



Red Hat OpenShift Container Storage 4.5

デプロイメントのプランニング

RHOCS 4.5 をデプロイする際の重要な考慮事項

Red Hat OpenShift Container Storage 4.5 デプロイメントのプランニング

RHOCS 4.5 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

第1章 RED HAT OPENSIFT CONTAINER STORAGE の概要	3
第2章 OPENSIFT CONTAINER STORAGE のアーキテクチャー	4
2.1. OPERATOR	4
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	5
2.2.1. 内部アプローチ	5
2.2.2. 外部アプローチ	6
2.3. ノードのタイプ	6
第3章 内部ストレージサービス	8
第4章 外部ストレージサービス	9
第5章 セキュリティーに関する考慮事項	10
5.1. FIPS-140-2	10
5.2. プロキシ環境	10
第6章 サブスクリプション	11
6.1. サブスクリプションのオフアリング	11
6.2. 障害復旧サブスクリプション	11
6.3. コア対 VCPU およびハイパースレッディング	11
6.4. コアの分割	11
6.5. サブスクリプションの要件	12
第7章 インフラストラクチャーの要件	13
7.1. プラットフォーム要件	13
7.1.1. Amazon EC2	13
7.1.2. ベアメタル	13
7.1.3. VMware	13
7.1.4. Google Cloud [テクノロジープレビュー]	14
7.1.5. Microsoft Azure [テクノロジープレビュー]	14
7.1.6. Red Hat Virtualization Platform [テクノロジープレビュー]	14
7.1.7. Red Hat OpenStack Platform [テクノロジープレビュー]	14
7.2. 外部モード要件	14
7.2.1. Red Hat Ceph Storage	14
7.3. リソース要件	14
7.4. POD の配置ルール	15
7.5. ストレージデバイスの要件	15
7.5.1. 動的ストレージデバイス	16
7.5.2. ローカルストレージデバイス	16
7.5.3. 容量のプランニング	16
第8章 非接続環境	18
第9章 次のステップ	19

第1章 RED HAT OPENSIFT CONTAINER STORAGE の概要

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。

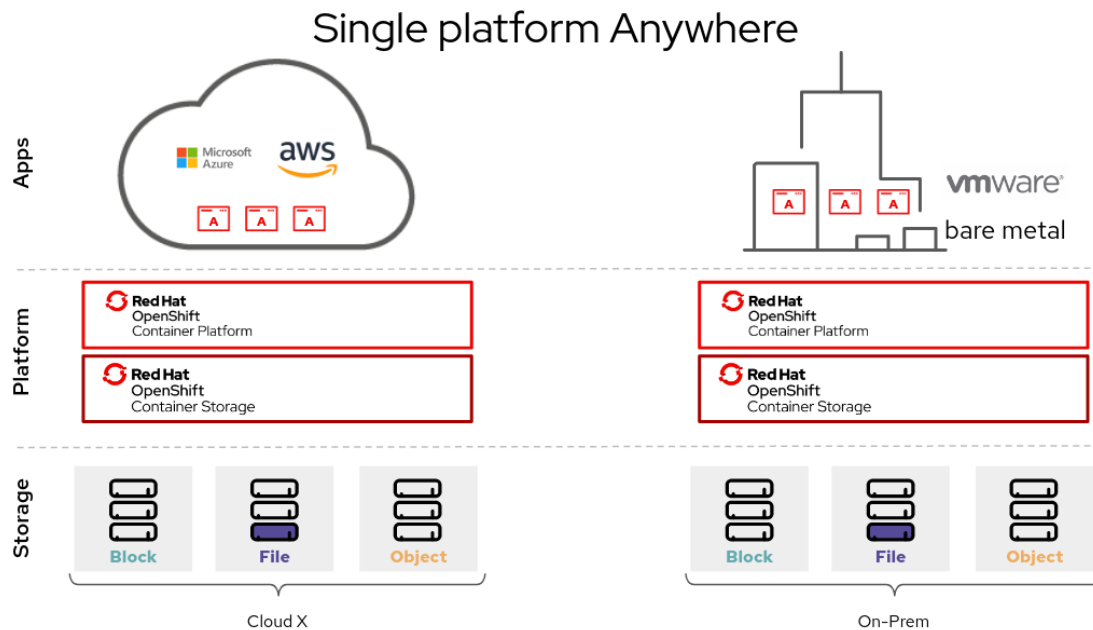
- Ceph。ブロックストレージ、共有および分散ファイルシステム、およびオンプレミスのオブジェクトストレージを提供します。
- Ceph CSI。永続ボリュームおよび要求のプロビジョニングおよびライフサイクルを管理します。
- NooBaa。Multicloud Object Gateway を提供します。

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 アーキテクチャー](#)』を参照してください。

2.1. OPERATOR

Red Hat OpenShift Container Storage は主に 3 つの Operator で構成されており、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 (OCS) 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。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、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。たとえば、EC2 の EBS ボリューム、または VMware の vSphere Virtual Volume。

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)

OpenShift Container Storage Operator のインストール後に、インストールウィザードを使用して OpenShift コンソールから Storage Cluster リソースを作成するか、または内部または外部モードで手動で送信できます。

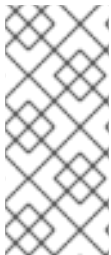
2.3. ノードのタイプ

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)	<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 クラスターではオプションですが、仮想化環境およびクラウド環境の OpenShift Container Storage 用に推奨されます。</p> <p>インフラストラクチャーノードを作成するには、Infra とラベルが付けられた新規ノードをプロビジョニングできます。「How to label Red Hat OpenShift Storage "nodes" as an infra node?」を参照してください。</p>

ノードタイプ	説明
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構成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されます。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ルが可能になります。</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、または VMware の vSphere Virtual Volume 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
- Google Cloud [テクノロジープレビュー]
- Microsoft Azure [テクノロジープレビュー]
- VMware vSphere

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 クラスターで外部 Red Hat Ceph Storage クラスターのサービスを利用できるようにします。

- ベアメタル
- VMware vSphere

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。外部クラスターリソースは、追加のストレージクラスをアプリケーションで利用できるようにします。外部クラスターは 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。

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アリング

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 コアのサブスクリプションの場合、任意の数の仮想マシンで利用できる 20 コアが提供されます。
- 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かにかかわらず、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。

6.4. 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、「[コア対 vCPU およびハイパースレッディング](#)」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。

6.5. 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。ワーカーノードが OpenShift Container Storage コンポーネントに使用される場合、それらのノードには OpenShift Container Platform と OpenShift Container Storage の両方のサブスクリプションが必要です。インフラストラクチャーノードが使用される場合、それらのノードには OpenShift Container Storage サブスクリプションのみが必要となります。ラベルは、ノードをワーカーノードまたはインフラストラクチャーノードとみなされるようにするどうかを示唆するために使用されます。

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 は、OpenShift Container Storage バージョンより1マイナーリリース前または先の OpenShift Container Platform リリースと組み合わせることができます。

OpenShift Container Storage 4.5 は以下で実行できます。

- OpenShift Container Platform 4.4 (1 バージョン前)(内部モードの場合のみ)
- OpenShift Container Platform 4.5 (同じバージョン)
- OpenShift Container Platform 4.6 (1 バージョン先)

外部クラスターの場合、OpenShift Container Platform バージョン 4.5 以降は Red Hat OpenShift Container Storage 4.5 をデプロイするために必要です。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ティクル](#) 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。

- EBS ストレージ (aws-ebs プロビジョナー経由)
- インスタンスストレージ (ローカルストレージ Operator 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

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. 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)

7.1.5. Microsoft Azure [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。

- Azure ディスク (azure-disk プロビジョナー経由)

7.1.6. Red Hat Virtualization Platform [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、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。

- Cinder プロビジョナーを使用した標準ディスク

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 4.1.1 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とこれに続く追加のデバイスセットで構成されます。これらの OpenShift Container Storage サービス Pod はすべて「[Pod の配置ルール](#)」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。

表7.1 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	<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	<ul style="list-style-type: none"> ● 4 個の CPU (論理) ● 16 GiB メモリー 	該当なし

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 の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。

例: 単一デバイスセットを持つ 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。CPU の 30 ユニットは 15 コアに相当し、これは Red Hat OpenShift Container Storage（2 コア）の ~8 サブスクリプションに相当します。

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。

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 ノードの）3 の倍数に拡張する方法は、[Pod の配置ルール](#)に簡単に従える方法です。



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 ノードの）3 の倍数に拡張する方法は、[Pod の配置ルール](#)に簡単に従える方法です。



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。ストレージ領域が不足する場合は、[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7.23 つのノードで構成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7.3 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 非接統環境

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](#) [記事](#) および「[Configuring chrony time service](#)」を参照してください。

詳細は、「[Preparing to deploy OpenShift Container Storage in disconnected environments](#)」を参照してください。

第9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](#)
- [Google Cloud を使用した OpenShift Container Storage のデプロイ](#) [テクノロジープレビュー]
- [Microsoft Azure を使用した OpenShift Container Storage のデプロイ](#) [テクノロジープレビュー]
- [VMware vSphere を使用した OpenShift Container Storage のデプロイ](#)
- [Red Hat Virtualization Platform を使用した OpenShift Container Storage のデプロイ](#) [テクノロジープレビュー]
- [Red Hat OpenStack Platform を使用した OpenShift Container Storage のデプロイ](#) [テクノロジープレビュー]

外部モード

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