Red Hat OpenShift Data Foundation 4.16

デプロイメントのプランニング

Red Hat OpenShift Data Foundation 4.16 をデプロイする際の重要な考慮事項

Last Updated: 2024-08-17

Red Hat OpenShift Data Foundation 4.16 デプロイメントのプランニング

Red Hat OpenShift Data Foundation 4.16 をデプロイする際の重要な考慮事項

法律上の通知

概要

このドキュメントは、Red Hat OpenShift Data Foundation のデプロイメントを計画する際の重要な考慮事項を説明します。

目次

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 プロパティー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、用語の置き換えは、今後の複数のリリースにわたって段階的に実施されます。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)

Red Hat ドキュメントに対する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。

フィードバックを送信するには、Bugzilla チケットを作成します。

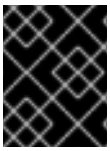
1. Bugzilla の Web サイトに移動します。
2. **Component** セクションで、**documentation** を選択します。
3. **Description** フィールドに、ドキュメントの改善に向けたご提案を記入してください。ドキュメントの該当部分へのリンクも記載してください。
4. **Submit Bug** をクリックします。

第1章 OPENSHIFT DATA FOUNDATION の紹介

Red Hat OpenShift Data Foundation は、Red Hat OpenShift Container Platform のクラウドストレージおよびデータサービスの集合です。これは、単純なデプロイメントや管理を容易に実行できるように Operator として同梱されており、Red Hat OpenShift Container Platform サービスカタログの一部として提供されます。

Red Hat OpenShift Data Foundation サービスは、主に以下のコンポーネントを表すストレージクラスを使用してアプリケーションで使用できます。

- ブロックストレージデバイス。主にデータベースのワークロードに対応します。主な例には、Red Hat OpenShift Container Platform のロギングおよびモニタリング、および PostgreSQL などがあります。



重要

ブロックストレージは、複数のコンテナー間でデータを共有する必要がない場合にのみ、ワークロードに使用する必要があります。

- 共有および分散ファイルシステム。主にソフトウェア開発、メッセージング、およびデータ集約のワークロードに対応します。これらの例には、Jenkins ビルドソースおよびアーティファクト、Wordpress のアップロードコンテンツ、Red Hat OpenShift Container Platform レジストリー、および JBoss AMQ を使用したメッセージングが含まれます。

- Multicloud オブジェクトストレージ。複数のクラウドオブジェクトストアからのデータの保存および取得を抽象化できる軽量 S3 API エンドポイントを特長としています。

- オンプレミスオブジェクトストレージ。主にデータ集約型アプリケーションをターゲットとする数十ペタバイトおよび数十億のオブジェクトにスケーリングする堅牢な S3 API エンドポイントを特長としています。これらの例には、Spark、Presto、Red Hat AMQ Streams (Kafka) などのアプリケーションや、TensorFlow や Pytorch などのマシンラーニングフレームワークを使用した行、列、および半構造化データの保存およびアクセスが含まれます。

注記

CephFS 永続ボリューム上での PostgresSQL ワークロードの実行はサポートされていないため、RADOS Block Device (RBD) ボリュームを使用することを推奨します。詳細は、ナレッジベースソリューション ODF Database Workloads Must Not Use CephFS PV/PVC を参照してください。

Red Hat OpenShift Data Foundation バージョン 4.x は、以下を含むソフトウェアプロジェクトのコレクションを統合します。

- Ceph。ブロックストレージ、共有および分散ファイルシステム、およびオンプレミスのオブジェクトストレージを提供します。

- Ceph CSI。永続ボリュームおよび要求のプロビジョニングおよびライフサイクルを管理します。

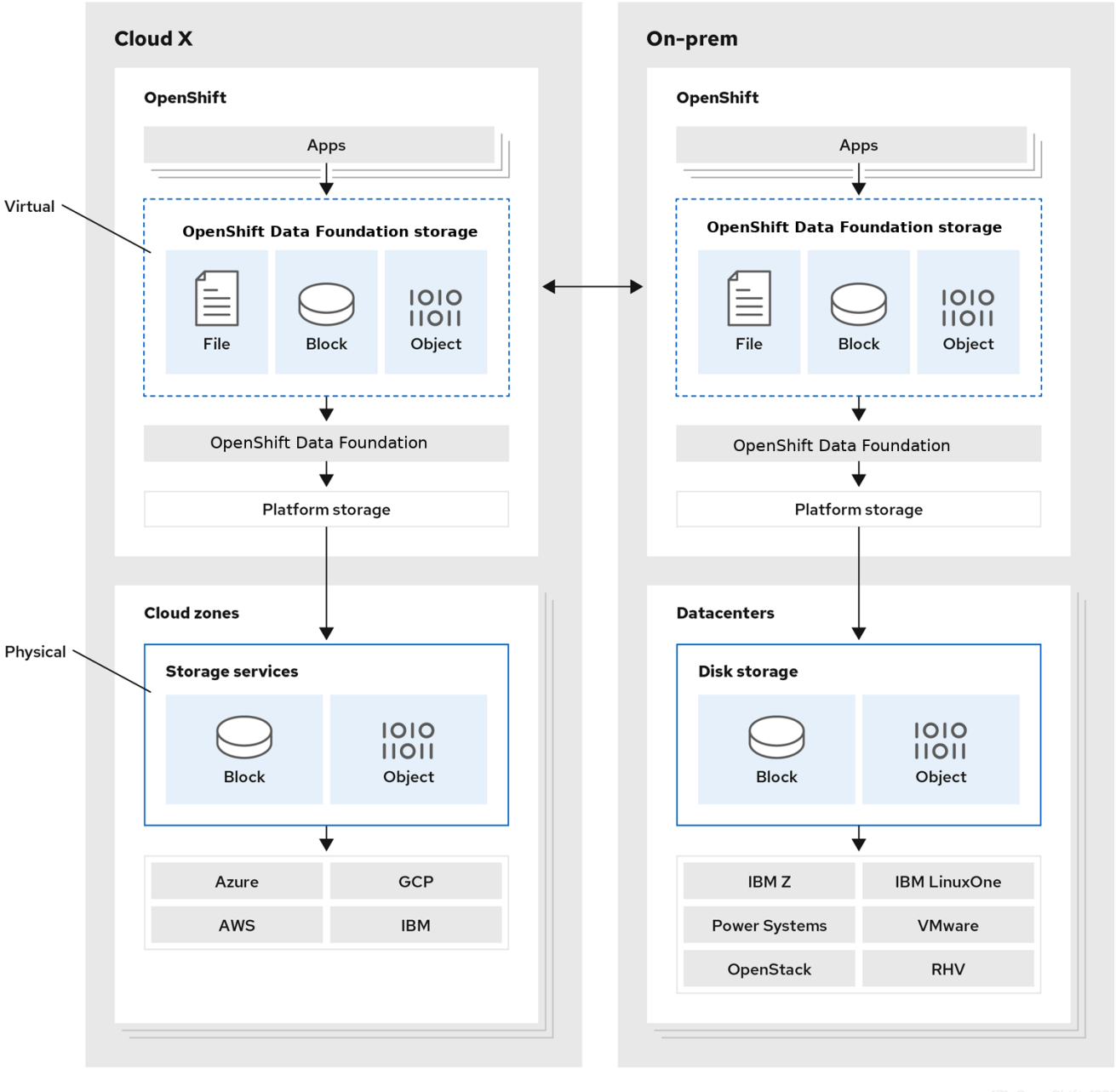
- NooBaa。Multicloud Object Gateway を提供します。

- OpenShift Data Foundation サービスを初期化し、管理する OpenShift Data Foundation、Rook-Ceph、および NooBaa Operator。

第2章 OPENSHIFT DATA FOUNDATION のアーキテクチャー

Red Hat OpenShift Data Foundation は、Red Hat OpenShift Container Platform のサービスを提供し、Red Hat OpenShift Container Platform の内部で実行できます。

図2.1 Red Hat OpenShift Data Foundation アーキテクチャー



Red Hat OpenShift Data Foundation は、インストーラーでプロビジョニングされるインフラストラクチャー、またはユーザーによってプロビジョニングされるインフラストラクチャーでデプロイされる Red Hat OpenShift Container Platform クラスターへのデプロイメントをサポートします。

これら 2 つの方法は、OpenShift Container Platform のインストールプロセス を参照してください。

Red Hat OpenShift Data Foundation と Red Hat OpenShift Container Platform のコンポーネントの相互運用性の詳細は、Red Hat OpenShift Data Foundation のサポート性および相互運用性チェッカー を参照してください。

OpenShift Container Platform のアーキテクチャーおよびライフサイクルの詳細は、OpenShift Container Platform アーキテクチャー を参照してください。

ヒント

IBM Power は、OpenShift Container Platform インストールプロセス を参照してください。

2.1. OPERATOR

Red Hat OpenShift Data Foundation は 3 つの主要な Operator で構成されています。これらの Operator は、管理タスクとカスタムリソースをコード化して、タスクとリソースの特性を簡単に自動化できるようにします。管理者はクラスターの必要な最終状態を定義し、OpenShift Data Foundation Operator は管理者の介入を最小限に抑えてクラスターをその状態にするか、その状態に近づけるようにします。

OpenShift Data Foundation Operator

サポートされる Red Hat OpenShift Data Foundation のデプロイメントの推奨事項と要件を成文化し、実施するために、特定のテストされた方法で他の Operator を利用するメタ Operator です。rook-ceph および noobaa operator は、これらのリソースをラップするストレージクラスターリソースを提供します。

Rook-ceph operator

この Operator は、永続ストレージおよびファイル、ブロックおよびオブジェクトサービスのパッケージ化、デプロイメント、管理、アップグレード、およびスケーリングを自動化します。これは、すべての環境用にブロックおよびファイルストレージクラスを作成し、オンプレミス環境でオブジェクトストレージクラスおよびサービスオブジェクトバケット要求 (OBC) を作成します。

さらに、内部モードクラスターの場合、以下を表すデプロイメントおよびサービスを管理する ceph クラスターリソースを提供します。

- オブジェクトストレージデーモン (OSD)
- モニター (MON)
- マネージャー (MGR)
- メタデータサーバー (MDS)
- オンプレミスのみの RADOS オブジェクトゲートウェイ (RGW)

Multicloud Object Gateway operator

この Operator は、Multicloud Object Gateway (MCG) オブジェクトサービスのパッケージ化、デプロイメント、管理、アップグレード、およびスケーリングを自動化します。オブジェクトストレージクラスを作成し、それに対して作成された OBC にサービスを提供します。

さらに、これは NooBaa クラスターリソースを提供します。このクラスターリソースは、NooBaa コア、データベースおよびエンドポイントのデプロイメントとサービスを管理します。

2.2. ストレージクラスターのデプロイメントアプローチ

モード数 (operating modalities) が増えていることからも、柔軟性が Red Hat OpenShift Data Foundation の主な特徴であることが分かります。本セクションでは、お使いの環境に最も適した方法を選択するのに役立つ情報を提供します。

Red Hat OpenShift Data Foundation は OpenShift Container Platform 内で完全にデプロイ (内部アプローチ) することも、OpenShift Container Platform 外で実行されるクラスターからサービスを利用可能する方法 (外部アプローチ) を実行することもできます。

2.2.1. 内部アプローチ

Red Hat OpenShift Data Foundation を Red Hat OpenShift Container Platform 内にすべてデプロイすると、Operator ベースのデプロイメントおよび管理からのすべての利点が得られます。グラフィカルユーザーインターフェイス (GUI) で内部接続デバイス方式を使用すると、Local Storage Operator とローカルストレージデバイスを使用して、Red Hat OpenShift Data Foundation を内部モードでデプロイできます。

デプロイメントおよび管理の容易性は、OpenShift Data Foundation サービスを OpenShift Container Platform の内部で実行することに関する主な特長となっています。Red Hat OpenShift Data Foundation が完全に Red Hat OpenShift Container Platform 内で実行されている場合、以下の 2 つのデプロイメントモードを使用できます。

- Simple (単純)
- Optimized (最適化)

Simple (単純) デプロイメント

Red Hat OpenShift Data Foundation サービスは、アプリケーションと共存する形で実行されます。Red Hat OpenShift Container Platform の operator がこのようなアプリケーションを管理します。

Simple (単純) デプロイメントは、以下のような場合に最も適しています。

- ストレージ要件が明確ではない。
- Red Hat OpenShift Data Foundation サービスは、アプリケーションと共存して実行されている。
- 特定のサイズのノードインスタンスを作成することが困難である (例: ベアメタル)。

Red Hat OpenShift Data Foundation をアプリケーションと共存させるには、ノードにローカルストレージデバイス、または EC2 の EBS ボリューム、VMware の vSphere 仮想ボリューム、SAN ボリュームなどのポータブルストレージデバイスを動的に接続する必要があります。

注記

PowerVC は SAN ボリュームを動的にプロビジョニングします。

Optimized (最適化) デプロイメント

Red Hat OpenShift Data Foundation サービスは、専用のインフラストラクチャーノードで実行します。Red Hat OpenShift Container Platform がこのようなインフラストラクチャーノードを管理します。

最適化アプローチは、以下の場合に最も適しています。

- ストレージ要件が明確である。
- Red Hat OpenShift Data Foundation サービスが、専用のインフラストラクチャーノードで実行されている。
- 特定サイズのノードインスタンスの作成が容易である (例: クラウド、仮想化環境など)。

2.2.2. 外部アプローチ

Red Hat OpenShift Data Foundation は、OpenShift Container Platform クラスター外で実行されている Red Hat Ceph Storage サービスをストレージクラスとして公開します。

以下の場合に外部アプローチが最も適しています。

- ストレージ要件の規模が大きい (600 以上のストレージデバイス)。
- 複数の OpenShift Container Platform クラスターが共通の外部クラスターからストレージサービスを使用する必要がある。
- 別のチームであるサイトリライアビリティーエンジニアリング (SRE)、ストレージなどは、ストレージサービスを提供する外部クラスターを管理する必要がある。(すでに存在している場合があります)。

2.3. ノードのタイプ

ノードはコンテナーランタイムとサービスを実行し、コンテナーが実行中の状態にし、Pod 間のネットワーク通信および分離を保ちます。OpenShift Data Foundation には、3 種類のノードがあります。

表2.1 ノードの種類

ノードタイプ	説明
マスター	これらのノードは、Kubernetes API を公開し、新たに作成された Pod を監視およびスケジュールし、ノードの正常性および数量を維持し、基礎となるクラウドプロバイダーとの対話を制御するプロセスを実行します。
インフラストラクチャー (インフラ)	インフラストラクチャーノードは、ロギング、メトリクス、レジストリー、およびルーティングなどのクラスターレベルのインフラストラクチャーサービスを実行します。これらは OpenShift Container Platform クラスターではオプションです。OpenShift Data Foundation レイヤーワークロードをアプリケーションから分離するには、仮想化環境およびクラウド環境で OpenShift Data Foundation に infra ノードを使用することを確認します。 **infra** というラベルが付けられた新規ノードをプロビジョニングして、インフラストラクチャーノードを作成できます。詳細は、Red Hat OpenShift Data Foundation に専用のワーカーノードを使用する方法 を参照してください。

ノードタイプ	説明
ワーカー	ワーカーノードは、アプリケーションを実行するため、アプリケーションノードとしても知られています。 OpenShift Data Foundation が内部モードでデプロイされている場合、3 つのワーカーノードの最小クラスターが必要です。可用性を確保するために、ノードが 3 つの異なるラックまたはアベイラビリティーゾーンに分散していることを確認してください。OpenShift Data Foundation をワーカーノードで実行するには、ローカルストレージデバイスまたはポータブルストレージデバイスをワーカーノードに動的に接続する必要があります。 OpenShift Data Foundation が外部モードでデプロイされている場合、複数のノードで実行されます。これにより、Kubernetes は、障害が発生した場合に使用可能なノードでスケジュールを変更できます。

注記

OpenShift Data Foundation には、OpenShift Container Platform と同じ数のサブスクリプションが必要です。ただし、OpenShift Data Foundation がインフラノードで実行されている場合、OpenShift はこれらのノードに OpenShift Container Platform サブスクリプションを必要としません。したがって、OpenShift Data Foundation コントロールプレーンには、追加の OpenShift Container Platform および OpenShift Data Foundation サブスクリプションは必要ありません。詳細は、6章 *サブスクリプション* を参照してください。

第3章 内部ストレージサービス

Red Hat OpenShift Data Foundation サービスは、以下のインフラストラクチャーで実行されている Red Hat OpenShift Container Platform の内部で利用できます。

- Amazon Web Services (AWS)
- ベアメタル
- VMware vSphere
- Microsoft Azure
- Google Cloud
- Red Hat OpenStack 13 以降 (インストーラーがプロビジョニングしたインフラストラクチャー) [テクノロジープレビュー]
- IBM Power
- IBM Z および IBM® LinuxONE

内部クラスターリソースを作成すると、OpenShift Data Foundation ベースサービスの内部プロビジョニングが実行され、追加のストレージクラスがアプリケーションで使用可能になります。

第4章 外部ストレージサービス

Red Hat OpenShift Data Foundation は、IBM FlashSystems を使用するか、外部の Red Hat Ceph Storage クラスターからのサービスを以下のプラットフォームで実行されている OpenShift Container Platform クラスターを介して利用できるようにします。

- VMware vSphere
- ベアメタル
- Red Hat OpenStack Platform (テクノロジープレビュー)
- IBM Power
- IBM Z

OpenShift Data Foundation Operator は、外部サービスに対する永続ボリューム (PV) およびオブジェクトバケット要求 (OBC) を満たすためにサービスを作成し、管理します。外部クラスターは、OpenShift Container Platform で実行されているアプリケーションのブロック、ファイル、およびオブジェクトストレージクラスを提供できます。Operator は、外部クラスターをデプロイまたは管理しません。

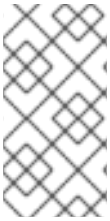
第5章 セキュリティーに関する考慮事項

5.1. FIPS-140-2

FIPS-140-2 (Federal Information Processing Standard Publication 140-2) は、暗号モジュールの使用に関する一連のセキュリティー要件を定義する標準です。法律は、米国政府機関および請負業者にこの基準を義務付けており、他の国際および業界固有の基準でも参照されています。

Red Hat OpenShift Data Foundation は、FIPS 検証済みの暗号化モジュールを使用するようになりました。Red Hat Enterprise Linux OS/CoreOS(RHCOS) は、これらのモジュールを提供します。

現在、暗号化モジュール検証プログラム (CMVP) は暗号化モジュールを処理します。これらのモジュールの状態は、プロセスリストのモジュール で確認できます。最新情報は、Red Hat ナレッジベースソリューションの RHEL コア暗号化コンポーネント を参照してください。



注記

OpenShift Data Foundation をインストールする前に、OpenShift Container Platform で FIPS モードを有効にします。この機能は Red Hat Enterprise Linux 7 (RHEL 7) での OpenShift Data Foundation デプロイメントをサポートしていないため、OpenShift Container Platform は RHCOS ノードで実行する必要があります。

詳細は、OpenShift Container Platform ドキュメントの インストールガイド に記載されている **FIPS モードでのクラスターのインストール** および **FIPS 暗号化のサポート**を参照してください。

5.2. プロキシー環境

プロキシー環境は、インターネットへの直接アクセスを拒否し、代わりに利用可能な HTTP または HTTPS プロキシーを提供する実稼働環境です。Red Hat Openshift Container Platform は、既存クラスターのプロキシーオブジェクトを変更するか、新規クラスターについて install-config.yaml ファイルでプロキシーを設定してプロキシーを使用するように設定されます。

Red Hat は、OpenShift Container Platform が クラスター全体のプロキシーの設定 に従って設定されている場合に、プロキシー環境での OpenShift Data Foundation のデプロイメントをサポートします。

5.3. データ暗号化オプション

暗号化を使用すると、必要な暗号化キーがなければデータを読み取ることができないようにデータをエンコードできます。このメカニズムは、物理メディアが手元から離れるような物理的なセキュリティー違反の発生時にもデータの機密性を保護できます。PV ごとの暗号化は、同じ OpenShift Container Platform クラスター内の他の namespace からのアクセス保護も提供します。データはディスクに書き込まれる際に暗号化され、ディスクから読み取られる際に復号化されます。暗号化されたデータを使用すると、パフォーマンスに小規模なペナルティーのみが発生する可能性があります。

暗号化は、Red Hat OpenShift Data Foundation 4.6 以降を使用してデプロイされる新規クラスターでのみサポートされます。外部鍵管理システム (KMS) を使用していない既存の暗号化されたクラスターは、外部 KMS を使用するように移行できません。

以前は、HashiCorp Vault はクラスター全体の暗号化および永続ボリュームの暗号化で唯一サポートされている KMS です。OpenShift Data Foundation 4.7.0 および 4.7.1 では、HashiCorp Vault Key/Value (KV) シークレットエンジン API (バージョン 1) のみがサポートされます。OpenShift Data Foundation

4.7.2 以降では、HashiCorp Vault KV シークレットエンジン API (バージョン 1 および 2) がサポートされるようになりました。OpenShift Data Foundation 4.12 の時点で、サポートされる追加の KMS として Thales CipherTrust Manager が導入されました。

重要

- KMS は StorageClass 暗号化に必須であり、クラスター全体の暗号化にはオプションです。
- まず、ストレージクラスの暗号化には、有効な Red Hat OpenShift Data Foundation Advanced サブスクリプションが必要です。詳細は、OpenShift Data Foundation サブスクリプションに関するナレッジベースの記事 を参照してください。

Red Hat はテクノロジーパートナーと連携して、このドキュメントをお客様へのサービスとして提供します。ただし、Red Hat では、Hashicorp 製品のサポートを提供していません。この製品に関するテクニカルサポートについては、Hashicorp にお問い合わせください。

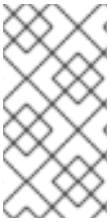
5.3.1. クラスター全体の暗号化

Red Hat OpenShift Data Foundation は、ストレージクラスター内のディスクおよび Multicloud Object Gateway 操作のすべてに対して、クラスター全体の暗号化 (保存時の暗号化、encryption-at-rest) をサポートします。OpenShift Data Foundation は、キーのサイズが 512 ビットの Linux Unified Key System (LUKS) バージョン 2 ベースの暗号化と、各デバイスが異なる暗号化キーを持つ **aes-xts-plain64** 暗号を使用します。このキーは Kubernetes シークレットまたは外部 KMS を使用して保存されます。どちらのメソッドも同時に使用できず、メソッド間の移行はできません。

ブロックおよびファイルストレージの暗号化は、デフォルトで無効になっています。デプロイメント時にクラスターの暗号化を有効にできます。MultiCloud Object Gateway は、デフォルトで暗号化をサポートしています。詳細は、デプロイメントガイドを参照してください。

クラスター全体の暗号化は、鍵管理システム (KMS) を使用しない OpenShift Data Foundation 4.6 でサポートされます。OpenShift Data Foundation 4.7 以降では、HashiCorp Vault KMS の有無にかかわらずサポートされます。OpenShift Data Foundation 4.12 以降では、HashiCorp Vault KMS および Thales CipherTrust Manager KMS の有無にかかわらずサポートされます。

一般的なセキュリティープラクティスでは、定期的な暗号鍵のローテーションが求められます。Red Hat OpenShift Data Foundation は、Kubernetes シークレット (KMS 以外) に保存されている暗号鍵を毎週自動的にローテーションします。



注記

有効な Red Hat OpenShift Data Foundation Advanced サブスクリプションが必要です。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。

HashiCorp Vault KMS を使用したクラスター全体の暗号化には、次の 2 つの認証方法があります。

- **トークン**: このメソッドでは、vault トークンを使用した認証が可能です。Vault トークンを含む kubernetes シークレットは、openshift-storage namespace で作成され、認証に使用されます。この認証方法を選択した場合、管理者は、暗号化キーが保存されている Vault のバックエンドパスへのアクセスを提供する Vault トークンを提供する必要があります。
- **Kubernetes**: このメソッドでは、serviceaccounts を使用して Vault で認証できます。この認証

方法を選択した場合、管理者は、暗号化キーが保存されているバックエンドパスへのアクセスを提供する Vault で設定されたロールの名前を指定する必要があります。次に、このロールの値が **ocs-kms-connection-details** config map に追加されます。このメソッドは、OpenShift Data Foundation 4.10 から利用できます。
現時点で、HashiCorp Vault は唯一サポートされている KMS です。OpenShift Data Foundation 4.7.0 および 4.7.1 では、HashiCorp Vault KV シークレットエンジン API (バージョン 1) のみがサポートされます。OpenShift Data Foundation 4.7.2 以降では、HashiCorp Vault KV シークレットエンジン API (バージョン 1 および 2) がサポートされるようになりました。

注記

IBM Cloud プラットフォーム上の OpenShift Data Foundation は、HashiCorp Vault KMS に加えて、暗号化ソリューションとして Hyper Protect Crypto Services (HPCS)Key Management Services (KMS) をサポートします。

重要

Red Hat はテクノロジーパートナーと連携して、このドキュメントをお客様へのサービスとして提供します。ただし、Red Hat では、Hashicorp 製品のサポートを提供していません。この製品に関するテクニカルサポートについては、Hashicorp にお問い合わせください。

5.3.2. ストレージクラスの暗号化

デバイスの暗号化キーを保存するために外部の鍵管理システム (KMS) を使用して、ストレージクラスの暗号化で永続ボリューム (ブロックのみ) を暗号化できます。永続ボリュームの暗号化は RADOS Block Device (RBD) 永続ボリュームでのみ利用できます。永続ボリュームの暗号化を使用したストレージクラスの作成方法 を参照してください。

ストレージクラスの暗号化は、HashiCorp Vault KMS を使用する OpenShift Data Foundation 4.7 以降でサポートされます。ストレージクラスの暗号化は、HashiCorp Vault KMS と Thales CipherTrust Manager KMS の両方を使用する OpenShift Data Foundation 4.12 以降でサポートされます。

注記

有効な Red Hat OpenShift Data Foundation Advanced サブスクリプションが必要です。OpenShift Data Foundation のサブスクリプションの仕組みを確認するには、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。

5.3.3. CipherTrust マネージャー

Red Hat OpenShift Data Foundation バージョン 4.12 では、デプロイメントの追加の鍵管理システム (KMS) プロバイダーとして Thales CipherTrust Manager が導入されています。Thales CipherTrust Manager は、一元化された鍵のライフサイクル管理を提供します。CipherTrust Manager は、鍵管理システム間の通信を可能にする Key Management Interoperability Protocol (KMIP) をサポートしています。

CipherTrust Manager は、デプロイメント時に有効になります。

5.3.4. Red Hat Ceph Storage のメッセンジャーバージョン 2 プロトコル (msgr2) を使用したデータの暗号化

OpenShift Data Foundation バージョン 4.14 以降、Red Hat Ceph Storage のメッセンジャーバージョン 2 プロトコルを使用して、転送中のデータを暗号化できるようになりました。これにより、インフラストラクチャーに重要なセキュリティー要件が提供されます。

転送中の暗号化は、クラスターの作成中、デプロイ時に有効にできます。クラスターの作成中に転送中のデータ暗号化を有効にする手順については、ご使用の環境の デプロイメントガイド を参照してください。

msgr2 プロトコルは、次の 2 つの接続モードをサポートしています。

crc

- cephx で接続を確立する際に、強力な初期認証を提供します。
- ビットフリップから保護する crc32c 整合性チェックを提供します。
- 悪意のある中間者攻撃に対する保護を提供しません。
- 盗聴者がすべての認証後のトラフィックを見るのを妨げません。

secure

- cephx で接続を確立する際に、強力な初期認証を提供します。
- 認証後トラフィックをすべて完全に暗号化します。
- 暗号化整合性チェックを提供します。

デフォルトのモードは **crc** です。

5.4. 転送中での暗号化

OVN-Kubernetes Container Network Interface (CNI) クラスターネットワーク上のノード間のすべてのネットワークトラフィックが暗号化されたトンネルを通過するように、IPsec を有効にする必要があります。

デフォルトでは、IPsec は無効になっています。クラスターのインストール中またはインストール後に有効化できます。クラスターのインストール後に IPsec を有効にする必要がある場合は、IPsec ESP IP ヘッダーのオーバーヘッドを考慮して、まずクラスター MTU のサイズを変更する必要があります。

IPsec 暗号化の設定方法の詳細は、OpenShift Container Platform ドキュメントの ネットワークガイド の **IPsec 暗号化の設定** を参照してください。

第6章 サブスクリプション

6.1. サブスクリプションのオファリング

Red Hat OpenShift Data Foundation のサブスクリプションは、OpenShift Container Platform と同様に “コアのペア” をベースとして提供されます。Red Hat OpenShift Data Foundation 2 コアサブスクリプションは、OpenShift Container Platform が実行されるシステムの CPU 上の論理コア数をベースとしています。

以下の点は、OpenShift Container Platform と同様です。

- OpenShift Data Foundation サブスクリプションは、大規模なホストに対応するようにスタック可能です。
- コアは、必要に応じて多数の仮想マシン (VM) に分散できます。たとえば、10 の 2 コアサブスクリプションは 20 コアを提供し、IBM Power の場合、SMT レベル 8 の 2 コアのサブスクリプションは、任意の数の仮想マシンで使用できる 2 コアまたは 16 vCPU が提供されます。
- OpenShift Data Foundation サブスクリプションは、Premium または Standard サポートで利用できます。

6.2. 障害復旧サブスクリプションの要件

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をすべて満たす必要があります。

- 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

ソースまたは宛先としてアクティブレプリケーションに参加している PV を含む Red Hat OpenShift Data Foundation クラスターには、OpenShift Data Foundation Advanced エンタイトルメントが必要です。このサブスクリプションは、ソースクラスターと宛先クラスターの両方でアクティブにする必要があります。

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。

6.3. コア対 VCPU およびハイパースレッディング

現時点で特定のシステムが 1 つまたは複数のコアを消費するかどうかに関する決定は、そのシステムでハイパースレッディング機能を利用できるかどうかによって異なります。ハイパースレッディングは Intel CPU のみの機能です。Red Hat カスタマーポータルにアクセスし、特定のシステムがハイパースレッディングをサポートしているかどうかを判断します。

ハイパースレッディングが有効にされており、1 つのハイパースレッドが 1 つの利用可能なシステムコアに等しいシステムの場合、コアの計算 は 2 コア対 4 vCPU の比率になります。したがって、2 コアのサブスクリプションは、ハイパースレッドシステムの 4 vCPU に対応します。大規模な仮想マシン (VM) には、4 サブスクリプションコアに相当する 8 vCPU がある場合があります。サブスクリプションは 2 コア単位で提供されるため、4 コアまたは 8 vCPU に対応するには 2 つの 2 コアサブスクリプションが必要になります。

ハイパースレッディングが有効にされていない場合や、表示される各システムのコアが基礎となる物理コアに直接相関する場合、コアの計算は 2 コア対 2 vCPU の比率になります。

6.3.1. IBM Power のコア対 vCPU および同時マルチスレッド (SMT)

現時点で特定のシステムが 1 つまたは複数のコアを消費するかどうかに関する決定は、同時マルチスレッド (SMT) のレベルによって異なります。IBM Power は、以下の表にあるように、各コアの同時マルチスレッドレベルの 1、2、4、または 8 を提供します。これは vCPU の数に対応します。

表6.1 さまざまな SMT レベルとそれに対応する vCPU

SMT レベル	SMT=1	SMT=2	SMT=4	SMT=8
1 コア	# vCPU=1	# vCPU=2	# vCPU=4	# vCPU=8
2 コア	# vCPU=2	# vCPU=4	# vCPU=8	# vCPU=16
4 コア	# vCPU=4	# vCPU=8	# vCPU=16	# vCPU=32

SMT が設定されたシステムでは、サブスクリプションに必要なコア数の計算は SMT レベルによって異なります。したがって、2 コアのサブスクリプションは、上記の表に示すように SMT レベル 1 の 2 vCPU に対応し、SMT レベル 2 の 4 vCPU に、SMT レベル 4 では 8 vCPU に、SMT レベル 8 では 16 vCPU に対応します。大規模な仮想マシン (VM) には 16 vCPU が含まれる場合があります。この場合、SMT レベル 8 では SMT レベルで vCPU の # を除算した計算により 2 コアサブスクリプションが必要になります (SMT-8 の場合: 16 vCPU/8 = 2)。サブスクリプションは 2 コア単位で提供されるため、これらの 2 コアまたは 16 vCPU に対応するには 1 つの 2 コアサブスクリプションが必要になります。

6.4. コアの分割

奇数のコアを必要とするシステムの場合でも、2 コアのサブスクリプションを使用する必要があります。たとえば、必要なコアが 1 つだけであると計算されたシステムは、登録およびサブスクライブされると、最終的に 2 コアのサブスクリプションを完全に消費することになります。

2 vCPU を持つ単一の仮想マシン (VM) がハイパースレッディング機能を使用し、1 vCPU が計算される場合、2 コアサブスクリプションが必要になります。単一の 2 コアサブスクリプションは、ハイパースレッディングを使用する 2 vCPU を持つ 2 仮想マシン間で分割することはできません。詳細は、コア対 vCPU およびハイパースレッディング セクションを参照してください。

そのため、仮想インスタンスは偶数のコアを必要とするようにサイズ設定することが推奨されます。

6.4.1. IBM Power 用の共有プロセッサープール

IBM Power には、共有プロセッサープールの概念があります。共用プロセッサープール内のプロセッサーは、クラスター内のノード間で共有できます。Red Hat OpenShift Data Foundation に必要な集約コンピュート容量はコアのペアの倍数である必要があります。

6.5. サブスクリプションの要件

Red Hat OpenShift Data Foundation コンポーネントは、Red Hat CoreOS (RHCOS) または Red Hat Enterprise Linux (RHEL) 8.4 のいずれかをホストのオペレーティングシステムとして使用できる OpenShift Container Platform ワーカーまたはインフラストラクチャーノードのいずれかで実行できます。RHEL 7 は非推奨になりました。OpenShift Data Foundation サブスクリプションは、すべての OpenShift Container Platform をサブスクライブするコアに 1:1 の割合で必要です。

インフラストラクチャーノードを使用する場合、OpenShift Container Platform または OpenShift Data Foundation サブスクリプションが必要でなくても、OpenShift Data Foundation のすべての OpenShift

ワーカーノードコアをサブスクライブするルールが適用されます。ラベルを使用して、ノードがワーカーノードまたはインフラストラクチャーノードであるかどうか示すことができます。

詳細は、ストレージリソースの管理および割り当て ガイドの Red Hat OpenShift Data Foundation に専用のワーカーノードを使用する方法 を参照してください。

第7章 インフラストラクチャーの要件

7.1. プラットフォーム要件

Red Hat OpenShift Data Foundation 4.16 は、OpenShift Container Platform バージョン 4.16 およびその次のマイナーバージョンでのみサポートされます。

以前のバージョンの Red Hat OpenShift Data Foundation に関するバグ修正は、バグ修正バージョンとしてリリースされます。詳細は、Red Hat OpenShift Container Platform ライフサイクルポリシー を参照してください。

外部クラスターのサブスクリプション要件については、Red Hat ナレッジベースの記事 OpenShift Data Foundation Subscription Guide を参照してください。

サポートされているプラットフォームバージョンの完全なリストについては、Red Hat OpenShift Data Foundation Supportability and Interoperability Checker を参照してください。

7.1.1. Amazon EC2

Red Hat OpenShift Data Foundation 内部クラスターのみをサポートします。

内部クラスターは、ストレージデバイス要件 の両方を満たし、aws-ebs プロビジョナー経由で EBS ストレージを提供するストレージクラスを備えている必要があります。

OpenShift Data Foundation は、Amazon Web Services (AWS) によって導入された **gp2-csi** および **gp3-csi** ドライバーをサポートします。これらのドライバーは、より優れたストレージ拡張機能と割引された月額料金 (**gp3-csi**) を提供します。ストレージクラスを選択するときに、新しいドライバーを選択できるようになりました。高いスループットが必要な場合は、OpenShift Data Foundation をデプロイするときに **gp3-csi** を使用することを推奨します。

高い 1 秒あたりの入出力操作 (IOPS) が必要な場合、推奨される EC2 インスタンスタイプは **D2** または **D3** です。

7.1.2. ベアメタル

内部クラスターをサポートし、外部クラスターの使用をサポートします。

内部クラスターは、ストレージデバイスの要件 を満たし、Local Storage Operator 経由でローカル SSD (NVMe/SATA/SAS、SAN) を提供するストレージクラスを備えている必要があります。

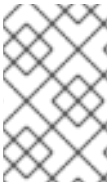
7.1.3. VMware vSphere

内部クラスターをサポートし、外部クラスターの使用をサポートします。

推奨されるバージョン:

- vSphere 6.7、Update2 以降
- vSphere 7.0 以降

詳細は、VMware vSphere インフラストラクチャーの要件 を参照してください。



注記

VMware ESXi がデバイスをフラッシュとして認識しない場合は、それらをフラッシュデバイスとしてマークします。Red Hat OpenShift Data Foundation をデプロイする前に、ストレージデバイスをフラッシュとしてマーク を参照してください。

さらに、内部クラスターは、ストレージデバイス要件 を満たし、次のいずれかを提供するストレージクラスを備えている必要があります。

- vSAN または VMFS データストア (vsphere-volume プロビジョナー経由)
- VMDK、RDM、または DirectPath ストレージデバイス (Local Storage Operator 経由)

7.1.4. Microsoft Azure

Red Hat OpenShift Data Foundation 内部クラスターのみをサポートします。

内部クラスターは、ストレージデバイス要件 を満たし、azure-disk プロビジョナー経由で Azure ディスクを提供するストレージクラスを備えている必要があります。

7.1.5. Google Cloud

Red Hat OpenShift Data Foundation 内部クラスターのみをサポートします。

内部クラスターは、ストレージデバイス要件 の両方を満たし、gce-pd プロビジョナー経由で GCE Persistent Disk を提供するストレージクラスを備えている必要があります。

7.1.6. Red Hat OpenStack Platform [テクノロジープレビュー]

内部 Red Hat OpenShift Data Foundation クラスターをサポートし、外部クラスターを使用します。

内部クラスターは、ストレージデバイス要件 を満たし、Cinder プロビジョナー経由で標準ディスクを提供するストレージクラスを備えている必要があります。

7.1.7. IBM Power

内部 Red Hat OpenShift Data Foundation クラスターをサポートし、外部クラスターを使用します。

内部クラスターは、ストレージデバイスの要件 を満たし、Local Storage Operator 経由でローカル SSD (NVMe/SATA/SAS、SAN) を提供するストレージクラスを備えている必要があります。

7.1.8. IBM Z および IBM® LinuxONE

内部 Red Hat OpenShift Data Foundation クラスターをサポートします。また、Red Hat Ceph Storage が x86 上で実行される外部モードもサポートします。

内部クラスターは、ストレージデバイスの要件 を満たし、Local Storage Operator 経由でローカル SSD (NVMe/SATA/SAS、SAN) を提供するストレージクラスを備えている必要があります。

7.1.9. 任意のプラットフォーム

内部クラスターをサポートし、外部クラスターの使用をサポートします。

内部クラスターは、ストレージデバイスの要件 を満たし、Local Storage Operator 経由でローカル SSD (NVMe/SATA/SAS、SAN) を提供するストレージクラスを備えている必要があります。

7.2. 外部モード要件

7.2.1. Red Hat Ceph Storage

外部モードの Red Hat OpenShift Data Foundation および Red Hat Ceph Storage (RHCS) のサポートと相互運用性を確認するには、Red Hat OpenShift Data Foundation サポートおよび相互運用性チェッカー のラボにアクセスします。

1. **ODF as Self-Managed Service** として **Service Type** を選択します。
2. ドロップダウンから適切な **Version** を選択します。
3. Versions タブで、**Supported RHCS Compatibility** タブをクリックします。

RHCS クラスターのインストール方法は、インストールガイド を参照してください。

7.2.2. IBM FlashSystem

IBM FlashSystem を他のプロバイダーのプラグ可能な外部ストレージとして使用するには、最初に OpenShift Data Foundation をデプロイする必要があります。これは、IBM FlashSystem ストレージクラスをバッキングストレージとして使用します。

サポートされている最新の FlashSystem ストレージシステムとバージョンについては、IBM ODF FlashSystem ドライバーのドキュメント を参照してください。

OpenShift Data Foundation をデプロイする方法は、Creating an OpenShift Data Foundation Cluster for external IBM FlashSystem storage を参照してください。

7.3. リソース要件

Red Hat OpenShift Data Foundation のサービスは、ベースサービスの初期セットで構成されており、追加のデバイスセットで拡張できます。これらの Red Hat OpenShift Data Foundation サービス Pod はすべて、OpenShift Container Platform ノード上の kubernetes によってスケジュールされます。クラスターを (障害ドメインごとに 1 ノード) 3 の倍数に拡張する方法は、Pod の配置ルールを簡単に満たす方法です。

重要

これらの要件は、OpenShift Data Foundation サービスのみに関連し、これらのノードで実行している他のサービス、Operator、またはワークロードには関連しません。

表7.1 Red Hat OpenShift Data Foundation のみの利用可能なリソース要件の集約

デプロイメントモード	ベースサービス	追加のデバイスセット

デプロイメントモード	ベースサービス	追加のデバイスセット
内部	• 30 個の CPU (論理) • 72 GiB メモリー • 3 ストレージデバイス	• 6 個の CPU (論理) • 15 GiB メモリー • 3 ストレージデバイス
外部	• 4 個の CPU (論理) • 16 GiB メモリー	該当なし

例: 単一デバイスセットを持つ内部モードデプロイメントの 3 ノードクラスターの場合、最小の 3 x 10 = 30 ユニットの CPU が必要です。

詳細は、6章 *サブスクリプション* および CPU ユニット を参照してください。

Red Hat OpenShift Data Foundation クラスターの設計に関する追加のガイダンスは、ODF Sizing Tool を参照してください。

CPU ユニット

本セクションでは、1 CPU ユニットは Kubernetes コンセプトの 1 CPU ユニットにマップされます。

- CPU の 1 ユニットは、ハイパースレッディングされていない CPU の 1 コアに相当します。
- CPU の 2 ユニットは、ハイパースレッディングされている CPU の 1 コアに相当します。
- Red Hat OpenShift Data Foundation コアベースのサブスクリプションは常にペア (2 コア) で提供されます。

表7.2 IBM Power の最小リソース要件の集約

デプロイメントモード	ベースサービス
内部	• 48 CPU (論理) • 192 GiB メモリー • 3 つのストレージデバイス (それぞれに追加の 500GB ディスクが含まれる)
外部	• 24 個の CPU (論理) • 48 GiB メモリー

例: 内部接続デバイスモードのデプロイメントの 3 ノードクラスターの場合、最小の 3 x 16 = 48 ユニットの CPU、および 3 x 64 = 192 GB が必要です。

7.3.1. IBM Z および IBM LinuxONE インフラストラクチャーのリソース要件

Red Hat OpenShift Data Foundation のサービスは、ベースサービスの初期セットで構成されており、追加のデバイスセットで拡張できます。

これらの Red Hat OpenShift Data Foundation サービス Pod はすべて、OpenShift Container Platform ノード上の kubernetes によってスケジュールされます。クラスターを (障害ドメインごとに 1 ノード) 3 の倍数に拡張する方法は、Pod の配置ルールを簡単に満たす方法です。

表7.3 Red Hat OpenShift Data Foundation でのみ利用可能なリソース要件を集約 (IBM Z および IBM® LinuxONE)

デプロイメントモード	ベースサービス	追加のデバイスセット	IBM Z および IBM® LinuxONE の最小ハードウェア要件
内部	• 30 個の CPU (論理) ◦ それぞれ 10 個の CPU (論理) を備えた 3 つのノード • 72 GiB メモリー • 3 ストレージデバイス	• 6 個の CPU (論理) • 15 GiB メモリー • 3 ストレージデバイス	1 IFL
外部	• 4 個の CPU (論理) • 16 GiB メモリー	該当なし	該当なし

CPU

ハイパーバイザー、IBM Z/VM、カーネル仮想マシン (KVM)、またはその両方で定義されている仮想コアの数です。

IFL(Linux 向けの統合機能)

IBM Z および IBM® LinuxONE の物理コアです。

最小システム環境

- 1 つの論理パーティション (LPAR) で最小クラスターを動作させるには、6 つの IFL の上に追加の IFL が必要です。OpenShift Container Platform は、これらの IFL を使用します。

7.3.2. デプロイメントリソースの最小要件

OpenShift Data Foundation クラスターは、標準のデプロイメントリソース要件を満たしていない場合に、最小の設定でデプロイされます。

重要

これらの要件は、OpenShift Data Foundation サービスのみに関連し、これらのノードで実行している他のサービス、Operator、またはワークロードには関連しません。

表7.4 OpenShift Data Foundation のみのリソース要件の集約

デプロイメントモード	ベースサービス
内部	• 24 個の CPU (論理) • 72 GiB メモリー • 3 ストレージデバイス

デバイスセットを追加する場合は、最小デプロイメントを標準デプロイメントに変換することが推奨されます。

7.3.3. コンパクトなデプロイメントリソース要件

Red Hat OpenShift Data Foundation は、3 ノードの OpenShift のコンパクトなベアメタルクラスターにインストールできます。ここでは、すべてのワークロードが 3 つの強力なマスターノードで実行されます。ワーカーノードまたはストレージノードは含まれません。

重要

これらの要件は、OpenShift Data Foundation サービスのみに関連し、これらのノードで実行している他のサービス、Operator、またはワークロードには関連しません。

表7.5 OpenShift Data Foundation のみのリソース要件の集約

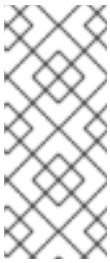
デプロイメントモード	ベースサービス	追加のデバイスセット
内部	• 24 個の CPU (論理) • 72 GiB メモリー • 3 ストレージデバイス	• 6 個の CPU (論理) • 15 GiB メモリー • 3 ストレージデバイス

コンパクトのベアメタルクラスターで OpenShift Container Platform を設定するには、3 ノードクラスターの設定 について、また エッジデプロインメントの 3 ノードアーキテクチャーの提供 について参照してください。

7.3.4. MCG のみのデプロイメントのリソース要件

Multicloud Object Gateway (MCG) コンポーネントのみを使用してデプロイされた OpenShift Data Foundation クラスターは、デプロイメントに柔軟性を提供し、リソース消費を削減するのに役立ちます。

表7.6 MCG のみのデプロイメントの総リソース要件

デプロイメントモード	コア	データベース (DB)	Endpoint (エンドポイント)
内部	• 1 CPU • 4 GiB メモリー	• 0.5 CPU • 4 GiB メモリー	• 1 CPU • 2 GiB メモリー  **注記** デフォルトオートスケールは 1〜2 です。

7.3.5. ネットワークファイルシステムを使用するためのリソース要件

Network File System (NFS) を使用してエクスポートを作成すると、OpenShift クラスターから外部からアクセスできます。この機能を使用する場合、NFS サービスは 3 つの CPU と 8Gi の RAM を消費します。NFS はオプションであり、デフォルトでは無効になっています。

NFS ボリュームには、次の 2 つの方法でアクセスできます。

- クラスター内: Openshift クラスター内のアプリケーション Pod による。
- cluster: Openshift クラスター外から。

NFS 機能の詳細は、NFS を使用したエクスポートの作成 を参照してください。

7.3.6. パフォーマンスプロファイルのリソース要件

OpenShift Data Foundation は、クラスターのパフォーマンスを向上させる 3 つのパフォーマンスプロファイルを提供します。利用可能なリソースと、デプロイ時またはデプロイ後に必要なパフォーマンスレベルに基づいて、これらのプロファイルの 1 つを選択できます。

表7.7 さまざまなパフォーマンスプロファイルの推奨リソース要件

パフォーマンスプロファイル	CPU	メモリー
Lean	24	72 GiB
Balanced	30	72 GiB
パフォーマンス	45	96 GiB

重要

すでに他のワークロードを実行している可能性があるため、利用可能な空きリソースに基づいてプロファイルを選択してください。

7.4. POD の配置ルール

Kubernetes は、宣言型の配置ルールに基づいて Pod の配置を行います。内部クラスターの Red Hat OpenShift Data Foundation ベースサービスの配置ルールは、以下のように要約できます。

- ノードには **cluster.ocs.openshift.io/openshift-storage** キーでラベルが付けられます。
- ノードは、擬似障害ドメインに分類されます (何も存在しない場合)。
- 高可用性が必要なコンポーネントは障害ドメインに分散されます。
- ストレージデバイスはそれぞれの障害ドメインでアクセスできる必要があります。

これにより、少なくとも 3 つのノードがあり、既存の トポロジーラベル が存在する場合にノードは 3 つの異なるラックまたはゾーン障害ドメインにある必要があります。

追加のデバイスセットについては、3 つの障害ドメインのそれぞれにストレージデバイスがあり、Pod が消費するのに十分なリソースが必要になります。手動の配置ルールはデフォルトの配置ルールを上書きするのに使用できますが、通常この方法はベアメタルのデプロイメントにのみ適しています。

7.5. ストレージデバイスの要件

このセクションでは、内部モードのデプロイメントおよびアップグレードの計画時に考慮できる各種のストレージ容量の要件を説明します。通常、ノードごとにデバイスを 12 個以下にすることを推奨します。この推奨事項により、ノードがクラウドプロバイダーの動的ストレージデバイスの割り当て制限下にあり、ローカルストレージデバイスに関連してノードに障害が発生した後の復旧時間を制限できます。クラスターを (障害ドメインごとに 1 ノード) 3 の倍数に拡張する方法は、Pod の配置ルール を簡単に満たす方法です。

ストレージノードには少なくとも 2 つのディスクが必要です。1 つはオペレーティングシステム用で、残りのディスクは OpenShift Data Foundation コンポーネント用です。

注記

ストレージ容量は、インストール時に選択した容量の増分値でのみ拡張できます。

7.5.1. 動的ストレージデバイス

Red Hat OpenShift Data Foundation では、動的ストレージデバイスサイズの要求サイズとして 0.5 TiB、2 TiB または 4 TiB の容量を選択できます。ノードごとに実行できる動的ストレージデバイスの数は、ノードのサイズ、基盤となるプロビジョナーの制限、および リソース要件 によって決まります。

7.5.2. ローカルストレージデバイス

ローカルストレージのデプロイメントの場合、16 TiB 以下のディスクサイズを使用でき、すべてのディスクが同じサイズおよび種類である必要があります。ノードごとに実行できるローカルストレージデバイスの数は、ノードのサイズと リソース要件 によって決まります。クラスターを (障害ドメインごとに 1 ノード) 3 の倍数に拡張する方法は、Pod の配置ルール を簡単に満たす方法です。

注記

ディスクのパーティション設定はサポートされません。

7.5.3. 容量のプランニング

使用する前に、利用可能なストレージ容量を必ず確保するようにしてください。利用可能なストレージ容量が完全に使い切られる場合はリカバリーが難しく、単に容量を追加したり、コンテンツを削除したり、移行したりするよりも多くの介入が必要になります。

容量アラートは、クラスターストレージ容量が合計容量の 75% (ほぼ一杯) および 85% (一杯) になると発行されます。容量に関する警告に常に迅速に対応し、ストレージを定期的に確認して、ストレージ領域が不足しないようにします。75% (ほぼフル) に達したら、スペースを解放するか、クラスターを拡張します。85% (フル) アラートに達すると、ストレージ領域が完全に不足していて、標準コマンドを使用して領域を解放できないことが示唆されます。この時点で、Red Hat カスタマーサポート にお問い合わせください。

次の表は、動的ストレージデバイスを使用した Red Hat OpenShift Data Foundation のノード設定の例を示しています。

表7.8 3 つのノードで設定される初期設定の例

ストレージデバイスのサイズ	ノードあたりのストレージデバイス	合計容量	利用可能なストレージ容量
0.5 TiB	1	1.5 TiB	0.5 TiB
2 TiB	1	6 TiB	2 TiB
4 TiB	1	12 TiB	4 TiB

表7.9 30 ノード (N) で拡張された設定の例

ストレージデバイスのサイズ (D)	ノードごとのストレージデバイス (M)	合計容量 (D * M * N)	使用可能なストレージ容量 (D*M*N/3)
0.5 TiB	3	45 TiB	15 TiB
2 TiB	6	360 TiB	120 TiB
4 TiB	9	1080 TiB	360 TiB

第8章 ネットワーク要件

このセクションを使用して、デプロイメントを計画する際のさまざまなネットワークの考慮事項を理解してください。

8.1. IPV6 サポート

Red Hat OpenShift Data Foundation バージョン 4.12 では、IPv6 のサポートが導入されました。IPv6 はシングルスタックでのみサポートされ、IPv4 と同時に使用することはできません。Openshift Container Platform で IPv6 がオンになっている場合は、IPv6 が OpenShift Data Foundation のデフォルトの動作です。

Red Hat OpenShift Data Foundation バージョン 4.14 では、IPv6 の自動検出と設定が導入されています。IPv6 を使用するクラスターは、それに応じて自動的に設定されます。

Red Hat OpenShift Data Foundation IPv4 を使用した OpenShift Container Platform デュアルスタックは、バージョン 4.13 以降でサポートされます。Red Hat OpenShift Data Foundation IPv6 でのデュアルスタックはサポートされていません。

8.2. マルチネットワークプラグイン (MULTUS) のサポート

OpenShift Data Foundation は、ベアメタルインフラストラクチャー上でマルチネットワークプラグイン Multus を使用する機能をサポートし、さまざまなタイプのネットワークトラフィックを分離することでセキュリティーとパフォーマンスを向上させます。Multus を使用すると、OpenShift Data Foundation 専用に、ホスト上の 1 つ以上のネットワークインターフェイスを予約できますが、

Multus を使用するには、まず Multus 前提条件検証ツールを実行します。ツールの使用方法は、OpenShift Data Foundation - Multus 前提条件検証ツール を参照してください。Multus ネットワークの詳細は、複数ネットワーク を参照してください。

Multus ネットワークを IPv4 または IPv6 を使用するように設定できます (テクノロジープレビュー機能)。これは、純粋な IPv4 または純粋な IPv6 の Multus ネットワークでのみ機能します。ネットワークは混在モードにできません。

重要

テクノロジープレビュー機能は、近々発表予定の製品イノベーションをリリースに先駆けてご提供することにより、お客様は機能性をテストし、開発プロセス中にフィードバックをお寄せいただくことができます。ただし、この機能は Red Hat のサービスレベルアグリーメントで完全にサポートされていないため、機能的に完全でない可能性があり、実稼働環境での使用を目的としていません。Red Hat ではテクノロジープレビュー機能を今後も繰り返し使用することで一般提供に移行できると考えていることから、お客様がこの機能を使用する際に発生する問題の解決に取り組みます。

詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

8.2.1. Multus の前提条件

Ceph-CSI が Multus 対応の CephCluster と通信するには、Kubernetes ホストに設定が必要です。

以下の前提条件を満たすには、Multus ネットワークの設定方法と Rook による使用方法の理解が必要です。このセクションは、発生する可能性のある疑問を明確にするのに役立ちます。

次の 2 つの基本要件を満たす必要があります。

- OpenShift ホストが、Multus パブリックネットワークに正常にルーティングできる必要があります。
- Multus パブリックネットワーク上の Pod が、OpenShift ホストに正常にルーティングできる必要があります。

これら 2 つの要件は、さらに次のように細分化できます。

- Kubernetes ホストを Multus パブリックネットワークにルーティングするには、各ホストで次の点を確認する必要があります。
  - ホストに、Multus パブリックネットワークに接続されたインターフェイス ("public-network-interface") が必要です。
  - "public-network-interface" には IP アドレスが必要です。
  - Multus パブリックネットワーク上の Pod 宛てのトラフィックを "public-network-interface" 経由で送信するためのルートが存在する必要があります。
- Multus パブリックネットワーク上の Pod を Kubernetes ホストにルーティングするには、パブリック NetworkAttachmentDefinition を設定して、次の点を確認する必要があります。
  - 定義に、ネットワークを介してノード宛てのトラフィックをルーティングするように IP アドレス管理 (IPAM) が設定されている必要があります。
- 2 つのネットワーク間のルーティングが適切に機能するように、ノードに割り当てられた IP アドレスが、Multus パブリックネットワーク上の Pod に割り当てられた IP アドレスと重複していない必要があります。
- 通常、NetworkAttachmentDefinition とノード設定の両方で、Multus パブリックネットワークに接続するために同じネットワークテクノロジー (Macvlan) を使用する必要があります。

ノード設定と Pod 設定は相互に関連しており、密接に結び付いています。両方を同時に計画する必要があります。どちらも OpenShift Data Foundation で Multus パブリックネットワークをサポートするのに不可欠です。

実際には、“public-network-interface” は両方の設定で同じである必要があります。一般的に、接続テクノロジー (Macvlan) も両方で同じである必要があります。NetworkAttachmentDefinition 内の IP 範囲は、ノード上のルートとしてエンコードする必要があります。ミラーでは、ノードの IP 範囲は NetworkAttachmentDefinition 内のルートとしてエンコードする必要があります。

概念的には、Pod とノードの両方に同じパブリックネットワーク IP アドレス範囲を使用するのが最も簡単ですが、すべてのインストール環境でそうすることが望まれるわけではありません。Pod とノードの範囲が異なる場合は、各範囲が他の範囲にルーティングされ、単一の連続したネットワークとして機能するように、さらなる注意を払う必要があります。

これらの要件を満たすには慎重な計画が必要ですが、方法によっては他の方法よりも簡単にこれらの要件を満たすことができます。これらの要件を理解して実装するには、Multus の例 を参照してください。

ヒント

多くの場合、ストレージノードごとに 10 個以上の OpenShift Data Foundation Pod が存在することに注意してください。通常、Pod のアドレス空間は、ホストのアドレス空間よりも数倍 (またはそれ以上) 大きくする必要があります。

OpenShift Container Platform では、ホストの要件を満たすようにホストを設定する適切な方法として、NMState Operator の NodeNetworkConfigurationPolicies を使用することを推奨しています。必要または希望に応じて、他の方法も使用できます。

8.2.1.1. Multus ネットワークのアドレス空間サイズの設定

Multus ネットワークのアドレス空間サイズを決定するには、見通しと計画が必要です。ネットワークには、ネットワークにアタッチされるストレージ Pod の数に対応できる十分な数のアドレスと、フェイルオーバーイベントに対応できる追加のスペースが必要です。

将来のストレージクラスターの拡張についても事前に計画し、OpenShift Container Platform および OpenShift Data Foundation クラスターが将来どの程度大きくなる可能性があるかを予測することを強く推奨します。将来の拡張を見越してアドレスを予約しておくと、拡張時に IP アドレスプールが予期せず枯渇するリスクが低くなります。

最も安全なのは、ストレージクラスターの稼働期間中に同時に必要になると予想されるアドレスの最大合計数よりも 25% (またはそれ以上) 多くのアドレスを割り当てることです。これにより、フェイルオーバーやメンテナンス中に IP アドレスプールが枯渇するリスクが軽減されます。

対応するネットワーク CIDR 設定を簡単に記述できるように、合計を最も近い 2 の累乗に切り上げることも推奨されます。

次の 3 つの範囲を計画する必要があります。

- パブリック Network Attachment Definition のアドレス空間を使用する場合、openshift-storage namespace で実行されている ODF Pod の合計数に応じた十分な IP がアドレス空間に含まれている必要があります。
- クラスター Network Attachment Definition のアドレス空間を使用する場合、openshift-storage namespace で実行されている OSD Pod の合計数に応じた十分な IP がアドレス空間に含まれている必要があります。
- Multus パブリックネットワークを使用する場合、ノードパブリックネットワークのアドレス空間に、Multus パブリックネットワークに接続されている OpenShift ノードの合計数に応じた十分な IP が含まれている必要があります。

注記

クラスターで、パブリック Network Attachment Definition とノードパブリックネットワークアタッチメントに 1 つのアドレス空間を使用する場合は、上記の 2 つの要件を加算してください。これは、DHCP を使用してパブリックネットワークの IP を管理する場合などに関係します。

8.2.1.1.1. 推奨設定

ほとんどの組織では、次の推奨設定で十分です。この推奨設定では、範囲の先頭が使用中であると想定して、または望ましいと想定して、予約済みプライベートアドレス空間 (192.168.0.0/16) の最後の 6.25% (1/16) を使用します。おおよその最大値 (25% のオーバーヘッドを考慮) を示します。

表8.1 Multus の推奨設定

ネットワーク	ネットワーク範囲 CIDR	おおよその最大値
パブリック Network Attachment Definition	192.168.240.0/21	合計 1,600 個の ODF Pod

ネットワーク	ネットワーク範囲 CIDR	おおよその最大値
クラスター Network Attachment Definition	192.168.248.0/22	800 個の OSD
ノードパブリックネットワークアタッチメント	192.168.252.0/23	合計 400 ノード

8.2.1.1.2. 計算

より詳細なアドレス空間のサイズは次のように決定できます。

1. 将来必要になる可能性のある OSD の最大数を決定します。25% を足してから 5 を足します。結果を最も近い 2 の累乗に切り上げます。これがクラスターのアドレス空間のサイズです。

2. ステップ 1 で計算した四捨五入されていない数値から始めます。64 を足してから 25% を足します。結果を最も近い 2 の累乗に切り上げます。これが Pod のパブリックアドレス空間のサイズです。

3. 将来必要になる可能性のある OpenShift ノード (ストレージノードを含む) の合計最大数を決定します。25% を足します。結果を最も近い 2 の累乗に切り上げます。これがノードのパブリックアドレス空間のサイズです。

8.2.1.2. 要件が満たされていることを確認する

ノードを設定し、Multus のパブリック NetworkAttachmentDefinition を作成した後 (ネットワークアタッチメント定義の作成 を参照)、ノード設定と NetworkAttachmentDefinition 設定に互換性があることを確認します。これを行うには、各ノードがパブリックネットワーク経由で Pod に **ping** できることを検証します。

次の例のような daemonset を起動します。

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
 name: multus-public-test
 namespace: openshift-storage
 labels:
   app: multus-public-test
spec:
 selector:
   matchLabels:
     app: multus-public-test
 template:
   metadata:
     labels:
       app: multus-public-test
     annotations:
       k8s.v1.cni.cncf.io/networks: openshift-storage/public-net #
   spec:
     containers:
       - name: test
         image: quay.io/ceph/ceph:v18 # image known to have ‘ping’ installed
         command:
```

```
      - sleep
      - infinity
    resources: {}
```

以下のようなコマンドを使用して、テスト Pod に割り当てられた Multus パブリックネットワーク IP をリスト表示します。この例のコマンドは、すべてのテスト Pod に割り当てられたすべての IP を短時間でリスト表示します (各 Pod に 2 つの IP があります)。出力から、Multus パブリックネットワークに関連付けられた IP を手動で簡単に抽出できます。

```
$ oc -n openshift-storage describe pod -l app=multus-public-test | grep -o -E 'Add .* from .*'
Add eth0 [10.128.2.86/23] from ovn-kubernetes
Add net1 [192.168.20.22/24] from default/public-net
Add eth0 [10.129.2.173/23] from ovn-kubernetes
Add net1 [192.168.20.29/24] from default/public-net
Add eth0 [10.131.0.108/23] from ovn-kubernetes
Add net1 [192.168.20.23/24] from default/public-net
```

上記の例では、Multus パブリックネットワーク上のテスト Pod の IP は次のとおりです。

- 192.168.20.22
- 192.168.20.29
- 192.168.20.23

各ノード (NODE) がパブリックネットワーク経由ですべてのテスト Pod IP にアクセスできることを確認します。

```
$ oc debug node/NODE
Starting pod/NODE-debug ...
To use host binaries, run `chroot /host`
Pod IP: ****
If you don't see a command prompt, try pressing enter.


sh-5.1# chroot /host


sh-5.1# ping 192.168.20.22
PING 192.168.20.22 (192.168.20.22) 56(84) bytes of data.
64 bytes from 192.168.20.22: icmp_seq=1 ttl=64 time=0.093 ms
64 bytes from 192.168.20.22: icmp_seq=2 ttl=64 time=0.056 ms
^C
--- 192.168.20.22 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1046ms
rtt min/avg/max/mdev = 0.056/0.074/0.093/0.018 ms


sh-5.1# ping 192.168.20.29
PING 192.168.20.29 (192.168.20.29) 56(84) bytes of data.
64 bytes from 192.168.20.29: icmp_seq=1 ttl=64 time=0.403 ms
64 bytes from 192.168.20.29: icmp_seq=2 ttl=64 time=0.181 ms
^C
--- 192.168.20.29 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1007ms
rtt min/avg/max/mdev = 0.181/0.292/0.403/0.111 ms
```

```
sh-5.1# ping 192.168.20.23
PING 192.168.20.23 (192.168.20.23) 56(84) bytes of data.
64 bytes from 192.168.20.23: icmp_seq=1 ttl=64 time=0.329 ms
64 bytes from 192.168.20.23: icmp_seq=2 ttl=64 time=0.227 ms
^C
--- 192.168.20.23 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1047ms
rtt min/avg/max/mdev = 0.227/0.278/0.329/0.051 ms
```

いずれかのノードが実行中の Pod への ping に成功しない場合、次に進むのは安全ではありません。問題を診断して修正し、このテストを繰り返してください。ここでは、さまざまな理由で問題が発生する可能性があります。理由としては次のようなものが挙げられます。

- ホストが Multus パブリックネットワークに (Macvlan 経由で) 正しくアタッチされていない可能性がある
- ホストが Pod の IP 範囲にルーティングするように適切に設定されていない可能性がある
- パブリック NetworkAttachmentDefinition がホストの IP 範囲にルーティングするように適切に設定されていない可能性がある
- ホストのファイアウォールルールが、どちらかの方向の接続をブロックしている可能性がある
- ネットワークスイッチのファイアウォールまたはセキュリティールールが、接続をブロックしている可能性がある

推奨されるデバッグ手順:

- パブリックネットワークの “shim” IP を使用してノードが相互に ping を実行できることを確認する
- **ip address** の出力を確認する

8.2.2. Multus の例

このクラスターに関連するネットワーク計画は、次のようなものになります。

- 専用の NIC が Multus パブリックネットワークに eth0 を提供します。
- Macvlan を使用して OpenShift Pod を eth0 にアタッチします。
- この例のクラスターでは、IP 範囲 192.168.0.0/16 が空いています。この IP 範囲を、Pod とノードが Multus パブリックネットワーク上で共有します。
- ノードは IP 範囲 192.168.252.0/22 を取得します (これにより、この例の組織が必要とする数よりも多い、最大 1024 個の Kubernetes ホストを使用できます)。
- Pod は残りの範囲 (192.168.0.1 から 192.168.251.255) を取得します。
- この例の組織では、必要がない限り DHCP を使用しません。そのため、ノードには、NMState Operator の NodeNetworkConfigurationPolicy リソースを使用して静的に割り当てられる Multus ネットワーク上の IP (eth0 経由) が与えられます。
- DHCP を使用できないため、すぐに使用できる Whereabouts を使用して、Multus パブリックネットワークに IP を割り当てます。

- OpenShift クラスターには、3 つのコンピュートノード (compute-0、compute-1、compute-2) があります。このクラスター上で OpenShift Data Foundation も実行されます。

ノードのネットワークポリシーは、Multus パブリックネットワーク上の Pod にルーティングするように設定する必要があります。

Pod は Macvlan 経由で接続され、Macvlan ではホストと Pod が相互にルーティングできないため、ホストも Macvlan 経由で接続する必要があります。一般的に、ホストは Pod と同じテクノロジーを使用して Multus パブリックネットワークに接続する必要があります (Pod 接続は Network Attachment Definition で設定されることに注意してください)。

ホストの IP 範囲は、範囲全体のサブセットであるため、ホストは IP 割り当てだけでは Pod にルーティングできません。ホストが 192.168.0.0/16 の範囲全体にルーティングできるようにするには、ホストにルートを追加する必要があります。

NodeNetworkConfigurationPolicy の **desiredState** 仕様は次のようになります。

```
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
 name: ceph-public-net-shim-compute-0
 namespace: openshift-storage
spec:
 nodeSelector:
   node-role.kubernetes.io/worker: ""
   kubernetes.io/hostname: compute-0
 desiredState:
   interfaces:
     - name: odf-pub-shim
       description: Shim interface used to connect host to OpenShift Data Foundation public Multus
network
       type: mac-vlan
       state: up
       mac-vlan:
         base-iface: eth0
         mode: bridge
         promiscuous: true
       ipv4:
         enabled: true
         dhcp: false
         address:
           - ip: 192.168.252.1 # STATIC IP FOR compute-0
             prefix-length: 22
   routes:
     config:
       - destination: 192.168.0.0/16
         next-hop-interface: odf-pub-shim
---
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
 name: ceph-public-net-shim-compute-1
 namespace: openshift-storage
spec:
 nodeSelector:
   node-role.kubernetes.io/worker: ""
```

```
    kubernetes.io/hostname: compute-1
  desiredState:
    interfaces:
      - name: odf-pub-shim
        description: Shim interface used to connect host to OpenShift Data Foundation public Multus
network
        type: mac-vlan
        state: up
        mac-vlan:
          base-iface: eth0
          mode: bridge
          promiscuous: true
        ipv4:
          enabled: true
          dhcp: false
          address:
            - ip: 192.168.252.1 # STATIC IP FOR compute-1
              prefix-length: 22
    routes:
      config:
        - destination: 192.168.0.0/16
          next-hop-interface: odf-pub-shim
---
apiVersion: nmstate.io/v1
kind: NodeNetworkConfigurationPolicy
metadata:
  name: ceph-public-net-shim-compute-2 # [1]
  namespace: openshift-storage
spec:
  nodeSelector:
    node-role.kubernetes.io/worker: ""
    kubernetes.io/hostname: compute-2 # [2]
  desiredState:
    Interfaces: [3]
      - name: odf-pub-shim
        description: Shim interface used to connect host to OpenShift Data Foundation public Multus
network
        type: mac-vlan # [4]
        state: up
        mac-vlan:
          base-iface: eth0 # [5]
          mode: bridge
          promiscuous: true
        ipv4: # [6]
          enabled: true
          dhcp: false
          address:
            - ip: 192.168.252.2 # STATIC IP FOR compute-2 # [7]
              prefix-length: 22
    routes: # [8]
      config:
        - destination: 192.168.0.0/16 # [9]
          next-hop-interface: odf-pub-shim
```

1. 静的 IP 管理の場合、各ノードに異なる NodeNetworkConfigurationPolicy が必要です。

2. 静的ネットワークを設定するには、ポリシーごとに個別のノードを選択します。

3. “shim” インターフェイスは、Network Attachment Definition が使用するのと同じテクノロジーを使用して、ホストを Multus パブリックネットワークに接続するために使用されます。

4. ホストの “shim” は、Pod 用に計画したものと同じタイプ (この例では **macvlan**) である必要があります。

5. インターフェイスは、計画時に選択した Multus パブリックネットワークインターフェイス (この例では **eth0**) と同じである必要があります。

6. **ipv4** (または **ipv6**) セクションでは、Multus パブリックネットワーク上のノード IP アドレスを設定します。

7. このノードの shim に割り当てられる IP が、計画と同じである必要があります。この例では、Multus パブリックネットワーク上のノード IP として 192.168.252.0/22 を使用します。

8. 静的 IP 管理の場合は、必ず各ノードの IP を変更してください。

9. **routes** セクションでは、Multus パブリックネットワーク上の Pod に到達する方法をノードに指示します。

10. ルートの宛先は、Pod 用に計画した CIDR 範囲と同じである必要があります。この場合、192.168.0.0/16 の範囲全体を使用しても、ノードが “shim” インターフェイスを介して他のノードに到達する能力に影響しないため、安全です。通常、これは Multus パブリック NetworkAttachmentDefinition で使用される CIDR と一致する必要があります。

パブリックネットワークの NetworkAttachmentDefinition は、次のようになります。ここでは、**range** 要求を簡素化するために、Whereabouts の **exclude** オプションを使用しています。Whereabouts の **routes[].dst** オプションにより、Pod が Multus パブリックネットワーク経由でホストにルーティングされるようになります。

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
 name: public-net
 namespace: openshift-storage
spec:
 config: '{
    "cniVersion": "0.3.1",
    "type": "macvlan", # [1]
    "master": "eth0", # [2]
    "mode": "bridge",
    "ipam": {
      "type": "whereabouts", # [3]
      "range": "192.168.0.0/16", # [4]
      "exclude": [
         "192.168.252.0/22" # [5]
      ],
      "routes": [             # [6]
        {"dst": "192.168.252.0/22"} # [7]
      ]
    }
  }'
```

1. これは、Pod を Multus パブリックネットワークにアタッチする方法の計画と同じである必要があります。ノードは同じテクノロジー (Macvlan) を使用してアタッチする必要があります。
2. インターフェイスは、計画時に選択した Multus パブリックネットワークインターフェイス (この例では **eth0**) と同じである必要があります。
3. この例の計画では、Pod に IP を割り当てるために、DHCP ではなく Whereabouts を使用します。
4. この例では、ノードに割り当てられる一部の範囲を除いて (5 を参照)、192.168.0.0/16 の範囲内の任意の IP を Pod に割り当てることができると決定しました。
5. Whereabouts は、ノードに割り当てられる範囲をプールから簡単に除外できる **exclude** ディレクティブを提供します。これにより、**range** ディレクティブ (4 を参照) を簡素化できます。
6. **routes** セクションでは、Multus パブリックネットワーク上のノードに到達する方法を Pod に指示します。
7. ルートの宛先 (**dst**) は、ノード用に計画した CIDR 範囲と同じである必要があります。

8.2.3. ホルダー Pod の非推奨化

アップグレード時にホルダー Pod のメンテナンスの影響が繰り返し発生するため (ホルダー Pod は Multus が有効な場合に存在します)、ホルダー Pod は ODF v4.16 リリースで非推奨となり、ODF v4.17 リリースで削除される予定です。この非推奨化に伴い、ホルダー Pod を削除する前に、追加のネットワーク設定アクションを完了する必要があります。ODF v4.15 の Multus が有効になっているクラスターを、標準のアップグレード手順に従って v4.16 にアップグレードします。ODF クラスター (Multus が有効) を v4.16 に正常にアップグレードした後、管理者は記事 Disabling Multus holder pods に記載されている手順を完了して、ホルダー Pod を無効化および削除する必要があります。この無効化手順は時間がかかることに注意してください。ただし、v4.16 にアップグレードした直後にすべての手順を完了する必要はありません。ODF を v4.17 にアップグレードする前に、このプロセスを完了することが重要です。

8.2.4. Multus を使用したストレージトラフィックの分離

デフォルトで、Red Hat OpenShift Data Foundation は Red Hat OpenShift Software Defined Network (SDN) を使用するように設定されています。デフォルトの SDN には、以下のトラフィックタイプがあります。

- Pod 間のトラフィック
- Pod からストレージへのトラフィック (ストレージが OpenShift Data Foundation の場合はパブリックネットワークトラフィックと呼ばれます)
- OpenShift Data Foundation の内部レプリケーションおよびリバランストラフィック (クラスターネットワークトラフィックと呼ばれます)

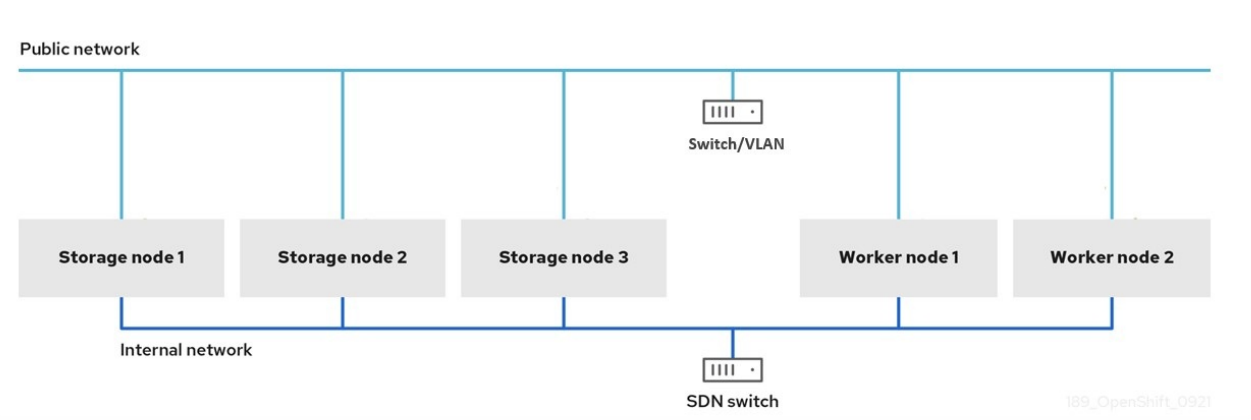
OpenShift Data Foundation を OpenShift のデフォルトネットワークから分離する方法は 3 つあります。

1. OpenShift Data Foundation のパブリックネットワーク用にホスト上でネットワークインターフェイスを予約する
    - Pod からストレージへのトラフィックと内部ストレージのレプリケーショントラフィックは、Pod 間のネットワークトラフィックから分離されたネットワーク上に共存します。

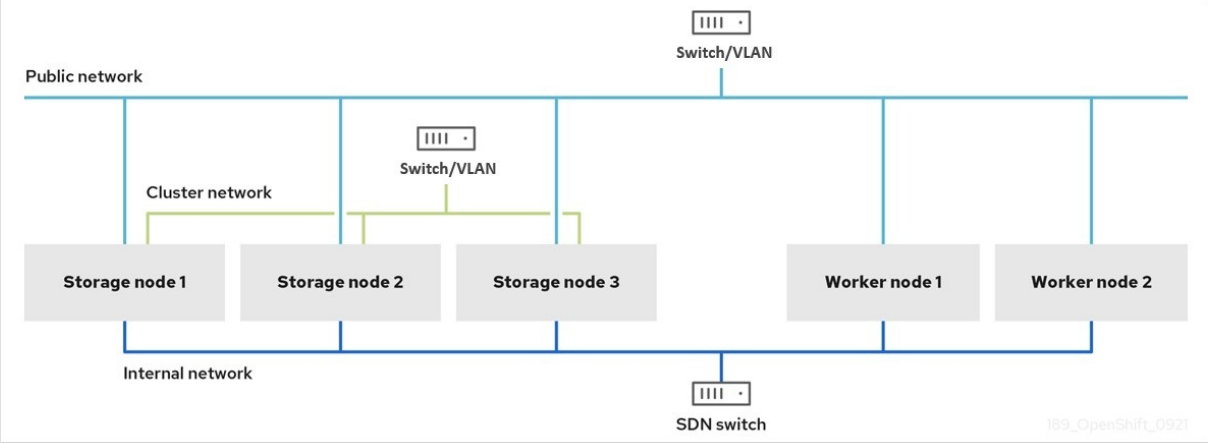
- OpenShift Data Foundation クラスターが正常な場合、アプリケーション Pod は最大のパブリックネットワークストレージ帯域幅にアクセスできます。
- ただし、OpenShift Data Foundation クラスターが障害から回復している場合、進行中のレプリケーションとトラフィックの再バランスにより、アプリケーション Pod の帯域幅が減少します。

2. OpenShift Data Foundation のクラスターネットワーク用にホスト上のネットワークインターフェイスを予約します。
   - Pod 間のトラフィックと Pod からストレージへのトラフィックはどちらも OpenShift のデフォルトネットワークを使用し続けます。
   - Pod からストレージまでの帯域幅は、OpenShift Data Foundation クラスターの正常性の影響をあまり受けません。
   - Pod 間および Pod からストレージへの OpenShift Data Foundation トラフィックは、OpenShift クラスターがビジーな場合に、ネットワーク帯域幅をめぐって競合する可能性があります。
   - ストレージの内部ネットワークには、障害時の使用のために予約された、未使用の帯域幅が過剰に存在することがよくあります。
3. OpenShift Data Foundation 用にホスト上で 2 つのネットワークインターフェイス (1 つはパブリックネットワーク用で、もう 1 つはクラスターネットワーク用) を予約します。
   - Pod から Pod、Pod からストレージ、ストレージの内部トラフィックはすべて分離されており、どのトラフィックタイプもリソースをめぐって競合することはありません。
   - すべてのトラフィックタイプのサービスレベルアグリーメントを確保できます。
   - 正常な実行時には、より多くのネットワーク帯域幅が予約されますが、3 つのネットワークすべてで使用されません。

デュアルネットワークインターフェイスの分離設定の概略例:



デュアルネットワークインターフェイスの分離設定の概略例:



8.2.5. Multus を使用する場合

以下が必要な場合は、OpenShift Data Foundation に Multus を使用します。

遅延の改善 - ODF を使用した Multus は常に遅延を改善します。ホストインターフェイスをホストネットワークに近い速度で使用し、OpenShift のソフトウェア定義の Pod ネットワークをバイパスします。各インターフェイスのインターフェイスレベルごとの Linux チューニングを実行することもできます。

帯域幅の向上 - OpenShift Data Foundation クライアントデータトラフィックと内部データトラフィックの専用インターフェイス。これらの専用インターフェイスは、完全な帯域幅を予約します。

セキュリティーの向上 - Multus は、ストレージネットワークトラフィックをアプリケーションネットワークトラフィックから分離して、セキュリティーを強化します。ネットワークがインターフェイスを共有している場合、帯域幅またはパフォーマンスが分離されない場合がありますが、QoS またはトラフィックシェーピングを使用して、共有インターフェイス上の帯域幅に優先順位を付けることができます。

8.2.6. Multus 設定

Multus を使用するには、OpenShift Data Foundation クラスターをデプロイする前にネットワーク接続定義 (NAD) を作成する必要があります。これは後でクラスターに接続されます。詳細は、ネットワークアタッチメント定義の作成 を参照してください。

追加のネットワークを Pod に割り当てるには、インターフェイスの割り当て方法を定義する設定を作成する必要があります。それぞれのインターフェイスは、**NetworkAttachmentDefinition** カスタムリソース (CR) を使用して指定します。これらの各 CR 内のコンテナーネットワークインターフェイス (CNI) 設定は、対象のインターフェイスの作成方法を定義します。

OpenShift Data Foundation は、次の機能を含む **macvlan** ドライバーをサポートしています。

- 各接続は、独自の MAC アドレスを持つ親インターフェイスのサブインターフェイスを取得し、ホストネットワークから分離されます。

- Linux ブリッジや **ipvlan** よりも CPU の使用量が少なく、スループットが向上します。
- ほとんどの場合、ブリッジモードが最適な選択肢です。
- ネットワークインターフェイスカード (NIC) がハードウェアで仮想ポート/仮想ローカルエリアネットワーク (VLAN) をサポートする場合の、ホストに近いパフォーマンス。

OpenShift Data Foundation は、次の 2 つのタイプの IP アドレス管理をサポートします。

whereabouts	DHCP
OpenShift/Kubernetes **リース** を使用して、Pod ごとに一意の IP アドレスを選択します。	**range** フィールドは必要ありません。
Pod に IP を提供するために DHCP サーバーを必要としません。	ネットワーク DHCP サーバーは、Multus Pod および同じネットワーク上の他のホストに同じ範囲を割り当てることができます。

注意

DHCP サーバーがある場合は、ネットワーク上の複数の MAC アドレスに同じ IP が割り当てられないように、Multus で設定された IPAM により、同じ範囲が設定されないようにします。

8.2.7. Multus 設定の要件

前提条件

- パブリックネットワークに使用されるインターフェイスは、各 OpenShift ストレージノードとワーカーノードで同じインターフェイス名を持つ必要があり、インターフェイスはすべて同じ基盤ネットワークに接続されている必要があります。
- クラスターネットワークに使用されるインターフェイスは、各 OpenShift ストレージノード上で同じインターフェイス名を持つ必要があり、インターフェイスはすべて同じ基盤ネットワークに接続されている必要があります。クラスターネットワークインターフェイスは OpenShift ワーカーノード上に存在する必要はありません。
- パブリックネットワークまたはクラスターネットワークに使用される各ネットワークインターフェイスは、少なくとも 10 ギガビットのネットワーク速度に対応できる必要があります。
- 各ネットワークには、個別の仮想ローカルエリアネットワーク (VLAN) またはサブネットが必要です。

ベアメタルで Multus ベースの設定に必要な手順については、Multus ネットワークの作成 を参照してください。

第9章 障害復旧

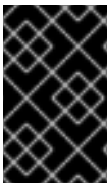
障害復旧 (DR) は、中断または障害が発生する場合に、組織がビジネスクリティカルな機能または通常の運用を回復し、再開するのに役立ちます。OpenShift Data Foundation は、ステートフルアプリに高可用性 (HA) および DR ソリューションを提供します。これらのソリューションは、大きく 2 つのカテゴリーに分類されます。

- **Metro-DR**: データ損失のない単一リージョンおよびクロスデータセンターの保護
- **Regional-DR** データ損失の可能性を最小限に抑えたクロスリージョン保護。
- **ストレッチクラスターを使用する Disaster Recovery**: 単一の OpenShift Data Foundation クラスターが 2 つの異なる場所間でストレッチされ、ストレージインフラストラクチャーにディザスターリカバリー機能を提供します。

9.1. METRO-DR

Metropolitan disaster recovery (Metro-DR) は、Red Hat Advanced Cluster Management for Kubernetes (RHACM)、Red Hat Ceph Storage、OpenShift Data Foundation コンポーネントで構成されており、OpenShift Container Platform クラスター全体のアプリケーションとデータのモビリティーを実現します。

このリリースの Metro-DR ソリューションは、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、これらはアベイラビリティーゾーンの障害からの保護に似ています。Metro-DR は、データセンターが利用できない場合でも、データを失うことなくビジネスの継続性を保証します。このソリューションには、Red Hat Advanced Cluster Management (RHACM) と OpenShift Data Foundation Advanced SKU および関連するバンドルが含まれています。



重要

OpenShift Data Foundation を使用して、OpenShift 仮想化テクノロジーに基づいたワークロード用の Metro 災害復旧ソリューションを簡単に設定できるようになりました。詳細は、ナレッジベースの記事 を参照してください。

前提条件

- 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をすべて満たす必要があります。
  - 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の仕組みを確認するには、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。
- プライマリー管理対象クラスター (Site-1) はアクティブ RHACM ハブクラスターと共存し、パッシブハブクラスターはセカンダリー管理対象クラスター (Site-2) とともに配置していることを確認します。あるいは、アクティブな RHACM ハブクラスターを、Site-1 のプライマリー管理対象クラスターまたは Site-2 のセカンダリークラスターの障害の影響を受けない中立サイト (サイト 3) に配置することもできます。このような状況では、パッシブハブクラスターを使用する場合は、Site-2 のセカンダリークラスターと一緒に配置できます。

注記

Metro-DR のハブリカバリーはテクノロジープレビュー機能であり、テクノロジープレビューのサポート制限の対象となります。

詳細なソリューション要件については、Metro-DR の要件、arbiter を使用して Red Hat Ceph Storage ストレッチクラスターをデプロイするための要件、および RHACM の要件 を参照してください。

9.2. REGIONAL-DR

Regional disaster recovery (Regional-DR) は、Red Hat Advanced Cluster Management for Kubernetes (RHACM) と OpenShift Data Foundation コンポーネントで構成されており、OpenShift Container Platform クラスター全体のアプリケーションとデータのモビリティーを実現します。非同期データレプリケーションに基づいて構築されているため、データが失われる可能性がありますが、さまざまな障害に対する保護を提供します。

Red Hat OpenShift Data Foundation は、ストレージプロバイダーとして Ceph に支えられており、そのライフサイクルは Rook によって管理されており、次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
- 永続ボリューム要求のミラーリングごとに管理する csi アドオンを提供する

このリリースの Regional-DR は、さまざまなリージョンおよびデータセンターにデプロイメントされるマルチクラスター設定をサポートします。たとえば、2 つの異なるリージョンまたはデータセンターにある 2 つのマネージドクラスターでの 2 方向のレプリケーションをサポートします。このソリューションには、Red Hat Advanced Cluster Management (RHACM) と OpenShift Data Foundation Advanced SKU および関連するバンドルが含まれています。

重要

OpenShift Data Foundation を使用して、OpenShift 仮想化テクノロジーに基づいたワークロード用の Regional disaster recovery ソリューションを簡単に設定できるようになりました。詳細は、ナレッジベースの記事 を参照してください。

前提条件

- 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をすべて満たす必要があります。
  - 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の仕組みを確認するには、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。
- プライマリー管理対象クラスター (Site-1) はアクティブ RHACM ハブクラスターと共存し、パッシブハブクラスターはセカンダリー管理対象クラスター (Site-2) とともに配置していることを確認します。あるいは、アクティブな RHACM ハブクラスターを、Site-1 のプライマリー管理対象クラスターまたは Site-2 のセカンダリークラスターの障害の影響を受けない中立サイト (サイト 3) に配置することもできます。このような状況では、パッシブハブクラスターを使用する場合は、Site-2 のセカンダリークラスターと一緒に配置できます。

詳細なソリューション要件は、Regional-DR 要件 および RHACM 要件 を参照してください。

9.3. ストレッチクラスターを使用した障害復旧

この例では、3 番目のゾーンを Arbiter の場所とした上で、単一クラスターが 2 つのゾーンにデプロイメントされます。この機能は現在、オンプレミスおよび同じ場所にある OpenShift Container Platform へのデプロイメントを目的としています。このソリューションは、複数のデータセンターにわたるデプロイメントには推奨できません。代わりに、複数のデータセンターにデプロイされており、ネットワークのレイテンシーが低く、データ損失がない DR ソリューションの 1 番のオプションとして、Metro-DR を検討してください。

注記

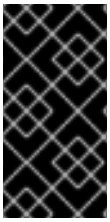
ストレッチクラスターソリューションは、データボリュームを含むゾーン間の遅延が 10 ミリ秒の最大ラウンドトリップ時間 (RTT) を超えないデプロイメント向けに設計されています。Arbiter ノードは、etcd に指定されたレイテンシー要件に従います。詳細は、Guidance for Red Hat OpenShift Container Platform Clusters - Deployments Spanning Multiple Sites (Data Centers/Regions) を参照してください。より高いレイテンシーでデプロイする予定がある場合は、Red Hat カスタマーサポート にお問い合わせください。

ストレッチクラスターを使用するには、以下を実行します。

- 3 つのゾーンには、最低でも 5 つのノードが必要です。ここでは、以下のようになります。
  - データセンターゾーンごとに 2 つのノードが使用され、arbiter ゾーンにはノードが 1 つ含まれる、追加ゾーンが 1 つ使用されます (arbiter はマスターノード上にある場合があります)。
- すべてのノードには、クラスターの作成前にゾーンのラベルを手動で付ける必要があります。たとえば、ゾーンには以下のようにラベル付けできます。
  - **topology.kubernetes.io/zone=arbiter** (マスターまたはワーカーノード)
  - **topology.kubernetes.io/zone=datacenter1** (2 つ以上のワーカーノード)
  - **topology.kubernetes.io/zone=datacenter2** (2 つ以上のワーカーノード)

詳細は、ストレッチクラスター用の OpenShift Data Foundation の設定 を参照してください。

OpenShift Data Foundation のサブスクリプションの仕組みを確認するには、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。



重要

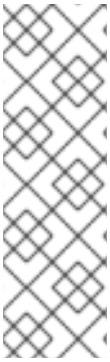
OpenShift Data Foundation を使用して、OpenShift 仮想化テクノロジーに基づいたワークロード用のストレッチクラスターによる障害復旧を簡単に設定できるようになりました。詳細は、OpenShift Container Platform ガイドの OpenShift Virtualization を参照してください。

第10章 非接続環境

非接続環境は、Operator Lifecycle Manager (OLM) がインターネット接続が必要なデフォルトの Operator Hub およびイメージレジストリーにアクセスできないネットワークが制限された環境です。

Red Hat は、OpenShift Container Platform がネットワークが制限された環境にインストールされた非接続環境での OpenShift Data Foundation のデプロイメントをサポートします。

切断された環境に OpenShift Data Foundation をインストールするには、OpenShift Container Platform のドキュメントの Operator ガイド の **制限付きネットワークでの Operator Lifecycle Manager の使用**を参照してください。



注記

OpenShift Data Foundation をネットワークが制限された環境でインストールする場合は、デフォルトでインターネット接続が OpenShift Container Platform で想定され、**chronyd** が ***.rhel.pool.ntp.org** サーバーを使用するように設定されるため、カスタム Network Time Protocol (NTP) 設定をノードに適用します。

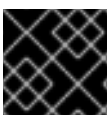
詳細は、Red Hat ナレッジベースソリューションの A newly deployed OCS 4 cluster status shows as "Degraded", Why?、および、OpenShift Container Platform ドキュメントの インストールガイド の **chrony の時間サービスの設定** を参照してください。

Red Hat OpenShift Data Foundation バージョン 4.12 では、非接続環境のデプロイメント用にエージェントベースのインストーラーが導入されています。エージェントベースのインストーラーを使用すると、非接続インストールにミラーレジストリーを使用できます。詳細は、エージェントベースのインストーラーによるインストールの準備 を参照してください。

OpenShift Data Foundation に含まれるパッケージ

redhat-operator インデックスイメージをプルーニングするときは、OpenShift Data Foundation デプロイメント用の以下のパッケージのリストを含めます。

- **ocs-operator**
- **odf-operator**
- **mcg-operator**
- **odf-csi-addons-operator**
- **odr-cluster-operator**
- **odr-hub-operator**
- オプション: **local-storage-operator**
  ローカルストレージデプロイメントの場合のみ。
- オプション: **odf-multicluster-orchestrator**
  Regional Disaster Recovery (Regional-DR) 設定の場合のみ。



重要

CatalogSource に **redhat-operators** という名前を付けます。

第11章 IBM POWER および IBM Z でサポートされている機能およびサポートされていない機能

表11.1 IBM Power および IBM Z でサポートされている機能およびサポートされていない機能の一覧

機能	IBM Power	IBM Z
コンパクトなデプロイメント	サポート対象外	サポート対象外
動的ストレージデバイス	サポート対象外	サポート対象
ストレッチクラスター - Arbiter	サポート対象	サポート対象外
Federal Information Processing Standard Publication (FIPS)	サポート対象外	サポート対象外
プール圧縮メトリクスを表示する機能	サポート対象	サポート対象外
Multicloud Object Gateway (MCG) エンドポイント Pod の自動スケーリング	サポート対象	サポート対象外
オーバープロビジョニングを制御するアラート	サポート対象	サポート対象外
Ceph Monitor がスペースを使い果たしたときにアラート	サポート対象	サポート対象外
IBM Flashsystem などのプラグ可能な外部ストレージを可能にする拡張 OpenShift Data Foundation コントロールプレーン	サポート対象外	サポート対象外
IPV6 サポート	サポート対象外	サポート対象外
Multus	サポート対象外	サポート対象外
Multicloud Object Gateway (MCG) バケットのレプリケーション	サポート対象	サポート対象外
オブジェクトデータのクォータサポート	サポート対象	サポート対象外
最小限のデプロイメント	サポート対象外	サポート対象外

機能	IBM Power	IBM Z
Red Hat Advanced Cluster Management (RHACM) を使用した Regional-Disaster Recovery (Regional-DR)	サポート対象	サポート対象外
RHACM を使用した Metro-Disaster Recovery (Metro-DR) の複数クラスター	サポート対象	サポート対象
無線アクセスネットワーク (RAN) のシングルノードソリューション	サポート対象外	サポート対象外
ネットワークファイルシステム (NFS) サービスのサポート	サポート対象	サポート対象外
Multicloud Object Gateway (MCG) アカウントの認証情報を変更する機能	サポート対象	サポート対象外
Red Hat Advanced Cluster Management コンソールでのマルチクラスター監視	サポート対象	サポート対象外
Multicloud Object Gateway ライフサイクルでの期限切れオブジェクトの削除	サポート対象	サポート対象外
OpenShift でサポートされているプラットフォームでの OpenShift Data Foundation の依存デプロイメント	サポート対象外	サポート対象外
ベアメタルインフラストラクチャーを使用した OpenShift Data Foundation のインストーラープロビジョニングデプロイメント	サポート対象外	サポート対象外
IPv4 を使用した OpenShift Data Foundation による Openshift デュアルスタック	サポート対象外	サポート対象外
デプロイメント時に Multicloud Object Gateway 外部サービスを無効にする機能	サポート対象外	サポート対象外
デフォルトの NooBaa バッキングストアの上書きを許可する機能	サポート対象	サポート対象外

機能	IBM Power	IBM Z
ocs-operator による 1 つのアクティブおよび 1 つのスタンバイ Pod の 2 つの MGR Pod のデプロイを許可する機能	サポート対象	サポート対象外
ブラウンフィールドデプロイメントのための障害復旧	サポート対象外	サポート対象
RGW の自動スケーリング	サポート対象外	サポート対象外

第12章 次のステップ

OpenShift Data Foundation のデプロイを開始するには、OpenShift Container Platform 内で内部モードを使用するか、外部モードを使用して OpenShift Container Platform の外部で実行されているクラスターからサービスを使用できるようにします。

要件に応じて、それぞれのデプロイメントガイドを参照します。

内部モード

- Amazon Web サービスを使用した OpenShift Data Foundation のデプロイ
- ベアメタルを使用した OpenShift Data Foundation のデプロイ
- VMWare vSphere を使用した OpenShift Data Foundation のデプロイ
- Microsoft Azure を使用した OpenShift Data Foundation のデプロイ
- Google Cloud を使用した OpenShift Data Foundation のデプロイ
- Red Hat OpenStack Platform を使用した OpenShift Data Foundation のデプロイ [テクノロジープレビュー]
- IBM Power での OpenShift Data Foundation のデプロイ
- IBM Z での OpenShift Data Foundation のデプロイ
- 任意のプラットフォームへの OpenShift Data Foundation のデプロイ

外部モード

- 外部モードでの OpenShift Data Foundation のデプロイ

内部または外部

複数のクラスターをデプロイする場合は、複数の OpenShift Data Foundation クラスターのデプロイ を参照してください。