



Red Hat OpenShift Container Storage 4.8

デプロイメントのプランニング

RHOCS 4.8 をデプロイする際の重要な考慮事項

Red Hat OpenShift Container Storage 4.8 デプロイメントのプランニング

RHOCS 4.8 をデプロイする際の重要な考慮事項

Enter your first name here. Enter your surname here.

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

Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Planning_your_deployment.ent file |.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本書は、Red Hat OpenShift Container Storage のデプロイメントを計画する際の重要な考慮事項について説明します。

目次

多様性を受け入れるオープンソースの強化	4
RED HAT ドキュメントへのフィードバック (英語のみ)	5
第1章 OPENSIFT CONTAINER STORAGE の紹介	6
第2章 OPENSIFT CONTAINER STORAGE のアーキテクチャー	7
2.1. OPERATOR	7
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	8
2.2.1. 内部アプローチ	8
2.2.2. 外部アプローチ	9
2.3. ノードのタイプ	9
第3章 内部ストレージサービス	11
第4章 外部ストレージサービス	12
第5章 セキュリティーに関する考慮事項	13
5.1. FIPS-140-2	13
5.2. プロキシ環境	13
5.3. データ暗号化オプション	13
5.3.1. クラスタ全体の暗号化	14
5.3.2. ストレージクラスの暗号化	14
第6章 サブスクリプション	15
6.1. サブスクリプションのオフアリング	15
6.2. 障害復旧サブスクリプション	15
6.3. コア対 VCPU およびハイパースレッディング	15
6.3.1. IBM Power Systems のコア対 vCPU および同時マルチスレッド (SMT)	15
6.4. コアの分割	16
6.4.1. IBM Power Systems の共有プロセッサプール	16
6.5. サブスクリプションの要件	16
第7章 インフラストラクチャーの要件	17
7.1. プラットフォーム要件	17
7.1.1. Amazon EC2	17
7.1.2. ベアメタル	17
7.1.3. VMware vSphere	17
7.1.4. Microsoft Azure	17
7.1.5. Google Cloud [テクノロジープレビュー]	18
7.1.6. Red Hat Virtualization プラットフォーム	18
7.1.7. Red Hat OpenStack Platform [テクノロジープレビュー]	18
7.1.8. IBM Power Systems	18
7.1.9. IBM Z および LinuxONE	18
7.2. 外部モード要件	18
7.2.1. Red Hat Ceph Storage	18
7.3. リソース要件	19
7.3.1. IBM Z および LinuxONE インフラストラクチャーのリソース要件	20
7.3.2. デプロイメントリソースの最小要件 [テクノロジープレビュー]	21
7.3.3. コンパクトなデプロイメントリソース要件	22
7.4. POD の配置ルール	23
7.5. ストレージデバイスの要件	23
7.5.1. 動的ストレージデバイス	23

7.5.2. ローカルストレージデバイス	23
7.5.3. 容量のプランニング	23
7.6. マルチネットワークプラグイン (MULTUS) のサポート	24
7.6.1. 複数ネットワークについて	25
7.6.1.1. 追加ネットワークの使用シナリオ	25
7.6.2. Multus を使用したストレージトラフィックの分離	25
7.6.3. Multus 設定の推奨ネットワーク設定および要件	26
第8章 障害復旧	28
第9章 非接続環境	30
第10章 IBM POWER SYSTEMS および IBM Z インフラストラクチャーでサポートされない機能	31
第11章 次のステップ	32

多様性を受け入れるオープンソースの強化

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

RED HAT ドキュメントへのフィードバック (英語のみ)

弊社のドキュメントについてのご意見をお聞かせください。ドキュメントの改善点があれば、ぜひお知らせください。フィードバックをお寄せいただくには、以下をご確認ください。

- 特定の部分についての簡単なコメントをお寄せいただく場合は、以下をご確認ください。
 1. ドキュメントの表示が **Multi-page HTML** 形式になっていることを確認してください。ドキュメントの右上隅に **Feedback** ボタンがあることを確認してください。
 2. マウスカーソルを使用して、コメントを追加するテキストの部分を強調表示します。
 3. 強調表示されたテキストの下に表示される **Add Feedback** ポップアップをクリックします。
 4. 表示される指示に従ってください。
- より詳細なフィードバックをお寄せいただく場合は、Bugzilla のチケットを作成してください。
 1. [Bugzilla](#) の Web サイトに移動します。
 2. **Component** セクションで、**documentation** を選択します。
 3. **Description** フィールドに、ドキュメントの改善に向けたご提案を記入してください。ドキュメントの該当部分へのリンクも追加してください。
 4. **Submit Bug** をクリックします。

第1章 OPENSIFT CONTAINER STORAGE の紹介

Red Hat OpenShift Container Storage 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および管理を容易に実行するために Operator としてパッケージ化され、Red Hat OpenShift Container Platform サービスカタログの一部として利用できます。

Red Hat OpenShift Container Storage 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。
- 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。

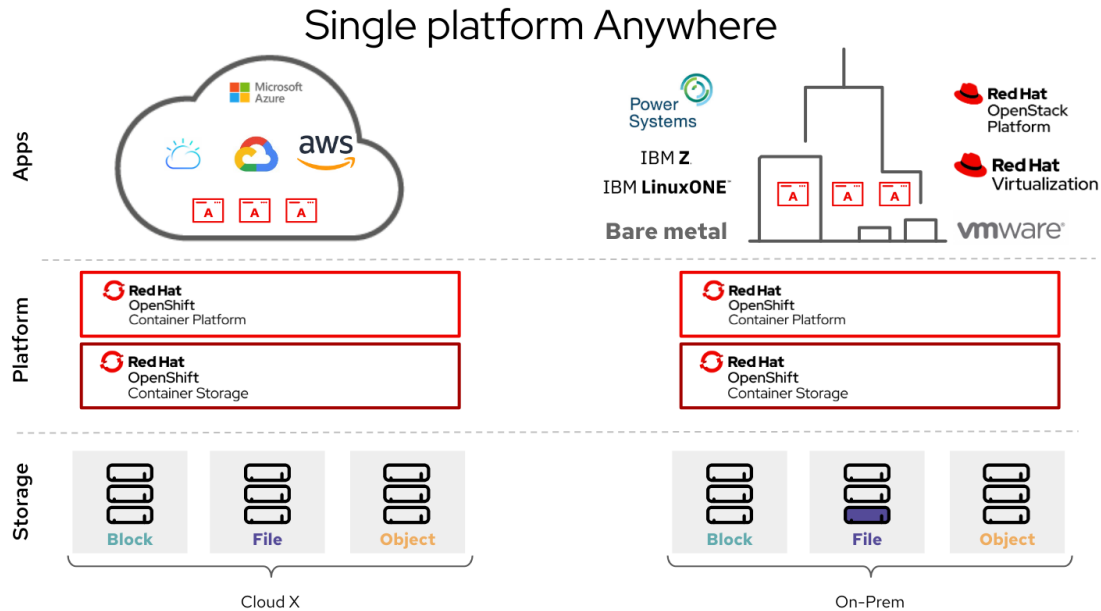
Red Hat OpenShift Container Storage バージョン 4.x は、以下を含むソフトウェアプロジェクトのコレクションを統合します。

- 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 Container Storage、Rook-Ceph、および NooBaa Operator。OpenShift Container Storage サービスを初期化し、管理します。

第2章 OPENSIFT CONTAINER STORAGE のアーキテクチャー

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

Red Hat OpenShift Container Storage アーキテクチャー



Red Hat OpenShift Container Storage 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つの方法については、[OpenShift Container Platform のインストールプロセス](#) について参照してください。Red Hat OpenShift Container Storage と Red Hat OpenShift Container Platform のコンポーネントの相互運用性についての詳細は、[相互運用性マトリックス](#) 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についての詳細は、[OpenShift Container Platform architecture](#) を参照してください。



注記

IBM Power Systems の場合は、[OpenShift Container Platform のインストールプロセス](#) について参照してください。

2.1. OPERATOR

Red Hat OpenShift Container Storage は主に3つの Operator で設定されており、管理タスクとカスタムリソースをコード化し、タスクおよびリソースの特性をより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 Container Storage 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、またはその状態に近づけるようにします。

OpenShift Container Storage Operator

他の Operator を特定のテストされた方法で利用してサポートされている Red Hat OpenShift Container Storage デプロイメントの推奨事項および要件を定め、実施するメタ Operator。この Operator は、Rook-Ceph および NooBaa Operator によって提供される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)
- オブジェクトゲートウェイ (RGW): オンプレミスの場合のみ

NooBaa Operator

この Operator は、Multicloud Object Gateway 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、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を作成します。

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

柔軟性は、複数モード (operating modalities) 一覧の拡張によって確認されるように Red Hat OpenShift Container Storage の中心的な特徴です。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。OpenShift Container Storage 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 Container Storage を Red Hat OpenShift Container Platform 内に完全にデプロイすると、Operator ベースのデプロイメントおよび管理による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で、内部でアタッチされているデバイスのアプローチを使用して、ローカルストレージ Operator とローカルストレージデバイスで、Red Hat OpenShift Container Storage を内部モードでデプロイできます。

Red Hat OpenShift Container Storage 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 Container Storage サービスは、Red Hat OpenShift Container Platform の Operator によって管理されるアプリケーションと共存する形で実行されます。

Simple (単純) デプロイメントは、以下の状況で最適です。

- ストレージ要件が明確ではない。
- OpenShift Container Storage サービスは、アプリケーションと共存する形で実行される。
- 特定サイズのノードインスタンスの作成が容易ではない (ベアメタル)。

Red Hat OpenShift Container Storage がアプリケーションと共存する形で実行されるには、EC2 の EBS ボリューム、VMware の vSphere Virtual Volume、または PowerVC によって動的にプロビジョニングされる SAN ボリュームのように、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がそれらに動的に接続される必要があります。

Optimized (最適化) デプロイメント

OpenShift Container Storage サービスは、Red Hat OpenShift Container Platform で管理される専用のインフラストラクチャーノードで実行されます。

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
- OpenShift Container Storage サービスを専用のインフラストラクチャーノードで実行する
- 特定サイズのノードインスタンスの作成が容易である (クラウド、仮想化環境など)

2.2.2. 外部アプローチ

Red Hat OpenShift Container Storage 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 (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)

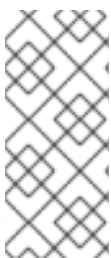
2.3. ノードのタイプ

ノードはコンテナランタイムとサービスを実行し、コンテナ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 Container Storage 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リングし、ノードの正常性および数を維持し、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。

ノードタイプ	説明
インフラストラクチャー (インフラ)	<p>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 クラスターでオプションになります。仮想化環境およびクラウド環境で OpenShift Container Storage 用に infra ノードを使用することが推奨されます。</p> <p>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することができます。 How to use dedicated worker nodes for Red Hat OpenShift Container Storage? を参照してください。</p>
ワーカー	<p>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。</p> <p>OpenShift Container Storage が内部モードでデプロイされる場合、3つのワーカーノードで設定される最小クラスターが必要になります。ここで、可用性を確保するためにノードは3つの別々のラックまたはアベイラビリティゾーンに分散することを推奨します。OpenShift Container Storage がワーカーノードで実行されるようにするには、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のいずれかがそれらに動的に接続される必要があります。</p> <p>外部モードでデプロイされると、複数のノードで実行され、障害の発生時に利用可能なノードでの K8S による再スケジューリングが可能になります。</p> <p>ポータブルストレージデバイスの例には、EC2 の EBS ボリューム、VMware の vSphere Virtual Volume、または PowerVC に動的にプロビジョニングされる SAN ボリュームがあります。</p>



注記

ストレージワークロードのみを実行するノードには、Red Hat OpenShift Container Storage のサブスクリプションが必要です。ストレージワークロードに加えて他のワークロードを実行するノードには、Red Hat OpenShift Container Storage および Red Hat OpenShift Container Platform サブスクリプションの両方が必要です。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 Container Storage 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
- ベアメタル
- VMware vSphere
- Microsoft Azure
- Google Cloud [テクノロジープレビュー]
- Red Hat Virtualization 4.4.x 以降 (IPI)
- Red Hat OpenStack 13 以降 (IPI) [テクノロジープレビュー]
- IBM Power Systems
- IBM Z および LinuxONE

デプロイメントおよび管理の容易性は、OpenShift Container Storage サービスを OpenShift Container Platform の内部で実行することについての主な特長となっています。内部クラスターリソースを作成すると、OpenShift Container Storage 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 Container Storage は、以下のプラットフォームで実行されている OpenShift Container Platform クラスタで外部 Red Hat Ceph Storage クラスタのサービスを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)

OpenShift Container Storage Operator は、外部サービスに対する永続ボリュームおよびオブジェクトバケット要求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。外部クラスタは Operator によってデプロイされず、管理されません。

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ついての一連のセキュリティー要件を定義する標準です。この標準は、米国の政府機関や請負業者に対して法律で義務付けられており、他の国際標準および業界固有の標準でも参照されています。

Red Hat OpenShift Container Storage は、Red Hat Enterprise Linux OS/CoreOS (RHCOS) によって配信される FIPS で検証済みの暗号モジュールを使用するようになりました。

現時点で、暗号モジュールは Cryptographic Module Validation Program (CMVP) によって処理され、それらの状態は [Modules in Process List](#) で確認できます。最新の情報は、[ナレッジベースのアーティクル](#) を参照してください。



注記

OpenShift Container Storage をインストールする前に、FIPS モードを OpenShift Container Platform で有効にする必要があります。OpenShift Container Platform は、この機能に対して RHEL 7 での OpenShift Container Storage デプロイメントがサポートされていないため、RHCOS ノードで実行される必要があります。

詳細は、[installing a cluster in FIPS mode](#) および [support for FIPS cryptography](#) を参照してください。

5.2. プロキシ環境

プロキシ環境は、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オブジェクトを変更するか、または新規クラスターについて `install-config.yaml` ファイルでプロキシを設定してプロキシを使用するように設定されます。

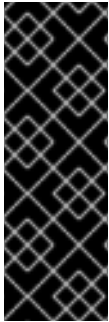
Red Hat では、OpenShift Container Platform が [クラスター全体のプリキシーの設定](#) に応じて設定されている場合に、プロキシ環境での OpenShift Container Storage バージョン 4.5 以上のデプロイメントをサポートします。

5.3. データ暗号化オプション

暗号化により、データをエンコードし、必要な暗号化キーなしに読み取れないようにすることができます。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。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、OpenShift Container Storage 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。

現時点で、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 Container Storage 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 Container Storage 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。



重要

- KMS は、永続ボリューム (PV) 暗号化に必要であり、クラスター全体の暗号化にはオプションです。
- Red Hat はテクノロジーパートナーと連携して、本書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。

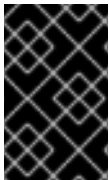
5.3.1. クラスター全体の暗号化

Red Hat OpenShift Container Storage 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 Container Storage 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。どちらのメソッドでも相互排他的であり、メソッド間の移行はできません。

暗号化はデフォルトでは無効にされます。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。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、OpenShift Container Storage 4.6 では鍵管理システム (KMS) を使用せずにサポートされますが、OpenShift Container Storage 4.7 では KMS の有無にかかわらずサポートされます。

現時点で、HashiCorp Vault は唯一サポートされている KMS です。OpenShift Container Storage 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 Container Storage 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。



重要

Red Hat はテクノロジーパートナーと連携して、本書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](#) について参照してください。

ストレージクラスの暗号化は OpenShift Container Storage 4.7 以降でサポートされます。

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リング

Red Hat OpenShift Container Storage 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 Container Storage 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 Container Storage 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 Systems 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 Container Storage 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

Red Hat OpenShift Container Storage は、障害復旧 (DR)、コールドバックアップその他のサブスクリプションタイプを提供しません。OpenShift Container Storage がインストールされたシステムでは、電源がオン/オフであるか、ワークロードを実行中かどうにかかわらず、アクティブなサブスクリプションが必要になります。

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ついての決定は、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関連する場合、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 Systems 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ついての決定は、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 Systems 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: $16 \text{ vCPU} / 8 = 2$)。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、1 コアのみを必要とするように計算されたシステムは、登録およびサブスクライブ後に 2 コアのサブスクリプションすべてを使用します。

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、[Cores versus vCPUs and hyperthreading](#) の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 Systems の共有プロセッサプール

IBM Power Systems には、共有プロセッサプールの概念があります。共有プロセッサプール内のプロセッサは、クラスター内のノード間で共有できます。OpenShift Container Storage 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

OpenShift Container Storage コンポーネントは、Red Hat CoreOS (RHCOS) または Red Hat Enterprise Linux (RHEL) 7 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。OpenShift Container Storage 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 Container Storage サブスクリプションが必要でなくても、OpenShift Container Storage 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を示すことができます。

詳細は、[Managing and Allocating Storage Resources](#) ガイドの [How to use dedicated worker nodes for Red Hat OpenShift Container Storage](#) 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 Container Storage 4.8 は、OpenShift Container Platform バージョン 4.8 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 OpenShift Container Storage 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、この [Red Hat ナレッジベースアール](#) 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のバージョンについての詳細の一覧は、[Red Hat OpenShift Container Storage and Red Hat OpenShift Container Platform interoperability matrix](#) を参照してください。

7.1.1. Amazon EC2

内部 Red Hat OpenShift Container Storage 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の要件](#) を満たし、以下を提供するストレージクラスがなければなりません。

- EBS ストレージ (aws-ebs プロビジョナー経由)

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、ローカルストレージ 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がなければなりません。

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、Update 2

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。

さらに、内部クラスターは [ストレージデバイスの要件](#) を満たし、以下のいずれかを提供するストレージクラスがなければなりません。

- 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

内部 Red Hat OpenShift Container Storage 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の要件](#) を満たし、以下を提供するストレージクラスがなければなりません。

- Azure ディスク (azure-disk プロビジョナー経由)

7.1.5. Google Cloud [テクノロジープレビュー]

内部 Red Hat OpenShift Container Storage 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の要件](#) を満たし、以下を提供するストレージクラスがなければなりません。

- GCE Persistent Disk (gce-pd プロビジョナー経由)

7.1.6. Red Hat Virtualization プラットフォーム

内部 Red Hat OpenShift Container Storage 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の要件](#) を満たし、ローカルストレージ 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がなければなりません。

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

内部 Red Hat OpenShift Container Storage 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、以下を提供するストレージクラスがなければなりません。

- Cinder プロビジョナーを使用した標準ディスク

7.1.8. IBM Power Systems

内部 Red Hat OpenShift Container Storage 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の要件](#) を満たし、ローカルストレージ 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がなければなりません。

7.1.9. IBM Z および LinuxONE

内部 Red Hat OpenShift Container Storage 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の要件](#) を満たし、ローカルストレージ 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がなければなりません。

7.2. 外部モード要件

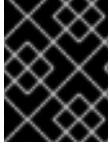
7.2.1. Red Hat Ceph Storage

Red Hat Ceph Storage (RHCS) バージョン 4.2z1 以降が必要です。サポートされるバージョンについては、この [Red Hat Ceph Storage リリースおよび対応する Ceph パッケージバージョンについてのナレッジベースのアーティクル](#) を参照してください。

RHCS 4 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。

7.3. リソース要件

OpenShift Container Storage サービスは、ベースサービスの初期セットで設定され、追加のデバイスセットで拡張できます。これらすべての OpenShift Container Storage サービス Pod は、[リソース要件](#) に応じて OpenShift Container Platform ノードの 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 Container Storage 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、オペレーター、またはワークロードには関連しません。

表7.1 OpenShift Container Storage のみの使用可能なリソース要件を集約します。

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	<ul style="list-style-type: none"> ● 30 個の CPU (論理) ● 72 GiB メモリー ● 3 ストレージデバイス 	<ul style="list-style-type: none"> ● 6 個の CPU (論理) ● 15 GiB メモリー ● 3 ストレージデバイス
外部	<div style="display: flex; align-items: center;"> <div> <p>重要</p> <p>外部 RedHat Ceph Storage クラスタは、通常の Ceph サイジングルールを使用します。詳細は、ハードウェアガイド を参照してください。</p> </div> </div> <ul style="list-style-type: none"> ● 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 $3 \times 10 = 30$ 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。

OpenShift Container Storage クラスタの設計に関する追加のガイダンスは、[OCS 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。
- OpenShift Container Storage 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 Systems 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	<ul style="list-style-type: none"> ● 48 CPU (論理) ● 192 GB 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	<ul style="list-style-type: none"> ● 該当なし

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 $3 \times 16 = 48$ ユニットの CPU、および $3 \times 64 = 192$ GB が必要です。

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

OpenShift Container Storage サービスは、ベースサービスの初期セットで設定され、追加のデバイスセットで拡張できます。

これらすべての OpenShift Container Storage サービス Pod は、[リソース要件](#) に応じて OpenShift Container Platform ノードの 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 OpenShift Container Storage のみ (IBM Z および LinuxONE) の使用可能なリソース要件を集約します。

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 LinuxONE の最小ハードウェア要件
------------	---------	------------	--------------------------------

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 LinuxONE の最小ハードウェア要件
内部	<ul style="list-style-type: none"> ● 30 個の CPU (論理) <ul style="list-style-type: none"> ○ 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 	<ul style="list-style-type: none"> ● 6 個の CPU (論理) ● 15 GiB メモリー ● 3 ストレージデバイス 	1IFL
外部	<ul style="list-style-type: none"> ● 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)

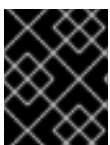
IBM Z および LinuxONE の物理コアです。

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 [テクノロジープレビュー]

OpenShift Container Storage 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 Container Storage 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、オペレーター、またはワークロードには関連しません。

表7.4 OpenShift Container Storage のみのリソース要件の集計

デプロイメントモード	ベースサービス
------------	---------

デプロイメントモード	ベースサービス
内部	<ul style="list-style-type: none"> ● 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。



重要

最小設定での OpenShift Container Storage のデプロイはテクノロジープレビュー機能です。テクノロジープレビュー機能は Red Hat の実稼働環境でのサービスレベルアグリーメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。Red Hat は実稼働環境でこれらを使用することを推奨していません。これらの機能は、近々発表予定の製品機能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。

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。

7.3.3. コンパクトなデプロイメントリソース要件

OpenShift Container Storage 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るようになりました。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 Container Storage 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、オペレーター、またはワークロードには関連しません。

表7.5 OpenShift Container Storage のみのリソース要件の集計

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	<ul style="list-style-type: none"> ● 24 個の CPU (論理) ● 72 GiB メモリー ● 3 ストレージデバイス 	<ul style="list-style-type: none"> ● 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 を設定する場合は、[Configuring a three-node cluster](#) および [Delivering a Three-node Architecture for Edge Deployments](#) を参照してください。

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 OpenShift Container Storage 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について説明します。通常、ノードごとに9以下のデバイスが推奨されます。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であり、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することができます。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。



注記

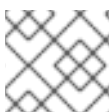
ストレージ容量は、インストール時に選択した容量の増分値でのみ拡張できます。

7.5.1. 動的ストレージデバイス

Red Hat OpenShift Container Storage 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、ノードサイズ、基礎となるプロビジョナー制限、および **リソース要件** の関数です。

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、4 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** の関数です。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。容量についての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) に達する場合、領域を解放するか、または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 カスタマーポータル](#) にお問い合わせください。

以下の表は、動的なストレージデバイスを持つ Red Hat OpenShift Container Storage のノードの設定例を示しています。

表7.6 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7.7 30 ノード (N) で拡張された設定の例

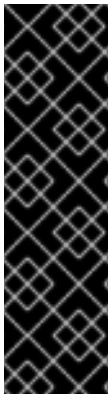
ストレージデバイスのサイズ (D)	ノードごとのストレージデバイス (M)	合計容量 (D * M * N)	使用可能なストレージ容量 (D*M*N/3)
0.5 TiB	3	45 TiB	15 TiB
2 TiB	6	360 TiB	120 TiB
4 TiB	9	1080 TiB	360 TiB

7.6. マルチネットワークプラグイン (MULTUS) のサポート

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

- Pod 対 Pod のトラフィック
- Pod 対 OpenShift Container Storage のトラフィック (OpenShift Container Storage のパブリックネットワークトラフィックとして知られる)
- OpenShift Container Storage クラスターネットワークトラフィックとして知られる OpenShift Container Storage のレプリケーションおよびリバランス

ただし、OpenShift Container Storage 4.8 以降では、テクノロジープレビューとして、さまざまな種類のネットワークトラフィックを分離することで、Multus を使用してセキュリティーやパフォーマンスを強化する機能がサポートされます。



重要

Multus のサポートはテクノロジープレビュー機能としてのみサポートされ、ベアメタルおよび VMWare デプロイメントでテストされます。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は実稼働環境でこれらを使用することを推奨していません。これらの機能は、近々発表予定の製品機能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。

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。

7.6.1. 複数ネットワークについて

Kubernetes では、コンテナネットワークは Container Network Interface (CNI) を実装するネットワークプラグインに委任されます。

OpenShift Container Platform は、Multus CNI プラグインを使用して CNI プラグインのチェーンを許可します。クラスターのインストール時に、**デフォルト** の Pod ネットワークを設定します。デフォルトのネットワークは、クラスターのすべての通常のネットワークトラフィックを処理します。利用可能な CNI プラグインに基づいて **追加のネットワーク** を定義し、1つまたは複数のネットワークを Pod に割り当てることができます。必要に応じて、クラスターの複数のネットワークを追加で定義することができます。これは、スイッチまたはルーティングなどのネットワーク機能を提供する Pod を設定する場合に柔軟性を実現します。

7.6.1.1. 追加ネットワークの使用シナリオ

データプレーンとコントロールプレーンの分離など、ネットワークの分離が必要な状況で追加のネットワークを使用することができます。トラフィックの分離は、以下のようなパフォーマンスおよびセキュリティー関連の理由で必要になります。

パフォーマンス

各プレーンのトラフィック量を管理するために、2つの異なるプレーンにトラフィックを送信できます。

セキュリティー

機密トラフィックは、セキュリティー上の考慮に基づいて管理されているネットワークに送信でき、テナントまたはカスタマー間で共有できないプライベートを分離することができます。

クラスターのすべての Pod はクラスター全体のデフォルトネットワークを依然として使用し、クラスター全体での接続性を維持します。すべての Pod には、クラスター全体の Pod ネットワークに割り当てられる **eth0** インターフェイスがあります。Pod のインターフェイスは、**oc exec -it <pod_name> -- ip a** コマンドを使用して表示できます。Multus CNI を使用するネットワークを追加する場合、それらの名前は **net1**、**net2**、...、**netN** になります。

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 CR のそれぞれにある CNI 設定は、インターフェイスの作成方法を定義します。

7.6.2. Multus を使用したストレージトラフィックの分離

Multus を使用するには、デプロイメントの前に、後にクラスターにアタッチされる Network attachment definition (NAD) を作成する必要があります。詳細は以下を参照してください。

- ベアメタルの場合は [Creating network attachment definitions](#)
- VMware の場合は [Creating network attachment definitions](#)

Multus を使用すると、ハードウェアの設定や VMWare インスタンスのネットワーク設定に応じて、以下の設定が可能です。

- デュアルネットワークインターフェイスを持つノードの推奨設定
 - 分離されたストレージトラフィック
 - OpenShift SDN に1つのインターフェイスを設定 (Pod 対 Pod のトラフィック)
 - すべての OpenShift Container Storage トラフィックに対して1つのインターフェイスを設定
- トリプルネットワークインターフェイスを持つノードの推奨設定
 - 完全なトラフィック分離
 - OpenShift SDN に1つのインターフェイスを設定 (Pod 対 Pod のトラフィック)
 - OpenShift Container Storage トラフィックへのすべての Pod に1つのインターフェイスを設定 (OpenShift Container Storage のパブリックトラフィック)
 - すべての OpenShift Container Storage レプリケーションおよびリバランストラフィックに1つのインターフェイスを設定 (OpenShift Container Storage クラスタートラフィック)

7.6.3. Multus 設定の推奨ネットワーク設定および要件

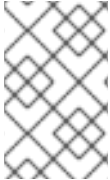
Multus 設定を利用する場合は、以下の前提条件を満たす必要があります。

- Multus 設定を完全に機能させるため、OpenShift Container Storage のデプロイに使用されるすべてのノードには、同じネットワークインターフェイス設定が必要です。すべてのノードのネットワークインターフェイス名は同じにする必要があります。Multus パブリックネットワークおよび Multus クラスタネットワークの同じ基礎となるスイッチメカニズムに接続されている必要があります。
- Multus 設定を完全に機能させるため、永続ストレージに OpenShift Container Storage を利用するアプリケーションをデプロイするために使用されるワーカーノードはすべて、同じネットワークインターフェイス設定を持つ必要があります。2つのインターフェイスの1つは、Storage ノードに Multus パブリックネットワークを設定するために使用されるインターフェイス名と同じである必要があります。すべてのワーカーネットワークインターフェイスが、Storage ノードの Multus パブリックネットワークに使用されるものと同じ基礎となるスイッチメカニズムに接続されている必要があります。



注記

ネットワークソフトウェアは、スイッチ内または SDN のいずれであっても、不明な MAC アドレスから送信されたイーサネットフレームをフィルタリングまたはドロップすることがあります。この機能に対する Cisco の用語は「ポートセキュリティー」です。ベアメタル環境では、ネットワーク管理者は、Multus が管理する物理ネットワークインターフェイスに対応するポートに対して、ポートセキュリティーまたは同等の機能が無効になっていることを確認する必要があります。同じことがソフトウェア定義ネットワークにも当てはまります。VMware の場合、MAC アドレスの変更を防ぐために、ポートグループのセキュリティーポリシーを設定してはなりません。



注記

OpenShift Container Storage OSD が実行されているストレージノードのみに、Multus によって設定された OpenShift Container Storage クラスターネットワークへのアクセスが必要になります。

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。

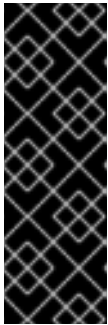
VMware で Multus ベースの設定に必要な手順については、[Multus ネットワークの作成](#) を参照してください。

第8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。

Red Hat OpenShift Container Storage は 2 つのフレーバーを提供します。

- **Regional-DR:** 2 つの Openshift Container Platform クラスタを提供する 2 つの Openshift Container Storage クラスタ全体でのストレージボリュームのマルチクラスタの非同期レプリケーションです。ステートレスアプリケーションを含むステートフルなアプリケーションには、ピアクラスタに同じものをデプロイする前に準備が必要です。



重要

これは開発者プレビュー機能であり、開発者プレビューのサポート制限の対象となります。開発者プレビューリリースは、実稼働環境で実行することは意図されておらず、Red Hat カスタマーポータルの場合管理システムではサポートされません。開発者のプレビュー機能に関してサポートが必要な場合には、ocs-devpreview@redhat.com メーリングリストに連絡してください。Red Hat Development Team のメンバーが稼働状況とスケジュールに応じて可能な限り迅速に対応します。

- **Metro-DR (Stretched Cluster - Arbiter):** この例では、3 番目のゾーンを Arbiter の場所とした上で、単一クラスタが 2 つのゾーンに展開されます。これはテクノロジープレビュー機能であり、現時点では OpenShift Container Platform オンプレミスでのデプロイメント用とされています。



注記

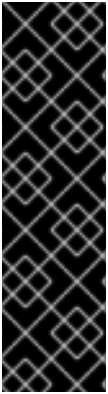
このソリューションは、レイテンシーが、ロケーション間で 4 ミリ秒のラウンドトリップタイム (RTT) を超えない場合にデプロイするように設計されています。より高いレイテンシーでデプロイする予定がある場合は、[Red Hat カスタマーサポート](#) にお問い合わせください。

Arbiter ストレッチクラスタを使用するには、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。
 - 各データセンターゾーンにはゾーンごとに 2 つのノードが使用されます。
 - 1 つのノードを含む 1 つの追加ゾーンは、arbiter ゾーンに使用されます (arbiter はマスターノードに配置できます)。
- すべてのノードには、クラスタ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)

詳細は以下を参照してください。

- [Configuring OpenShift Container Storage for Metro-DR stretch cluster](#)
- [Recovering a Metro-DR stretch cluster](#)



重要

これはテクノロジープレビュー機能であり、ローカルストレージデバイスを使用するデプロイメントでのみ利用できます。テクノロジープレビュー機能は Red Hat の実稼働環境でのサービスレベルアグリーメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。Red Hat は実稼働環境でこれらを使用することを推奨していません。これらの機能は、近々発表予定の製品機能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。

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。

第9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る非接続環境での OpenShift Container Storage のデプロイメントをサポートします。



注記

OpenShift Container Storage 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 ナレッジベースの記事](#) および [Configuring chrony time service](#) を参照してください。

詳細は、[Operators guide to using Operator Lifecycle Manager on restricted networks](#) を参照してください。

第10章 IBM POWER SYSTEMS および IBM Z インフラストラクチャーでサポートされない機能

- FIPS は、FIPS 自体が RHEL ではサポートされていないため、IBM Power Systems および IBM Z インフラストラクチャーの OpenShift Container Storage 4.8 ではサポートされません。
- IBM Power Systems および IBM Z インフラストラクチャーには、外部モードは使用できません。
- クラスタ全体の暗号化は、IBM Z インフラストラクチャーではサポートされていません。
- IBM Z インフラストラクチャーでは、デプロイメント中のストレージデータ暗号化はサポートされません。
- IBM Power Systems の OpenShift Container Storage 4.8 は、動的ストレージデバイスをサポートしません。
- IBM Power Systems および IBM Z インフラストラクチャーの OpenShift Container Storage 4.8 は Disaster Recovery をサポートしません。
- コンパクトなデプロイメントは、IBM Power Systems および IBM Z インフラストラクチャーには適用できません。
- IBM Power Systems および IBM Z インフラストラクチャーの OpenShift Container Storage 4.8 は Multus をサポートしません。

第11章 次のステップ

OpenShift Container Storage 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 Container Storage のデプロイ](#)
- [ベアメタルを使用した OpenShift Container Storage のデプロイ](#)
- [VMWare vSphere を使用した OpenShift Container Storage のデプロイ](#)
- [Microsoft Azure を使用した OpenShift Container Storage のデプロイ](#)
- [Deploying OpenShift Container Storage using Google Cloud \[テクノロジープレビュー\]](#)
- [Deploying OpenShift Container Storage using Red Hat OpenStack Platform \[テクノロジープレビュー\]](#)
- [Red Hat Virtualization Platform を使用した OpenShift Container Storage のデプロイ](#)
- [IBM Power Systems での OpenShift Container Storage のデプロイ](#)
- [IBM Z インフラストラクチャーへの OpenShift Container Storage のデプロイ](#)

外部モード

- [外部モードでの OpenShift Container Storage のデプロイ](#)