** のミラーリングを有効にします。
 - **PVC** および **PV** メタデータの各マネージドクラスターで Multicloud Object Gateway (MCG) を使用してオブジェクトバケットを作成します。
 - **openshift-dr-system** プロジェクトの **ハブクラスター** でバケットアクセス用のキーを持つ新しいオブジェクトバケットごとに **シークレット** を作成します。
 - 各 **SchedulingIntervals** (例: 5m、15m、30m) に対して、**プライマリ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** で **VolumeReplicationClass** を作成します。
 - ハブクラスター上の **ramen-hub-operator-config** ConfigMap を変更し、s3StoreProfiles エントリーを追加します。
- **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 4.10 および RHACM の最新バージョンを使用して 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 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. RHACM オペレーターのインストール、OpenShift Container Platform の RHACM ハブおよびネットワーク設定への作成またはインポートを含む 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. OpenShift Data Foundation 4.10 をプライマリ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します。[マネージドクラスターへの OpenShift Data Foundation のインストール](#) を参照してください。
3. Openshift DR Hub Operator をハブクラスターにインストールします。[Installing OpenShift DR Hub Operator on Hub cluster](#) を参照してください。
4. 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。
5. 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。
6. 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。



注記

複数のポリシーが存在する場合があります。

7. OpenShift DR Cluster オペレーターの自動インストールと、マネージドクラスターでの S3 シークレットの自動転送を有効にします。手順は、[OpenShift DR クラスターオペレーターの自動インストールの有効化](#) および [マネージドクラスターでの S3 シークレットの自動転送の有効化](#) を参照してください。
8. フェイルオーバーと再配置のテストをテストするために、RHACM コンソールを使用してサンプルアプリケーションを作成します。手順については、[サンプルアプリケーションの作成、アプリケーションフェイルオーバー](#)、およびマネージドクラスター間での [アプリケーションの再配置](#) 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

Red Hat OpenShift Data Foundation でサポートされる障害復旧機能では、障害復旧ソリューションを正常に実装するために以下の前提条件がすべて必要になります。

- サブスクリプションの要件
 - 有効な Red Hat OpenShift Data Foundation Advanced エンタイトルメント
 - 有効な Red Hat Advanced Cluster Management for Kubernetes サブスクリプション

OpenShift Data Foundation のサブスクリプションがどのように機能するかを知るには、[OpenShift Data Foundation subscriptions に関するナレッジベースの記事](#) を参照してください。

- 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。
 - **プライマリー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 オペレーターと Multicloudhub 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。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。
 - OpenShift 認証情報を使用して RHACM コンソールにログインします。
 - 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="{.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** をインポートまたは作成していることを確認します。
- マネージドクラスターには、重複しないネットワークが必要です。
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** を使用して接続する必要があります。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。

第3章 マネージドクラスターへの OPENSIFT DATA FOUNDATION のインストール

手順

1. 各マネージドクラスターに OpenShift Data Foundation バージョン 4.10 をインストールします。
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。
2. 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}'
```

および Multicloud Object Gateway (MCG) キーの場合:

```
$ oc get noobaa -n openshift-storage noobaa -o jsonpath='{.status.phase}'
```

両クエリーのステータスが **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** 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4章 ハブクラスターへの OPENSIFT-DR HUB OPERATOR のインストール

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. 次のコマンドを使用して、オペレーター Pod が **Running** 状態にあることを確認します。

```
$ oc get pods -n openshift-dr-system
```

出力例:

NAME	READY	STATUS	RESTARTS	AGE
ramen-hub-operator-898c5989b-96k65	2/2	Running	0	4m14s

第5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。

5.1. 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** が正常にインストールされたことを確認します。
 - a. **View Operator** を選択できるようにして、インストールが成功したことを検証します。
 - b. オペレーター Pod が **Running** 状態にあることを確認します。

```
$ oc get pods -n openshift-operators
```

出力例:

NAME	READY	STATUS	RESTARTS	AGE
odfmo-controller-manager-65946fb99b-779v8	1/1	Running	0	5m3s

5.2. 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. <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** にコピーして保存します。

```
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
  manageS3: true
  schedulingIntervals:
    - 5m
    - 15m
```



注記

schedulingIntervals の間隔時間 (たとえば 5m) は、永続ボリュームを複製するための目的の間隔を設定するために使用されます。これらの値は、重要なアプリケーションの Recovery Point Objective (RPO) にマッピングできます。アプリケーションの要件に合うように **schedulingIntervals** の値を変更します。最小値は **1m** で、デフォルトは **5m** です。

4. 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。
5. YAML ビュー画面の下部にある **Create** をクリックします。
6. 続行する前に、Phase 状態を **ExchangedSecret** として表示できることを確認します。

5.3. マネージドクラスターでの CEPH ミラーリングの検証

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 で次の検証を実行して、Ceph ミラーリングがアクティブであることを確認します。

1. デフォルトの **Ceph block pool** で **mirroring** 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}'{"\n"}
```

出力例:

```
true
```

2. **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
```

3. **daemon** ヘルスのステータスをチェックして、問題がないことを確認します。

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

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。10 分後にステータスが OK にならない場合は、Advanced Cluster Manager コンソールを使用して、**submariner add-on** の接続がまだ正常な状態にあることを確認します。

4. **VolumeReplicationClass** が、MirrorPeer にリストされている **schedulingIntervals** (5m、15m など) ごとに、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に作成されていることを確認します。

```
$ oc get volumereplicationclass
```

出力例:

NAME	PROVISIONER
rbd-volumereplicationclass-1625360775	openshift-storage.rbd.csi.ceph.com
rbd-volumereplicationclass-539797778	openshift-storage.rbd.csi.ceph.com



注記

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。

5.4. オブジェクトバケットと S3STOREPROFILES の検証

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 で次の検証を実行して、Ceph ミラーリングがアクティブであることを確認します。

手順

1. プライマリーマネージドクラスター と **openshift-storage** 名前空間の セカンダリーマネージドクラスター に新しい オブジェクトバケットクレーム と対応する オブジェクトバケット があることを確認します。

```
$ oc get obc,ob -n openshift-storage
```

出力例:

```
NAME                                STORAGE-CLASS          PHASE  AGE
objectbucketclaim.objectbucket.io/odrbucket-21eb5332f6b6  openshift-storage.noobaa.io
Bound  13m
```

```
NAME                                STORAGE-CLASS          CLAIM-
NAMESPACE CLAIM-NAME RECLAIM-POLICY PHASE  AGE
objectbucket.objectbucket.io/obc-openshift-storage-odrbucket-21eb5332f6b6  openshift-
storage.noobaa.io                                Delete  Bound  13m
```

2. 新しいオブジェクトバケットクラスごとにアクセスキーとシークレットキーを含む **ハブクラスター** の **openshift-dr-system** 名前空間に 2 つの新しい **シークレット** があることを確認します。

```
$ oc get secrets -n openshift-dr-system | grep Opaque
```

出力例:


```
8b3fb9ed90f66808d988c7edfa76eba35647092 Opaque 2    16m
af5f82f21f8f77faf3de2553e223b535002e480 Opaque 2    16m
```

3. OBC とシークレットは、新しく作成された **s3StoreProfiles** セクションのハブクラスターの ConfigMap **ramen-hub-operator-config** に書き込まれます。

```
$ oc get cm ramen-hub-operator-config -n openshift-dr-system -o yaml | grep -A 14 s3StoreProfiles
```

出力例:

```
s3StoreProfiles:
- s3Bucket: odrbucket-21eb5332f6b6
  s3CompatibleEndpoint: https://s3-openshift-storage.apps.perf2.example.com
  s3ProfileName: s3profile-ocp4perf2-ocs-storagecluster
  s3Region: noobaa
  s3SecretRef:
    name: 8b3fb9ed90f66808d988c7edfa76eba35647092
    namespace: openshift-dr-system
- s3Bucket: odrbucket-21eb5332f6b6
  s3CompatibleEndpoint: https://s3-openshift-storage.apps.perf1.example.com
  s3ProfileName: s3profile-ocp4perf1-ocs-storagecluster
  s3Region: noobaa
```



```
s3SecretRef:  
  name: af5f82f21f8f77faf3de2553e223b535002e480  
  namespace: openshift-dr-system
```



注記

s3ProfileName の名前を記録します。これらは DRPolicy リソースで使用されます。

第6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7章 S3 エンドポイント間の SSL アクセスの設定

s3 エンドポイント 間のネットワーク (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 object bucket** 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の有効な証明書セットを使用してデプロイされる場合は、このセクションを省略できます。

手順

1. 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
```

2. 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
```

3. 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-cert.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。また、以前作成した **primary.crt** ファイルおよび **secondary.crt** ファイルから、証明書の内容をコピーして貼り付けた後に、証明書の内容が正しくインデントされていることを確認します。

```
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
```

4. 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に ConfigMap 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もあります。

5. 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パッチを適用します。

```
$ oc patch proxy cluster --type=merge --patch='{"spec":{"trustedCA":{"name":"user-ca-bundle"}}}'
```

出力例:

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


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 ACM 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。**<string_value_1>** および **<string_value_2>** は、一意である限り任意の値に置き換えます (例: east と west)。**SchedulingInterval** は、以前に **MirrorPeer** で設定された値の 1 つである必要があります (例: 5m)。

```
apiVersion: ramendr.openshift.io/v1alpha1
kind: DRPolicy
metadata:
  name: odr-policy-5m
spec:
  drClusterSet:
    - name: <cluster1>
      region: <string_value_1>
      s3ProfileName: s3profile-<cluster1>-ocs-storagecluster
    - name: <cluster2>
      region: <string_value_2>
      s3ProfileName: s3profile-<cluster2>-ocs-storagecluster
  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 の **schedulingInterval** は、MirrorPeer リソースで設定されている値の1つ (5m など) と一致する**必要があります**。MirrorPeer で設定されたボリュームレプリケーションに他の **schedulingIntervals** のいずれかを使用するには、新しい値 (つまり、15m) で追加の DRPolicy リソースを作成する必要があります。DRPolicy **name**を一意で、レプリケーション間隔の識別に役立つように変更してください (例: odr-policy-15m)。

6. 作成された DRPolicy リソースごとに **ハブクラスター** で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。この例は、**odr-policy-5m** の場合です。

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

出力例:

```
Succeeded
```

第9章 OPENSIFT DR クラスターオペレーターの自動インストールを有効にする

DRPolicy が正常に作成されると、**OpenShift DR クラスターオペレーター** を **openshift-dr-system** 名前空間のプライー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できます。

手順

1. ハブクラスターで ConfigMap **ramen-hub-operator-config** を編集して、次のように **deploymentAutomationEnabled=true** を追加します。

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

```
apiVersion: v1
data:
  ramen_manager_config.yaml: |
    apiVersion: ramendr.openshift.io/v1alpha1
    drClusterOperator:
      deploymentAutomationEnabled: true ## <-- Add to enable installation of ODR Cluster
operator on managed clusters
  catalogSourceName: redhat-operators
  catalogSourceNamespaceName: openshift-marketplace
  channelName: stable-4.10
  clusterServiceVersionName: odr-cluster-operator.v4.10.0
  namespaceName: openshift-dr-system
  packageName: odr-cluster-operator
[...]
```

2. プライーマリーマネージドクラスターでインストールが成功したことを確認し、セカンダリーマネージドクラスターで次のコマンドを実行します。

```
$ oc get csv,pod -n openshift-dr-system
```

出力例:

NAME	DISPLAY	VERSION
REPLACES PHASE		
clusterserviceversion.operators.coreos.com/odr-cluster-operator.v4.10.0		Openshift DR
Cluster Operator 4.10.0	Succeeded	

NAME	READY	STATUS	RESTARTS	AGE
pod/ramen-dr-cluster-operator-5564f9d669-f6lbc	2/2	Running	0	5m32s

各マネージドクラスターの Operator Hub に移動して、**OpenShift DR Cluster Operator** がインストールされているかどうかを確認することもできます。

第10章 マネージドクラスターへの S3SECRETS の自動転送の有効化

この手順に従って、必要な OpenShift DR クラスターコンポーネントへの s3Secrets の自動転送を有効にします。OpenShift DR 設定マップ内の s3Profiles にアクセスするために必要な s3Secrets を使用して、OpenShift DR クラスターの名前空間を更新します。

手順

1. 次のように、ハブクラスターの ConfigMap **ramen-hub-operator-config** 設定を編集して、**s3SecretDistributionEnabled=true** を追加します。

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

```
apiVersion: v1
data:
  ramen_manager_config.yaml: |
    apiVersion: ramendr.openshift.io/v1alpha1
    drClusterOperator:
      deploymentAutomationEnabled: true
      s3SecretDistributionEnabled: true ## <-- Add to enable automatic transfer of s3secrets
      catalogSourceName: redhat-operators
      catalogSourceNamespaceName: openshift-marketplace
      channelName: stable-4.10
      clusterServiceVersionName: odr-cluster-operator.v4.10.0
      namespaceName: openshift-dr-system
      packageName: odr-cluster-operator
  [...]
```

2. 両方のマネージドクラスターでこのコマンドを実行して、シークレットの転送が成功したことを確認します。

```
$ oc get secrets -n openshift-dr-system | grep Opaque
```

出力例:

```
8b3fb9ed90f66808d988c7edfa76eba35647092 Opaque      2      11m
af5f82f21f8f77faf3de2553e223b535002e480 Opaque      2      11m
```

第11章 サンプルアプリケーションの作成

プライマリーマネージドクラスターからセカンダリーマネージドクラスターへの **failover** およびフェイルバックをテストするには、単純な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** に保存します。目的のレプリケーション間隔を持つ **DRPolicy** の **drPolicyRef** 名を変更します。

```
apiVersion: ramendr.openshift.io/v1alpha1
kind: DRPlacementControl
metadata:
  labels:
    app: busybox-sample
    name: busybox-drpc
spec:
  drPolicyRef:
    name: odr-policy-5m    ## <-- Modify to specify desired DRPolicy and RPO
  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。
次に表示される画面で下部までスクロールします。アプリケーショントポロジーのチェックマークがすべて緑であることが確認できるはずです。

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。

11.1. 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。



注記

サンプルアプリケーションを削除する手順は、フェイルオーバーとフォールバック (再配置) のテストが完了し、アプリケーションを RHACM とマネージドクラスターから削除する準備ができるまで実行しないでください。

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12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.v4.10.0](#) > [DRPlacementControl details](#)**DRPC** **busybox-drpc** Deployed[Details](#) [YAML](#) [Resources](#) [Events](#)

```

2  kind: DRPlacementControl
3  metadata:
4    resourceVersion: '2773813'
5    name: busybox-drpc
6    uid: d18afdba-97fb-4072-8e23-6acd0c07c356
7    creationTimestamp: '2022-03-02T01:10:33Z'
8    generation: 3
9  > managedFields: ""
83 namespace: busybox-sample
84 finalizers:
85   - drpc.ramendr.openshift.io/finalizer
86 labels:
87   app: busybox-sample
88   cluster.open-cluster-management.io/backup: resource
89 spec:
90   drPolicyRef:
91     name: odr-policy-5m
92   action: Failover
93   failoverCluster: ocp4perf2
94   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。

第13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.v4.10.0](#) > [DRPlacementControl details](#)**DRPC** busybox-drpc FailedOver[Details](#) [YAML](#) [Resources](#) [Events](#)

```

7   creationTimestamp: "2022-03-02T01:10:33Z"
8   generation: 4
9   > managedFields: ...
84  namespace: busybox-sample
85  finalizers:
86    - drpc.ramendr.openshift.io/finalizer
87  labels:
88    app: busybox-sample
89    cluster.open-cluster-management.io/backup: resource
90  spec:
91    action: Relocate
92    drPolicyRef:
93      name: odr-policy-5m
94    failoverCluster: ocp4perf2
95    placementRef:
96      kind: PlacementRule
97      name: busybox-placement
98      namespace: busybox-sample
99    preferredCluster: ocp4perf1
100   pvcSelector:
101     matchLabels:

```

Save

Reload

Cancel

5. **Save** をクリックします。
6. アプリケーション **busybox** がプライマリーマネージドクラスターで実行されているかどうかを確認します。フェイルオーバー操作の前にアプリケーションが実行していた YAML ファイルで指定されている **preferredCluster ocp4perf1** へのフェイルバックが行われます。

```
$ 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。