



Red Hat OpenShift Data Foundation 4.10

고급 클러스터 관리를 사용하여 **region-DR**에 대한 OpenShift Data Foundation 구성

개발자 프리뷰: Regional-DR 기능을 사용한 OpenShift Data Foundation 설정 지침. 이 솔루션은 개발자 프리뷰 기능이며 프로덕션 환경에서 실행할 수 없습니다.

Red Hat OpenShift Data Foundation 4.10 고급 클러스터 관리를 사용하여 region-DR에 대한 OpenShift Data Foundation 구성

개발자 프리뷰: Regional-DR 기능을 사용한 OpenShift Data Foundation 설정 지침. 이 솔루션은 개발자 프리뷰 기능이며 프로덕션 환경에서 실행할 수 없습니다.

법적 공지

Copyright © 2023 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

초록

이 솔루션 가이드에서는 고급 클러스터 관리를 통해 재해 복구를 위해 OpenShift Data Foundation을 배포하여 고가용성 스토리지 인프라를 달성하는 데 필요한 단계를 자세히 설명합니다. 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.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체	3
RED HAT 문서에 관한 피드백 제공	4
1장. REGION-DR 소개	5
1.1. REGION-DR 솔루션의 구성 요소	5
1.2. REGIONAL-DR 배포 워크플로	6
2장. REGION-DR 활성화 요구 사항	8
3장. 관리형 클러스터에 OPENSIFT DATA FOUNDATION 설치	10
4장. HUB 클러스터에 OPENSIFT DR HUB OPERATOR 설치	11
5장. 다중 사이트 스토리지 복제 구성	12
5.1. OPENSIFT DATA FOUNDATION 다중 클러스터 오케스트레이션 설치	12
5.2. HUB 클러스터에서 미러 피어 생성	12
5.3. 관리형 클러스터에서 CEPH 미러링 검증	13
5.4. 오브젝트 버킷 및 S3STOREPROFILE 검증	14
6장. 미러링 STORAGECLASS 리소스 생성	16
7장. S3 끝점 간 SSL 액세스 구성	17
8장. HUB 클러스터에서 재해 복구 정책 생성	19
9장. OPENSIFT DR 클러스터 OPERATOR 자동 설치 활성화	21
10장. S3SECRETS를 관리 클러스터로 자동 전송 활성화	22
11장. 샘플 애플리케이션 생성	23
11.1. 샘플 애플리케이션 삭제	26
12장. 관리되는 클러스터 간 애플리케이션 페일오버	28
13장. 관리 클러스터 간 애플리케이션 재배포	31

보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 [CTO Chris Wright의 메시지](#)를 참조하십시오.

RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. Red Hat이 이를 개선하는 방법을 알려 주십시오. 피드백을 제공하려면 다음을 수행하십시오.

- 특정 문구에 대한 간단한 의견 작성 방법은 다음과 같습니다.
 1. 문서가 *Multi-page HTML* 형식으로 표시되는지 확인합니다. 또한 문서 오른쪽 상단에 **피드백** 버튼이 있는지 확인합니다.
 2. 마우스 커서를 사용하여 주석 처리하려는 텍스트 부분을 강조 표시합니다.
 3. 강조 표시된 텍스트 아래에 표시되는 **피드백 추가** 팝업을 클릭합니다.
 4. 표시된 지침을 따릅니다.
- 보다 상세하게 피드백을 제출하려면 다음과 같이 Bugzilla 티켓을 생성하십시오.
 1. [Bugzilla](#) 웹 사이트로 이동하십시오.
 2. 구성 요소 섹션에서 **설명서** 를 선택합니다.
 3. **Description** 필드에 문서 개선을 위한 제안 사항을 기입하십시오. 관련된 문서의 해당 부분 링크를 알려주십시오.
 4. **Submit Bug**를 클릭하십시오.

1장. REGION-DR 소개

재해 복구 기능은 자연 또는 사람이 생성한 재해로부터 비즈니스 크리티컬 애플리케이션을 복구하고 지속하는 기능입니다. 이는 주요 부정적 이벤트 중에 비즈니스 운영의 연속성을 유지하기 위해 설계된 모든 주요 조직의 전반적인 비즈니스 연속성 전략입니다.

regional-DR 기능은 지리적으로 분산된 사이트 간에 볼륨 영구 데이터 및 메타데이터 복제를 제공합니다. 퍼블릭 클라우드에서 이는 지역 실패로부터 보호하는 것과 유사합니다. regional-DR은 지리적으로 사용할 수 없는 동안에도 비즈니스 연속성을 보장하고 일부 데이터 손실을 예측 가능한 양으로 허용합니다. 이는 일반적으로 RPO(Recovery Point Objective) 및 복구 시간 목표(RTO)로 표시됩니다.

- RPO는 영구 데이터의 백업 또는 스냅샷을 생성하는 빈도를 측정한 것입니다. 실제로 RPO는 손실 될 데이터 양을 나타내거나 중단 후 다시 입력해야 함을 나타냅니다.
- RTO는 기업이 허용할 수 있는 다운타임의 양입니다. RTO는 "비즈니스 중단 통지를 받은 후 시스템을 복구하는 데 얼마나 오래 걸릴 수 있습니까?"라는 질문에 대답합니다.

이 가이드의 목적은 재해 복구를 활성화하기 위해 인프라를 구성하는 데 필요한 단계와 명령을 자세히 설명하는 것입니다.

1.1. REGION-DR 솔루션의 구성 요소

regional-DR은 OpenShift Container Platform 클러스터에서 애플리케이션 및 데이터 이동성을 제공하기 위해 RHACM(Red Hat Advanced Cluster Management for Kubernetes) 및 OpenShift Data Foundation 구성 요소로 구성됩니다.

Red Hat Advanced Cluster Security for Kubernetes

Red Hat Advanced Cluster Management는 여러 클러스터 및 애플리케이션 라이프사이클을 관리할 수 있는 기능을 제공합니다. 따라서 다중 클러스터 환경에서 컨트롤 플레인 역할을 합니다.

RHACM은 다음 두 부분으로 나뉩니다.

- RHACM Hub: 다중 클러스터 컨트롤 플레인에서 실행되는 구성 요소가 포함되어 있습니다.
- 관리형 클러스터: 관리되는 클러스터에서 실행되는 구성 요소가 포함되어 있습니다.

이 제품에 대한 자세한 내용은 RHACM [설명서](#) 및 RHACM ["Managing Applications" 설명서를 참조하십시오.](#)

OpenShift Data Foundation

OpenShift Data Foundation은 OpenShift Container Platform 클러스터에서 상태 저장 애플리케이션의 스토리지를 프로비저닝하고 관리하는 기능을 제공합니다.

OpenShift Data Foundation은 Ceph에서 OpenShift Data Foundation 구성 요소 스택의 Rook에서 라이프 사이클을 관리하는 스토리지 프로바이더로 지원합니다. Ceph-CSI는 상태 저장 애플리케이션의 영구 볼륨을 프로비저닝 및 관리합니다.

OpenShift Data Foundation 스택이 다음과 같은 기능으로 향상되었습니다.

- 미러링을 위해 풀 활성화
- RBD 블록 풀의 이미지 자동 미러링
- PVC(영구 볼륨 클레임) 미러링에 사용할 csi-addons 제공

OpenShift DR

OpenShift DR은 RHACM을 사용하여 배포 및 관리되는 피어 OpenShift 클러스터 세트에 대한 상태 저장 애플리케이션의 재해 복구 오케스트레이터이며 영구 불륨에서 애플리케이션 상태의 라이프 사이클을 오케스트레이션하는 클라우드 네이티브 인터페이스를 제공합니다. 여기에는 다음이 포함됩니다.

- OpenShift 클러스터에서 애플리케이션 상태 관계 보호
- 피어 클러스터로 애플리케이션 상태를 장애 조치
- 이전에 배포된 클러스터에 애플리케이션 상태 재배포

OpenShift DR은 다음 세 가지 구성 요소로 나뉩니다.

- **ODF 멀티클러스터 오케스트레이터:** 다중 클러스터 컨트롤 플레인(RHACM Hub)에 설치되어 있으며 다음 작업도 수행합니다.
 - 부트스트랩 토큰을 생성하고 이 토큰을 관리형 클러스터 간에 교환합니다.
 - 관리형 클러스터에서 기본 **CephBlockPool**에 대한 미러링을 활성화합니다.
 - **PVC** 및 **PV** 메타데이터의 각 관리형 클러스터에서 MCG(Multicloud Object Gateway)를 사용하여 오브젝트 버킷을 생성합니다.
 - **openshift-dr-system** 프로젝트의 **Hub** 클러스터에서 버킷 액세스 키가 있는 새 오브젝트 버킷마다 보안을 생성합니다.
 - 기본 관리 클러스터에서 **VolumeReplicationClass**를 만들고 각 스케줄링 간격(예: 5m, 15m, 30m)에 대한 보조 관리 클러스터를 생성합니다.
 - **Hub** 클러스터에서 **ramen-hub-operator-config** ConfigMap을 수정하고 **s3StoreProfiles** 항목을 추가합니다.
- **OpenShift DR Hub Operator:** hub 클러스터에 설치되어 애플리케이션의 장애 조치 및 재배포를 관리합니다.
- **OpenShift DR Cluster Operator:** 각 관리 클러스터에 설치되어 애플리케이션의 모든 PVC의 라이프 사이클을 관리합니다.

1.2. REGIONAL-DR 배포 워크플로

이 섹션에서는 두 개의 개별 OpenShift Container Platform 클러스터에서 OpenShift Data Foundation 버전 4.10 및 RHACM 최신 버전을 사용하여 Regional-DR 기능을 구성하고 배포하는 데 필요한 단계를 간략하게 설명합니다. 두 개의 관리형 클러스터 외에도 고급 클러스터 관리를 배포하려면 세 번째 OpenShift Container Platform 클러스터가 필요합니다.

인프라를 구성하려면 지정된 순서대로 다음 단계를 수행하십시오.

1. RHACM Operator 설치, 생성 또는 가져오기가 포함된 KnativeServing-DR 요구 사항을 RHACM 허브 및 네트워크 구성으로 충족해야 합니다. [Regional-DR 활성화에 대한 요구 사항을 참조하십시오.](#)
2. 기본 및 보조 관리 클러스터에 OpenShift Data Foundation 4.10을 설치합니다. [관리형 클러스터에 OpenShift Data Foundation 설치](#)를 참조하십시오.
3. Hub 클러스터에 OpenShift DR Hub Operator를 설치합니다. [Hub 클러스터에 OpenShift DR Hub Operator 설치](#)를 참조하십시오.

4. 두 OpenShift Data Foundation 관리 클러스터 간에 미러링 관계를 생성하여 다중 사이트 스토리지 복제를 구성합니다. [다중 사이트 스토리지 복제 구성](#)을 참조하십시오.
5. 미러링이 활성화된 블록 볼륨에 대해 새 **imageFeatures** 를 지원하는 각 관리형 클러스터에 미러링 StorageClass 리소스를 만듭니다. [미러링 StorageClass 리소스 생성](#)을 참조하십시오.
6. 관리 클러스터 전체에서 워크로드를 배포, 페일오버 및 재배치하는 데 사용되는 허브 클러스터에 DRPolicy 리소스를 생성합니다. [Hub 클러스터에서 재해 복구 정책 생성](#)을 참조하십시오.



참고

하나의 정책 이상의 것이 있을 수 있습니다.

7. OpenShift DR Cluster Operator 자동 설치를 활성화하고 관리 클러스터에서 S3 시크릿을 자동으로 전송할 수 있습니다. 자세한 내용은 [관리 클러스터에서 S3 시크릿 자동 전송 활성화 및 OpenShift DR 클러스터 Operator 자동 설치 활성화](#)를 참조하십시오.
8. 장애 조치 및 재배치 테스트를 테스트하기 위해 RHACM 콘솔을 사용하여 샘플 애플리케이션을 생성합니다. 자세한 내용은 [관리형 클러스터 간에 샘플 애플리케이션 생성, 애플리케이션 장애 조치 및 애플리케이션 재배치](#)를 참조하십시오.

2장. REGION-DR 활성화 요구 사항

Red Hat OpenShift Data Foundation에서 지원하는 재해 복구 기능을 사용하려면 재해 복구 솔루션을 성공적으로 구현하기 위해 다음과 같은 사전 요구 사항이 모두 필요합니다.

- 서브스크립션 요구 사항
 - 유효한 Red Hat OpenShift Data Foundation Advanced Entitlement
 - 유효한 Red Hat Advanced Cluster Management for Kubernetes 서브스크립션

OpenShift Data Foundation의 서브스크립션이 작동하는 방법을 알아보려면 [OpenShift Data Foundation 서브스크립션에 대한 기술 자료 문서](#)를 참조하십시오.

- 네트워크로 연결할 수 있는 세 개의 OpenShift 클러스터가 있어야 합니다.
 - RHACM(Advanced Cluster Management for Kubernetes), ODF Multicluster Orchestrator 및 OpenShift DR Hub 컨트롤러가 설치된 Hub 클러스터.
 - OpenShift Data Foundation, OpenShift DR Cluster 컨트롤러 및 애플리케이션이 설치된 주요 관리형 클러스터.
 - OpenShift Data Foundation, OpenShift DR Cluster 컨트롤러 및 애플리케이션이 설치된 보조 관리형 클러스터.
- RHACM Operator 및 MultiClusterResourceOverride가 Hub 클러스터에 설치되어 있는지 확인합니다. 자세한 내용은 [RHACM 설치 가이드](#)를 참조하십시오.
 - OpenShift 인증 정보를 사용하여 RHACM 콘솔에 로그인합니다.
 - Advanced Cluster Manager 콘솔에 대해 생성된 경로를 찾습니다.

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

출력 예:

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

OpenShift 자격 증명을 사용하여 로그인하면 로컬 클러스터가 가져와야 합니다.

- RHACM 콘솔을 사용하여 기본 관리 클러스터와 보조 관리 클러스터를 가져오거나 생성했는지 확인합니다.
- 관리형 클러스터에 오버라이딩 이외의 네트워크가 있어야 합니다. Submariner 애드온을 사용하여 관리형 OpenShift 클러스터 및 서비스 네트워크를 연결하려면 관리되는 각 클러스터에 대해 다음 명령을 실행하여 두 클러스터에 오버레이되지 않은 네트워크가 있는지 확인해야 합니다.

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

cluster1의 출력 예(예: ocp4perf1)

```
{
  "clusterNetwork": [
    {
```

```

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

```

cluster2의 출력 예 (예: ocp4perf2)

```

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

```

자세한 내용은 [Submariner add-ons 설명서를 참조하십시오.](#)

- 관리형 클러스터가 **Submariner** 애드온 을 사용하여 연결할 수 있는지 확인합니다. 클러스터 및 서비스 네트워크의 오버레이 범위가 아닌지 확인하고 확인한 후 RHACM 콘솔 및 클러스터 세트를 사용하여 각 관리 클러스터에 대해 **Submariner** 애드온 을 설치합니다. 자세한 내용은 [Submariner 문서를 참조하십시오.](#)

3장. 관리형 클러스터에 OPENSIFT DATA FOUNDATION 설치

절차

1. 각 관리형 클러스터에 OpenShift Data Foundation 버전 4.10을 설치합니다.
OpenShift Data Foundation 배포에 대한 자세한 내용은 [인프라별 배포 가이드](#) (예: AWS, VMware, Bare Metal, Azure)를 참조하십시오.
2. 다음 명령을 사용하여 각 관리 클러스터에서 성공적으로 배포를 확인합니다.

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

Multicloud Object Gateway(MCG)의 경우:

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

상태 결과가 **Primary** 관리 클러스터 및 보조 관리 클러스터의 쿼리 모두에 **Ready** 인 경우 계속 관리 클러스터에서 미러링을 활성화합니다.

4장. HUB 클러스터에 OPENSIFT DR HUB OPERATOR 설치

절차

1. Hub 클러스터에서 OperatorHub로 이동하고 OpenShift DR Hub Operator 에 대한 검색 필터를 사용합니다.
2. 화면 지침에 따라 **openshift-dr-system** 프로젝트에 운영자를 설치합니다.
3. 다음 명령을 사용하여 Operator Pod가 **Running** 상태인지 확인합니다.

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

출력 예:

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

5장. 다중 사이트 스토리지 복제 구성

미러링 또는 복제는 피어 관리 클러스터 내에서 **CephBlockPool** 기준으로 활성화되며 풀 내의 특정 이미지 하위 집합에 구성할 수 있습니다. **rbd-mirror** 데몬은 로컬 피어 클러스터에서 원격 클러스터에서 동일한 이미지로 이미지 업데이트를 복제합니다.

이러한 지침에서는 두 OpenShift Data Foundation 관리 클러스터 간에 미러링 관계를 생성하는 방법을 자세히 설명합니다.

5.1. OPENSIFT DATA FOUNDATION 다중 클러스터 오케스트레이션 설치

OpenShift Data Foundation Multicluster Orchestrator는 Hub 클러스터의 OpenShift Container Platform의 OperatorHub에서 설치된 컨트롤러입니다. 이 멀티클러스터 오케스트레이션 컨트롤러는 mirrorPeer 사용자 정의 리소스와 함께 부트스트랩 토큰을 생성하고 관리 클러스터 간에 이 토큰을 교환합니다.

절차

1. Hub 클러스터에서 OperatorHub 로 이동하고 키워드 필터를 사용하여 ODF Multicluster Orchestrator 를 검색합니다.
2. ODF 멀티클러스터 오케스트레이션 타일을 클릭합니다.
3. 모든 기본 설정을 유지하고 설치를 클릭합니다.
Operator 리소스는 **openshift-operators** 에 설치되고 모든 네임스페이스에서 사용할 수 있습니다.
4. ODF Multicluster Orchestrator 가 설치되었는지 확인합니다.
 - a. Operator 보기를 선택하여 설치를 검증합니다.
 - b. Operator Pod가 **Running** 상태인지 확인합니다.

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

출력 예:

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

5.2. HUB 클러스터에서 미러 피어 생성

mirror Peer는 피어 투 피어 관계가 될 관리 클러스터에 대한 정보를 보유하는 클러스터 범위 리소스입니다.

사전 요구 사항

- ODF Multicluster Orchestrator 가 Hub 클러스터에 설치되어 있는지 확인합니다.
- Mirror Peer당 두 개의 클러스터만 있어야 합니다.
- 각 클러스터에 **ocp4perf1** 및 **ocp4perf 2** 와 같은 고유하게 식별 가능한 클러스터 이름이 있는지 확인합니다.

절차

1. **ODF(ODF) Multicluster Orchestrator** (다중 클러스터 오케스트레이션)를 클릭하여 운영자 세부 정보를 확인합니다.
Multicluster Orchestrator를 성공적으로 설치한 후 **View Operator** (운영자 보기)를 클릭할 수도 있습니다.
2. **Mirror Peer API Create** 인스턴스를 클릭한 다음 **YAML 보기**를 선택합니다.
3. **<cluster1>**을 교체한 후 **<cluster1>** 및 **<cluster 2>**를 RHACM 콘솔에서 관리 클러스터의 올바른 이름으로 교체한 후 다음 **YAML**을 **mirror-peer.yaml**에 복사하고 저장합니다.

```

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

```



참고

schedulingIntervals의 시간 값(예: 5m)은 영구 볼륨 복제에 필요한 간격을 구성하는데 사용됩니다. 이러한 값은 중요한 응용 프로그램을 위해 RPO (recovery Point Purpose)에 매핑 될 수 있습니다. 애플리케이션 요구 사항에 맞게 **schedulingIntervals**의 값을 수정합니다. 최소값은 1m이며 기본값은 5m입니다.

4. 고유한 **mirror-peer.yaml** 파일의 콘텐츠를 **YAML 보기**로 복사합니다. 원본 콘텐츠를 완전히 교체해야 합니다.
5. **YAML 보기** 화면 하단에서 **Create(생성)**를 클릭합니다.
6. 진행하기 전에 단계 상태를 **ExchangedSecret**으로 볼 수 있는지 확인합니다.

5.3. 관리형 클러스터에서 CEPH 미러링 검증

Ceph 미러링이 활성화되어 있는지 확인하려면 기본 관리 클러스터와 보조 관리 클러스터에서 다음 검증을 수행합니다.

1. 미러링이 기본 **Ceph** 블록 풀에서 활성화되었는지 확인합니다.

```

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

```

출력 예:

```
true
```

2. **rbd-mirror** 포드가 실행 중인지 확인합니다.

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

출력 예:

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

3. 데몬 상태의 상태를 확인하여 OK인지 확인합니다.

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

출력 예:

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



참고

daemon_health 및 **health** 필드가 **Warning** 에서 **OK** 로 변경되는 데 최대 10분이 걸릴 수 있습니다. 10분 후에 상태가 OK가 되지 않으면 **Advanced Cluster Manager** 콘솔을 사용하여 하위 애드온 연결이 여전히 정상 상태인지 확인합니다.

4. **Primary managed cluster**에 **VolumeReplicationClass** 가 생성되었고, **MirrorTiB**(예: 5m, 15m)에 나열된 각 예약 간격에 대한 보조 관리 클러스터가 생성되었는지 확인합니다.

```
$ oc get volumereplicationclass
```

출력 예:

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



참고

VolumeReplicationClass 는 복제할 각 볼륨의 **mirroringMode** 와 로컬 클러스터에서 원격 클러스터로 볼륨 또는 이미지가 복제되는 빈도(예: 5분마다)를 지정하는데 사용됩니다.

5.4. 오브젝트 버킷 및 S3STOREPROFILE 검증

Ceph 미러링이 활성화되어 있는지 확인하려면 기본 관리 클러스터와 보조 관리 클러스터에서 다음 검증을 수행합니다.

절차

1. 기본 관리 클러스터에 새 **Object** 버킷 클레임 및 해당 **Object** 버킷 이 있고 **openshift-storage** 네임스페이스에 보조 관리 클러스터 가 있는지 확인합니다.

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

출력 예:

```
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. Hub 클러스터 **openshift-dr-system** 네임스페이스에 새 오브젝트 버킷 클래스 각각에 대한 액세스 및 시크릿 키가 포함된 두 개의 새 시크릿이 있는지 확인합니다.

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

출력 예:

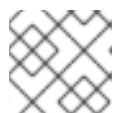
```
8b3fb9ed90f66808d988c7edfa76eba35647092 Opaque 2   16m
af5f82f21f8f77faf3de2553e223b535002e480 Opaque 2   16m
```

3. **OBC** 및 **Secrets**는 새로 생성된 **s3StoreProfiles** 섹션에 있는 Hub 클러스터의 **ConfigMap ramen-hub-operator-config** 에 작성됩니다.

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

출력 예:

```
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
```



참고

s3ProfileName 의 이름을 기록합니다. 이는 **DRPolicy** 리소스에서 사용됩니다.

6장. 미러링 STORAGECLASS 리소스 생성

미러링을 통해 관리되는 클러스터 간에 이미지 복제를 더 빠르게 생성하기 위해 추가 **imageFeatures** 가 필요한 새 **StorageClass** 를 사용하여 블록 볼륨을 생성해야 합니다. 새로운 기능은 **베타적 잠금, 개체 맵 및 fast-diff** 입니다. 기본 OpenShift Data Foundation StorageClass **ocs-storagecluster-ceph-rbd** 에는 이러한 기능이 포함되어 있지 않습니다.



참고

이 리소스는 기본 관리 클러스터와 보조 관리 클러스터에서 생성해야 합니다.

절차

1. 다음 YAML을 파일 이름 **ocs-storagecluster-ceph-rbdmirror.yaml** 에 저장합니다.

```
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. 관리 클러스터 모두에서 파일을 생성합니다.

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

출력 예:

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

7장. S3 끝점 간 SSL 액세스 구성

s3 끝점 간 네트워크(SSL) 액세스를 구성하여 오브젝트 버킷에 대한 액세스를 검증하기 위해 Hub 클러스터에서 **MCG** 오브젝트 버킷의 대체 클러스터에 메타데이터를 저장할 수 있습니다.



참고

환경에 대해 서명된 유효한 인증서 세트를 사용하여 모든 OpenShift 클러스터를 배포하는 경우 이 섹션을 건너뛸 수 있습니다.

절차

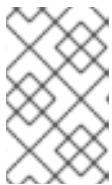
1. 기본 관리 클러스터의 수신 인증서를 추출하고 출력을 **primary.crt**에 저장합니다.

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

2. **Secondary** 관리 클러스터의 수신 인증서를 추출하고 출력을 **secondary.crt**에 저장합니다.

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

3. 기본 관리 클러스터, **Secondary** 관리 클러스터, **Hub** 클러스터의 파일 이름 **cm-clusters.crt.yaml**로 원격 클러스터의 인증서 번들을 보관할 새 **ConfigMap**을 만듭니다.



참고

이 예제 파일에 표시된 대로 각 클러스터에 대해 인증서가 3개 미만일 수 있습니다. 또한 이전에 생성된 **primary.crt** 및 **secondary.crt** 파일에서 복사하여 붙여넣은 후 인증서 콘텐츠의 들여쓰기가 올바른지 확인합니다.

```
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. 기본 관리 클러스터, 보조 관리 클러스터 및 Hub 클러스터에 ConfigMap 파일을 만듭니다.

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

출력 예:

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



중요

Hub 클러스터에서 DRPolicy 리소스를 사용하여 오브젝트 버킷에 대한 액세스 권한을 확인하려면 Hub 클러스터에서 동일한 ConfigMap **cm-clusters.crt.yaml**도 생성해야 합니다.

5. 기본 관리 클러스터, 보조 관리 클러스터 및 Hub 클러스터에서 기본 프록시 리소스를 패치합니다.

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

출력 예:

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

8장. HUB 클러스터에서 재해 복구 정책 생성

OpenShift DR은 RHACM 허브 클러스터에서 DRPolicy(Disaster Recovery Policy) 리소스(클러스터 범위)를 사용하여 관리 클러스터 전체에 워크로드를 배포, 장애 조치 및 재배치합니다.

사전 요구 사항

- 두 개의 클러스터 집합이 스토리지 수준 복제를 위해 피어링되고 CSI 볼륨 복제가 활성화되었는지 확인합니다.
- DRPolicy를 사용하여 워크로드에 대한 중복 복구 지점 목표(RPO) 역할을 하는 빈도 데이터 복제를 결정하는 스케줄링 간격이 있는지 확인합니다.
- 정책에 있는 각 클러스터에 OpenShift DR 클러스터 및 허브 운영자의 ConfigMap을 사용하여 구성된 S3 프로파일 이름이 할당되었는지 확인합니다.

절차

1. Hub 클러스터에서 **openshift-dr-system** 프로젝트에서 Installed Operators로 이동하여 OpenShift DR Hub Operator 를 클릭합니다. 두 개의 사용 가능한 API, DRPolicy 및 DRPlacementControl이 표시되어야 합니다.
2. DRPolicy의 인스턴스 생성을 클릭하고 YAML 보기를 클릭합니다.
3. <cluster1> 및 <cluster 2>를 ACM의 관리형 클러스터의 올바른 이름으로 교체한 후 파일 이름 **drpolicy.yaml** 에 다음 YAML을 복사하고 저장합니다. <string_value_1> 및 <string_value_2>를 고유한 값(예: east 및 west)으로 바꿉니다. **schedulingInterval** 은 이전에짐에 구성된 값 중 하나여야 합니다(예: 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
```



참고

DRPolicy는 클러스터 범위 리소스이므로 이 리소스를 생성하기 위해 네임스페이스를 지정할 필요가 없습니다.

4. 고유한 **drpolicy.yaml** 파일의 콘텐츠를 YAML 보기에 복사합니다. 원본 콘텐츠를 완전히 교체해야 합니다.
5. YAML 보기 화면에서 Create(만들기)를 클릭합니다.



중요

DRPolicy `schedulingInterval` 은 Mirror sandbox 리소스에 구성된 값 중 하나와 일치해야 합니다(예: 5m). Mirror sandbox에 구성된 블록 복제에 다른 `schedulingIntervals` 중 하나를 사용하려면 새 값을 사용하여 DRPolicy 리소스를 추가로 생성해야 합니다(예: 15m). DRPolicy 이름을 고유하게 변경하고 복제 간격(예: odr-policy-15m)을 식별하는 데 유용합니다.

6. 생성된 각 DRPolicy 리소스에 대해 Hub 클러스터에서 명령을 실행하여 DRPolicy 가 생성되었는지 확인합니다. 이 예제는 `odr-policy-5m` 입니다.

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

출력 예:

```
Succeeded
```


9장. OPENSIFT DR 클러스터 OPERATOR 자동 설치 활성화

DRPolicy가 성공적으로 생성되면 **OpenShift DR Cluster Operator** 를 **openshift-dr-system** 네임스페이스의 기본 관리 클러스터와 보조 관리 클러스터에 설치할 수 있습니다.

절차

1. Hub 클러스터에서 **ConfigMap ramen-hub-operator-config** 를 편집하여 다음과 같이 **deploymentAutomationEnabled=true** 를 추가합니다.

```
$ 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. 기본 관리 클러스터에서 설치가 성공적으로 수행되었으며 보조 관리 클러스터에서 다음 명령을 수행하는지 확인합니다.

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

출력 예:

```
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
```

각 관리 클러스터에서 **OperatorHub**로 이동하여 **OpenShift DR Cluster Operator** 가 설치되었는지 확인할 수도 있습니다.

10장. S3SECRETS를 관리 클러스터로 자동 전송 활성화

s3Secrets를 필수 OpenShift DR 클러스터 구성 요소로 자동 전송할 수 있도록 하려면 다음 절차를 따르십시오. OpenShift DR 구성 맵의 s3Profiles에 액세스하는 데 필요한 s3Secrets로 OpenShift DR 클러스터 네임스페이스를 업데이트합니다.

절차

1. 다음과 같이 Hub 클러스터에서 ConfigMap `ramen-hub-operator-config` 를 편집하여 `s3SecretDistributionEnabled=true` 를 추가합니다.

```
$ 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. 두 관리 클러스터에서 이 명령을 실행하여 시크릿 전송에 성공했는지 확인합니다.

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

출력 예:

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

11장. 샘플 애플리케이션 생성

기본 관리 클러스터에서 보조 관리 클러스터로 장애 조치를 테스트하고 다시 간단한 애플리케이션이 필요합니다. 예제로 **busybox** 라는 샘플 애플리케이션을 사용합니다.

절차

1. **busybox** 샘플 애플리케이션에 사용할 Hub 클러스터에 네임스페이스 또는 프로젝트를 생성합니다.

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



참고

원하는 경우 **busybox-sample** 이외의 다른 프로젝트 이름을 사용할 수 있습니다. 이 단계에서 생성되는 프로젝트 이름과 동일한 프로젝트 이름을 사용하도록 Advanced Cluster Manager 콘솔을 통해 샘플 애플리케이션을 배포할 때 확인합니다.

2. DRPlacementControl 리소스 만들기

DRPlacementControl은 OpenShift DR Hub Operator가 Hub 클러스터에 설치된 후 사용할 수 있는 API입니다. DRPolicy의 일부인 클러스터 전체의 데이터 가용성을 기반으로 배치 결정을 조정하는 고급 Cluster Manager PlacementRule 조정기입니다.

- a. Hub 클러스터에서 **busybox-sample** 프로젝트에서 Installed Operators로 이동하여 OpenShift DR Hub Operator 를 클릭합니다. 두 개의 사용 가능한 API, DRPolicy 및 DRPlacementControl이 표시되어야 합니다.
- b. DRPlacementControl 인스턴스를 만든 다음 YAML 보기로 이동합니다. **busybox-sample** 프로젝트가 선택되었는지 확인합니다.
- c. <cluster1>을 *Advanced Cluster Manager*에서 관리 클러스터의 올바른 이름으로 대체한 후 **busybox-drpc.yaml** 파일 이름에 다음 YAML을 저장합니다. 원하는 복제 간격이 있는 DRPolicy 의 **drPolicyRef** 이름을 수정합니다.

```
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. 고유한 **busybox-drpc.yaml** 파일의 콘텐츠를 YAML 보기(원래 콘텐츠 전체 교체)로 복사합니다.

- e. YAML 보기 화면에서 **Create(만들기)**를 클릭합니다.
다음 CLI 명령을 사용하여 이 리소스를 생성할 수도 있습니다.

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

출력 예:

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



중요

이 리소스는 **busybox-sample** 네임스페이스 또는 이전에 생성한 네임스페이스에서 생성해야 합니다.

3. 리소스 템플릿을 배포할 수 있는 대상 클러스터를 정의하는 배치 규칙 리소스를 만듭니다. 배치 규칙을 사용하여 애플리케이션의 다중 클러스터 배포를 용이하게 합니다.

- a. 다음 YAML을 복사하여 **busybox-placementrule.yaml** 파일 이름에 저장합니다.

```
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** 애플리케이션의 배치 규칙 리소스를 생성합니다.

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

출력 예:

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



중요

이 리소스는 **busybox-sample** 네임스페이스 또는 이전에 생성한 네임스페이스에서 생성해야 합니다.

4. RHACM 콘솔을 사용하여 샘플 애플리케이션 생성

- a. 아직 로그인하지 않은 경우 OpenShift 자격 증명을 사용하여 RHACM 콘솔에 로그인합니다.

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

출력 예:

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

- b. Applications (애플리케이션) 로 이동하여 Create Application(애플리케이션 만들기)을 클릭합니다.
- c. Subscription (서브스크립션)으로 type(유형)을 선택합니다.
- d. 애플리케이션 이름 (예: **busybox**) 및 네임스페이스 (예: **busybox-sample**)를 입력합니다.
- e. resources의 Repository location(리포지토리 위치) 섹션에서 Repository type **Git**(리포지토리 유형 **Git**)을 선택합니다.
- f. 샘플 애플리케이션의 Git 리포지토리 URL, 리소스 **busybox** Pod 및 PVC가 생성될 github 분기 및 경로를 입력합니다.
분기 가 주 이며 경로가 **busybox-odr** 인 <https://github.com/RamenDR/ocm-ramen-samples> 샘플 애플리케이션 리포지토리를 사용합니다.



중요

계속하기 전에 새 StorageClass **ocs-storagecluster-ceph-rbdmirror** 가 상제한 대로 생성되었는지 확인합니다.

https://access.redhat.com/documentation/en-us/red_hat_openshift_data_foundation/4.10/html-single/configuring_openshift_data_foundation_for_regional-dr_with_advanced_cluster_management#proc_creating-mirroring-storageclass-resource_rhodf

다음 명령을 사용하여 생성되었는지 확인합니다.

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

출력 예:

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

- g. 양식을 아래로 스크롤하여 Select an existing placement configuration(클러스터 선택)을 클릭하고 Select an existing placement configuration(기존 배치 구성 선택)을 클릭합니다.
- h. 드롭다운 목록에서 an Existing Placement Rule (예: **busybox-placement**)을 선택합니다.
- i. 저장을 클릭합니다.
후속 화면에서 아래쪽으로 스크롤합니다. On the follow-on screen scroll to the bottom. 애플리케이션 토폴로지에 모든 그린 확인 표시가 있는지 확인해야 합니다.



참고

자세한 내용을 보려면 토폴로지 요소 중 하나를 클릭하면 토폴로지 보기 오른쪽에 창이 표시됩니다.

5. 샘플 애플리케이션 배포 및 복제를 확인합니다.

busybox 애플리케이션이 기본 클러스터에 배포되었으므로(DRPlacementControl에 지정됨) 배포를 검증할 수 있습니다.

- a. RHACM에서 **busybox** 를 배포한 관리형 클러스터에 로그인합니다.

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

출력 예:

```
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. **busybox** PVC에 대한 복제 리소스도 생성되었는지 확인합니다.

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

출력 예:

```
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. 기본 관리 클러스터와 보조 관리 클러스터에서 다음 명령을 실행하여 **busybox** 볼륨이 대체 클러스터에 복제되었는지 확인합니다.

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

출력 예:

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



참고

두 관리 클러스터 모두 새 상태가 **"states":{"replaying":2}**인 것과 정확히 동일해야 합니다.

11.1. 샘플 애플리케이션 삭제

RHACM 콘솔을 사용하여 샘플 애플리케이션 **busybox** 를 삭제할 수 있습니다.



참고

샘플 애플리케이션을 삭제하는 지침은 장애 조치(failover) 및 실패(relocate) 테스트가 완료되고 애플리케이션이 RHACM 및 관리되는 클러스터에서 제거할 준비가 될 때까지 실행되지 않아야 합니다.

절차

1. RHACM 콘솔에서 Applications (애플리케이션)로 이동합니다.
2. 삭제할 샘플 애플리케이션(예: **busybox**)을 검색합니다.
3. 삭제할 애플리케이션 옆에 있는 작업 메뉴(kube)를 클릭합니다.
4. Delete application (애플리케이션 삭제)을 클릭합니다.
애플리케이션을 삭제하면 애플리케이션 관련 리소스도 삭제해야 하는지를 묻는 새 화면이 나타납니다.
5. Remove application related resources(애플리케이션 관련 리소스 제거) 확인란을 선택하여 Subscription(서브스크립션) 및 PlacementRule(배치)을 삭제합니다.
6. 삭제를 클릭합니다. 그러면 기본 관리 클러스터(또는 애플리케이션이 실행 중인 모든 클러스터)에서 busybox 애플리케이션이 삭제됩니다.
7. RHACM 콘솔을 사용하여 삭제된 리소스 외에도 **busybox** 애플리케이션을 삭제한 직후 **DRPlacementControl** 도 삭제해야 합니다.
 - a. Hub 클러스터의 OpenShift 웹 콘솔에 로그인하고 **busybox-sample** 프로젝트에 대해 설치된 Operator로 이동합니다.
 - b. OpenShift DR Hub Operator 를 클릭한 다음 DRPlacementControl 탭을 클릭합니다.
 - c. 삭제할 **busybox** 애플리케이션 DRPlacementControl 옆에 있는 작업 메뉴(kube)를 클릭합니다.
 - d. Delete DRPlacementControl(DRPlacementControl 삭제)를 클릭합니다.
 - e. 삭제를 클릭합니다.



참고

이 프로세스는 **DRPlacementControl** 리소스로 애플리케이션을 삭제하는 데 사용할 수 있습니다. CLI를 사용하여 애플리케이션 네임스페이스에서도 **DRPlacementControl** 리소스를 삭제할 수 있습니다.

12장. 관리되는 클러스터 간 애플리케이션 페일오버

이 섹션에서는 `busybox` 샘플 애플리케이션을 장애 조치하는 방법에 대한 지침을 제공합니다. `Regional-DR`의 장애 조치 방법은 애플리케이션을 기반으로 합니다. 이러한 방식으로 보호할 각 애플리케이션에는 `DR` 테스트용 샘플 애플리케이션 생성 섹션에 표시된 대로 해당 `DRPlacementControl` 리소스와 애플리케이션 네임스페이스에 생성된 `PlacementRule` 리소스가 있어야 합니다.

절차

1. Hub 클러스터에서 `Installed Operators`(설치된 운영자)로 이동한 다음 `Openshift DR Hub Operator`를 클릭합니다.
2. `DRPlacementControl` 탭을 클릭합니다.
3. `DRPC busybox-drpc`를 클릭한 다음 `YAML 보기`를 클릭합니다.
4. 아래 스크린샷과 같이 `action` 및 `failoverCluster` 세부 정보를 추가합니다. `failoverCluster`는 보조 관리 클러스터의 `ACM` 클러스터 이름입니다.

`DRPlacementControl` 추가 작업 실패

Project: busybox-sample ▾

Installed Operators > odr-hub-operator.v4.10.0 > DRPlacementControl details

DRPC busybox-drpc DeployedDetails 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

- 저장을 클릭합니다.
- YAML 파일에 지정된 페일오버 클러스터 **ocp4perf2** 인 Secondary managed 클러스터에서 이제 애플리케이션 **busybox** 가 실행 중인지 확인합니다.

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

출력 예:

```

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** 가 더 이상 기본 관리 클러스터에서 실행되지 않는지 확인합니다.

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

출력 예:

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



중요

릴리스 노트의 [Known issues](#) 섹션에 설명된 알려진 Region-DR 문제에 대해 유의하십시오.

13장. 관리 클러스터 간 애플리케이션 재배포

재배포 작업은 장애 조치(failover)와 매우 유사합니다. 재배포는 애플리케이션을 기반으로 하며 DRPlacementControl를 사용하여 재배포를 트리거합니다. 재배포의 주요 차이점은 2차 관리 클러스터에 저장된 새 애플리케이션 데이터가 즉시 복제 되도록 재동기화 하여 기본 관리 클러스터에 복제된 미러링 일정 간격을 기다리지 않도록 한다는 것입니다.

절차

1. Hub 클러스터에서 Installed Operators(설치된 운영자)로 이동한 다음 Openshift DR Hub Operator 를 클릭합니다.
2. DRPlacementControl 탭을 클릭합니다.
3. DRPC `busybox-drpc`를 클릭한 다음 YAML 보기를 클릭합니다.
4. 재배포하기 위한 작업 수정

재배포하기 위한 DRPlacementControl 수정 작업

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

```

Save Reload Cancel

- 저장을 클릭합니다.
- 이제 애플리케이션이 Primary managed cluster에서 실행 중인지 확인합니다. failback은 YAML 파일에 지정된 대로 기본Cluster ocp4perf1 에 해당합니다. 장애 조치(failover) 작업 전에 애플리케이션이 실행 중인 위치인 YAML 파일에 지정된 대로 failback은 기본 클러스터 ocp4perf1에 해당합니다.

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

출력 예:

```
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 가 보조 관리 클러스터에서 실행 중인지 확인합니다. **busybox** 애플리케이션은 이 관리 클러스터에서 더 이상 실행되지 않아야 합니다.

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

출력 예:

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

중요

릴리스 노트의 [Known issues](#) 섹션에 설명된 알려진 **Region-DR** 문제에 대해 유의하십시오.