



SUSE Enterprise Storage 7

# 導入ガイド

# 導入ガイド

## SUSE Enterprise Storage 7


著者: Tomáš Bažant、Alexandra Settle、Liam Proven

発行日: 11/12/2025

<https://documentation.suse.com> 

Copyright © 2020–2025 SUSE LLC and contributors. All rights reserved.

特に明記されている場合を除き、本書はクリエイティブ・コモンズ表示-継承4.0国際(CC-BY-SA 4.0)に基づいてライセンスされています。 <https://creativecommons.org/licenses/by-sa/4.0/legalcode>  を参照してください。

SUSEの商標については、<http://www.suse.com/company/legal/> を参照してください。サードパーティ各社とその製品の商標は、所有者であるそれぞれの会社に所属します。商標記号(®、™など)は、SUSEおよび関連会社の商標を示します。アスタリスク(\*)は、第三者の商標を示します。

本書のすべての情報は、細心の注意を払って編集されています。しかし、このことは絶対に正確であることを保証するものではありません。SUSE LLC、その関係者、著者、翻訳者のいずれも誤りまたはその結果に対して一切責任を負いかねます。

# 目次

## このガイドについて viii

- 1 利用可能なマニュアル viii
- 2 フィードバックの提供 ix
- 3 マニュアルの表記規則 x
- 4 製品のライフサイクルとサポート xii
  - SUSEによるサポートの定義 xii • SUSE Enterprise Storageのサポートステートメント xii • 技術レビュー xiii
- 5 Cephの貢献者 xiv
- 6 このガイドで使用するコマンドとコマンドプロンプト xiv
  - Salt関連のコマンド xiv • Ceph関連のコマンド xv • 一般的なLinuxコマンド xvi • 追加情報 xvi

## I SES (SUSE ENTERPRISE STORAGE)の概要 1

### 1 SESとCeph 2

- 1.1 Cephの特徴 2
- 1.2 Cephのコアコンポーネント 3
  - RADOS 3 • CRUSH 4 • Cephのノードとデーモン 5
- 1.3 Cephのストレージ構造 6
  - プール 6 • 配置グループ 7 • 例 7
- 1.4 BlueStore 9
- 1.5 追加情報 10

### 2 ハードウェア要件と推奨事項 11

- 2.1 ネットワーク概要 11
  - ネットワーク推奨事項 12
- 2.2 複数のアーキテクチャの設定 14

2.3	ハードウェア設定 15
	最小クラスタ構成 15 • 運用クラスタの推奨設定 17 • マルチパス設定 18
2.4	オブジェクトストレージノード 19
	Minimum requirements 19 • 最小ディスクサイズ 20 • BlueStoreのWALおよびDBデバイスの推奨サイズ 20 • WAL/DBパーティション用SSD 21 • 推奨されるディスクの最大数 21
2.5	Monitorノード 22
2.6	Object Gatewayノード 22
2.7	メタデータサーバノード 22
2.8	管理ノード 23
2.9	iSCSI Gatewayノード 23
2.10	SESおよび、その他のSUSE製品 23
	SUSE Manager 23
2.11	名前の制限 23
2.12	OSDとモニタでの1台のサーバの共有 23
<b>3</b>	<b>管理ノードのHAセットアップ 25</b>
3.1	管理ノードのHAクラスタの概要 25
3.2	管理ノードを有するHAクラスタの構築 26
<b>II</b>	<b>CEPHクラスタの展開 28</b>
<b>4</b>	<b>導入と共通タスク 29</b>
4.1	リリースノートの確認 29
<b>5</b>	<b>cephadmによる展開 30</b>
5.1	SUSE Linux Enterprise Serverのインストールと設定 30
5.2	Saltの展開 31

- 5.3 Cephクラスタの展開 34
  - ceph-saltのインストール 34 • クラスタプロパティの設定 35 • ノードの更新と最小クラスタのブートストラップ 47 • 最終ステップの確認 49
- 5.4 サービスとゲートウェイの展開 50
  - ceph orchコマンド 50 • サービス仕様と配置仕様 52 • Cephサービスの展開 55
- III 追加のサービスのインストール 65
- 6 iSCSIゲートウェイのインストール 66
  - 6.1 iSCSIブロックストレージ 66
    - LinuxカーネルiSCSIターゲット 67 • iSCSIイニシエータ 67
  - 6.2 ceph-iscsiに関する一般情報 68
  - 6.3 展開に関する考慮事項 69
  - 6.4 インストールと設定 70
    - CephクラスタへのiSCSI Gatewayの展開 70 • RBDイメージの作成 70 • iSCSIを経由したRBDイメージのエクスポート 71 • 認証とアクセス制御 72 • 高度な設定 74
  - 6.5 tcmu-runnerを使用したRADOS Block Deviceイメージのエクスポート 77
- IV 古いリリースからのアップグレード 79
- 7 前回リリースからのアップグレード 80
  - 7.1 アップグレード実行前の確認事項 80
    - 考慮すべきポイント 81 • クラスタ設定とデータのバックアップ 82 • 前回のアップグレード手順の確認 83 • クラスタノードのアップデートとクラスタのヘルスの確認 83 • ソフトウェアリポジトリとコンテナイメージへのアクセス確認 83
  - 7.2 Salt Masterのアップグレード 84
  - 7.3 MON、MGR、OSDノードのアップグレード 86
  - 7.4 ゲートウェイノードのアップグレード 87

7.5	ceph-saltのインストールと、クラスタ設定の適用	88
7.6	監視スタックのアップグレードと導入	90
7.7	ゲートウェイサービスの再展開	91
	Object Gatewayのアップグレード	91
	• NFS Ganeshaのアップグレード	92
	• メタデータサーバのアップグレード	96
	• iSCSI Gatewayのアップグレード	96
7.8	アップグレード後のクリーンアップ	98
<b>A</b>	<b>アップストリーム「Octopus」ポイントリリースに基づくCeph保守更新</b>	<b>100</b>
<b>B</b>	<b>マニュアルの更新</b>	<b>103</b>
	<b>用語集</b>	<b>104</b>

# このガイドについて

このガイドでは基本的なCephクラスタの展開方法と、追加のサービスの展開方法に焦点を当てて説明しています。また、旧バージョンの製品をSUSE Enterprise Storage 7にアップグレードする手順についても説明しています。

SUSE Enterprise Storage 7はSUSE Linux Enterprise Server 15 SP2の拡張機能です。Ceph (<http://ceph.com/>) ストレージプロジェクトの機能に、SUSEのエンタープライズエンジニアリングとサポートが組み合わされています。SUSE Enterprise Storage 7により、IT組織は、コモディティハードウェアプラットフォームを使用して多様な使用事例に対応できる分散ストレージアーキテクチャを展開できます。

## 1 利用可能なマニュアル



### 注記: オンラインマニュアルと最新のアップデート

製品に関するマニュアルは、<https://documentation.suse.com> からご利用いただけます。最新のアップデートもご利用いただけるほか、マニュアルをさまざまな形式でブラウズおよびダウンロードすることができます。最新のマニュアルアップデートは英語版で検索できます。

また、製品マニュアルは、`/usr/share/doc/manual`の下にあるインストール済みシステムから入手できます。製品マニュアルは `ses-manual_LANG_CODE` という名前のRPMパッケージに含まれています。システム上にマニュアルが存在しない場合は、たとえば次のコマンドを使用してインストールしてください。

```
root # zypper install ses-manual_en
```

この製品の次のマニュアルを入手できます。

導入ガイド (<https://documentation.suse.com/ses/html/ses-all/book-storage-deployment.html>)


このガイドでは基本的なCephクラスタの展開方法と、追加のサービスの展開方法に焦点を当てて説明しています。また、旧バージョンの製品をSUSE Enterprise Storage 7にアップグレードする手順についても説明しています。

**運用と管理ガイド** (<https://documentation.suse.com/ses/html/ses-all/book-storage-admin.html>) 

このガイドでは、基本的なCephクラスタを展開した後に、管理者として実行する必要があるルーチンタスク(日常的な管理)に焦点を当てて説明しています。また、サポートされている、Cephクラスタに保存されたデータにアクセスする方法をすべて説明しています。

**Security Hardening Guide** (<https://documentation.suse.com/ses/html/ses-all/book-storage-security.html>) 

このガイドでは、クラスタのセキュリティを確保する方法に焦点を当てて説明しています。

**トラブルシューティングガイド** (<https://documentation.suse.com/ses/html/ses-all/book-storage-troubleshooting.html>) 

このガイドでは、SUSE Enterprise Storage 7を実行する際のさまざまな一般的な問題と、CephやObject Gatewayのような関連コンポーネントに関する問題について説明しています。


**SUSE Enterprise Storage for Windows Guide** (<https://documentation.suse.com/ses/html/ses-all/book-storage-windows.html>) 


このガイドでは、Windowsドライバを使用したMicrosoft Windows環境とSUSE Enterprise Storageの統合、インストール、および設定について説明しています。

## 2 フィードバックの提供


このドキュメントに対するフィードバックや貢献を歓迎します。次のチャンネルがあります。

### サービス要求およびサポート

ご使用の製品に利用できるサービスとサポートのオプションについては、<http://www.suse.com/support/> を参照してください。

サービス要求を開くには、SUSE Customer Centerに登録されたSUSEの購読が必要です。<https://scc.suse.com/support/requests> に移動して、ログインし、新規作成をクリックします。

### バグレポート

<https://bugzilla.suse.com/> にあるドキュメントで問題を報告します。レポートिंगの問題には、Bugzillaアカウントが必要です。

このプロセスを簡略化するために、このドキュメントのHTMLバージョンの見出しの横にあるReport Documentation Bug (ドキュメントバグの報告)リンクを使用できます。これらにより、Bugzillaで適切な製品とカテゴリが事前に選択され、現在のセクションへのリンクが追加されます。バグレポートの入力を直ちに開始できます。

## 貢献内容

このドキュメントに貢献するには、このドキュメントのHTMLバージョンの見出しの横にあるEdit Source (ソースの編集)リンクを使用してください。GitHubのソースコードに移動し、そこでプル要求を開くことができます。貢献にはGitHubアカウントが必要です。

このドキュメントに使用されるドキュメント環境に関する詳細については、<https://github.com/SUSE/doc-ses>にあるリポジトリのREADMEを参照してください。

## メール

ドキュメントに関するエラーの報告やフィードバックは[doc-team@suse.com](mailto:doc-team@suse.com)宛に送信してください。ドキュメントのタイトル、製品のバージョン、およびドキュメントの発行日を記載してください。また、関連するセクション番号とタイトル(またはURL)、問題の簡潔な説明も記載してください。

# 3 マニュアルの表記規則

このマニュアルでは、次の通知と表記規則が使用されています。

- `/etc/passwd`: ディレクトリ名とファイル名
- `PLACEHOLDER`: `PLACEHOLDER`は、実際の値で置き換えられます。
- `PATH`: 環境変数
- `ls`、`--help`: コマンド、オプション、およびパラメータ
- `user`: ユーザまたはグループの名前
- `package_name`: ソフトウェアパッケージの名前
- `Alt`、`Alt + F1`: 押すキーまたはキーの組み合わせ。キーはキーボードのように大文字で表示されます。
- ファイル、ファイル > 名前を付けて保存: メニュー項目、ボタン
- `AMD/Intel` この説明は、Intel 64/AMD64アーキテクチャにのみ当てはまります。矢印は、テキストブロックの先頭と終わりを示します。◁

**IBM Z, POWER** この説明は、IBM ZおよびPOWERの各アーキテクチャにのみ当てはまります。矢印は、テキストブロックの先頭と終わりを示します。◁□

- 第1章、「章の例」：このガイドの別の章への相互参照。
- root特権で実行する必要があるコマンド。多くの場合、これらのコマンドの先頭にsudoコマンドを置いて、特権のないユーザとしてコマンドを実行することもできます。

```
root # command
tux > sudo command
```

- 特権のないユーザでも実行できるコマンド。

```
tux > command
```

- 通知



### 警告: 警告の通知

続行する前に知っておくべき、無視できない情報。セキュリティ上の問題、データ損失の可能性、ハードウェアの損傷、または物理的な危険について警告します。



### 重要: 重要な通知

続行する前に知っておくべき重要な情報です。



### 注記: メモの通知

追加情報。たとえば、ソフトウェアバージョンの違いに関する情報です。



### ヒント: ヒントの通知

ガイドラインや実地的なアドバイスなどの役に立つ情報です。

- コンパクトな通知



追加情報。たとえば、ソフトウェアバージョンの違いに関する情報です。



ガイドラインや実地的なアドバイスなどの役に立つ情報です。

## 4 製品のライフサイクルとサポート

SUSE製品が異なれば、製品のライフサイクルも異なります。SUSE Enterprise Storageのライフサイクルの正確な日付を確認するには、<https://www.suse.com/lifecycle/>を参照してください。

### 4.1 SUSEによるサポートの定義

サポートポリシーとオプションについては、<https://www.suse.com/support/policy.html>および<https://www.suse.com/support/programs/long-term-service-pack-support.html>を参照してください。

### 4.2 SUSE Enterprise Storageのサポートステートメント

サポートを受けるには、SUSEの適切な購読が必要です。利用可能なサポートサービスを具体的に確認するには、<https://www.suse.com/support/>にアクセスして製品を選択してください。

サポートレベルは次のように定義されます。

#### L1

問題の判別。互換性情報、使用サポート、継続的な保守、情報収集、および利用可能なドキュメントを使用した基本的なトラブルシューティングを提供するように設計されたテクニカルサポートを意味します。

#### L2

問題の切り分け。データの分析、お客様の問題の再現、問題領域の特定、レベル1で解決できない問題の解決、またはレベル3の準備を行うように設計されたテクニカルサポートを意味します。

#### L3

問題解決。レベル2サポートで特定された製品の欠陥を解決するようにエンジニアリングに依頼して問題を解決するように設計されたテクニカルサポートを意味します。

契約されているお客様およびパートナーの場合、SUSE Enterprise Storageでは、次のものを除くすべてのパッケージに対してL3サポートを提供します。

- 技術レビュー
- サウンド、グラフィック、フォント、およびアートワーク
- 追加の顧客契約が必要なパッケージ
- モジュール「Workstation Extension」の一部として出荷される一部のパッケージは、L2サポートのみです。
- 名前が `-devel` で終わるパッケージ(ヘッダファイルなどの開発用リソースが含まれるパッケージ)のサポートを受けるには、そのメインパッケージが必要です。

SUSEは、元のパッケージの使用のみをサポートします。つまり、変更も、再コンパイルもされないパッケージをサポートします。

## 4.3 技術レビュー

技術レビューとは、今後のイノベーションを垣間見ていただくための、SUSEによって提供されるパッケージ、スタック、または機能を意味します。技術レビューは、ご利用中の環境で新しい技術をテストする機会を参考までに提供する目的で収録されています。私たちはフィードバックを歓迎しています。技術レビューをテストする場合は、SUSEの担当者に連絡して、経験や使用例をお知らせください。ご入力いただいた内容は今後の開発のために役立たせていただきます。

技術レビューには、次の制限事項があります。

- 技術レビューはまだ開発中です。したがって、機能が不完全であったり、不安定であったり、何らかの理由で運用環境での使用には適していなかったりする場合があります。
- 技術レビューにはサポートが提供されません。
- 技術レビューは、特定のハードウェアアーキテクチャでしか利用できないことがあります。

- 技術プレビューの詳細および機能は、変更される場合があります。そのため、今後リリースされる技術プレビューへのアップグレードができない場合や、再インストールが必要となる場合があります。
- 技術プレビューは製品から予告なく削除される可能性があります。将来的にこうした技術に対応したバージョンを提供することを、SUSEはお約束しません。たとえば、プレビューがお客様や市場のニーズを満たしていない、またはエンタープライズ基準に準拠していないとSUSEが判断した場合です。

ご使用の製品に付属している技術プレビューの概要については、[https://www.suse.com/releases/notes/x86\\_64/SUSE-Enterprise-Storage/7](https://www.suse.com/releases/notes/x86_64/SUSE-Enterprise-Storage/7)にあるリリースノートを参照してください。

## 5 Cephの貢献者

Cephプロジェクトとそのドキュメントは、数百人の貢献者と組織の作業の結果です。詳しくは「<https://ceph.com/contributors/>」を参照してください。

## 6 このガイドで使用されるコマンドとコマンドプロンプト

Cephクラスタ管理者は、特定のコマンドを実行して、クラスタの動作を設定および調整します。必要になるコマンドには、次のようにいくつかの種類があります。

### 6.1 Salt関連のコマンド

これらのコマンドは、Cephクラスタノードを展開する場合や、クラスタノードの一部(または全部)で同時にコマンドを実行する場合、クラスタノードを追加または削除する場合に役立ちます。最も頻繁に使用されるコマンドは**ceph-salt**と**ceph-salt config**です。Salt Masterノードでは、Saltコマンドは**root**として実行する必要があります。これらのコマンドは、次のプロンプトで示されます。

```
root@master #
```

例:

```
root@master # ceph-salt config ls
```

## 6.2 Ceph関連のコマンド

これらは、**ceph**、**cephadm**、**rbd**、または**radosgw-admin**など、コマンドラインでクラスタとそのゲートウェイのすべての側面を設定および微調整するための下位レベルのコマンドです。

Ceph関連のコマンドを実行するには、Cephキーの読み取りアクセス権が必要です。このキーの機能により、Ceph環境内におけるユーザの特権が定義されます。1つのオプションは、**root**として(または**sudo**を使用して)Cephコマンドを実行し、制限のないデフォルトのキーリング「**ceph.client.admin.keyring**」を使用します。

より安全な推奨オプションは、各管理者ユーザに対してより制限の厳しい個別のキーを作成し、そのキーを、各ユーザが読み取ることができるディレクトリに保存することです。次に例を示します。

```
~/ceph/ceph.client.USERNAME.keyring
```



### ヒント: Cephキーのパス

カスタムの管理者ユーザとキーリングを使用するには、**ceph**コマンドを実行するたびに、**-n client.USER\_NAME**オプションと**--keyring PATH/TO/KEYRING**オプションを使用して、ユーザ名とプールのパスを指定する必要があります。

これを回避するには、個々のユーザの**~/.bashrc**ファイルで**CEPH\_ARGS**変数にこれらのオプションを含めてください。

Ceph関連のコマンドは任意のクラスタノードで実行できますが、管理ノードで実行することをお勧めします。このドキュメントでは、**cephadm**ユーザを使用してコマンドを実行するので、コマンドは次のプロンプトが表示されます。

```
cephuser@adm >
```

例:

```
cephuser@adm > ceph auth list
```



### ヒント: 特定のノード用のコマンド

クラスタノードに対して特定の役割でコマンドを実行するようドキュメントで指示されている場合は、プロンプトによって示されます。以下に例を示します。

```
cephuser@mon >
```

### 6.2.1 **ceph-volume**の実行

SUSE Enterprise Storage 7から、Cephサービスはコンテナ化された状態で実行されます。OSDノード上で**ceph-volume**を実行する必要がある場合は、**cephadm**コマンドに付加する必要があります。たとえば、次のようになります。

```
cephuser@adm > cephadm ceph-volume simple scan
```

## 6.3 一般的なLinuxコマンド

**mount**、**cat**、または**openssl**など、Cephに関連しないLinuxコマンドは、関連するコマンドに必要な特権に応じて、**cephuser@adm >**または**root #**のいずれかで導入されます。

## 6.4 追加情報

Cephのキー管理の詳細については、『運用と管理ガイド』、第30章「cephxを使用した認証」、30.2項「キー管理」を参照してください。

# I SES (SUSE Enterprise Storage)の概要

- 1 SESとCeph 2
- 2 ハードウェア要件と推奨事項 11
- 3 管理ノードのHAセットアップ 25

# 1 SESとCeph

SUSE Enterprise Storageは、スケーラビリティ、信頼性、およびパフォーマンスを目的として設計された、Cephテクノロジーに基づく分散ストレージシステムです。Cephクラスタは、Ethernetなどの一般的なネットワーク内にあるコモディティサーバで実行できます。クラスタは、数千台のサーバ(以降、ノードと呼びます)とペタバイトの域にまで容易に拡張できます。データを保存および取得するためのアロケーションテーブルを持つ従来のシステムとは異なり、Cephは決定的アルゴリズムを使用してデータの記憶域を割り当て、集中化された情報構造を持ちません。Cephでは、Storage Cluster内でのハードウェアの追加や削除は、例外ではなく標準の動作であると想定されています。Cephクラスタは、データの分散と再分散、データの複製、障害検出、回復などの管理タスクを自動化します。Cephは自己修復機能と自己管理機能の両方を備えているため、管理と予算のオーバーヘッドが削減されます。

この章では、SUSE Enterprise Storage 7の大まかな概要と、最も重要なコンポーネントについて簡単に説明します。

## 1.1 Cephの特徴

Ceph環境には次のような特徴があります。

### 拡張性

Cephは数千台のノードにまで拡張でき、ペタバイトの域のストレージを管理できます。

### コモディティハードウェア

Cephクラスタを実行するのに特別なハードウェアは必要ありません。詳細については、[第2章「ハードウェア要件と推奨事項」](#)を参照してください。

### 自己管理

Cephクラスタは自己管理型です。ノードが追加または削除された場合、あるいはノードに障害が発生した場合、クラスタは自動的にデータを再分散します。過負荷状態のディスクを認識する機能もあります。

### SPOF (Single Point of Failure)を排除

クラスタ内のノードが重要な情報を単独で保存することはありません。冗長性の数は設定が可能です。

### オープンソースのソフトウェア

Cephは、特定のハードウェアやベンダーとは無関係のオープンソースソフトウェアソリューションです。

## 1.2 Cephのコアコンポーネント

Cephの能力を最大限活用するには、その基本的なコンポーネントと概念を理解する必要があります。このセクションでは、他の章で頻繁に参照されるCephの機能をいくつか紹介します。

### 1.2.1 RADOS

Cephの基本コンポーネントは「RADOS (Reliable Autonomic Distributed Object Store)」と呼ばれます。これは、クラスタに保存されるデータの管理を受け持ちます。通常、Ceph内のデータはオブジェクトとして保存されています。各オブジェクトはIDとデータで構成されます。

RADOSは、保存オブジェクトへのアクセス方法として次の方法を備えており、さまざまな使用事例に対応します。

#### Object Gateway

Object Gatewayは、RADOS Object StoreのHTTP RESTゲートウェイです。これにより、Cephクラスタに保存されているオブジェクトへの直接アクセスが可能になります。

#### RADOS Block Device

RADOS Block Device (RBD)には他のブロックデバイスと同じようにアクセスできます。たとえば、仮想化を行う場合、RBDとlibvirtを組み合わせで使用できます。

#### CephFS

Ceph File SystemはPOSIX互換のファイルシステムです。

#### librados

libradosは、Storage Clusterを直接操作できるアプリケーションを作成するためのライブラリで、さまざまなプログラミング言語で使用できます。

Object GatewayとRBDはlibradosを使用するのに対し、CephFSはRADOSと直接対話します。図1.1「[Ceph Object Storeのインタフェース](#)」を参照してください。

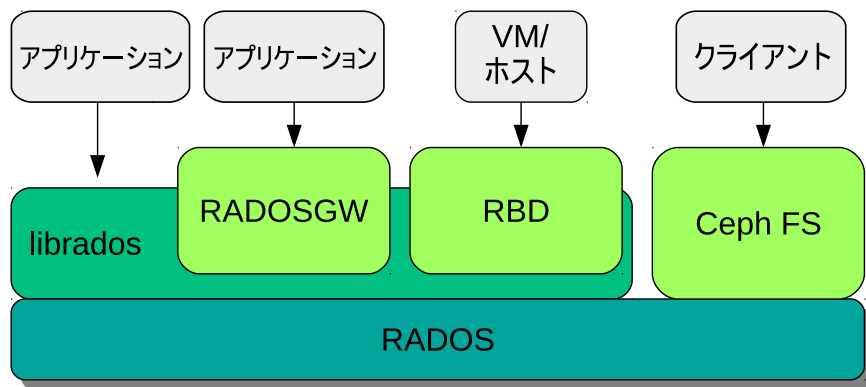


図 1.1: CEPH OBJECT STOREのインタフェース

## 1.2.2 CRUSH

Cephクラスタの中核を成すのは「CRUSH」アルゴリズムです。CRUSHは「Controlled Replication Under Scalable Hashing」の略語です。CRUSHはストレージの割り当てを扱う機能で、指定が必要なパラメータは比較的少なくなっています。つまり、オブジェクトの保存位置を計算するのに必要な情報はごくわずかです。そのパラメータは、ヘルス状態を含むクラスタの現在のマップ、管理者定義の配置ルール、および保存または取得する必要があるオブジェクトの名前です。この情報により、Cephクラスタ内のすべてのノードは、オブジェクトとそのレプリカが保存されている場所を計算できます。このため、データの読み書きが非常に効率化されます。CRUSHは、データをクラスタ内のすべてのノードに均等に分散しようとしています。

「CRUSHマップ」には、クラスタにオブジェクトを保存するための、すべてのストレージノードと管理者定義の配置ルールが記述されています。CRUSHマップは階層構造を定義し、その階層構造は通常、クラスタの物理構造に対応します。たとえば、データが含まれるディスクがホストにあり、ホストがラックに格納されているとします。さらに、ラックは複数の列に収容されていて、ラック列はデータセンターにあるとします。この構造を使用して「障害ドメイン」を定義できます。Cephは、それに従ってレプリケーションが特定の障害ドメインの異なるブランチに保存されるようにします。

障害ドメインがラックに設定されている場合、オブジェクトのレプリケーションは異なるラックに分散されます。これによって、ラック内のスイッチの障害によって発生する停止を緩和できます。1台の配電ユニットで1つのラック列に電力を供給している場合は、障害ドメインを列に設定できます。配電ユニットに障害が発生しても、引き続き他の列で複製されたデータを利用できます。

### 1.2.3 Cephのノードとデーモン

Cephでは、ノードとはクラスタを形成しているサーバです。ノードでは複数の種類のデーモンを実行できます。各ノードで実行するデーモンは1種類だけにするをお勧めします。ただし、Ceph Managerデーモンは例外で、Ceph Monitorと一緒に配置できます。各クラスタには、少なくともCeph Monitor、Ceph Manager、およびCeph OSDデーモンが必要です。

#### 管理ノード

「管理ノード」とは、コマンドを実行してクラスタを管理するCephクラスタノードを指します。管理ノードはCephクラスタの中心となる場所です。これは、Salt Minionサービスに対してクエリと命令を行って他のクラスタノードを管理する役割があるためです。

#### Ceph Monitor

「Ceph Monitor」(多くの場合、「MON」と省略)ノードは、クラスタのヘルス状態に関する情報、すべてのノードのマップ、およびデータ分散ルールを維持します(1.2.2項「CRUSH」を参照してください)。

障害または衝突が発生した場合、クラスタ内のCeph Monitorノードは、どの情報が正しいかを多数決で決定します。必ず多数決が得られるように、奇数個(少なくとも3個以上)のCeph Monitorノードを設定することをお勧めします。

複数のサイトを使用する場合、Ceph Monitorノードは奇数個のサイトに分散する必要があります。サイトあたりのCeph Monitorノードの数は、1つのサイトに障害が発生したときに、50%を超えるCeph Monitorノードの機能が維持される数にする必要があります。

#### Ceph Manager

Ceph Managerはクラスタ全体から状態情報を収集します。Ceph ManagerデーモンはCeph Monitorデーモンと共に動作します。追加のモニタリング機能を提供し、外部のモニタリングシステムや管理システムとのインターフェースとして機能します。これには、他のサービスも含まれます。たとえば、CephダッシュボードWeb UIはCeph Managerと同じノードで実行されます。

Ceph Managerには、動作確認以外の追加設定は必要ありません。

#### Ceph OSD

「Ceph OSD」は、「オブジェクトストレージデバイス」を処理するデーモンです。OSDは、物理ストレージユニットまたは論理ストレージユニット(ハードディスクまたはパーティション)になります。オブジェクトストレージデバイスは、物理ディスク/パーティションにも、論理ボリュームにもできます。このデーモンはほかにも、データのレプリケーションや、ノードが追加または削除された場合のリバランスも処理します。

Ceph OSDデーモンはモニターデーモンと通信して、他のOSDデーモンの状態をモニターデーモンに提供します。

CephFS、Object Gateway、NFS Ganesha、またはiSCSI Gatewayを使用するには、追加のノードが必要です。

### MDS (メタデータサーバ)

CephFSのメタデータは自身のRADOSプールに保存されます(1.3.1項「プール」を参照してください)。メタデータサーバはスマートなメタデータのキャッシュ層として機能し、必要に応じてアクセスをシリアル化します。これにより、明示的な同期を取ることなく多数のクライアントからの同時アクセスが可能となります。

### Object Gateway

Object Gatewayは、RADOS Object StoreのHTTP RESTゲートウェイです。OpenStack SwiftおよびAmazon S3と互換性があり、独自のユーザ管理機能を持ちます。

### NFS Ganesha

NFS Ganeshaは、Object GatewayまたはCephFSにNFSアクセスを提供します。カーネル空間ではなくユーザ空間で動作し、Object GatewayまたはCephFSと直接対話します。

### iSCSI Gateway

iSCSIは、クライアントからリモートサーバ上のSCSIストレージデバイス(ターゲット)にSCSIコマンドを送信できるようにするストレージネットワークプロトコルです。

### Sambaゲートウェイ

Sambaゲートウェイは、CephFSに保存されているデータにSambaからアクセスできるようにします。

## 1.3 Cephのストレージ構造

### 1.3.1 プール

Cephクラスタに保存されるオブジェクトは「プール」に配置されます。プールは、外部環境に対してはクラスタの論理パーティションを表します。各プールに対して一連のルールを定義できます。たとえば、各オブジェクトのレプリケーションがいくつ存在する必要があるかななどを定義できます。プールの標準設定を「複製プール」と呼びます。

通常、プールにはオブジェクトが含まれていますが、RAID 5と同様の動作をするように設定することもできます。この設定の場合、オブジェクトは追加のコーディングチャンクと共にチャンクで保存されます。コーディングチャンクには冗長な情報が含まれます。データとコーディングチャンクの数管理者が定義できます。この設定の場合、プールは「イレージャコーディングプール(ECプール)」と呼ばれます。

### 1.3.2 配置グループ

「配置グループ」(PG)は、プール内でデータを分散するために使用されます。プールの作成時に、一定数の配置グループが設定されます。配置グループは、オブジェクトをグループ化するために内部的に使用され、Cephクラスタのパフォーマンスにおける重要な要因です。オブジェクトのPGはオブジェクトの名前によって決定されます。

### 1.3.3 例

このセクションでは、Cephのデータ管理の仕組みを簡単な例で説明します(図1.2「小規模なCephの例」を参照してください)。この例は、Cephクラスタの推奨設定を表すものではありません。このハードウェアセットアップは、3つのストレージノードまたはCeph OSD (ホスト1、ホスト2、ホスト3)で構成されます。各ノードにはハードディスクが3つあり、それぞれがOSDとして使用されます(osd.1～osd.9)。この例では、Ceph Monitorノードを無視しています。



#### 注記: Ceph OSDとOSDの違い

「Ceph OSD」または「Ceph OSDデーモン」は、ノード上で実行されるデーモンを指すのに対し、「OSD」という語はそのデーモンが対話する論理ディスクを指します。

クラスタにはプールAとプールBの2つのプールがあります。プールAはオブジェクトを2回だけ複製するのに対し、プールBの災害耐性はより重要であるため、各オブジェクトのレプリケーションを3つ保持します。

たとえばREST API経由でアプリケーションがオブジェクトをプールに配置すると、プールとオブジェクト名に基づいて配置グループ(PG1～PG4)が選択されます。続いて、CRUSHアルゴリズムにより、オブジェクトが含まれている配置グループに基づいて、オブジェクトを保存するOSDが計算されます。

この例では、障害ドメインはホストに設定されています。これにより、オブジェクトのレプリケーションが確実に別のホストに保存されるようにします。プールに設定されているレプリケーションレベルに応じて、オブジェクトは、配置グループによって使用される2つまたは3つのOSDに保存されます。

オブジェクトを書き込むアプリケーションは、プライマリCeph OSDである1つのCeph OSDとのみ対話します。プライマリCeph OSDはレプリケーションを処理し、他のすべてのOSDがオブジェクトを保存したら、書き込みプロセスの完了を確認します。

osd.5に障害が発生した場合、PG1のすべてのオブジェクトはosd.1で引き続き利用可能です。OSDに障害が発生したことをクラスタが認識するとすぐに、別のOSDが処理を引き継ぎます。この例では、osd.4がosd.5の代わりとして使用されます。その後、osd.1に保存されているオブジェクトがosd.4に複製され、レプリケーションレベルが復元されます。

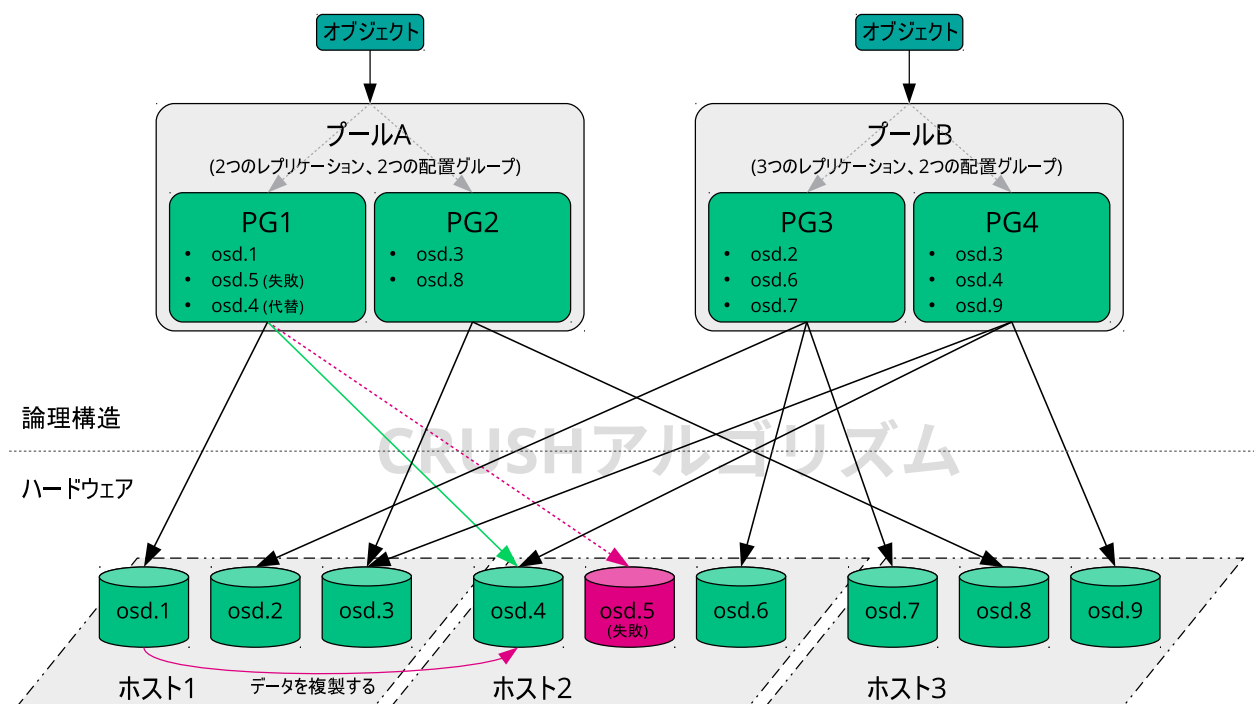


図 1.2: 小規模なCEPHの例

新しいOSDを持つ新しいノードがクラスタに追加されると、クラスタマップが変更されます。それに従って、CRUSH機能はオブジェクトに対して別の場所を返します。新しい場所を受け取ったオブジェクトは、別の場所に移動されます。このプロセスにより、すべてのOSDがバランス良く使用されます。

## 1.4 BlueStore

SES 5から、BlueStoreがCephの新しいデフォルトストレージバックエンドになりました。BlueStoreはFileStoreよりもパフォーマンスが高く、データの完全なチェックサムや組み込みの圧縮機能を備えています。

BlueStoreは、1つ、2つ、または3つのいずれかのストレージデバイスを管理します。最もシンプルなケースでは、BlueStoreは1つのプライマリストレージデバイスを使用します。通常、ストレージデバイスは、次の2つの部分にパーティション分割されます。

1. BlueFSという名前の小容量のパーティション。RocksDBで必要な、ファイルシステムに似た機能を実装します。
2. 通常、デバイスの残りの部分は、その部分を占有する大容量のパーティションになります。これはBlueStoreによって直接管理され、実際のデータがすべて含まれます。通常、このプライマリデバイスは、データディレクトリ内ではブロックシンボリックリンクで識別されます。

次のように、2つの追加デバイスにわたってBlueStoreを展開することもできます。

「WALデバイス」は、BlueStoreの内部ジャーナルまたは先書きログに使用できます。これは、データディレクトリでは、シンボリックリンク`block.wal`で識別されます。別個のWALデバイスを使用すると役立つのは、そのデバイスがプライマリデバイスまたはDBデバイスより高速な場合だけです。たとえば、次のような場合です。

- WALデバイスがNVMeで、DBデバイスがSSD、データデバイスがSSDまたはHDDである。
- WALデバイスとDBデバイスの両方が別個のSSDで、データデバイスがSSDまたはHDDである。

「DBデバイス」は、BlueStoreの内部メタデータを保存するために使用できます。BlueStore (または埋め込みのRocksDB)は、パフォーマンスを向上させるため、できる限り多くのメタデータをDBデバイス上に配置します。ここでも、共有DBデバイスをプロビジョニングすると役に立つのは、そのデバイスがプライマリデバイスより高速な場合だけです。





### ヒント: DBサイズの計画

DBデバイスが十分なサイズになるよう慎重に計画してください。DBデバイスがいっぱいになると、メタデータがプライマリデバイスにあふれ、OSDのパフォーマンスが大きく低下します。

`ceph daemon osd.ID perf dump`コマンドを使用して、WAL/DBパーティションがいっぱいであふれそうかどうかを調べることができます。`slow_used_bytes`の値に、あふれているデータの量が表示されます。

```
cephuser@adm > ceph daemon osd.ID perf dump | jq '.bluefs'
{
  "db_total_bytes": 1073741824,
  "db_used_bytes": 33554432,
  "wal_total_bytes": 0,
  "wal_used_bytes": 0,
  "slow_total_bytes": 554432,
  "slow_used_bytes": 554432,
}
```

## 1.5 追加情報

- コミュニティプロジェクトとして、Cephには、独自の広範なオンラインヘルプがあります。このマニュアルに記載されていないトピックについては、<https://docs.ceph.com/en/octopus/>  を参照してください。
- S.A. Weil、S.A. Brandt、E.L. Miller、C. Maltzahnによる元のドキュメント『CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data』には、Cephの内部動作に関する有益な洞察が記載されています。特に大規模クラスタを展開する場合には、これを一読することをお勧めします。このドキュメントは<http://www.ssrc.ucsc.edu/papers/weil-sc06.pdf>  にあります。
- SUSE Enterprise Storageは、SUSE OpenStack以外の配布パッケージで使用できません。Cephクライアントは、SUSE Enterprise Storageと互換性があるレベルである必要があります。



### 注記

SUSEはCeph展開のサーバコンポーネントをサポートし、クライアントはOpenStack配布パッケージベンダーによってサポートされます。

## 2 ハードウェア要件と推奨事項

Cephのハードウェア要件は、IOワークロードに大きく依存します。次のハードウェア要件と推奨事項は、詳細な計画を立てる際の起点と考えてください。

一般的に、このセクションで説明する推奨事項はプロセスごとの推奨事項です。同じマシンに複数のプロセスがある場合は、CPU、RAM、ディスク、およびネットワークの各要件を追加する必要があります。

### 2.1 ネットワーク概要

Cephには次のような論理ネットワークが含まれます。

- パブリックネットワークと呼ばれるフロントエンドのネットワーク
- クラスタネットワークと呼ばれる、バックエンドの信頼済み内部ネットワークこれは必要な場合のみ指定してください。
- 1つ以上のゲートウェイ用クライアントネットワーク。これは必要な場合のみ指定してください。また、この章では扱いません。

パブリックネットワークはCephデーモンが他のCephデーモンやクライアントと通信するネットワークです。つまり、クラスタネットワークが構成されている場合を除き、すべてのCephクラスタのトラフィックはこのパブリックネットワークを通ります。

クラスタネットワークはOSDノード間のバックエンドネットワークであり、レプリケーション、再バランシング、リカバリに使用されます。このオプションのネットワークを構成した場合、理論上、デフォルトの3方向レプリケーションを使用するパブリックネットワークの2倍の帯域幅が生まれます。これは、プライマリOSDが2つのコピーを他のOSDに送る際に、このネットワークを使用するためです。一方、パブリックネットワークはクライアントとゲートウェイ間のネットワークであり、Monitor、マネージャ、MDSノード、OSDノードとの通信に使用されます。パブリックネットワークはMonitor、マネージャ、MDSノードがOSDノードと通信する際にも使用されます。

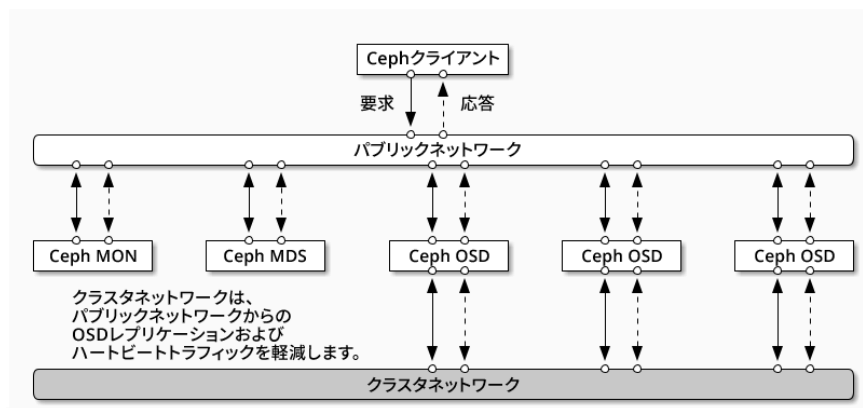


図 2.1: ネットワーク概要

### 2.1.1 ネットワーク推奨事項

十分に要件を満たせるような帯域幅を備えた、フォールトトレラントな単一のネットワークをお勧めします。Cephパブリックネットワーク環境では、802.3ad (LACP)を使用してボンディングされた2つの25GbE (またはそれ以上)のネットワークインターフェイスをお勧めします。この環境がCephの最小セットアップとみなされます。クラスタネットワークを使用する場合、25GbEのネットワークインターフェイスを4つボンディングすることをお勧めします。2つ以上のネットワークインターフェイスをボンディングすることで、リンクアグリゲーションによりスループットが改善されます。さらに、リンクとスイッチに冗長性が与えられるため、耐障害性と保守性が向上します。

VLANを作成することで、ボンディングした機器を通過する異なるタイプのトラフィックを分離することもできます。たとえば、1つはパブリックネットワーク用、もう1つはクラスタネットワーク用として、計2つのVLANインターフェイスを提供するようにボンディングを設定できます。しかしながら、これはCephネットワークを構成する際の必須事項ではありません。ネットワークインターフェイスのボンディングの詳細については、<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-network.html#sec-network-iface-bonding>を参照してください。

耐障害性はコンポーネントを障害ドメインに隔離することで強化できます。ネットワークの耐障害性を改善する目的で、別々のネットワークインターフェイスカード(NIC)2枚を1つのインターフェイスにボンディングすると、単一のNICの障害に対する保護が提供されます。同様に、2つのスイッチをボンディングすると、単一のスイッチの障害に対する保護が提供されます。必要なレベルの耐障害性を構築するために、ネットワーク機器のベンダーに相談することをお勧めします。

## ！ 重要: 管理ネットワークはサポート対象外

管理ネットワークのセットアップ(たとえば、分離されたSSH、Salt、またはDNSのネットワーク形成を可能とするもの)の追加はテストされておらず、サポート対象外です。

## 💡 ヒント: DHCP経由で設定されたノード

ストレージノードがDHCP経由で設定されている場合、さまざまCephデーモンが起動する前にネットワークを正しく設定するのにデフォルトのタイムアウトでは十分でないことがあります。この場合、CephのMONとOSDを正常に起動できません(**systemctl status ceph\\***を実行すると「unable to bind」エラーになります)。この問題を回避するには、ストレージクラスタの各ノードでDHCPクライアントのタイムアウト時間を少なくとも30秒まで延ばすことをお勧めします。このためには、各ノードで以下の設定を変更します。

`/etc/sysconfig/network/dhcp`で、以下を設定します。

```
DHCLIENT_WAIT_AT_BOOT="30"
```

`/etc/sysconfig/network/config`で、以下を設定します。

```
WAIT_FOR_INTERFACES="60"
```

### 2.1.1.1 実行中のクラスタにプライベートネットワークを追加

Cephの展開中にクラスタネットワークを指定しない場合、単一のパブリックネットワーク環境と想定されます。Cephはパブリックネットワークで問題なく動作しますが、2つ目のプライベートクラスタネットワークを設定すると、パフォーマンスとセキュリティが向上します。2つのネットワークをサポートするには、各Cephノードに少なくとも2つのネットワークカードが必要です。

各Cephノードに以下の変更を適用する必要があります。これらの変更は、小規模なクラスタの場合は比較的短時間で済みますが、数百または数千のノードで構成されるクラスタの場合は非常に時間がかかる可能性があります。

1. 次のコマンドを使用して、クラスタネットワークを設定します。

```
root # ceph config set global cluster_network MY_NETWORK
```

OSDを再起動し、指定したクラスタネットワークにバインドします。

```
root # systemctl restart ceph-*@osd.*.service
```

2. プライベートクラスタネットワークがOSレベルで想定どおりに動作していることを確認します。

### 2.1.1.2 異なるサブネット上のモニタノード

モニタノードが複数のサブネット上に存在する場合(たとえば、別の部屋に配置されていたり、別のスイッチによってサービスを提供されていたりする場合)、CIDR表記でパブリックネットワークアドレスを指定する必要があります。

```
cephuser@adm > ceph config set mon public_network  
"MON_NETWORK_1, MON_NETWORK_2, MON_NETWORK_N"
```

以下に例を示します。

```
cephuser@adm > ceph config set mon public_network "192.168.1.0/24, 10.10.0.0/16"
```



#### 警告

このセクションで説明するように、パブリックネットワーク(またはクラスタネットワーク)用に複数のネットワークセグメントを指定する場合、これらのサブネットはいずれも他のすべてのサブネットにルーティングできる必要があります。さもなければ、異なるネットワークセグメント上に存在するMONなどのCephデーモンが通信できず、クラスタが分割されてしまいます。また、ファイアウォールを使用している場合、必要に応じて、iptablesに各IPアドレスやサブネットを含め、すべてのノードでこれらのアドレスに対してポートを開放したかを確認してください。

## 2.2 複数のアーキテクチャの設定

SUSE Enterprise Storageでは、x86とArmの両方のアーキテクチャをサポートしています。各アーキテクチャを検討する際は、OSDあたりのコア数、周波数、およびRAMの観点から見ると、サイジングに関して、CPUアーキテクチャ間に実質的な差異はないことに注意することが重要です。

小型のx86プロセッサ(サーバ以外)と同様に、パフォーマンスの低いArmベースのコアは、特にイレージャコーディングプールに使用する場合は最適なエクスペリエンスを提供できない可能性があります。



## 注記

このドキュメントでは、x86やArmと書く代わりにSYSTEM-ARCHと表記します。

## 2.3 ハードウェア設定

最高のプロダクトエクスペリエンスのために、推奨されるクラスタ構成から始めることをお勧めします。テスト用クラスタや、それほどパフォーマンスが要求されないクラスタのために、サポートされる最小限のクラスタ設定を記載しています。

### 2.3.1 最小クラスタ構成

最小限の運用クラスタは、次のような構成です。

- サービスを共同配置した、最低でも4つの物理ノード(OSDノード)
- ボンディングされたネットワークを構成する、デュアルポート10Gb Ethernet
- 分離された管理ノード(外部ノードに仮想化してもよい)

詳細な構成は以下の通りです。

- 4GBのRAM、4コア、1TBのストレージ容量を備えた別個の管理ノード。これは通常、Salt Masterノードです。Ceph Monitor、メタデータサーバ、Ceph OSD、Object Gateway、NFS GaneshaなどのCephサービスとゲートウェイは管理ノードではサポートされません。これは、管理ノードがクラスタの更新をオーケストレートし、個別にプロセスをアップグレードする必要があるためです。
- それぞれ8つのOSDディスクを備えた、最低でも4つの物理OSDノード。要件は2.4.1項「Minimum requirements」を参照してください。  
クラスタの合計容量は、1つのノードが使用不能になることを想定したサイズとする必要があります。また、使用済み容量(冗長分を含む)の合計が80%を超えないようにする必要があります。
- 3つのCeph MonitorインスタンスMonitorはレイテンシの問題からHDDではなくSSD/NVMeストレージから実行する必要があります。
- Monitor、メタデータサーバ、ゲートウェイは同じOSDノードに配置できます。Monitorの共同配置については、2.12項「OSDとモニタでの1台のサーバの共有」を参照してください。サービスを共同配置する場合、メモリとCPUの要件を積み増す必要があります。

- iSCSI Gateway、Object Gateway、メタデータサーバは最低でも4GBのRAMと4コアの追加が必要です。
- CephFS、S3/Swift、iSCSIを使用する場合、冗長性と可用性を確保するため、それに応じた役割のインスタンス(メタデータサーバ、Object Gateway、iSCSI)が最低でも2つ必要です。
- ノードはSUSE Enterprise Storage専用とし、他のいかなる物理的ワークロード、コンテナ化ワークロード、仮想化ワークロードにも使用しないでください。
- VM上になんらかのゲートウェイ(iSCSI、Object Gateway、NFS Ganesha、メタデータサーバなど)が展開される場合、こうしたVMをクラスタの他の役割を担う物理マシンにホストしないでください(ゲートウェイは共同配置サービスとしてサポートされているため、これは不要です)。
- コアとなる物理クラスタの外のハイパーバイザでサービスをVMとして展開する場合、障害ドメインは冗長性の確保を考慮する必要があります。  
たとえば、同じハイパーバイザに同じタイプの複数の役割(複数のMONやMDSインスタンスなど)を展開しないでください。
- VM上に展開する場合、優れたネットワーク接続性と上手く動作する時刻同期機能をノードに持たせることがきわめて重要です。
- ハイパーバイザノードは、他のワークロードがCPU、RAM、ネットワーク、ストレージなどのリソースを消費することによる干渉を避けるため、適切なサイズにする必要があります。

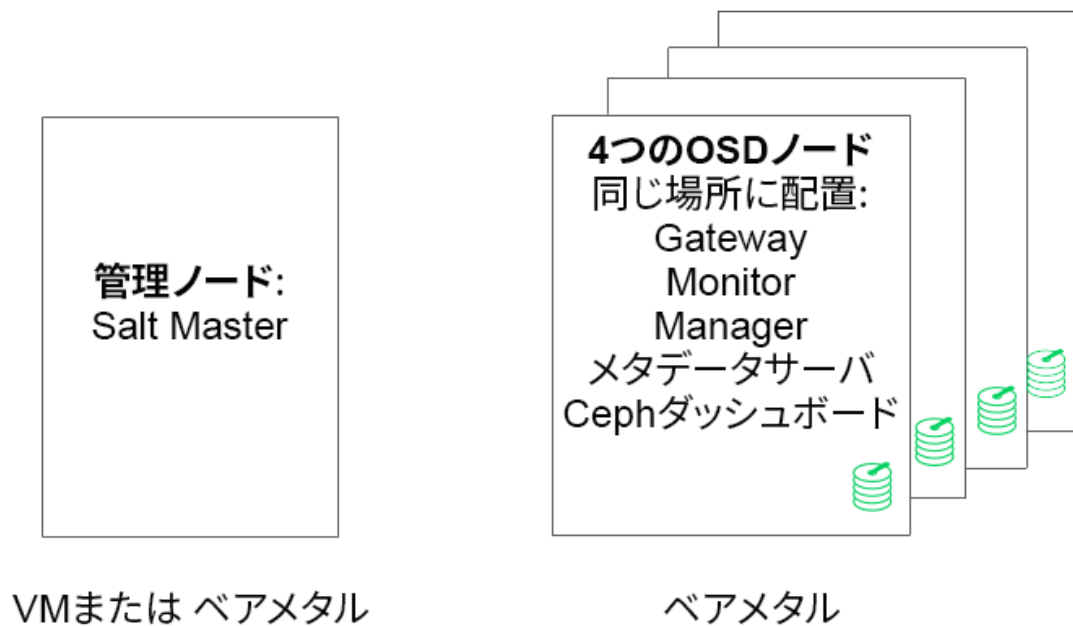


図 2.2: 最小クラスタ構成

### 2.3.2 運用クラスタの推奨設定

クラスタを拡張した場合、耐障害性を高めるため、Ceph Monitor、メタデータサーバ、ゲートウェイを別々のノードに再配置することをお勧めします。

- 7つのオブジェクトストレージノード
  - 1つのノードが最大で合計ストレージの15%を超えないこと。
  - クラスタの合計容量は、1つのノードが使用不能になることを想定したサイズとする必要があります。また、使用済み容量(冗長分を含む)の合計が80%を超えないようにする必要があります。
  - 内部クラスタと外部パブリックネットワークをそれぞれボンディングする、25Gb Ethernet(またはそれ以上)。
  - Storage Clusterあたり56以上のOSD
  - 推奨設定の詳細については2.4.1項「[Minimum requirements](#)」を参照してください。
- 専用の物理インフラストラクチャノード

- 3つのCeph Monitorノード: 4GBのRAM、4コアプロセッサ、ディスク用のRAID 1 SSD。  
推奨設定の詳細については2.5項「Monitorノード」を参照してください。
- Object Gatewayノード: 32GBのRAM、8コアプロセッサ、ディスク用のRAID 1 SSD。  
推奨設定の詳細については2.6項「Object Gatewayノード」を参照してください。
- iSCSI Gatewayノード: 16GBのRAM、8コアプロセッサ、ディスク用のRAID 1 SSD。  
推奨設定の詳細については2.9項「iSCSI Gatewayノード」を参照してください。
- メタデータサーバノード(アクティブ x 1/ホットスタンバイ x 1): 32GBのRAM、8コアプロセッサ、ディスク用のRAID 1 SSD。  
推奨設定の詳細については2.7項「メタデータサーバノード」を参照してください。
- 1つのSES管理ノード: 4GBのRAM、4コアプロセッサ、ディスク用のRAID1 SSD。

### 2.3.3 マルチパス設定

マルチパスハードウェアを使用したい場合、設定ファイルのdevicesセクションでLVMがmultipath\_component\_detection = 1を認識するようにしてください。この設定は、**lvm config**コマンドで確認できます。

もしくは、LVMのフィルタ設定により、LVMがデバイスのmpathコンポーネントをフィルタするようにしてください。これはホスト固有です。



#### 注記

この方法はお勧めしません。multipath\_component\_detection = 1を設定できない場合にのみ検討する必要があります。

マルチパス設定の詳細については、<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-multipath.html#sec-multipath-lvm> を参照してください。

## 2.4 オブジェクトストレージノード

### 2.4.1 Minimum requirements

- 次のCPUの推奨事項は、Cephによる使用とは関係なくデバイスを考慮しています。
  - スピナあたり1つの2GHz CPUスレッド
  - SSDあたり2つの2GHz CPUスレッド
  - MVMaあたり4つの2GHz CPUスレッド
- 独立した10GbEネットワーク(パブリック/クライアントおよび内部)。4つの10GbEが必須、2つの25GbEを推奨。
- 必要なRAMの合計 = OSDの数 x (1GB + `osd_memory_target`) + 16GB  
`osd_memory_target`の詳細については、『運用と管理ガイド』、第28章「Cephクラスタの設定」、28.4.1項「自動キャッシュサイズ調整の設定」を参照してください。
- JBOD設定または個々のRAID-0設定のOSDディスク。
- OSDジャーナルはOSDディスクに配置可能。
- OSDディスクはSUSE Enterprise Storage専用である必要があります。
- オペレーティングシステム専用のディスクおよびSSD(できればRAID 1設定)。
- このOSDホストがキャッシュ階層化用のキャッシュプールの一部をホストする場合、最低でも4GBのRAMを追加で割り当て。
- Ceph Monitor、ゲートウェイ、およびメタデータサーバはオブジェクトストレージノードに配置可能。
- ディスク性能の理由から、OSDノードはベアメタルノードです。Ceph MonitorとCeph Managerの最小セットアップを除き、OSDノードで他のワークロードを実行しないでください。
- ジャーナル用のSSD(SSDジャーナルとOSDの比率は6:1)。



#### 注記

OSDノードにiSCSIやRADOS Block Deviceなどのネットワークブロックデバイスがマッピングされていないことを確認してください。

## 2.4.2 最小ディスクサイズ

OSD上で実行する必要があるディスク領域には2つのタイプがあります。WAL/DBデバイス用の領域と、保存データ用のプライマリ領域です。WAL/DBの最小値(デフォルト値)は6GBです。データ用の最小領域は5GBです。これは、5GB未満のパーティションには自動的に重み0が割り当てられるためです。

したがって、OSDの最小ディスク領域は11GBになりますが、テスト目的であっても20GB未満のディスクはお勧めしません。

## 2.4.3 BlueStoreのWALおよびDBデバイスの推奨サイズ



### ヒント: 詳細情報

BlueStoreの詳細については、[1.4項「BlueStore」](#)を参照してください。

- WALデバイス用に4GBを予約することをお勧めします。DBの推奨サイズは、ほとんどのワークロードにおいて64MBです。



### 重要

高負荷な展開では、DB容量をさらに大きくすることをお勧めします。特に、RGWやCephFSの使用量が高い場合です。必要に応じて、DB領域拡張用のハードウェアを設置する収容能力(スロット)をある程度確保してください。

- WALとDBデバイスを同じディスクに配置する予定の場合は、それぞれに別個のパーティションを設けるのではなく、両方のデバイスに対して単一のパーティションを使用することをお勧めします。これにより、CephはDBデバイスをWAL操作にも使用できます。Cephは必要時にのみDBパーティションをWALに使用するので、ディスク領域の管理が効率化します。もう1つの利点は、WALパーティションがいっぱいになる可能性は極めて低く、完全に使い切っていなければ、その領域が無駄になることはなく、DB操作に使用されることです。

DBデバイスをWALと共有するには、WALデバイスを「指定しない」で、DBデバイスのみを指定します。

OSDレイアウトを指定する方法の詳細については、『運用と管理ガイド』、第13章「運用タスク」、13.4.3項「DriveGroups仕様を用いたOSDの追加」を参照してください。

## 2.4.4 WAL/DBパーティション用SSD

SSD (ソリッドステートドライブ)には可動部品がありません。これは、ランダムアクセス時間と読み込みレイテンシを短縮すると同時に、データスループットを加速します。SSDの1MBあたりの価格は回転型ハードディスクの価格より大幅に高いため、SSDは小容量のストレージにのみ適しています。

WAL/DBパーティションをSSDに保存し、オブジェクトデータは別個のハードディスクに保存することで、OSDのパフォーマンスを大幅に向上させることができます。



### ヒント: 複数のWAL/DBパーティションでSSDを共有

WAL/DBパーティションは比較的小さい領域しか占有しないため、1つのSSDディスクを複数のWAL/DBパーティションで共有できます。WAL/DBパーティションを増やすほど、SSDディスクのパフォーマンスが低下することに注意してください。同じSSDディスクでWAL/DBパーティションを7つ以上共有することはお勧めしません。NVMeディスクの場合、13個以上の共有はお勧めしません。

## 2.4.5 推奨されるディスクの最大数

1台のサーバで、そのサーバで使える数だけのディスクを使用できます。サーバあたりのディスク数を計画する際には、考慮すべき点があります。

- 「ネットワーク帯域幅」サーバのディスクが増えるほど、ディスク書き込み操作のためにネットワークカード経由で転送しなければならないデータが増えます。
- 「メモリ」2GBを超えるRAMは、BlueStoreキャッシュに使用されません。`osd_memory_target`のデフォルトである4GBを使用すると、回転型メディアに適したキャッシュ開始サイズがシステムに設定されます。SSDまたはNVMEを使用する場合は、OSDあたりのキャッシュサイズとRAM割り当てを増やすことを検討します。
- 「耐障害性」サーバ全体に障害が発生した場合、搭載ディスクの数が多いほど、クラスタが一時的に失うOSDが増えます。さらに、レプリケーションルールを実行し続けるために、障害が発生したサーバからクラスタ内のほかのノードにすべてのデータをコピーする必要があります。

## 2.5 Monitorノード

- 少なくとも3つのMONノードが必要です。モニタの数は常に奇数( $1+2n$ )である必要があります。
- 4GBのRAM。
- 4つの論理コアを持つプロセッサ。
- 特に各モニタノードの`/var/lib/ceph`パスには、SSDか、その他の十分高速なストレージタイプを強くお勧めします。ディスクのレイテンシが大きいと、クォーラムが不安定になる可能性があるためです。冗長性を確保するには、RAID 1設定で2台のディスクを使用することをお勧めします。モニタが利用可能なディスク領域をログファイルの増大などから保護するため、モニタプロセスには、別個のディスク、または少なくとも別個のディスクパーティションを使用することをお勧めします。
- 各ノードのモニタプロセスは1つだけにする必要があります。
- OSDノード、MON、またはObject Gatewayノードを混在させることは、十分なハードウェアリソースが利用可能な場合にのみサポートされます。つまり、すべてのサービスの要件を合計する必要があります。
- 複数のスイッチにボンディングされた2つのネットワークインタフェース。

## 2.6 Object Gatewayノード

Object Gatewayノードには最低でも6コアCPUと32GBのRAMを使用する必要があります。同じマシンに他のプロセスも配置されている場合、それらの要件を合計する必要があります。

## 2.7 メタデータサーバノード

メタデータサーバノードの適切なサイズは、具体的な使用事例によって異なります。一般的には、メタデータサーバが処理する開いているファイルが多いほど、より多くのCPUとRAMが必要になります。最小要件は次のとおりです。

- 各メタデータサーバドメインに対して4GBのRAM。
- ボンディングされた2つのネットワークインタフェース。
- 2個以上のコアを持つ2.5GHzのCPU。

## 2.8 管理ノード

少なくとも4GBのRAMとクアッドコアCPUが必要です。これには、管理ノードにおけるSalt Masterの実行が含まれます。数百のノードで構成される大規模クラスタでは、6GBのRAMをお勧めします。

## 2.9 iSCSI Gatewayノード

iSCSI Gatewayノードには最低でも6コアCPUと16GBのRAMを使用すべきです。

## 2.10 SESおよび、その他のSUSE製品

このセクションには、SESと他のSUSE製品の統合に関する重要な情報が記載されています。

### 2.10.1 SUSE Manager

SUSE ManagerとSUSE Enterprise Storageは統合されていません。そのため、現在のところSUSE ManagerでSESクラスタを管理することはできません。

## 2.11 名前の制限

一般的に、Cephは、設定ファイル、プール名、ユーザ名などでASCII以外の文字をサポートしません。Cephクラスタを設定する場合、すべてのCephオブジェクト名/設定名で単純な英数字の文字(A～Z、a～z、0～9)と、最小限の句読点(「.」「-」「\_」)のみを使用することをお勧めします。

## 2.12 OSDとモニタでの1台のサーバの共有

テスト環境ではOSDとMONを同じサーバで実行することは技術的に可能ですが、運用ではモニタノードごとに別個のサーバを用意することを強くお勧めします。その主な理由はパフォーマンスです。クラスタのOSDが増えるほど、MONノードが実行しなければならないI/O操作が増えます。さらに、1台のサーバをMONノードとOSDで共有する場合、OSDのI/O操作がモニタノードにとって制限要因になります。

考慮すべきもう1つの点は、サーバ上のOSD、MONノード、およびオペレーティングシステムでディスクを共有するかどうかです。答えは単純です。可能であれば、OSDには別個の専用ディスクを使用し、モニタノードには別個の専用サーバを使用します。

CephはディレクトリベースのOSDをサポートしますが、OSDには、常にオペレーティングシステムのディスクではなく専用ディスクを使用する必要があります。



## ヒント

OSDとMONノードをどうしても同じサーバで実行する必要がある場合は、MONを別個のディスクで実行し、そのディスクを/var/lib/ceph/monディレクトリにマウントすることで、少しでもパフォーマンスを向上させます。

## 3 管理ノードのHAセットアップ

「管理ノード」はSalt Masterサービスが動作するCephクラスタノードです。管理ノードはSalt Minionサービスのクエリと命令を行うことで、他のクラスタノードを管理します。管理ノードには通常、ほかのサービスも含まれます。たとえば、「Prometheus」監視ツールキットに支援された「Grafana」ダッシュボードなどです。

管理ノードに障害が発生した場合、通常は、動作する新しいハードウェアをノードに提供し、最新のバックアップから完全なクラスタ設定スタックを復元する必要があります。この方法では時間がかかり、クラスタの停止も発生します。

管理ノードの障害によってCephクラスタのパフォーマンスにダウンタイムが発生するのを防止するため、Ceph管理ノードにはHA (高可用性)クラスタを利用することをお勧めします。

### 3.1 管理ノードのHAクラスタの概要

HAクラスタは、一方のクラスタノードで障害が発生した場合に、もう一方のノードがその役割(仮想化された管理ノードを含む)を自動的に引き継ぐという考えです。こうすることで、管理ノードに障害が発生したことを他のCephクラスタノードが認識することはありません。

管理ノード用の最小限のHAソリューションには、次のハードウェアが必要です。

- High Availability ExtensionをインストールしたSUSE Linux Enterpriseを実行し、管理ノードを仮想化できるベアメタルサーバ2台。
- 2つ以上の冗長ネットワーク通信パス。たとえば、ネットワークデバイスボンディングを使用します。
- 管理ノード仮想マシンのディスクイメージをホストするための共有ストレージ。この共有ストレージは両方のサーバからアクセス可能である必要があります。たとえば、NFSエクスポート、Samba共有、iSCSIターゲットなどを使用できます。

クラスタの要件の詳細については、<https://documentation.suse.com/sle-ha/15-SP2/html/SLE-HA-all/art-sleha-install-quick.html#sec-ha-inst-quick-req> を参照してください。

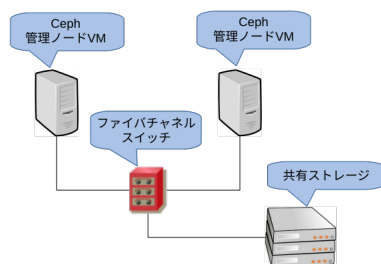




図 3.1: 管理ノードの2ノードHAクラスタ

## 3.2 管理ノードを有するHAクラスタの構築

次の手順は、管理ノードを仮想化するためのHAクラスタを構築する手順の中で最も重要なものをまとめたものです。詳細については、記載のリンクを参照してください。

1. 共有ストレージを使用する基本的な2ノードHAクラスタを設定します。 <https://documentation.suse.com/sle-ha/15-SP2/html/SLE-HA-all/art-sleha-install-quick.html> を参照してください。
2. 両方のクラスタノードに、KVMハイパーバイザを実行するために必要なすべてのパッケージとlibvirtツールキットをインストールします。 <https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-vt-installation.html#sec-vt-installation-kvm> を参照してください。
3. 1つ目のクラスタノードで、libvirtを利用する新しいKVM VM (仮想マシン)を作成します。 <https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-kvm-inst.html#sec-libvirt-inst-virt-install> を参照してください。事前設定済みの共有ストレージを使用して、VMのディスクイメージを保存します。
4. VMのセットアップが完了したら、その設定を共有ストレージ上のXMLファイルにエクスポートします。以下の構文を使用してください。

```
root # virsh dumpxml VM_NAME > /path/to/shared/vm_name.xml
```

5. 管理ノードVMのリソースを作成します。HAリソースの作成に関する全般的な情報については、<https://documentation.suse.com/sle-ha/15-SP2/html/SLE-HA-all/cha-conf-hawk2.html>  を参照してください。KVM仮想マシンのリソースの作成に関する詳細情報については、[http://www.linux-ha.org/wiki/VirtualDomain\\_%28resource\\_agent%29](http://www.linux-ha.org/wiki/VirtualDomain_%28resource_agent%29)  を参照してください。
6. 新しく作成したVMゲストで、管理ノードと、そこで必要な追加サービスを展開します。5.2項「Saltの展開」の関連手順に従います。同時に、非HAクラスタサーバに残りのCephクラスタノードを展開します。

## II Cephクラスタの展開

4 導入と共通タスク 29

5 cephadmによる展開 30

## 4 導入と共通タスク

SUSE Enterprise Storage 7から、CephサービスはRPMパッケージではなく、cephadmを利用したコンテナとして展開されます。詳細については、第5章「cephadmによる展開」を参照してください。

### 4.1 リリースノートの確認

リリースノートには、旧リリースのSUSE Enterprise Storageからの変更点に関する追加情報が記載されています。リリースノートを参照して以下を確認します。

- 使用しているハードウェアに特別な配慮が必要かどうか
- 使用しているソフトウェアパッケージに大幅な変更があるかどうか
- インストールのために特別な注意が必要かどうか

リリースノートには、マニュアルに記載できなかった情報が記載されています。また、既知の問題に関する注意も記載されています。

パッケージ `release-notes-ses` をインストールすると、リリースノートは、ローカルではディレクトリ `/usr/share/doc/release-notes` に、オンラインでは <https://www.suse.com/releasesnotes/> にあります。

## 5 cephadmによる展開

SUSE Enterprise Storage 7はSaltベースの`ceph-salt`ツールを使用して、参加している各クラスタノードのオペレーティングシステムをcephadmを介した展開用に準備します。cephadmはSSHを介してCeph Managerデーモンからホストに接続してCephクラスタを展開、管理します。cephadmはCephクラスタのライフサイクル全体を管理します。ライフサイクルは単独のノード(1つのMONおよびMGRサービス)に小規模なクラスタをブートストラップすることから始まります。その後、オーケストレーションインターフェイスを使用して、クラスタをすべてのホストを含むように拡張するとともに、Cephサービスをすべてプロビジョニングします。この作業はCeph CLI(コマンドラインインターフェイス)から実施できます。一部はCephダッシュボード(GUI)からも行えます。

### ！ 重要

Cephコミュニティのドキュメントでは、最初の展開の際に**`cephadm bootstrap`**コマンドを使用していることに注意してください。**`cephadm bootstrap`**コマンドは`ceph-salt`から呼び出されます。直接実行しないでください。**`cephadm bootstrap`**を使用する手動のCephクラスタ展開はサポートされていません。

cephadmを使用してCephクラスタを展開するには、以下のタスクを実行する必要があります。

1. すべてのクラスタノードで、ベースとなるオペレーティングシステム(SUSE Linux Enterprise Server 15 SP2)のインストールと基本的な設定を行います。
2. `ceph-salt`を介した初期展開の準備のために、すべてのクラスタノードにSaltインフラストラクチャを展開します。
3. `ceph-salt`経由でクラスタの基本的なプロパティを設定し、展開します。
4. cephadmを使用してクラスタに新しいノードと役割を追加し、サービスを展開します。

### 5.1 SUSE Linux Enterprise Serverのインストールと設定

1. 各クラスタノードでSUSE Linux Enterprise Server 15 SP2のインストールと登録を行います。SUSE Enterprise Storageのインストール中にアップデートリポジトリへのアクセスが必要なため、登録は必須です。最低でも、次のモジュールを導入します。

- Basesystem Module
- Server Applications Module

SUSE Linux Enterprise Serverのインストール方法の詳細については、<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-install.html> を参照してください。

2. 各クラスターノードに「SUSE Enterprise Storage 7」の拡張機能をインストールします。



### ヒント: SUSE Linux Enterprise Serverと共にSUSE Enterprise Storageをインストールする

SUSE Enterprise Storage 7の拡張機能は、SUSE Linux Enterprise Server 15 SP2のインストール後に分けてインストールすることも、SUSE Linux Enterprise Server 15 SP2のインストール手順の中で追加することもできます。

拡張機能のインストール方法の詳細については、<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-register-sle.html> を参照してください。

3. ネットワークを設定します。各ノードでDNS名が適切に解決されるようにする設定も含まれます。ネットワークの設定の詳細については、<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-network.html#sec-network-yast> を参照してください。DNSサーバの設定の詳細については、<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-dns.html> を参照してください。

## 5.2 Saltの展開

SUSE Enterprise Storageは最初のクラスターの準備にSaltと`ceph-salt`を使用します。Saltを使用すると、「Salt Master」と呼ばれる単独の専用ホストから複数のクラスターノードに対して、同時に設定やコマンドを実行できます。Saltの展開前に、次の重要な点を考慮してください。

- 「Salt Minion」は、Salt Masterと呼ばれる専用のノードによって制御されるノードです。
- 仮にSalt MasterホストがCephクラスターの一部である場合は、独自のSalt Minionを実行する必要があります。ただしこれは必須ではありません。



## ヒント: 1つのサーバで複数の役割を共有

各役割を別個のノードに展開すると、Cephクラスタで最適なパフォーマンスを実現できます。しかし、実際の展開では、1つのノードを複数の役割のために共有しなければならない場合があります。パフォーマンスやアップグレード手順で問題が起きないようにするため、Ceph OSD、メタデータサーバ、またはCeph Monitorの役割は管理ノードに展開しないでください。

- Salt Minionは、ネットワークでSalt Masterのホスト名を正しく解決する必要があります。Salt Minionは、デフォルトではsaltというホスト名を検索しますが、ネットワーク経由でアクセス可能なほかのホスト名を/etc/salt/minionファイルで指定できます。

1. Salt Masterノードにsalt-masterをインストールします。

```
root@master # zypper in salt-master
```

2. salt-masterサービスが有効になっていて起動していることを確認します。必要であれば、サービスを有効にして起動します。

```
root@master # systemctl enable salt-master.service
root@master # systemctl start salt-master.service
```

3. ファイアウォールを使用する場合は、Salt Masterノードのポート4505と4506がすべてのSalt Minionノードに対して開いていることを確認します。これらのポートが閉じている場合は、**yast2 firewall**コマンドを使用してポートを開き、salt-masterサービスに適切なゾーンを許可できます。たとえば、publicを許可します。

4. パッケージsalt-minionをすべてのミニオンノードにインストールします。

```
root@minion > zypper in salt-minion
```

5. /etc/salt/minionを編集し、次の行のコメントを解除します。

```
#log_level_logfile: warning
```

warningログレベルをinfoに変更します。



## 注記: log\_level\_logfileとlog\_level

`log_level`は、どのログメッセージが画面に表示されるかを制御します。一方、`log_level_logfile`は、どのログメッセージが`/var/log/salt/minion`に書き込まれるかを制御します。



## 注記

「すべて」のクラスタ(ミニオン)ノードのログレベルを変更したか確認してください。

- すべてのノードが他のノードの「完全修飾ドメイン名」をパブリッククラスタネットワークのIPアドレスに解決できることを確認します。
- すべてのミニオンをマスターに接続するように設定します。ホスト名`salt`でSalt Masterに接続できない場合は、ファイル`/etc/salt/minion`を編集するか、次の内容で新しいファイル`/etc/salt/minion.d/master.conf`を作成します。

```
master: host_name_of_salt_master
```

先に説明した設定ファイルを変更した場合は、すべての関連するSalt MinionのSaltサービスを再起動します。

```
root@minion > systemctl restart salt-minion.service
```

- すべてのノードで`salt-minion`サービスが有効になっていて起動していることを確認します。必要であれば、次のコマンドを使用して有効にして起動します。

```
root # systemctl enable salt-minion.service
root # systemctl start salt-minion.service
```

- 各Salt Minionの指紋を確認して、指紋が一致する場合、Salt Master上のすべてのSaltキーを受諾します。



## 注記

Salt Minionの指紋が空に戻る場合は、Salt MinionがSalt Masterの設定を持っていて、Salt Masterと通信できることを確認します。

各ミニオンの指紋を表示します。

```
root@minion > salt-call --local key.finger
local:
3f:a3:2f:3f:b4:d3:d9:24:49:ca:6b:2c:e1:6c:3f:c3:83:37:f0:aa:87:42:e8:ff...
```

すべてのSalt Minionの指紋を収集した後、Salt Master上の、受諾されていない全ミニオンキーの指紋を一覧にします。

```
root@master # salt-key -F
[...]
Unaccepted Keys:
minion1:
3f:a3:2f:3f:b4:d3:d9:24:49:ca:6b:2c:e1:6c:3f:c3:83:37:f0:aa:87:42:e8:ff...
```

ミニオンの指紋が一致する場合は、それらを受諾します。

```
root@master # salt-key --accept-all
```

10. キーが受諾されたことを確認します。

```
root@master # salt-key --list-all
```

11. すべてのSalt Minionが応答するかテストします。

```
root@master # salt-run manage.status
```

## 5.3 Cephクラスタの展開

このセクションでは、基本的なCephクラスタを展開する一連のプロセスを説明します。以下のサブセクションをよく読んで、記載されているコマンドを記載されている順番で実行してください。

### 5.3.1 ceph-saltのインストール

`ceph-salt`は`cephadm`に管理されるCephクラスタを展開するためのツールを提供します。`ceph-salt`はSaltインフラストラクチャを使用して、OSの管理(たとえば、ソフトウェアアップデートや時刻の同期)や、Salt Minionの役割の定義を行います。

Salt Master上で `ceph-salt` パッケージをインストールします。

```
root@master # zypper install ceph-salt
```

このコマンドは `ceph-salt-formula` を依存関係としてインストールします。この依存関係により、`/etc/salt/master.d`ディレクトリに追加のファイルを挿入することで、Salt Masterの設定が変更されます。変更を適用するには、`salt-master.service`を再起動し、Saltモジュールを同期させます。

```
root@master # systemctl restart salt-master.service
root@master # salt \* saltutil.sync_all
```

## 5.3.2 クラスタプロパティの設定

`ceph-salt config`コマンドを使用して、クラスタの基本的なプロパティを設定します。

### ！ 重要

`/etc/ceph/ceph.conf`ファイルは、`cephadm`で管理されており、ユーザは編集しないでください。Cephの設定パラメータは、新しい`ceph config`コマンドを使用して設定する必要があります。詳細については、『運用と管理ガイド』、第28章「Cephクラスタの設定」、28.2項「設定データベース」を参照してください。

### 5.3.2.1 ceph-saltシェルの使用

`ceph-salt config`をパスやサブコマンドを使わずに実行する場合、インタラクティブな`ceph-salt`シェルを入力します。このシェルは、1つのバッチで複数のプロパティを設定する必要がありますが、完全なコマンド構文を入力したくない場合に便利です。

```
root@master # ceph-salt config
/> ls
o- / ..... [...]
  o- ceph_cluster ..... [...]
    | o- minions ..... [no minions]
    | o- roles ..... [...]
    |   o- admin ..... [no minions]
    |   o- bootstrap ..... [no minion]
    |   o- cephadm ..... [no minions]
    |   o- tuned ..... [...]
    |     o- latency ..... [no minions]
    |     o- throughput ..... [no minions]
  o- cephadm_bootstrap ..... [...]
    | o- advanced ..... [...]
    | o- ceph_conf ..... [...]
```

```
| o- ceph_image_path ..... [ no image path]
| o- dashboard ..... [...]
| | o- force_password_update ..... [enabled]
| | o- password ..... [admin]
| | o- ssl_certificate ..... [not set]
| | o- ssl_certificate_key ..... [not set]
| | o- username ..... [admin]
| o- mon_ip ..... [not set]
o- containers ..... [...]
| o- registries_conf ..... [enabled]
| | o- registries ..... [empty]
| o- registry_auth ..... [...]
|   o- password ..... [not set]
|   o- registry ..... [not set]
|   o- username ..... [not set]
o- ssh ..... [no key pair set]
| o- private_key ..... [no private key set]
| o- public_key ..... [no public key set]
o- time_server ..... [enabled, no server host set]
  o- external_servers ..... [empty]
  o- servers ..... [empty]
  o- subnet ..... [not set]
```

ceph-saltのlsコマンドの出力を見るとわかるように、クラスタ構成がツリー構造に整理されます。ceph-saltシェルに含まれる、クラスタの特定のプロパティを設定するには、次の2つのオプションがあります。

- 現在位置からコマンドを実行し、第1引数としてプロパティへの絶対パスを入力する

```
/> /cephadm_bootstrap/dashboard ls
o- dashboard ..... [...]
  o- force_password_update ..... [enabled]
  o- password ..... [admin]
  o- ssl_certificate ..... [not set]
  o- ssl_certificate_key ..... [not set]
  o- username ..... [admin]
/> /cephadm_bootstrap/dashboard/username set ceph-admin
Value set.
```

- 設定する必要があるプロパティへのパスを変更してから、コマンドを実行する

```
/> cd /cephadm_bootstrap/dashboard/
/ceph_cluster/minions> ls
o- dashboard ..... [...]
  o- force_password_update ..... [enabled]
  o- password ..... [admin]
  o- ssl_certificate ..... [not set]
  o- ssl_certificate_key ..... [not set]
  o- username ..... [ceph-admin]
```



## ヒント: 設定スニペットの自動補完

`ceph-salt`シェルの中では自動補完機能を使用できます。これは、通常のLinuxシェル(Bash)の自動補完と同じようなものです。この機能は設定パス、サブコマンド、またはSalt Minion名を補完します。設定パスを自動補完する場合は、次の2つのオプションがあります。

- 現在位置からの相対的なパスをシェルに補完させる場合は、TABキー `<Tab>` を2回押します。
- シェルに絶対パスを補完させる場合は、`/` を入力してからTABキー `<Tab>` を2回押します。



## ヒント: 方向キーによる移動

`ceph-salt`シェルからパスを使用せずに`cd`コマンドを入力すると、ツリー構造のクラスタ構成が出力され、現在パスの行がアクティブになります。上下の方向キーを使用して、それぞれの行に移動できます。`Enter` を押して確定すると、アクティブ行に設定パスが変更されます。



## 重要: 表記

ドキュメントの整合性を維持するため、`ceph-salt`シェルを入力しない単一のコマンド構文を使用しています。たとえば、次のコマンドを使用してクラスタ構成のツリーを一覧にできます。

```
root@master # ceph-salt config ls
```

### 5.3.2.2 Salt Minionの追加

5.2項「Saltの展開」で展開し受諾したSalt Minionの全体またはサブセットをCephクラスタ構成に含めます。Salt Minionはフルネームで指定できます。また、「\*」と「?」のグロブ表現を使用することで複数のSalt Minionを同時に含めることもできます。`/ceph_cluster/minions`パスで`add`サブコマンドを使用します。次のコマンドは受諾済みのSalt Minionをすべて含めます。

```
root@master # ceph-salt config /ceph_cluster/minions add '*'
```

指定したSalt Minionが追加されたことを確認します。

```
root@master # ceph-salt config /ceph_cluster/minions ls
o- minions ..... [Minions: 5]
  o- ses-master.example.com ..... [no roles]
  o- ses-min1.example.com ..... [no roles]
  o- ses-min2.example.com ..... [no roles]
  o- ses-min3.example.com ..... [no roles]
  o- ses-min4.example.com ..... [no roles]
```

### 5.3.2.3 cephadmで管理するSalt Minionの指定

Cephクラスタに属し、cephadmで管理するノードを指定します。Cephサービスを実行するすべてのノードと、管理ノードを含めます。

```
root@master # ceph-salt config /ceph_cluster/roles/cephadm add '*'
```

### 5.3.2.4 管理ノードの指定

管理ノードは、`ceph.conf`設定ファイルとCeph管理キーリングがインストールされるノードです。通常、Ceph関連のコマンドは管理ノードで実行します。



#### ヒント: 同じノード上のSalt Masterと管理ノード

すべての、または、ほとんどのホストがSUSE Enterprise Storageに所属するような均質な環境では、Salt Masterと同じホストに管理ノードを置くことお勧めします。

あるSaltインフラストラクチャが複数のクラスタのホストとなるような異種環境(たとえば、SUSE Enterprise Storageと共にSUSE Managerを使用するような環境)では、Salt Masterと同じホストに管理ノードを置かないでください。

管理ノードを指定するには、次のコマンドを実行します。

```
root@master # ceph-salt config /ceph_cluster/roles/admin add ses-master.example.com
1 minion added.
root@master # ceph-salt config /ceph_cluster/roles/admin ls
o- admin ..... [Minions: 1]
  o- ses-master.example.com ..... [Other roles: cephadm]
```



## ヒント: ceph.confと管理キーリングを複数のノードにインストールする

展開が必要な場合は、Ceph設定ファイルと管理キーリングを複数のノードにインストールすることもできます。セキュリティ上の理由から、すべてのクラスタのノードにインストールすることは避けてください。

### 5.3.2.5 最初のMON/MGRノードの指定

クラスタをブートストラップするSalt Minionをクラスタ内から指定する必要があります。このミニオンはCeph MonitorとCeph Managerサービスを実行する最初のミニオンになります。

```
root@master # ceph-salt config /ceph_cluster/roles/bootstrap set ses-min1.example.com
Value set.
root@master # ceph-salt config /ceph_cluster/roles/bootstrap ls
o- bootstrap ..... [ses-min1.example.com]
```

さらに、`public_network`パラメータが正しく設定されていることを確認するために、パブリックネットワーク上のブートストラップMONのIPアドレスを指定する必要があります。たとえば、次のコマンドを実行します。

```
root@master # ceph-salt config /cephadm_bootstrap/mon_ip set 192.168.10.20
```

### 5.3.2.6 調整されるプロファイルの指定

クラスタの中から、アクティブに調整されるプロファイルを保有するミニオンを指定する必要があります。そのためには、次のコマンドを実行して役割を明示的に追加してください。



## 注記

1つのミニオンに`latency`と`throughput`の両方の役割を持たせることはできません。

```
root@master # ceph-salt config /ceph_cluster/roles/tuned/latency add ses-min1.example.com
Adding ses-min1.example.com...
1 minion added.
root@master # ceph-salt config /ceph_cluster/roles/tuned/throughput add ses-
min2.example.com
Adding ses-min2.example.com...
1 minion added.
```

### 5.3.2.7 SSHキーペアの生成

cephadmはSSHプロトコルを使用してクラスタノードと通信します。`cephadm`という名前のユーザアカウントが自動的に作成され、SSH通信に使用されます。

SSHキーペアの公開鍵と秘密鍵を生成する必要があります。

```
root@master # ceph-salt config /ssh generate
Key pair generated.
root@master # ceph-salt config /ssh ls
o- ssh ..... [Key Pair set]
  o- private_key ..... [53:b1:eb:65:d2:3a:ff:51:6c:e2:1b:ca:84:8e:0e:83]
  o- public_key ..... [53:b1:eb:65:d2:3a:ff:51:6c:e2:1b:ca:84:8e:0e:83]
```

### 5.3.2.8 タイムサーバの設定

すべてのクラスタノードは信頼できるタイムソースと時刻を同期する必要があります。時刻を同期するには、いくつかのシナリオがあります。

- 最適なNTPサービスを使用して時刻を同期するように、すべてのクラスタノードを設定済みの場合、タイムサーバ処理を完全に無効化します。

```
root@master # ceph-salt config /time_server disable
```

- お使いのサイトに単一のタイムソースがすでに存在する場合は、そのタイムソースのホスト名を指定します。

```
root@master # ceph-salt config /time_server/servers add time-server.example.com
```

- 別の方法として、`ceph-salt`にはSalt Minionの1つを残りのクラスタのタイムサーバとして機能するように設定する機能があります。この機能は「内部タイムサーバ」と呼ばれることもあります。このシナリオでは、`ceph-salt`は内部タイムサーバ(Salt Minionの1つであるはず)を、`pool.ntp.org`などの外部のタイムサーバと時刻を同期するように設定します。同時に、それ以外のミニオンを内部タイムサーバから時刻を取得するように設定します。この方法は、次のように実現できます。

```
root@master # ceph-salt config /time_server/servers add ses-master.example.com
root@master # ceph-salt config /time_server/external_servers add pool.ntp.org
```

`/time_server/subnet`オプションはサブネットを指定します。NTPクライアントはこのサブネットからNTPサーバへのアクセスを許可されます。サブネットは`/time_server/servers`を指定した際に自動で設定されます。変更や手動指定が必要な場合は、次のコマンドを実行します。

```
root@master # ceph-salt config /time_server/subnet set 10.20.6.0/24
```

次のコマンドでタイムサーバの設定を確認します。

```
root@master # ceph-salt config /time_server ls
o- time_server ..... [enabled]
  o- external_servers ..... [1]
    | o- pool.ntp.org ..... [...]
  o- servers ..... [1]
    | o- ses-master.example.com ..... [...]
  o- subnet ..... [10.20.6.0/24]
```

時刻同期設定の詳細については、<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-ntp.html#sec-ntp-yast> を参照してください。

### 5.3.2.9 Cephダッシュボードログインアカウント情報の設定

基本的なクラスタが展開されると、Cephダッシュボードが使用可能になります。アクセスするには、有効なユーザ名とパスワードを設定する必要があります。次に例を示します。

```
root@master # ceph-salt config /cephadm_bootstrap/dashboard/username set admin
root@master # ceph-salt config /cephadm_bootstrap/dashboard/password set PWD
```



#### ヒント: パスワードの更新を強制する

デフォルトでは、最初のダッシュボードユーザはダッシュボードに最初にログインの際にパスワードの変更を求められます。機能を無効化するには、次のコマンドを実行します。

```
root@master # ceph-salt config /cephadm_bootstrap/dashboard/force_password_update
disable
```

### 5.3.2.10 コンテナイメージへのパスの設定

cephadmは展開手順で使用するコンテナイメージへの有効なURIパスを認識する必要があります。デフォルトパスが設定されているかどうかを確認します。

```
root@master # ceph-salt config /cephadm_bootstrap/ceph_image_path ls
```

デフォルトパスが設定されていない場合や、展開時に特定のパスを必要とする場合は、次のように追加してください。

```
root@master # ceph-salt config /cephadm_bootstrap/ceph_image_path set registry.suse.com/ses/7/ceph/ceph
```



## 注記

監視スタックの場合、これ以外にもコンテナイメージが必要です。エアギャップに守られた環境で展開する場合や、ローカルレジストリから展開する場合に、対応するローカルレジストリを準備するために、こうしたコンテナイメージをこの時点で取得したい場合があります。

`ceph-salt`はこれらのコンテナイメージを展開に使用しないことに注意してください。これは、後の手順で監視コンポーネントの展開と移行のために`cephadm`を使用するための準備です。

監視スタックが使用するイメージとそのカスタマイズ方法の詳細については、『運用と管理ガイド』、第16章「監視とアラート」、16.1項「カスタムイメージまたはローカルイメージの設定」を参照してください。

### 5.3.2.11 コンテナレジストリの設定

必要に応じて、ローカルコンテナレジストリを設定できます。これは`registry.suse.com`レジストリのミラーとして機能します。`registry.suse.com`から更新されたコンテナを新しく入手できるようになった際には、ローカルレジストリを再同期する必要があることに注意してください。

ローカルレジストリを作成すると、以下のようなシナリオで役立ちます。

- 多数のクラスタノードを使用していて、コンテナイメージのローカルミラーを作成することで、ダウンロード時間と帯域幅を削減したい場合。
- クラスタがオンラインのレジストリにアクセスできないため(エアギャップ展開)、コンテナイメージを取得するローカルミラーを必要とする場合。
- 設定やネットワークの問題により、クラスタがセキュアリンク経由でリモートレジストリにアクセスできないため、ローカルの暗号化されていないレジストリが必要な場合。



## 重要

PTF(Program Temporary Fix)をサポートされたシステムに展開するには、ローカルコンテナレジストリを展開する必要があります。

アクセス資格情報と共にローカルレジストリのURLを設定するには、以下の手順に従います。

1. ローカルレジストリのURLを設定します。

```
cephuser@adm > ceph-salt config /containers/registry_auth/registry set REGISTRY_URL
```

2. ローカルレジストリにアクセスするためのユーザ名とパスワードを設定します。

```
cephuser@adm > ceph-salt config /containers/registry_auth/username  
set REGISTRY_USERNAME
```

```
cephuser@adm > ceph-salt config /containers/registry_auth/password  
set REGISTRY_PASSWORD
```

3. **ceph-salt apply**を実行して、すべてのミニオンのSalt Pillarを更新します。



## ヒント: レジストリキャッシュ

更新されたコンテナが新しく登場した際にローカルレジストリを再同期しないようにするには、「レジストリキャッシュ」を設定できます。

クラウドネイティブなアプリケーションの開発とデリバリーの手法は、コンテナイメージの開発と作成に、レジストリとCI/CD(継続的インテグレーション/デリバリー)インスタンスを必要とします。このインスタンス内でプライベートレジストリを使用できます。

### 5.3.2.12 データ転送中の暗号化(msgr2)の有効化

MSGR2プロトコルは、Cephの通信プロトコルです。このプロトコルは、ネットワークを通過するすべてのデータを暗号化するセキュリティモードを提供し、認証ペイロードをカプセル化し、新しい認証モード(Kerberosなど)を将来的に統合することを可能にします。



## 重要

現在のところ、CephFSやRADOS Block Deviceなどの、LinuxカーネルのCephFSクライアントはmsgr2をサポートしていません。

Cephデーモンは複数のポートにバインドできるため、古いCephクライアントとv2対応の新しいクライアントが同じクラスタに接続できます。デフォルトでは、MONはIANAが新しく割り当てた3300番ポート(CE4hまたは0xCE4)と、過去のデフォルトポートである6789番ポートにバインドされます。前者は新しいv2プロトコル用で、後者は旧式のv1プロトコル用です。

v2プロトコル(MSGR2)は2つの接続モードに対応しています。

## crcモード

接続確立時の整合性チェックと、CRC32Cによる完全性チェックが行われます。

## セキュアモード

接続確立時の厳重な初回認証と、認証後のすべてのトラフィックの完全な暗号化が行われます。これには、暗号の完全性チェックが含まれます。

ほとんどの接続については、オプションで使用するモードを制御できます。

## ms\_cluster\_mode

Cephデーモン間のクラスタ内通信に使用される接続モード(または許可モード)。複数のモードが記載されている場合は、先頭に記載されたものが優先されます。

## ms\_service\_mode

クライアントがクラスタに接続する際に使用する許可モードのリスト。

## ms\_client\_mode

Cephクラスタと通信する際にクライアントが使用(または許可)する、優先度順の接続モードのリスト。

Monitorだけに適用される、同様のオプションセットが存在します。これにより、管理者がMonitorとの通信に異なる要求(通常はより厳しい要求)を設定することができます。

## ms\_mon\_cluster\_mode

Monitor間の通信に使用される接続モード(または許可モード)。

## ms\_mon\_service\_mode

クライアントや他のCephデーモンがMonitorに接続する際に使用する許可モードのリスト。

## ms\_mon\_client\_mode

Monitorと通信する際にクライアントやMonitor以外のデーモンが使用する、優先度順の接続モードのリスト。

展開中にMSG2の暗号化モードを有効化するには、`ceph-salt`設定に設定オプションをいくつか追加してから**`ceph-salt apply`**を実行します。

`secure`モードを使用するには、次のコマンドを実行します。

`ceph-salt`設定ツールの`ceph_conf`にグローバルセクションを追加します。

```
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf add global
```

次のオプションを設定します。

```
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf/global set ms_cluster_mode "secure crc"
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf/global set ms_service_mode "secure crc"
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf/global set ms_client_mode "secure crc"
```



## 注記

crcの前にsecureがついているか確認してください。

secureモードを強制するには、次のコマンドを実行します。

```
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf/global set ms_cluster_mode secure
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf/global set ms_service_mode secure
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf/global set ms_client_mode secure
```



## ヒント: 設定の更新

上記の設定を変更したい場合、Monitor設定の保存先に変更内容を設定します。その手段として、**ceph config set**コマンドを使用します。

```
root@master # ceph config set global CONNECTION_OPTION CONNECTION_MODE [--force]
```

例:

```
root@master # ceph config set global ms_cluster_mode "secure crc"
```

現在値やデフォルト値を確認したい場合は、次のコマンドを実行します。

```
root@master # ceph config get CEPH_COMPONENT CONNECTION_OPTION
```

たとえば、OSDのms\_cluster\_modeを取得するには、次のコマンドを実行します。

```
root@master # ceph config get osd ms_cluster_mode
```

### 5.3.2.13 クラスタネットワークの設定

必要に応じて分離されたクラスタネットワークを実行する場合は、クラスタのネットワーク IP アドレスの末尾にスラッシュ記号で区切ったサブネットマスクを付加したアドレスを設定する必要があります場合があります。たとえば、192.168.10.22/24のようなアドレスです。

`cluster_network`を有効化するには、次のコマンドを実行します。

```
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf add global
root@master # ceph-salt config /cephadm_bootstrap/ceph_conf/global set
cluster_network NETWORK_ADDR
```

### 5.3.2.14 クラスタ設定の確認

最低限のクラスタ設定が完了しました。明らかな誤りがないか、確認してください。

```
root@master # ceph-salt config ls
o- / ..... [...]
  o- ceph_cluster ..... [...]
    | o- minions ..... [Minions: 5]
    | | o- ses-master.example.com ..... [admin]
    | | o- ses-min1.example.com ..... [bootstrap, admin]
    | | o- ses-min2.example.com ..... [no roles]
    | | o- ses-min3.example.com ..... [no roles]
    | | o- ses-min4.example.com ..... [no roles]
    | o- roles ..... [...]
    |   o- admin ..... [Minions: 2]
    |   | o- ses-master.example.com ..... [no other roles]
    |   | o- ses-min1.example.com ..... [other roles: bootstrap]
    |   o- bootstrap ..... [ses-min1.example.com]
    |   o- cephadm ..... [Minions: 5]
    |   o- tuned ..... [...]
    |     o- latency ..... [no minions]
    |     o- throughput ..... [no minions]
  o- cephadm_bootstrap ..... [...]
    | o- advanced ..... [...]
    | o- ceph_conf ..... [...]
    | o- ceph_image_path ..... [registry.suse.com/ses/7/ceph/ceph]
    | o- dashboard ..... [...]
    |   o- force_password_update ..... [enabled]
    |   o- password ..... [randomly generated]
    |   o- username ..... [admin]
    | o- mon_ip ..... [192.168.10.20]
  o- containers ..... [...]
    | o- registries_conf ..... [enabled]
    | | o- registries ..... [empty]
    | o- registry_auth ..... [...]
    |   o- password ..... [not set]
```

```
| o- registry ..... [not set]
| o- username ..... [not set]
o- ssh ..... [Key Pair set]
| o- private_key ..... [53:b1:eb:65:d2:3a:ff:51:6c:e2:1b:ca:84:8e:0e:83]
| o- public_key ..... [53:b1:eb:65:d2:3a:ff:51:6c:e2:1b:ca:84:8e:0e:83]
o- time_server ..... [enabled]
  o- external_servers ..... [1]
    | o- 0.pt.pool.ntp.org ..... [...]
  o- servers ..... [1]
    | o- ses-master.example.com ..... [...]
  o- subnet ..... [10.20.6.0/24]
```



## ヒント: クラスタ設定のステータス

次のコマンドを実行することで、クラスタ設定が有効かどうかを確認できます。

```
root@master # ceph-salt status
cluster: 5 minions, 0 hosts managed by cephadm
config: OK
```

### 5.3.2.15 クラスタ設定のエクスポート

基本的なクラスタの設定が完了し、設定が有効であることを確認したら、クラスタ設定をファイルにエクスポートするとよいでしょう。

```
root@master # ceph-salt export > cluster.json
```



## 警告

**ceph-salt export**の出力にはSSHの秘密鍵が含まれます。セキュリティ上の不安がある場合は、適切な予防策を講じるまではコマンドを実行しないでください。

クラスタ設定を破棄してバックアップの状態に戻す場合は、次のコマンドを実行します。

```
root@master # ceph-salt import cluster.json
```

### 5.3.3 ノードの更新と最小クラスタのブートストラップ

クラスタを展開する前に、すべてのノードのソフトウェアパッケージをすべて更新してください。

```
root@master # ceph-salt update
```

アップデート中にノードが `Reboot is needed` と報告した場合、重要なOSのパッケージ(カーネルなど)が新しいバージョンに更新されているため、ノードを再起動して変更を適用する必要があります。

再起動が必要なノードをすべて再起動するには、`--reboot` オプションを付加してください。

```
root@master # ceph-salt update --reboot
```

もしくは、個別に再起動してください。

```
root@master # ceph-salt reboot
```

## ！ 重要

Salt Masterは `ceph-salt update --reboot` や `ceph-salt reboot` コマンドでは再起動されません。Salt Masterの再起動が必要な場合、手動で再起動してください。

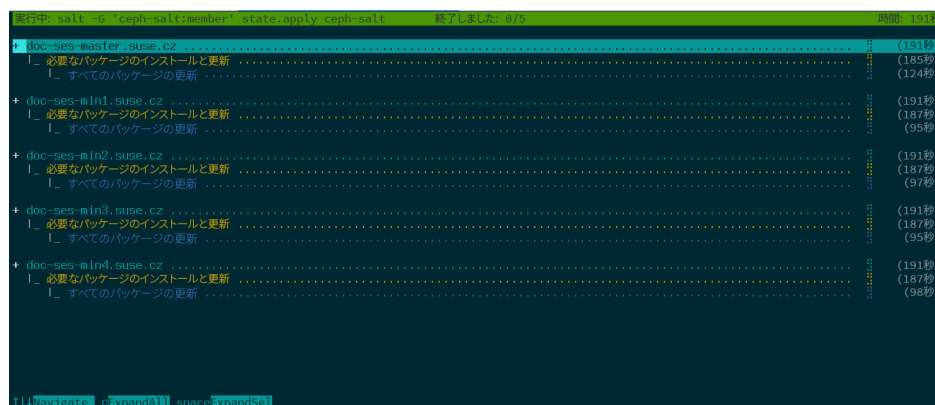
ノードの更新後、最小のクラスタをブートストラップします。

```
root@master # ceph-salt apply
```

## 📄 注記

ブートストラップが完了すると、クラスタにはCeph MonitorとCeph Managerが1つずつ含まれます。

先のコマンドを実行すると、インタラクティブなユーザインターフェイスが開かれ、各ミニオンの現在の進行状況が表示されます。



```
実行中: salt -G 'ceph-salt:member' state.apply ceph-salt 終了しました: 0/0 経過: 10/14
+ doc-ses-master.suse.cz ..... (191秒)
  | 必要なパッケージのインストールと更新 ..... (185秒)
  | 全てのパッケージの更新 ..... (124秒)
+ doc-ses-min1.suse.cz ..... (191秒)
  | 必要なパッケージのインストールと更新 ..... (187秒)
  | 全てのパッケージの更新 ..... (95秒)
+ doc-ses-min2.suse.cz ..... (191秒)
  | 必要なパッケージのインストールと更新 ..... (187秒)
  | 全てのパッケージの更新 ..... (97秒)
+ doc-ses-min3.suse.cz ..... (191秒)
  | 必要なパッケージのインストールと更新 ..... (187秒)
  | 全てのパッケージの更新 ..... (95秒)
+ doc-ses-min4.suse.cz ..... (191秒)
  | 必要なパッケージのインストールと更新 ..... (187秒)
  | 全てのパッケージの更新 ..... (98秒)

!!! Navigate | c:expandAll | space:expandSel
```

図 5.1: 最小クラスタの展開



## ヒント: 非インタラクティブモード

スクリプトから設定を適用する必要がある場合、非インタラクティブモードで展開することもできます。このモードは、リモートマシンからクラスタを展開する際にも有用です。ネットワーク経由で進捗状況を画面に更新し続けると、煩わしく感じる場合があるためです。

```
root@master # ceph-salt apply --non-interactive
```

### 5.3.4 最終ステップの確認

**ceph-salt apply** コマンドが完了すると、Ceph MonitorとCeph Managerが1つずつ存在するはずです。rootに相当するadminの役割を与えられたミニオンか、**sudo**を使用するcephadmユーザは、**ceph status** コマンドを正常に実行できるはずです。

次の手順には、cephadmを使用した追加のCeph Monitor、Ceph Manager、OSD、監視スタック、ゲートウェイの展開が含まれます。

続ける前に新しいクラスタのネットワーク設定を確認してください。この時点では、**ceph-salt** 設定の `/cephadm_bootstrap/mon_ip` に入力された内容に従って、**public\_network** 設定が読み込まれます。しかし、この設定はCeph Monitorにしか適用されません。次のコマンドを使用して、この設定を確認できます。

```
root@master # ceph config get mon public_network
```

これがCephの動作に必要な最低限の設定ですが、この**public\_network** 設定を**global** に設定することをお勧めします。つまり、この設定がMONだけでなく、すべてのタイプのCephデーモンにも適用されます。

```
root@master # ceph config set global public_network "$(ceph config get mon public_network)"
```



## 注記

この手順は必須ではありません。しかしながら、この設定を使用しないと、Ceph OSDと(Ceph Monitorを除く)その他のデーモンが「すべてのアドレス」をリスンすることになります。

完全に分離されたネットワークを使用して、OSDどうしを通信させたい場合は、次のコマンドを実行します。

```
root@master # ceph config set global cluster_network
"cluster_network_in_cidr_notation"
```

このコマンドを実行すると、展開中に作成されるOSDは最初から所定のクラスタネットワークを使用ようになります。

クラスタが高密度なノード(ホストあたりのOSDが62個を超える)から構成されるように設定する場合は、Ceph OSDに十分なポートを割り当ててください。デフォルトのポート範囲(6800～7300)のままで、ホストあたりのOSDは最大62個までです。高密度なノードを含むクラスタの場合、`ms_bind_port_max`の設定を適切な値に調整してください。各OSDは追加で8個のポートを使用します。たとえば、96個のOSDを実行するように設定されたホストの場合、768個のポートが必要になります。この場合、次のコマンドを実行して、`ms_bind_port_max`を少なくとも7568に設定する必要があります。

```
root@master # ceph config set osd.* ms_bind_port_max 7568
```

これを動作させるには、設定した値に応じてファイアウォールの設定も調整する必要があります。詳細については、『Troubleshooting Guide』、第13章「Hints and tips」、13.7項「Firewall settings for Ceph」を参照してください。

## 5.4 サービスとゲートウェイの展開

基本的なCephクラスタを展開した後、より多くのクラスタノードにコアサービスを展開します。クライアントからクラスタのデータにアクセスできるようにするには、追加のサービスも展開します。

現時点では、Cephオーケストレータ(`ceph orch`サブコマンド)を使用したコマンドライン上でのCephサービスの展開がサポートされています。

### 5.4.1 `ceph orch`コマンド

Cephオーケストレータコマンドである`ceph orch`は、新しいクラスタノード上で、クラスタコンポーネントの一覧とCephサービスの展開を行います。このコマンドは`cephadm`モジュールのインターフェイスです。

#### 5.4.1.1 オーケストレータステータスの表示

次のコマンドは、Cephオーケストレータの現在モードとステータスを表示します。

```
cephuser@adm > ceph orch status
```

### 5.4.1.2 デバイス、サービス、デーモンの一覧

すべてのディスクデバイスを一覧にするには、次のコマンドを実行します。

```
cephuser@adm > ceph orch device ls
Hostname Path      Type  Serial  Size  Health  Ident  Fault  Available
ses-master /dev/vdb hdd    0d8a... 10.7G Unknown N/A    N/A    No
ses-min1   /dev/vdc hdd    8304... 10.7G Unknown N/A    N/A    No
ses-min1   /dev/vdd hdd    7b81... 10.7G Unknown N/A    N/A    No
[...]
```



#### ヒント: サービスとデーモン

「サービス」とは、特定のタイプのCephサービスを指す総称です。たとえば、Ceph Managerなどです。

「デーモン」とは、サービスの特定のインスタンスを指します。たとえば、ses-min1という名前のノードで実行されるmgr.ses-min1.gdalcikプロセスなどです。

cephadmが認識しているすべてのサービスを一覧にするには、次のコマンドを実行します。

```
cephuser@adm > ceph orch ls
NAME  RUNNING  REFRESHED  AGE  PLACEMENT  IMAGE NAME  IMAGE ID
mgr    1/0      5m ago     -    <no spec>  registry.example.com/[...] 5bf12403d0bd
mon    1/0      5m ago     -    <no spec>  registry.example.com/[...] 5bf12403d0bd
```



#### ヒント

リストに特定のノードのサービスだけを表示するには、オプションの --host パラメータを使用します。特定のタイプのサービスだけを表示するには、オプションの --service-type パラメータを使用します(指定できるタイプは mon、osd、mgr、mds、rgw です)。

cephadmが展開した実行中のすべてのデーモンを一覧にするには、次のコマンドを実行します。

```
cephuser@adm > ceph orch ps
NAME           HOST      STATUS  REFRESHED  AGE  VERSION  IMAGE ID  CONTAINER ID
mgr.ses-min1.gd ses-min1  running  8m ago     12d  15.2.0.108 5bf12403d0bd b8104e09814c
mon.ses-min1    ses-min1  running  8m ago     12d  15.2.0.108 5bf12403d0bd a719e0087369
```



## ヒント

特定のデーモンのステータスを照会するには、`--daemon_type`と`--daemon_id`を使用します。OSDの場合、IDは数字のOSD IDです。MDSの場合、IDはファイルシステム名です。

```
cephuser@adm > ceph orch ps --daemon_type osd --daemon_id 0
cephuser@adm > ceph orch ps --daemon_type mds --daemon_id my_cephfs
```

## 5.4.2 サービス仕様と配置仕様

Cephサービスの展開内容を指定する方法としては、YAMLフォーマットのファイルを作成して、展開したいサービスの仕様を記載することをお勧めします。

### 5.4.2.1 サービス仕様の作成

サービスタイプごとに個別の仕様ファイルを作成できます。以下に例を示します。

```
root@master # cat nfs.yml
service_type: nfs
service_id: EXAMPLE_NFS
placement:
  hosts:
    - ses-min1
    - ses-min2
spec:
  pool: EXAMPLE_POOL
  namespace: EXAMPLE_NAMESPACE
```

もしくは、各サービスを実行するノードを記載した単一のファイル(`cluster.yml`など)により、複数の(または、すべての)サービスタイプを指定することもできます。それぞれのサービスタイプを3つのダッシュ記号(`---`)で区切ることを忘れないでください。

```
cephuser@adm > cat cluster.yml
service_type: nfs
service_id: EXAMPLE_NFS
placement:
  hosts:
    - ses-min1
    - ses-min2
spec:
  pool: EXAMPLE_POOL
```

```
namespace: EXAMPLE_NAMESPACE
---
service_type: rgw
service_id: REALM_NAME.ZONE_NAME
placement:
  hosts:
    - ses-min1
    - ses-min2
    - ses-min3
---
[...]
```

各プロパティが意味するものは、以下の通りです。

#### service\_type

サービスのタイプです。次のいずれかを指定できます。Cephサービス (mon、mgr、mds、crash、osd、rbd-mirror)、ゲートウェイ(nfs、rgw)、監視スタックの一部(alertmanager、grafana、node-exporter、prometheus)。

#### service\_id

サービスの名前です。次のサービスタイプについては、service\_idプロパティは不要です。mon、mgr、alertmanager、grafana、node-exporter、prometheus。

#### placement

どのノードがサービスを実行するかを指定します。詳細については、[5.4.2.2項「配置仕様の作成」](#)を参照してください。

#### spec

サービスタイプに関連する、追加仕様です。



### ヒント: 特定のサービスを適用する

通常、Cephクラスタのサービスには、いくつかの固有のプロパティがあります。個別のサービス仕様の例と詳細については、[5.4.3項「Cephサービスの展開」](#)を参照してください。

#### 5.4.2.2 配置仕様の作成

Cephサービスを展開するには、サービスの展開先ノードをcephadmが認識する必要があります。placementプロパティを使用して、サービスを適用するノードのホスト名の略称を列挙してください。

```
cephuser@adm > cat cluster.yml
[...]
placement:
  hosts:
    - host1
    - host2
    - host3
[...]
```

### 5.4.2.3 クラスタ仕様の適用

すべてのサービス仕様とサービスの配置仕様を記載した完全な`cluster.yml`ファイルの作成が完了したら、次のコマンドを実行して、クラスタに仕様を適用してください。

```
cephuser@adm > ceph orch apply -i cluster.yml
```

クラスタのステータスを確認するには、**`ceph orch status`**コマンドを実行します。詳細については、「[5.4.1.1項「オーケストレータステータスの表示」](#)」を参照してください。

### 5.4.2.4 実行中のクラスタ仕様のエクスポート

[5.4.2項「サービス仕様と配置仕様」](#)で説明した仕様ファイルを用いてCephクラスタにサービスを展開したにもかかわらず、運用中にクラスタの設定が元の仕様から変わる場合もあります。また、誤って仕様ファイルを削除してしまうことも考えられます。

実行中のクラスタからすべての仕様を取得するには、次のコマンドを実行してください。

```
cephuser@adm > ceph orch ls --export
placement:
  hosts:
    - hostname: ses-min1
      name: ''
      network: ''
service_id: my_cephfs
service_name: mds.my_cephfs
service_type: mds
---
placement:
  count: 2
service_name: mgr
service_type: mgr
---
[...]
```



## ヒント

`--format`オプションを付加することで、デフォルトのyaml出力フォーマットを変更できます。選択できるフォーマットは、`json`、`json-pretty`、`yaml`です。以下に例を示します。

```
ceph orch ls --export --format json
```

## 5.4.3 Cephサービスの展開

基本的なクラスタの実行後、他のノードにCephサービスを展開できます。

### 5.4.3.1 Ceph MonitorとCeph Managerの展開

Cephクラスタでは、3個または5個のMONを異なるノードに展開します。クラスタに5個以上のノードが含まれる場合、5個のMONを展開することをお勧めします。MONと同じノードにMGRを展開すると良いでしょう。



### 重要: ブートストラップMONを含める

MONとMGRを展開する際は、5.3.2.5項「最初のMON/MGRノードの指定」で基本的なクラスタを構成した際に追加した、最初のMONを忘れずに含めてください。

MONを展開するには、次の仕様を適用してください。

```
service_type: mon
placement:
  hosts:
    - ses-min1
    - ses-min2
    - ses-min3
```



## 注記

別のノードを追加する必要がある場合は、同じYAMLリストにホスト名を付加してください。以下に例を示します。

```
service_type: mon
placement:
  hosts:
```

```
- ses-min1
- ses-min2
- ses-min3
- ses-min4
```

同様に、MGRを展開するには次の仕様を適用してください。



## 重要

展開ごとに、少なくとも3個のCeph Managerが展開されているかを確認してください。

```
service_type: mgr
placement:
  hosts:
    - ses-min1
    - ses-min2
    - ses-min3
```



## ヒント

MONまたはMGRが同じサブネット上に存在しない場合、サブネットアドレスを付加する必要があります。以下に例を示します。

```
service_type: mon
placement:
  hosts:
    - ses-min1:10.1.2.0/24
    - ses-min2:10.1.5.0/24
    - ses-min3:10.1.10.0/24
```

### 5.4.3.2 Ceph OSDの展開



## 重要: ストレージデバイスが使用可能となる条件

以下の条件をすべて満たす場合、ストレージデバイスは「使用可能」とみなされます。

- デバイスにパーティションが作成されていない。
- デバイスがLVM状態ではない。

- デバイスがマウント先になっていない。
- デバイ스에 파일 시스템이 포함되지 않음.
- 디바이스에 BlueStore OSD가 포함되지 않음.
- 디바이스의 크기가 5GB를 초과하고 있음.

これらの条件が満たされない場合、CephはそのOSDのプロビジョニングを拒否します。

OSDを展開する方法は2つあります。

- 「使用可能」とみなされた未使用のストレージデバイスをすべて使用するように、Cephに指示する方法。

```
cephuser@adm > ceph orch apply osd --all-available-devices
```

- DriveGroupsを使用してデバイスを記述したOSD仕様を作成し、そのプロパティを基にデバイスを展開する方法(『運用と管理ガイド』、第13章「運用タスク」、13.4.3項「DriveGroups仕様を用いたOSDの追加」を参照してください)。プロパティの例としては、デバイスの種類(SSDまたはHDD)、デバイスのモデル名、サイズ、デバイスが存在するノードなどがあります。仕様の作成後、次のコマンドを実行して仕様を適用します。

```
cephuser@adm > ceph orch apply osd -i drive_groups.yml
```

### 5.4.3.3 メタデータサーバの展開

CephFSは1つ以上のMDS(メタデータサーバ)サービスを必要とします。CephFSを作成するには、まず以下の仕様を適用して、MDSサーバを作成する必要があります。



#### 注記

最低でも2つのプールを作成してから以下の仕様を適用してください。1つはCephFSのデータ用、もう1つはCephFSのメタデータ用のプールです。

```
service_type: mds
service_id: CEPHFS_NAME
placement:
```

```
hosts:
- ses-min1
- ses-min2
- ses-min3
```

MDSが機能したら、CephFSを作成します。

```
ceph fs new CEPHFS_NAME metadata_pool data_pool
```

#### 5.4.3.4 Object Gatewayの展開

cephadmはObject Gatewayを、特定の「レルム」と「ゾーン」を管理するデーモンのコレクションとして展開します。

Object Gatewayサービスを既存のレルムとゾーンに関連付けることも(詳細については、『運用と管理ガイド』、第21章「Ceph Object Gateway」、21.13項「マルチサイトObject Gateway」を参照してください)、存在しない`REALM_NAME`と`ZONE_NAME`を指定することもできます。後者の場合、次の設定を適用すると自動的にゾーンとレルムが作成されます。

```
service_type: rgw
service_id: REALM_NAME.ZONE_NAME
placement:
  hosts:
  - ses-min1
  - ses-min2
  - ses-min3
spec:
  rgw_realm: RGW_REALM
  rgw_zone: RGW_ZONE
```

##### 5.4.3.4.1 セキュアなSSLアクセスの使用

Object Gatewayへの接続にセキュアなSSL接続を使用するには、有効なSSL証明書とキーファイルのペアが必要です(詳細については、『運用と管理ガイド』、第21章「Ceph Object Gateway」、21.7項「Object GatewayでのHTTPS/SSLの有効化」を参照してください)。必要な作業は、SSLの有効化、SSL接続のポート番号の指定、SSL証明書とキーファイルの指定です。

SSLを有効化し、ポート番号を指定するには、仕様に次の内容を記載します。

```
spec:
  ssl: true
  rgw_frontend_port: 443
```

SSL証明書とキーを指定するには、YAML仕様ファイルに内容を直接ペーストすることができます。行末のパイプ記号(|)は、構文解析の際に複数行にまたがる文字列を1つの値として認識させるためのものです。以下に例を示します。

```
spec:
  ssl: true
  rgw_frontend_port: 443
  rgw_frontend_ssl_certificate: |
    -----BEGIN CERTIFICATE-----
    MIIFmjCCA4KgAwIBAgIJAIZ2n35bmwXTMA0GCSqGSIb3DQEBCwUAMGIXCzAJBgNV
    BAYTAkFVMQwwCgYDVQQIDANOU1cxHTAbBgNVBAoMFEV4YW1wbGUkXIFNTTCBp
    [...]
    -----END CERTIFICATE-----
  rgw_frontend_ssl_key: |
    -----BEGIN PRIVATE KEY-----
    MIIJRAIBADANBgkqhkiG9w0BAQEFAASCCS4wggkqAgEAAoICAQDLtFwg6LLl2j4Z
    BDV+iL4A07VZ9KbmWI37Ml2W6y2YeKX3Qwf+3eBz7TVHRldm6iPpCpqpQjXUsT9
    [...]
    -----END PRIVATE KEY-----
```



## ヒント

SSL証明書とキーファイルの内容をペーストする代わり

に、`rgw_frontend_ssl_certificate`:キーワードと`rgw_frontend_ssl_key`:キーワードを削除して、設定データベースにSSL証明書とキーファイルをアップロードすることもできます。

```
cephuser@adm > ceph config-key set rgw/cert/REALM_NAME/ZONE_NAME.crt \
-i SSL_CERT_FILE
cephuser@adm > ceph config-key set rgw/cert/REALM_NAME/ZONE_NAME.key \
-i SSL_KEY_FILE
```

### 5.4.3.4.2 サブクラスタを使用した展開

「サブクラスタ」はクラスタ内のノードの整理に役立ちます。これによりワークロードを分離することで、弾力的な拡張が容易になります。サブクラスタを使用して展開する場合は、次の設定を適用します。

```
service_type: rgw
service_id: REALM_NAME.ZONE_NAME.SUBCLUSTER
placement:
  hosts:
    - ses-min1
```

```
- ses-min2
- ses-min3
spec:
  rgw_realm: RGW_REALM
  rgw_zone: RGW_ZONE
  subcluster: SUBCLUSTER
```

### 5.4.3.5 iSCSI Gatewayの展開

cephadmが展開するiSCSI Gatewayは、クライアント(「イニシエータ」)から、リモートサーバ上のSCSIストレージデバイス(「ターゲット」)にSCSIコマンドを送信できるようにする、SAN(ストレージエリアネットワーク)プロトコルです。

展開するには以下の設定を適用します。trusted\_ip\_listにすべてのiSCSI GatewayノードとCeph ManagerノードのIPアドレスが含まれているか確認してください(以下の出力例を参照してください)。



#### 注記

以下の仕様を適用する前に、プールが作成されているか確認してください。

```
service_type: iscsi
service_id: EXAMPLE_ISCSI
placement:
  hosts:
    - ses-min1
    - ses-min2
    - ses-min3
spec:
  pool: EXAMPLE_POOL
  api_user: EXAMPLE_USER
  api_password: EXAMPLE_PASSWORD
  trusted_ip_list: "IP_ADDRESS_1,IP_ADDRESS_2"
```



#### 注記

trusted\_ip\_listに列挙されたIPについて、カンマ区切りの後にスペースが入っていないことを確認してください。

#### 5.4.3.5.1 セキュアなSSLの設定

セキュアなSSL接続をCephダッシュボードとiSCSIターゲットAPIの間で使用するには、有効なSSL証明書とキーファイルのペアが必要です。証明書とキーファイルは、CAが発行したものか自己署名したものを使用します(『運用と管理ガイド』、第10章「手動設定」、10.1.1項「自己署名証明書の作成」を参照してください)。SSLを有効化するには、仕様ファイルに`api_secure: true`設定を含めます。

```
spec:
  api_secure: true
```

SSL証明書とキーを指定するには、YAML仕様ファイルに内容を直接ペーストすることができます。行末のパイプ記号(`|`)は、構文解析の際に複数行にまたがる文字列を1つの値として認識させるためのものです。以下に例を示します。

```
spec:
  pool: EXAMPLE_POOL
  api_user: EXAMPLE_USER
  api_password: EXAMPLE_PASSWORD
  trusted_ip_list: "IP_ADDRESS_1,IP_ADDRESS_2"
  api_secure: true
  ssl_cert: |
    -----BEGIN CERTIFICATE-----
    MIIDtTCCAp2gAwIBAgIYMC4xNzc1NDQxNjEzMzc2MjMyXzxxvQ7EcMA0GCSqGSIb3
    DQEBChUAMG0xCzAJBgNVBAYTAlVTMQ0wCwYDVQQIDARVdGFoMRcwFQYDVQQHDA5T
    [...]
    -----END CERTIFICATE-----
  ssl_key: |
    -----BEGIN PRIVATE KEY-----
    MIIIEvQIBADANBgkqhkiG9w0BAQEFAASCBAcwggSjAgEAAoIBAQC5jdYbjtNTAKW4
    /CwQr/7w0iLGzVxChn3mmCIF3DwbL/qvTFTX2d8bDf6LjGwLYloXHscRfxszX/4h
    [...]
    -----END PRIVATE KEY-----
```

#### 5.4.3.6 NFS Ganeshaの展開

cephadmはNFS Ganeshaの展開に、事前定義されたRADOSプールとオプションのネームスペースを使用します。NFS Ganeshaを展開するには、次の仕様を適用してください。



#### 注記

事前定義されたRADOSプールが必要です。これが存在しない場合は、**ceph orch apply**処理に失敗します。プールの作成の詳細については、『運用と管理ガイド』、第18章「ストレージプールの管理」、18.1項「プールの作成」を参照してください。

```

service_type: nfs
service_id: EXAMPLE_NFS
placement:
  hosts:
    - ses-min1
    - ses-min2
spec:
  pool: EXAMPLE_POOL
  namespace: EXAMPLE_NAMESPACE

```

- EXAMPLE\_NFSにはNFSエクスポートを識別する任意の文字列を指定します。
- EXAMPLE\_POOLにはNFS GaneshaのRADOS設定オブジェクトを保存するプール名を指定します。
- EXAMPLE\_NAMESPACE(オプション)には、希望するObject GatewayのNFSネームスペースを指定します(ganeshaなど)。

#### 5.4.3.7 rbd-mirrorの展開

rbd-mirrorサービスは2つのCephクラスタ間でRADOS Block Deviceイメージの同期を行います(詳細については『運用と管理ガイド』、第20章「RADOS Block Device」、20.4項「RBDイメージのミラーリング」を参照してください)。rbd-mirrorを展開するには、次の仕様を使用してください。

```

service_type: rbd-mirror
service_id: EXAMPLE_RBD_MIRROR
placement:
  hosts:
    - ses-min3

```

#### 5.4.3.8 監視スタックの展開

監視スタックは、Prometheus、Prometheusエクスポータ、Prometheus Alertmanager、Grafanaから構成されます。Cephダッシュボードはこうしたコンポーネントを利用して、クラスタの使用量やパフォーマンスの詳細なメトリクスの保存と視覚化を行います。



## ヒント

展開に監視スタックサービスのカスタムコンテナイメージやローカルコンテナイメージを必要とする場合は、『運用と管理ガイド』、第16章「監視とアラート」、16.1項「カスタムイメージまたはローカルイメージの設定」を参照してください。

監視スタックを展開するには、以下の手順に従ってください。

1. Ceph Managerデーモンでprometheusモジュールを有効化します。これにより、Cephの内部メトリクスが公開され、Prometheusから読み取れるようになります。

```
cephuser@adm > ceph mgr module enable prometheus
```



## 注記

このコマンドはPrometheusの展開前に実行してください。展開前にコマンドを実行していない場合、Prometheusを再展開してPrometheusの設定を更新する必要があります。

```
cephuser@adm > ceph orch redeploy prometheus
```

2. 次のような内容を含む仕様ファイル(monitoring.yamlなど)を作成します。

```
service_type: prometheus
placement:
  hosts:
    - ses-min2
---
service_type: node-exporter
---
service_type: alertmanager
placement:
  hosts:
    - ses-min4
---
service_type: grafana
placement:
  hosts:
    - ses-min3
```

3. 次のコマンドを実行して、監視サービスを適用します。

```
cephuser@adm > ceph orch apply -i monitoring.yaml
```

監視サービスの展開には1、2分かかる場合があります。

## ！ 重要

Prometheus、Grafana、Cephダッシュボードは、お互いに通信できるようにすべて自動的に設定されます。そのため、この手順で展開されたとき、Cephダッシュボードには完全に機能するGrafanaが統合されています。

このルールの例外は、RBDイメージの監視だけです。詳細については、『運用と管理ガイド』、第16章「監視とアラート」、16.5.4項「RBDイメージ監視の有効化」を参照してください。

## III 追加のサービスのインストール

### 6 iSCSIゲートウェイのインストール 66

## 6 iSCSIゲートウェイのインストール

iSCSIは、クライアント(「イニシエータ」)から、リモートサーバ上のSCSIストレージデバイス(「ターゲット」)にSCSIコマンドを送信できるようにするSAN (ストレージエリアネットワーク)プロトコルです。SUSE Enterprise Storage 7には、Cephのストレージ管理をiSCSIプロトコル経由でMicrosoft Windows\*、VMware\* vSphereなどの異種クライアントから利用できるようにする機能が含まれています。マルチパスiSCSIアクセスによってこれらのクライアントの可用性とスケーラビリティが向上すると同時に、標準化されたiSCSIプロトコルがクライアントとSUSE Enterprise Storage 7クラスタ間に追加のセキュリティ分離層も提供します。この設定機能は`ceph-iscsi`という名前です。Cephストレージ管理者は、`ceph-iscsi`を使用して、シンプロビジョニングおよび複製された高可用性ボリュームを定義できます。これらのボリュームでは、Ceph RBD (RADOS Block Device)により、読み込み専用スナップショット、読み書きクローン、および自動サイズ調整がサポートされます。これにより、単一の`ceph-iscsi`ゲートウェイホスト、またはマルチパスフェールオーバーをサポートする複数のゲートウェイホストを通じてボリュームをエクスポートできます。iSCSIプロトコルによってボリュームを他のSCSIブロックデバイスと同じように利用できるようになり、Linux、Microsoft Windows、およびVMwareホストはiSCSIプロトコルを使用してボリュームに接続できます。つまり、SUSE Enterprise Storage 7の顧客は、従来のSANの特徴と利点をすべて備えた完全なブロックストレージインフラストラクチャサブシステムをCeph上で効果的に実行でき、将来の増加に対応できます。

この章では、CephクラスタインフラストラクチャをiSCSI Gatewayと共に設定し、クライアントホストがiSCSIプロトコルを使ってリモート保存データをローカルストレージデバイスとして使用できるようにするための情報について詳しく説明します。

### 6.1 iSCSIブロックストレージ

iSCSIは、IP (インターネットプロトコル)を使用するSCSI (Small Computer System Interface) コマンドセットを実装したもので、RFC 3720で規定されています。iSCSIはサービスとして実装され、クライアント(イニシエータ)はTCPポート3260でセッションを経由してサーバ(ターゲット)と通信します。iSCSIターゲットのIPアドレスとポートを「iSCSIポータル」と呼び、1つ以上のポータルを通じてターゲットを公開できます。ターゲットと1つ以上のポータルの組み合わせを「TPG」(ターゲットポータルグループ)と呼びます。

iSCSIの基礎となるデータリンク層プロトコルはほとんどの場合Ethernetです。具体的には、最新のiSCSIインフラストラクチャは、最適なスループットのために10ギガビットEthernetまたはより高速なネットワークを使用します。iSCSI GatewayとバックエンドのCephクラスタ間の接続には、10ギガビットEthernetを強くお勧めします。

## 6.1.1 LinuxカーネルiSCSIターゲット

LinuxカーネルiSCSIターゲットは元々、プロジェクトの発端となったドメインとWebサイト [linux-iscsi.org](http://linux-iscsi.org) にちなんでLIOと呼ばれていました。しばらくの間、競合するiSCSIターゲット実装がLinuxプラットフォームで4つも利用可能な状態が続いていましたが、最終的にはLIOがiSCSIの単一のリファレンスターゲットとして普及しました。LIOのメインラインカーネルコードは、シンプルではあるものの若干あいまいな「ターゲット」という名前を用いて、「ターゲットコア」と、さまざまなフロントエンド/バックエンドターゲットモジュールを区別しています。

最も一般的に用いられているフロントエンドモジュールはまず間違いなくiSCSIです。ただし、LIOはFC (ファイバチャネル)、FCoE (ファイバチャネルオーバーEthernet)、およびその他の複数のフロントエンドプロトコルもサポートしています。現在のところ、SUSE Enterprise StorageによってサポートされているのはiSCSIプロトコルのみです。

最もよく使用されるターゲットバックエンドモジュールは、ターゲットホスト上で利用可能なブロックデバイスを単に再エクスポートできるモジュールです。このモジュールは「iblock」という名前です。ただし、LIOには、RBDイメージへの並列化マルチパスI/Oアクセスをサポートする、RBD固有のバックエンドモジュールもあります。

## 6.1.2 iSCSIイニシエータ

このセクションでは、Linux、Microsoft Windows、およびVMwareの各プラットフォームで使用されているiSCSIイニシエータについて簡単に紹介します。

### 6.1.2.1 Linux

Linuxプラットフォームの標準のイニシエータはopen-iscsiです。open-iscsiはデーモンiscsidを起動し、ユーザはこのデーモンを使用して特定のポータル上のiSCSIターゲットを検出してターゲットにログインし、iSCSIボリュームをマップできます。iscsidはSCSIの中間層と通信して、カーネル内ブロックデバイスを作成します。これにより、カーネルはこのブロックデバイスをシステムの他のSCSIブロックデバイスと同じように扱うことができます。open-iscsiイニシエータをデバيسマッパーマルチパス(dm-multipath)機能と組み合わせて展開することで、高可用性iSCSIブロックデバイスを提供できます。

### 6.1.2.2 Microsoft WindowsとHyper-V

Microsoft WindowsオペレーティングシステムのデフォルトのiSCSIイニシエータは、Microsoft iSCSIイニシエータです。このiSCSIサービスはGUI (グラフィカルユーザインタフェース)を使用して設定でき、高可用性のためにマルチパスI/Oをサポートしています。

### 6.1.2.3 VMware

VMware vSphereおよびESXのデフォルトのiSCSIイニシエータは、VMware ESXソフトウェアiSCSIイニシエータ`vmkiscsi`です。これが有効な場合、vSphere Clientから、または`vmkiscsi-tool`コマンドを使用して設定できます。その後、vSphere iSCSIストレージアダプタを介してVMFSに接続されたストレージボリュームをフォーマットし、他のVMストレージデバイスと同じように使用できます。VMwareイニシエータも、高可用性のためにマルチパスI/Oをサポートしています。

## 6.2 ceph-iscsiに関する一般情報

`ceph-iscsi`は、RADOS Block Deviceの利点とiSCSIのユビキタスな汎用性を組み合わせたものです。iSCSIターゲットホスト(iSCSI Gatewayとして知られている)上で`ceph-iscsi`を使用することで、Cephクライアントプロトコルに対応していなくても、ブロックストレージを利用する必要があるすべてのアプリケーションがCephの利点を享受できます。代わりに、ユーザはiSCSIまたは他のターゲットフロントエンドプロトコルを使用してLIOターゲットに接続できます。これにより、そのターゲットがすべてのI/OをRBDストレージ操作に変換します。

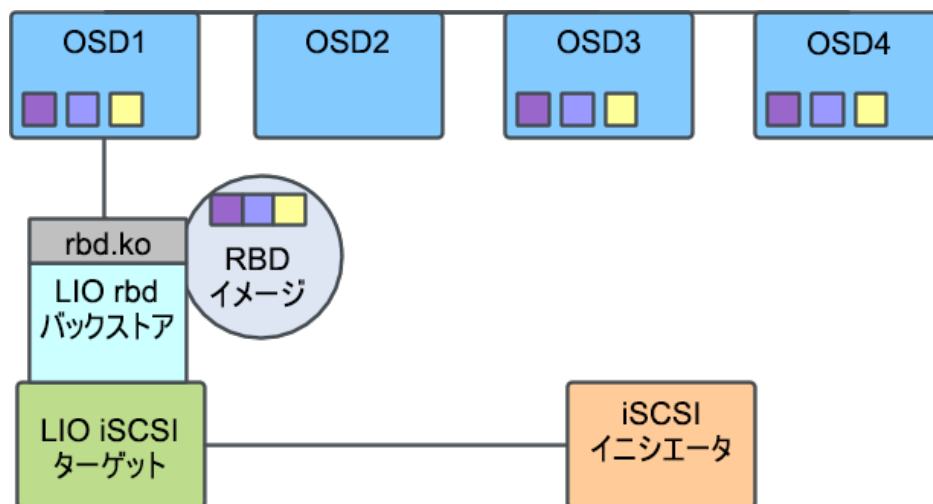


図 6.1: 1つのiSCSI GATEWAYで構成されるCEPHクラスタ

`ceph-iscsi`は本質的に高可用性であり、マルチパス操作をサポートしています。したがって、ダウンストリームのイニシエータホストは、複数のiSCSI Gatewayを使用して高可用性とスケーラビリティの両方を実現できます。複数のゲートウェイで構成されるiSCSI設定で通信する場合、イニシエータはiSCSI要求を複数のゲートウェイに負荷分散できます。ゲートウェイに障害が発生したり、一時的にアクセス不可能であったり、保守のために無効になっていたりする場合、I/Oは別のゲートウェイ経由で透過的に継続されます。

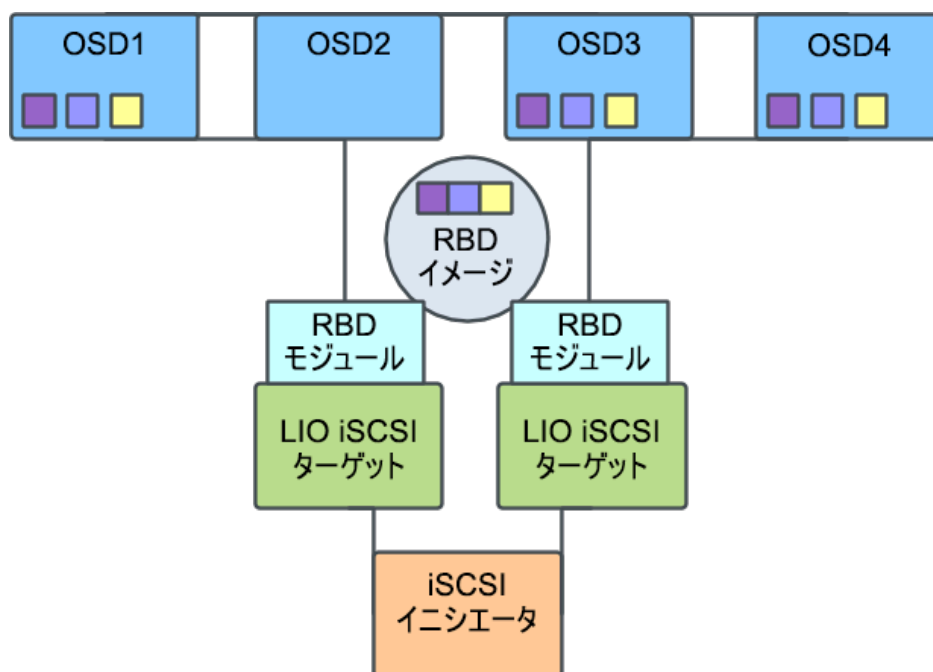


図 6.2: 複数のiSCSI GATEWAYで構成されるCEPHクラスタ

## 6.3 展開に関する考慮事項

SUSE Enterprise Storage 7と`ceph-iscsi`の最小設定は以下のコンポーネントで構成されます。

- Ceph Storage Cluster。Cephクラスタは、それぞれが8つ以上のOSD (オブジェクトストレージデーモン)をホストする少なくとも4台の物理サーバで構成されます。このような設定では、3つのOSDノードがモニタ(MON)ホストとしての役割も持ちます。
- LIO iSCSIターゲットを実行する1つのiSCSIターゲットサーバ。`ceph-iscsi`で設定します。
- 1つのiSCSIイニシエータホスト。`open-iscsi` (Linux)、Microsoft iSCSIイニシエーター (Microsoft Windows)、または互換性があるその他のiSCSIイニシエータ実装を実行します。

SUSE Enterprise Storage 7と`ceph-iscsi`の推奨運用設定は以下で構成されます。

- Ceph Storage Cluster。運用Cephクラスタは任意の数(通常は11以上)のOSDノードで構成されます。一般的にはそれぞれが10～12のOSD (オブジェクトストレージデーモン)を実行し、少なくとも3つの専用のMONホストを持ちます。
- LIO iSCSIターゲットを実行する複数のiSCSIターゲットサーバ。 `ceph-iscsi`で設定します。iSCSIのフェールオーバーと負荷分散を行うには、これらのサーバで、`target_core_rbd`モジュールをサポートするカーネルを実行する必要があります。更新パッケージはSUSE Linux Enterprise Server保守チャネルから入手できます。
- 任意の数のiSCSIイニシエータホスト。 `open-iscsi` (Linux)、Microsoft iSCSIイニシエータ(Microsoft Windows)、または互換性があるその他のiSCSIイニシエータ実装を実行します。

## 6.4 インストールと設定

このセクションでは、SUSE Enterprise StorageにiSCSI Gatewayをインストールして設定する手順について説明します。

### 6.4.1 CephクラスタへのiSCSI Gatewayの展開

Ceph iSCSI Gatewayの展開は他のCephサービスの展開と同じ手順で行われます。すなわち、`cephadm`を使用します。詳細については、「[5.4.3.5項 「iSCSI Gatewayの展開」](#)」を参照してください。

### 6.4.2 RBDイメージの作成

RBDイメージはCephストア内に作成され、その後iSCSIにエクスポートされます。この目的のため、専用のRADOSプールを使用することをお勧めします。Ceph `rbd`コマンドラインユーティリティを使用してStorage Clusterに接続できる任意のホストからボリュームを作成できます。このためには、クライアントが少なくとも最小限の`ceph.conf`設定ファイルとCephX認証資格情報を持っている必要があります。

以降iSCSI経由でエクスポートするために新しいボリュームを作成するには、`rbd create`コマンドを使用して、ボリュームサイズをメガバイト単位で指定します。たとえば、`iscsi-images`という名前のプールに`testvol`という名前の100GBのボリュームを作成するには、次のコマンドを実行します。

```
cephuser@adm > rbd --pool iscsi-images create --size=102400 testvol
```

### 6.4.3 iSCSIを経由したRBDイメージのエクスポート

iSCSI経由でRBDイメージをエクスポートするには、CephダッシュボードWebインタフェースか、`ceph-iscsi gwcli`ユーティリティのいずれかを使用できます。このセクションでは、`gwcli`にのみ焦点を当て、コマンドラインを使用してRBDイメージをエクスポートするiSCSIターゲットを作成する方法を示します。



#### 注記

次のプロパティを持つRBDイメージは、iSCSI経由ではエクスポートできません。

- `journaling`機能が有効化されたイメージ
- `stripe_unit`が4096バイト未満のイメージ

`root`として、iSCSI Gatewayのコンテナを入力します。

```
root # cephadm enter --name CONTAINER_NAME
```

`root`として、iSCSI Gatewayのコマンドラインインタフェースを起動します。

```
root # gwcli
```

`iscsi-targets`に移動して、次の名前のターゲットを作成します。 `iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol`

```
gwcli > /> cd /iscsi-targets
gwcli > /iscsi-targets> create iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol
```

iSCSI Gatewayの `name` と `ip` アドレスを指定して、ゲートウェイを作成します。

```
gwcli > /iscsi-targets> cd iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol/gateways
gwcli > /iscsi-target...tvol/gateways> create iscsi1 192.168.124.104
gwcli > /iscsi-target...tvol/gateways> create iscsi2 192.168.124.105
```



#### ヒント

現在の設定ノードで使用可能なコマンドのリストを表示するには、`help`コマンドを使用します。

`testvol`という名前のRBDイメージをプール`iscsi-images`に追加します。

```
gwcli > /iscsi-target...testvol/gateways> cd /disks
gwcli > /disks> attach iscsi-images/testvol
```

RBDイメージをターゲットにマップします。

```
gwcli > /disks> cd /iscsi-targets/iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol/
disks
gwcli > /iscsi-target...testvol/disks> add iscsi-images/testvol
```



## 注記

`targetcli`などの下位レベルのツールを使用してローカル設定を照会することができますが、設定を変更しないでください。



## ヒント

`ls`コマンドを使用して、設定を確認できます。一部の設定ノードは、`info`コマンドもサポートしています。このコマンドを使用すると、詳細情報を表示できます。

デフォルトではACL認証が有効になっているため、このターゲットにはまだアクセスできません。認証とアクセス制御の詳細については、[6.4.4項「認証とアクセス制御」](#)を確認してください。

## 6.4.4 認証とアクセス制御

iSCSI認証は柔軟性があり、多数の認証方法に対応しています。

### 6.4.4.1 ACL認証の無効化

「認証なし」とは、イニシエータが、対応するターゲット上のすべてのLUNにアクセスできることを意味します。「認証なし」を有効にするには、ACL認証を無効にします。

```
gwcli > /> cd /iscsi-targets/iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol/hosts
gwcli > /iscsi-target...testvol/hosts> auth disable_acl
```

#### 6.4.4.2 ACL認証の使用

イニシエータ名ベースのACL認証の使用時には、定義されたイニシエータのみが接続を許可されます。以下を実行して、イニシエータを定義できます。

```
gwcli > /> cd /iscsi-targets/iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol/hosts
gwcli > /iscsi-target...testvol/hosts> create iqn.1996-04.de.suse:01:e6ca28cc9f20
```

定義されているイニシエータは接続できますが、イニシエータに明示的に追加されたRBDイメージにのみアクセスできます。

```
gwcli > /iscsi-target...:e6ca28cc9f20> disk add rbd/testvol
```

#### 6.4.4.3 CHAP認証の有効化

ACLに加えて、各イニシエータのユーザ名とパスワードを指定して、CHAP認証を有効にできます。

```
gwcli > /> cd /iscsi-targets/iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol/hosts/
iqn.1996-04.de.suse:01:e6ca28cc9f20
gwcli > /iscsi-target...:e6ca28cc9f20> auth username=common12 password=pass12345678
```



#### 注記

ユーザ名は8～64文字の長さが必要で、英数字と記号「.」、「@」、「-」、「\_」、「:」を使用できます。

パスワードは12～16文字の長さが必要で、英数字と記号「@」、「-」、「\_」、「/」を使用できます。

必要に応じて、**auth**コマンドでmutual\_usernameパラメータとmutual\_passwordパラメータを指定して、CHAP相互認証を有効にすることもできます。

#### 6.4.4.4 検出認証と相互認証の設定

「検出認証」は、前の認証方法とは異なります。参照用の資格情報が必要です。これはオプションで、次のコマンドによって設定できます。

```
gwcli > /> cd /iscsi-targets
gwcli > /iscsi-targets> discovery_auth username=du123456 password=dp1234567890
```



## 注記

ユーザ名は8～64文字の長さが必要で、英数字と記号「.」、「@」、「-」、「\_」、「:」を使用できます。

パスワードは12～16文字の長さが必要で、英数字と記号「@」、「-」、「\_」、「/」を使用できます。

オプションで、**discovery\_auth**コマンドでmutual\_usernameパラメータとmutual\_passwordパラメータを指定することもできます。

検出認証は、次のコマンドを使用して無効にすることができます。

```
gwcli > /iscsi-targets> discovery_auth nochap
```

## 6.4.5 高度な設定

高度なパラメータを使用してceph-iscsiを設定し、設定したパラメータをその後LIO I/Oターゲットに渡すことができます。パラメータは、ターゲットのパラメータとディスクのパラメータに分かれています。



## 警告

特に明記されていない限り、これらのパラメータをデフォルト設定から変更することは推奨しません。

### 6.4.5.1 ターゲット設定の表示

**info**コマンドを使用して、これらの設定の値を表示できます。

```
gwcli > /> cd /iscsi-targets/iqn.2003-01.org.linux-iscsi.iscsi.SYSTEM-ARCH:testvol
gwcli > /iscsi-target...i.SYSTEM-ARCH:testvol> info
```

また、**reconfigure**コマンドを使用して、設定を変更します。

```
gwcli > /iscsi-target...i.SYSTEM-ARCH:testvol> reconfigure login_timeout 20
```

使用可能なターゲット設定は、次のとおりです。

#### default\_cmdsn\_depth

CmdSN (コマンドシーケンス番号)のデフォルトの深さ。特定の時点でiSCSIイニシエータが未処理の状態にしておくことができる要求の量を制限します。

**default\_eri**

デフォルトのエラー回復レベル。

**login\_timeout**

ログインタイムアウトの値(秒)。

**netif\_timeout**

NICの障害タイムアウト(秒)。

**prod\_mode\_write\_protect**

1に設定すると、LUNへの書き込みを防止します。

### 6.4.5.2 ディスク設定の表示

**info**コマンドを使用して、これらの設定の値を表示できます。

```
gwcli > /> cd /disks/rbd/testvol  
gwcli > /disks/rbd/testvol> info
```

また、**reconfigure**コマンドを使用して、設定を変更します。

```
gwcli > /disks/rbd/testvol> reconfigure rbd/testvol emulate_pr 0
```

使用可能なディスク設定は次のとおりです。

**block\_size**

基礎となるデバイスのブロックサイズ。

**emulate\_3pc**

1に設定すると、サードパーティコピーが有効になります。

**emulate\_caw**

1に設定すると、Compare and Writeが有効になります。

**emulate\_dpo**

1に設定すると、Disable Page Outがオンになります。

**emulate\_fua\_read**

1に設定すると、Force Unit Access読み込みが有効になります。

**emulate\_fua\_write**

1に設定すると、Force Unit Access書き込みが有効になります。

### **emulate\_model\_alias**

1に設定すると、モデルのエイリアスに対してバックエンドデバイス名が使用されます。

### **emulate\_pr**

0に設定すると、Persistent Group Reservationを含む、SCSI予約のサポートが無効になります。無効になっている間、SES iSCSI Gatewayは予約状態を無視できるため、要求の遅延が改善されます。



### **ヒント**

iSCSIイニシエータでSCSI予約のサポートが必要ない場合は、backstore\_emulate\_prを0に設定することをお勧めします。

### **emulate\_rest\_reord**

0に設定すると、Queue Algorithm ModifierにRestricted Reorderingが設定されます。

### **emulate\_tas**

1に設定すると、Task Aborted状態が有効になります。

### **emulate\_tpu**

1に設定すると、Thin Provisioning Unmapが有効になります。

### **emulate\_tpws**

1に設定すると、Thin Provisioning Write Sameが有効になります。

### **emulate\_ua\_intlck\_ctrl**

1に設定すると、Unit Attention Interlockが有効になります。

### **emulate\_write\_cache**

1に設定すると、Write Cache Enableが有効になります。

### **enforce\_pr\_isids**

1に設定すると、ISIDの永続的な予約が強制されます。

### **is\_nonrot**

1に設定すると、バックストアは非ローテーションデバイスになります。

### **max\_unmap\_block\_desc\_count**

UNMAPのブロック記述子の最大数。

### **max\_unmap\_lba\_count:**

UNMAPのLBAの最大数。

#### **max\_write\_same\_len**

WRITE\_SAMEの最大長。

#### **optimal\_sectors**

最適な要求サイズ(セクタ単位)。

#### **pi\_prot\_type**

DIF保護タイプ。

#### **queue\_depth**

キューの深さ。

#### **unmap\_granularity**

UNMAPの細分性。

#### **unmap\_granularity\_alignment**

UNMAPの細分性の配置。

#### **force\_pr\_aptpl**

有効にすると、クライアントが**aptpl=1**によって要求したかどうかに関係なく、LIOは常に永続ストレージに「永続予約」状態を書き出します。これは、LIOのカーネルRBDバックエンドには影響しません。常にPR状態を永続化します。これを`target_core_rbd`オプションで強制的に「1」に設定し、誰かが設定で無効にしようとした場合はエラーをスローするのが理想的です。

#### **unmap\_zeroes\_data**

LIOがLBPRZをSCSIイニシエータにアドバタイズするかどうかに影響します。これは、マップ解除ビットを使用したUNMAPまたはWRITE SAMEの後に、領域から0が読み込まれることを示します。

## 6.5 tcmu-runnerを使用したRADOS Block Deviceイメージのエクスポート

`ceph-iscsi`は、`rbd` (カーネルベース)および`user:rbd` (`tcmu-runner`)の両方のバックストアをサポートしており、すべての管理をバックストアから独立して透過的に実行できます。



### 警告: 技術プレビュー

`tcmu-runner`ベースのiSCSI Gatewayの展開は現在のところ技術プレビューです。

カーネルベースのiSCSI Gatewayの展開と異なり、tcmu-runnerベースのiSCSI Gatewayの展開では、マルチパスI/OやSCSIの永続的な予約はサポートされません。

tcmu-runnerを使用してRADOS Block Deviceをエクスポートするには、ディスクの接続時にuser: rbdバックストアを指定することのみが必要です。

```
gwcli > /disks> attach rbd/testvol backstore=user:rbd
```



## 注記

tcmu-runnerを使用する場合、エクスポートされたRBDイメージでexclusive-lock機能が有効になっている必要があります。

## IV 古いリリースからのアップグレード

### 7 前回リリースからのアップグレード 80

## 7 前回リリースからのアップグレード

この章では、SUSE Enterprise Storage 6をバージョン7にアップグレードする手順について説明します。

アップグレードには次のタスクが含まれます。

- Ceph NautilusからCeph Octopusへのアップグレード。
- RPMパッケージを介してCephのインストールと実行を行う環境から、コンテナ内で実行する環境への切り替え。
- DeepSeaを完全に消去し、`ceph-salt`と`cephadm`で置き換え。



### 警告

この章のアップグレード情報は、DeepSeaからcephadmへのアップグレードに「のみ」適用されます。SUSE CaaS Platform上にSUSE Enterprise Storageを展開する場合は、この章の手順を使用しないでください。



### 重要

6より古いバージョンのSUSE Enterprise Storageからのアップグレードはサポートされていません。まず、最新バージョンのSUSE Enterprise Storage 6にアップグレードしてから、この章の手順を実行する必要があります。

## 7.1 アップグレード実行前の確認事項

アップグレードの開始前に、必ず以下のタスクを完了させてください。このタスクはSUSE Enterprise Storage 6のライフタイムのどの時点でも実行できます。

- FileStoreからBlueStoreへのOSDマイグレーションは、必ずアップグレード前に行ってください。SUSE Enterprise Storage 7がFileStoreをサポートしていないためです。BlueStoreの詳細と、FileStoreからBlueStoreへのマイグレーション方法の詳細については、<https://documentation.suse.com/ses/6/html/ses-all/cha-ceph-upgrade.html#filestore2bluestore> を参照してください。
- `ceph-disk` OSDを使用するクラスタを実行中の場合は、アップグレード前に必ず `ceph-volume` へ切り替えてください。詳細については、<https://documentation.suse.com/ses/6/html/ses-all/cha-ceph-upgrade.html#upgrade-osd-deployment> を参照してください。

### 7.1.1 考慮すべきポイント

アップグレードの前に以下のセクションを熟読して、実行する必要があるすべてのタスクを十分に理解してください。

- 「リリースノートをお読みください」 - 旧リリースのSUSE Enterprise Storageからの変更点に関する追加情報が記載されています。リリースノート参照して以下を確認します。
  - 使用しているハードウェアに特別な配慮が必要かどうか
  - 使用しているソフトウェアパッケージに大幅な変更があるかどうか
  - インストールのために特別な注意が必要かどうか

リリースノートには、マニュアルに記載できなかった情報が記載されています。また、既知の問題に関する注意も記載されています。

SES 7のリリースノートは<https://www.suse.com/releasesnotes/> を参照してください。

あるいは、パッケージ `release-notes-ses` をSES 7リポジトリからインストールすると、ローカルディレクトリ `/usr/share/doc/release-notes` にリリースノートが置かれます。オンラインのリリースノート<https://www.suse.com/releasesnotes/> も利用できます。

- 第5章「`cephadm`による展開」を参照して、`ceph-salt`とCephオーケストレータについて理解してください。特に、サービス仕様に記載されている情報は重要です。
- クラスタのアップグレードには長い時間がかかることがあります。所要時間は、1台のマシンのアップグレード時間xクラスタノード数です。


- 最初にSalt Masterをアップグレードし、その後DeepSeaを `ceph-salt` と `cephadm` に置き換える必要があります。少なくとも、すべてのCeph Managerがアップグレードされるまで、`cephadm` オークストレータモジュールの使用を開始することはできません。
- NautilusのRPMを使用する環境からOctopusのコンテナ環境へアップグレードは、一度に行う必要があります。これは、一度に1つのデーモンではなく、ノード全体を一度にアップグレードすることを意味します。
- コアサービス(MON、MGR、OSD)のアップグレードは、順序立てて進めます。アップグレードの間も、各サービスは利用可能です。ゲートウェイサービス(メタデータサーバ、Object Gateway、NFS Ganesha、iSCSI Gateway)については、コアサービスのアップグレード後に再展開する必要があります。次に示すサービスごとに、ある程度のダウンタイムが発生します。

### ● 重要

メタデータサーバとObject Gatewaysについては、ノードをSUSE Linux Enterprise Server 15 SP1からSUSE Linux Enterprise Server 15 SP2にアップグレードする際にダウンします。ダウン状態はアップグレード手続きの最後にサービスが再展開されるまで続きます。特に注意が必要なのは、メタデータサーバとObject GatewaysをMON、MGR、OSDなどと同じ場所に配置している場合です。この場合、クラスタのアップグレードが完了するまでこれらのサービスを利用できない可能性があります。これが問題となる場合は、これらのサービスをアップグレード前に別のノードに分けて展開することを検討してください。そうすれば、ダウンタイムは最小限で済みます。この場合、ダウンする期間はクラスタ全体のアップグレード中ではなく、ゲートウェイノードのアップグレード中になります。

- NFS GaneshaとiSCSI Gatewaysについては、SUSE Linux Enterprise Server 15 SP1からSUSE Linux Enterprise Server 15 SP2へアップグレードする手続きの中で、ノードがリブートしている間にダウンします。また、各サービスがコンテナ化モードで再展開する際にも一時的にダウンします。

## 7.1.2 クラスタ設定とデータのバックアップ

SUSE Enterprise Storage 7にアップグレードする前に、すべてのクラスタの設定とデータをバックアップすることを強く推奨します。すべてのデータをバックアップする方法については、<https://documentation.suse.com/ses/6/html/ses-all/cha-deployment-backup.html>  を参照してください。

### 7.1.3 前回のアップグレード手順の確認

以前にバージョン5からアップグレードした場合は、バージョン6へのアップグレードが正常に完了していることを確認します。

`/srv/salt/ceph/configuration/files/ceph.conf.import`というファイルが存在することを確認します。

このファイルはSUSE Enterprise Storage 5から6へのアップグレード中に、engulfプロセスにより作成されます。`configuration_init: default-import`オプションは`/srv/pillar/ceph/proposals/config/stack/default/ceph/cluster.yml`に設定されます。

`configuration_init`がまだ`default-import`に設定されている場合、クラスタはその設定ファイルとして`ceph.conf.import`を使用しています。これは、`/srv/salt/ceph/configuration/files/ceph.conf.d/`にあるファイルからコンパイルされるDeepSeaのデフォルトの`ceph.conf`ではありません。

したがって、`ceph.conf.import`でカスタム設定を調べ、可能であれば、`/srv/salt/ceph/configuration/files/ceph.conf.d/`にあるファイルのいずれかに設定を移動する必要があります。

その後、`/srv/pillar/ceph/proposals/config/stack/default/ceph/cluster.yml`から`configuration_init: default-import`行を削除してください。

### 7.1.4 クラスタノードのアップデートとクラスタのヘルスの確認

SUSE Linux Enterprise Server 15 SP1とSUSE Enterprise Storage 6のすべての最新のアップデートがすべてのクラスタノードに適用されていることを確認してください。

```
root # zypper refresh && zypper patch
```

アップデートの適用後、クラスタのヘルスを確認してください。

```
cephuser@adm > ceph -s
```

### 7.1.5 ソフトウェアリポジトリとコンテナイメージへのアクセス確認

各クラスタノードがSUSE Linux Enterprise Server 15 SP2とSUSE Enterprise Storage 7のソフトウェアリポジトリとコンテナイメージのリポジトリにアクセスできることを確認してください。

### 7.1.5.1 ソフトウェアリポジトリ

すべてのノードがSCCに登録されている場合、**zypper migration**コマンドによるアップグレードが可能です。詳細については、<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-upgrade-online.html#sec-upgrade-online-zypper>を参照してください。

ノードがSCCに登録されていない場合は、すべての既存のソフトウェアリポジトリを無効化し、次に示す各エクステンションにPoolリポジトリとUpdatesリポジトリの両方を追加してください。

- SLE-Product-SLES/15-SP2
- SLE-Module-Basesystem/15-SP2
- SLE-Module-Server-Applications/15-SP2
- SUSE-Enterprise-Storage-7

### 7.1.5.2 コンテナイメージ

すべてのクラスタノードはコンテナイメージのレジストリにアクセスする必要があります。多くの場合、[registry.suse.com](https://registry.suse.com)のパブリックSUSEレジストリを使用します。必要なイメージは次のとおりです。

- [registry.suse.com/ses/7/ceph/ceph](https://registry.suse.com/ses/7/ceph/ceph)
- [registry.suse.com/ses/7/ceph/grafana](https://registry.suse.com/ses/7/ceph/grafana)
- [registry.suse.com/caasp/v4.5/prometheus-server](https://registry.suse.com/caasp/v4.5/prometheus-server)
- [registry.suse.com/caasp/v4.5/prometheus-node-exporter](https://registry.suse.com/caasp/v4.5/prometheus-node-exporter)
- [registry.suse.com/caasp/v4.5/prometheus-alertmanager](https://registry.suse.com/caasp/v4.5/prometheus-alertmanager)

もしくは、たとえばエアギャップ環境で展開したい場合などは、ローカルレジストリを設定し、正常なコンテナイメージのセットが利用できることを確認してください。ローカルなコンテナイメージのレジストリを設定する方法の詳細については、[5.3.2.11項「コンテナレジストリの設定」](#)を参照してください。

## 7.2 Salt Masterのアップグレード

Salt Masterのアップグレードプロセスを以下に示します。

1. 基礎となるOSをSUSE Linux Enterprise Server 15 SP2にアップグレードします。

- すべてのノードがSCCに登録されているクラスタの場合は、**zypper migration** コマンドを実行します。
- 手動で割り当てられたソフトウェアリポジトリを含むクラスタの場合は、**zypper dup** コマンドを実行した後、**reboot** コマンドを実行します。

2. 誤って使用しないように、DeepSeaステージを無効化します。/srv/pillar/ceph/stack/global.ymlに次の内容を追加します。

```
stage_prep: disabled
stage_discovery: disabled
stage_configure: disabled
stage_deploy: disabled
stage_services: disabled
stage_remove: disabled
```

ファイルを保存し、変更を適用します。

```
root@master # salt '*' saltutil.pillar_refresh
```

3. registry.suse.comのコンテナイメージではなく、ローカル環境に設定したレジストリを使用している場合は、/srv/pillar/ceph/stack/global.ymlを編集して、DeepSeaがどのCephコンテナイメージとレジストリを使用するか指定します。たとえば、192.168.121.1:5000/my/ceph/imageを使用する場合は、次に示す内容を追加します。

```
ses7_container_image: 192.168.121.1:5000/my/ceph/image
ses7_container_registries:
  - location: 192.168.121.1:5000
```

ファイルを保存し、変更を適用します。

```
root@master # salt '*' saltutil.refresh_pillar
```

4. 既存の設定を取り込みます。

```
cephuser@adm > ceph config assimilate-conf -i /etc/ceph/ceph.conf
```

5. アップグレードステータスを確認します。クラスタの設定によって、出力は異なる場合があります。

```
root@master # salt-run upgrade.status
The newest installed software versions are:
```

```

ceph: ceph version 15.2.2-60-gf5864377ab (f5864377abb5549f843784c93577980aa264b9bc)
octopus (stable)
os: SUSE Linux Enterprise Server 15 SP2
Nodes running these software versions:
  admin.ceph (assigned roles: master, prometheus, grafana)
Nodes running older software versions must be upgraded in the following order:
  1: mon1.ceph (assigned roles: admin, mon, mgr)
  2: mon2.ceph (assigned roles: admin, mon, mgr)
  3: mon3.ceph (assigned roles: admin, mon, mgr)
  4: data4.ceph (assigned roles: storage, mds)
  5: data1.ceph (assigned roles: storage)
  6: data2.ceph (assigned roles: storage)
  7: data3.ceph (assigned roles: storage)
  8: data5.ceph (assigned roles: storage, rgw)

```

## 7.3 MON、MGR、OSDノードのアップグレード

Ceph Monitor、Ceph Manager、OSDのノードを一度にアップグレードしてください。サービスごとに、次の手順に従います。

1. アップグレードするノードがOSDノードの場合、アップグレード中にOSDが`out`とマークされることを避けるため、次のコマンドを実行します。

```
cephuser@adm > ceph osd add-noout SHORT_NODE_NAME
```

`SHORT_NODE_NAME`はノードの略称で置き換えます。この名称が**`ceph osd tree`**コマンドの出力に表示されます。たとえば、以下の入力ではホスト名の略称が`ses-min1`と`ses-min2`の場合です。

```

root@master # ceph osd tree
ID   CLASS  WEIGHT  TYPE NAME        STATUS  REWEIGHT  PRI-AFF
-1               0.60405  root default
-11               0.11691  host ses-min1
  4   hdd   0.01949    osd.4      up     1.00000   1.00000
  9   hdd   0.01949    osd.9      up     1.00000   1.00000
 13   hdd   0.01949    osd.13     up     1.00000   1.00000
[...]
-5               0.11691  host ses-min2
  2   hdd   0.01949    osd.2      up     1.00000   1.00000
  5   hdd   0.01949    osd.5      up     1.00000   1.00000
[...]

```

2. 基礎となるOSをSUSE Linux Enterprise Server 15 SP2にアップグレードします。

- クラスタのノードがすべてSCCに登録されている場合は、**zypper migration**を実行します。
  - 手動で割り当てられたソフトウェアリポジトリを含むクラスタノードの場合は、**zypper dup**を実行した後、**reboot**コマンドを実行します。
3. ノードの再起動後、Salt Master上で以下のコマンドを実行して、ノード上のすべての既存のMONデーモン、MGRデーモン、OSDデーモンをコンテナ化します。

```
root@master # salt MINION_ID state.apply ceph.upgrade.ses7.adopt
```

MINION\_IDはアップグレードするミニオンのIDで置き換えます。Salt Master上で**salt-key -L**コマンドを実行することで、ミニオンIDのリストを取得できます。



## ヒント

「導入」の進捗状況を確認するには、Cephダッシュボードを確認するか、Salt Master上で以下のコマンドのいずれかを実行します。

```
root@master # ceph status
root@master # ceph versions
root@master # salt-run upgrade.status
```

4. OSDノードをアップグレード中の場合、導入が正常に完了した後でnooutフラグの設定を解除してください。

```
cephuser@adm > ceph osd rm-noout SHORT_NODE_NAME
```

## 7.4 ゲートウェイノードのアップグレード

次に、個別のゲートウェイノード(メタデータサーバ、Object Gateway、NFS Ganesha、iSCSI Gateway)をアップグレードしてください。各ノードに基礎となるOSをSUSE Linux Enterprise Server 15 SP2にアップグレードしてください。

- クラスタのノードがすべてSUSE Customer Centerに登録されている場合は、**zypper migration**コマンドを実行してください。
- 手動で割り当てられたソフトウェアリポジトリを含むクラスタノードの場合は、**zypper dup**コマンドを実行した後、**reboot**コマンドを実行してください。

この手順はクラスタの一部であるが、まだ役割を割り当てていないノードにも適用します(割り当てたかどうか不明な場合は、Salt Master上で`salt-key -L`コマンドを実行してホストのリストを取得し、`salt-run upgrade.status`コマンドの出力と比較してください)。

クラスタに含まれる全ノードのOSをアップグレードした後、次のステップで `ceph-salt` パッケージをインストールし、クラスタ設定を適用します。ゲートウェイサービスそのものは、アップグレード処理の最後にコンテナ化モードで再展開されます。



## 注記

メタデータサーバとObject Gatewayサービスは、SUSE Linux Enterprise Server 15 SP2へのアップグレードが始まると利用できなくなります。この状態はアップグレード処理の最後にサービスが再展開されるまで続きます。

## 7.5 `ceph-salt`のインストールと、クラスタ設定の適用

`ceph-salt`のインストールとクラスタ設定の適用を開始する前に、次のコマンドを実行してクラスタとアップグレードステータスを確認してください。

```
root@master # ceph status
root@master # ceph versions
root@master # salt-run upgrade.status
```

1. DeepSeaが作成した`rbd_exporter`と`rgw_exporter`というcron jobを削除します。Salt Master上で`root`として`crontab -e`コマンドを実行し、`crontab`を編集します。以下の項目が存在する場合は削除します。

```
# SALT_CRON_IDENTIFIER:deepsea rbd_exporter cron job
*/5 * * * * /var/lib/prometheus/node-exporter/rbd.sh > \
/var/lib/prometheus/node-exporter/rbd.prom 2> /dev/null
# SALT_CRON_IDENTIFIER:Prometheus rgw_exporter cron job
*/5 * * * * /var/lib/prometheus/node-exporter/ceph_rgw.py > \
/var/lib/prometheus/node-exporter/ceph_rgw.prom 2> /dev/null
```

2. 次のコマンドを実行して、DeepSeaからクラスタ設定をエクスポートします。

```
root@master # salt-run upgrade.ceph_salt_config > ceph-salt-config.json
root@master # salt-run upgrade.generate_service_specs > specs.yaml
```

3. DeepSeaをアンインストールし、`ceph-salt`をSalt Masterにインストールします。

```
root@master # zypper remove 'deepsea*'
root@master # zypper install ceph-salt
```

4. Salt Masterを再起動し、Saltモジュールを同期します。

```
root@master # systemctl restart salt-master.service
root@master # salt '*' saltutil.sync_all
```

5. DeepSeaのクラスタ設定を`ceph-salt`にインポートします。

```
root@master # ceph-salt import ceph-salt-config.json
```

6. クラスタノード通信用のSSHキーを作成します。

```
root@master # ceph-salt config /ssh generate
```



## ヒント

DeepSeaからクラスタ設定がインポートされたことを確認し、欠落している可能性のあるオプションを指定します。

```
root@master # ceph-salt config ls
```

クラスタ設定の詳細については[5.3.2項「クラスタプロパティの設定」](#)を参照してください。

7. 設定を適用し、`cephadm`を有効化します。

```
root@master # ceph-salt apply
```

8. ローカルのコンテナレジストリURLやアクセス資格情報を提供する場合がある場合は、[5.3.2.11項「コンテナレジストリの設定」](#)の手順に従ってください。
9. [registry.suse.com](#)のコンテナイメージではなく、ローカルに設定したレジストリを使う場合は、次のコマンドを実行してどのコンテナイメージを使用するかをCephに伝えます。

```
root@master # ceph config set global container_image IMAGE_NAME
```

次に例を示します。

```
root@master # ceph config set global container_image 192.168.121.1:5000/my/ceph/
image
```

10. SUSE Enterprise Storage 6のceph-crashデーモンを停止し、無効化します。これらのデーモンの新しくコンテナ化された形式は、後ほど自動的に起動します。

```
root@master # salt '*' service.stop ceph-crash
root@master # salt '*' service.disable ceph-crash
```

## 7.6 監視スタックのアップグレードと導入

以下の手順によって、監視スタックのすべてのコンポーネントを導入します(詳細については『運用と管理ガイド』、第16章「監視とアラート」を参照してください)。

1. オーケストレータを一時停止します。

```
cephuser@adm > ceph orch pause
```

2. Prometheus、Grafana、Alertmanagerがどのノードで実行されている場合でも(デフォルトではSalt Masterノード)、以下のコマンドを実行します。

```
cephuser@adm > cephadm adopt --style=legacy --name prometheus.${hostname}
cephuser@adm > cephadm adopt --style=legacy --name alertmanager.${hostname}
cephuser@adm > cephadm adopt --style=legacy --name grafana.${hostname}
```



### ヒント

registry.suse.comのデフォルトコンテナイメージレジストリを実行していない場合は、使用するイメージを指定する必要があります。以下に例を示します。

```
cephuser@adm > cephadm --image 192.168.121.1:5000/caasp/v4.5/prometheus-
server:2.18.0 \
  adopt --style=legacy --name prometheus.${hostname}
cephuser@adm > cephadm --image 192.168.121.1:5000/caasp/v4.5/prometheus-
alertmanager:0.16.2 \
  adopt --style=legacy --name alertmanager.${hostname}
cephuser@adm > cephadm --image 192.168.121.1:5000/ses/7/ceph/grafana:7.0.3 \
  adopt --style=legacy --name grafana.${hostname}
```

カスタムまたはローカルのコンテナイメージの使用方法的詳細については、『運用と管理ガイド』、第16章「監視とアラート」、16.1項「カスタムイメージまたはローカルイメージの設定」を参照してください。

3. Node-Exporterを削除します。Node-Exporterのマイグレーションは不要です。specs.yamlファイルが適用された際にコンテナとして再インストールされます。

```
tux > sudo zypper rm golang-github-prometheus-node_exporter
```

4. DeepSeaからエクスポートしておいたサービス仕様を適用します。

```
cephuser@adm > ceph orch apply -i specs.yaml
```

5. オーケストレータを再開します。

```
cephuser@adm > ceph orch resume
```

## 7.7 ゲートウェイサービスの再展開

### 7.7.1 Object Gatewayのアップグレード

SUSE Enterprise Storage 7においてObject Gatewayは常にレルムに設定されます。これにより、将来的なマルチサイトが可能になります(詳細については『運用と管理ガイド』、第21章「Ceph Object Gateway」、21.13項「マルチサイトObject Gateway」を参照してください)。SUSE Enterprise Storage 6でシングルサイトのObject Gateway設定を使用している場合は、以下の手順に従ってレルムを追加してください。マルチサイト機能を実際に使う予定がない場合は、レルム、ゾーングループ、ゾーンの名前に`default`を使用してもかまいません。

1. 新しいレルムを作成します。

```
cephuser@adm > radosgw-admin realm create --rgw-realm=REALM_NAME --default
```

2. 必要に応じて、デフォルトのゾーンとゾーングループの名前を変更します。

```
cephuser@adm > radosgw-admin zonegroup rename \  
--rgw-zonegroup default \  
--zonegroup-new-name=ZONEGROUP_NAME  
cephuser@adm > radosgw-admin zone rename \  
--rgw-zone default \  
--zone-new-name ZONE_NAME \  
--rgw-zonegroup=ZONEGROUP_NAME
```

3. マスターゾーングループを設定します。

```
cephuser@adm > radosgw-admin zonegroup modify \  
--rgw-realm=REALM_NAME \  
--rgw-zonegroup=ZONEGROUP_NAME \  
--endpoints http://RGW.EXAMPLE.COM:80 \  
--master --default
```

4. マスターゾーンを設定します。このとき、systemフラグが有効なObject GatewayユーザのACCESS\_KEYとSECRET\_KEYが必要になります。通常はadminユーザが該当します。ACCESS\_KEYとSECRET\_KEYを取得するには、radosgw-admin user info --uid adminを実行します。

```
cephuser@adm > radosgw-admin zone modify \  
--rgw-realm=REALM_NAME \  
--rgw-zonegroup=ZONEGROUP_NAME \  
--rgw-zone=ZONE_NAME \  
--endpoints http://RGW.EXAMPLE.COM:80 \  
--access-key=ACCESS_KEY \  
--secret=SECRET_KEY \  
--master --default
```

5. アップデートされた設定をコミットします。

```
cephuser@adm > radosgw-admin period update --commit
```

Object Gatewayサービスをコンテナ化するには、[5.4.3.4項「Object Gatewayの展開」](#)に記載されている仕様ファイルを作成し、適用します。

```
cephuser@adm > ceph orch apply -i RGW.yml
```

## 7.7.2 NFS Ganeshaのアップグレード

Ceph Nautilusを実行する既存のNFS Ganeshaサービスから、Ceph Octopusを実行するNFS Ganeshaコンテナにマイグレートする方法を次で説明します。



### 警告

以下の情報は、コアとなるCephサービスのアップグレードに成功していることを前提としたものです。

NFS Ganeshaはデーモンごとの追加設定を保存し、RADOSプールに設定をエクスポートします。設定済みのRADOSプールは、ganesha.confファイルのRADOS\_URLSブロックのwatch\_url行で確認できます。デフォルトでは、このプールはganesha\_configと名付けられます。

マイグレーションを試みる前に、RADOSプールに配置されたエクスポート設定オブジェクトとデーモン設定オブジェクトのコピーを作成しておくことを強く推奨します。設定済みのRADOSプールを検索するには、次のコマンドを実行します。

```
cephuser@adm > grep -A5 RADOS_URLS /etc/ganesha/ganesha.conf
```

RADOSプールの内容を一覧にするには、次のコマンドを実行します。

```
cephuser@adm > rados --pool ganesha_config --namespace ganesha ls | sort
conf-node3
export-1
export-2
export-3
export-4
```

RADOSオブジェクトをコピーするには、次のコマンドを実行します。

```
cephuser@adm > RADOS_ARGS="--pool ganesha_config --namespace ganesha"
cephuser@adm > OBJJS=$(rados $RADOS_ARGS ls)
cephuser@adm > for obj in $OBJJS; do rados $RADOS_ARGS get $obj $obj; done
cephuser@adm > ls -lah
total 40K
drwxr-xr-x 2 root root 4.0K Sep 8 03:30 .
drwx----- 9 root root 4.0K Sep 8 03:23 ..
-rw-r--r-- 1 root root 90 Sep 8 03:30 conf-node2
-rw-r--r-- 1 root root 90 Sep 8 03:30 conf-node3
-rw-r--r-- 1 root root 350 Sep 8 03:30 export-1
-rw-r--r-- 1 root root 350 Sep 8 03:30 export-2
-rw-r--r-- 1 root root 350 Sep 8 03:30 export-3
-rw-r--r-- 1 root root 358 Sep 8 03:30 export-4
```

ノードごとに既存のNFS Ganeshaサービスを停止して、cephadmが管理するコンテナに置き換える必要があります。

1. 既存のNFS Ganeshaサービスを停止および無効化します。

```
cephuser@adm > systemctl stop nfs-ganesha
cephuser@adm > systemctl disable nfs-ganesha
```

2. 既存のNFS Ganeshaサービスが停止すると、cephadmを用いてコンテナ内に新しいサービスを展開できます。そのためには、`service_id`を記述したサービス仕様を作成する必要があります。このIDはこの新しいNFSクラスタの特定、配置仕様にホストとして記載された、マイグレーション先ノードのホスト名の特定、設定済みNFSエクスポートオブジェクトを含むRADOSプールとネームスペースの特定に使用されます。次に例を示します。

```
service_type: nfs
service_id: SERVICE_ID
placement:
```

```
hosts:
- node2
pool: ganesha_config
namespace: ganesha
```

配置仕様の作成の詳細については、5.4.2項「サービス仕様と配置仕様」を参照してください。

3. 配置仕様を適用します。

```
cephuser@adm > ceph orch apply -i FILENAME.yaml
```

4. ホストでNFS Ganeshaデーモンが実行されていることを確認します。

```
cephuser@adm > ceph orch ps --daemon_type nfs
```

NAME	HOST	STATUS	REFRESHED	AGE	VERSION	IMAGE NAME
		IMAGE ID	CONTAINER ID			
nfs.foo.node2	node2	running (26m)	8m ago	27m	3.3	registry.suse.com/ses/7/ceph/ceph:latest
		8b4be7c42abd	c8b75d7c8f0d			

5. NFS Ganeshaノードごとに、これらの手順を繰り返します。ノードごとに別々のサービス仕様を作成する必要はありません。各ノードのホスト名を既存のNFSサービス仕様に追加し、再適用すれば十分です。

既存のエクスポートは次の2つの方法でマイグレートできます。

- Cephダッシュボードを使用して、手動で再作成と再適用を行う方法。
- デーモンごとのRADOSオブジェクトの内容を、新しく作成されたNFS Ganesha共通設定に手動でコピーする方法。

手順 7.1: エクスポートをNFS GANESHA共通設定ファイルに手動コピーする

1. デーモンごとのRADOSオブジェクトのリストを確認します。

```
cephuser@adm > RADOS_ARGS="--pool ganesha_config --namespace ganesha"
cephuser@adm > DAEMON_OBJS=$(rados $RADOS_ARGS ls | grep 'conf-')
```

2. デーモンごとのRADOSオブジェクトのコピーを作成します。

```
cephuser@adm > for obj in $DAEMON_OBJS; do rados $RADOS_ARGS get $obj $obj; done
cephuser@adm > ls -lah
total 20K
drwxr-xr-x 2 root root 4.0K Sep 8 16:51 .
drwxr-xr-x 3 root root 4.0K Sep 8 16:47 ..
-rw-r--r-- 1 root root 90 Sep 8 16:51 conf-nfs.SERVICE_ID
-rw-r--r-- 1 root root 90 Sep 8 16:51 conf-node2
-rw-r--r-- 1 root root 90 Sep 8 16:51 conf-node3
```

### 3. ソートとマージを行って、エクスポートを単一のリストにします。

```
cephuser@adm > cat conf-* | sort -u > conf-nfs.SERVICE_ID
cephuser@adm > cat conf-nfs.foo
%url "rados://ganesha_config/ganesha/export-1"
%url "rados://ganesha_config/ganesha/export-2"
%url "rados://ganesha_config/ganesha/export-3"
%url "rados://ganesha_config/ganesha/export-4"
```

### 4. 新しいNFS Ganesha共通設定ファイルを書き込みます。

```
cephuser@adm > rados $RADOS_ARGS put conf-nfs.SERVICE_ID conf-nfs.SERVICE_ID
```

### 5. NFS Ganeshaデーモンに通知します。

```
cephuser@adm > rados $RADOS_ARGS notify conf-nfs.SERVICE_ID conf-nfs.SERVICE_ID
```



## 注記

このアクションによって、デーモンは設定を再ロードします。

サービスのマイグレートに成功すると、NautilusベースのNFS Ganeshaサービスを削除できるようになります。

### 1. NFS Ganeshaを削除します。

```
cephuser@adm > zypper rm nfs-ganesha
Reading installed packages...
Resolving package dependencies...
The following 5 packages are going to be REMOVED:
  nfs-ganesha nfs-ganesha-ceph nfs-ganesha-rados-grace nfs-ganesha-rados-urls nfs-
ganesha-rgw
5 packages to remove.
After the operation, 308.9 KiB will be freed.
Continue? [y/n/v/...? shows all options] (y): y
(1/5) Removing nfs-ganesha-
ceph-2.8.3+git0.d504d374e-3.3.1.x86_64 .....
[done]
(2/5) Removing nfs-ganesha-
rgw-2.8.3+git0.d504d374e-3.3.1.x86_64 .....
[done]
(3/5) Removing nfs-ganesha-rados-
urls-2.8.3+git0.d504d374e-3.3.1.x86_64 .....
[done]
(4/5) Removing nfs-ganesha-rados-
grace-2.8.3+git0.d504d374e-3.3.1.x86_64 .....
[done]
```

```
(5/5) Removing nfs-  
ganesha-2.8.3+git0.d504d374e-3.3.1.x86_64 .....  
[done]  
Additional rpm output:  
warning: /etc/ganesha/ganesha.conf saved as /etc/ganesha/ganesha.conf.rpmsave
```

2. Cephダッシュボードから古いクラスタ設定を削除します。

```
cephuser@adm > ceph dashboard reset-ganesha-clusters-rados-pool-namespace
```

### 7.7.3 メタデータサーバのアップグレード

MON、MGR、OSDとは異なり、メタデータサーバをインプレース導入することはできません。その代わり、Cephオーケストレータを使用して、コンテナ内に再展開する必要があります。

1. **ceph fs ls** コマンドを実行して、ファイルシステムの名前を取得します。以下に例を示します。

```
cephuser@adm > ceph fs ls  
name: cephfs, metadata pool: cephfs_metadata, data pools: [cephfs_data ]
```

2. 5.4.3.3項「メタデータサーバの展開」に記載されている、新しいサービス仕様ファイル `mds.yml` を作成します。そのために、ファイルシステムの名前を `service_id` として使用して、MDSデーモンを実行するホストを指定します。以下に例を示します。

```
service_type: mds  
service_id: cephfs  
placement:  
  hosts:  
    - ses-min1  
    - ses-min2  
    - ses-min3
```

3. **ceph orch apply -i mds.yml** コマンドを実行して、サービス仕様を適用し、MDSデーモンを起動します。

### 7.7.4 iSCSI Gatewayのアップグレード

iSCSI Gatewayをアップグレードするには、Cephオーケストレータを使用してコンテナ内に再展開する必要があります。複数のiSCSI Gatewayを使用している場合はサービスのダウンタイムを短縮するために、iSCSI Gatewayを1つずつ再展開する必要があります。

1. 各iSCSI Gatewayノードで実行されている既存のiSCSIデーモンを停止し無効化するには、次のコマンドを実行します。

```
tux > sudo systemctl stop rbd-target-gw
tux > sudo systemctl disable rbd-target-gw
tux > sudo systemctl stop rbd-target-api
tux > sudo systemctl disable rbd-target-api
```

2. 5.4.3.5項「iSCSI Gatewayの展開」に記載されている、iSCSI Gateway用のサービス仕様を作成します。そのためには、既存の/etc/ceph/iscsi-gateway.cfgファイルからpool、trusted\_ip\_list、api\_\*という設定を取得する必要があります。SSLサポートを有効にしている場合(api\_secure = true)、SSL証明書(/etc/ceph/iscsi-gateway.crt)とキー(/etc/ceph/iscsi-gateway.key)も必要です。  
例として、/etc/ceph/iscsi-gateway.cfgが以下の内容を含む場合を考えます。

```
[config]
cluster_client_name = client.igw.ses-min5
pool = iscsi-images
trusted_ip_list = 10.20.179.203,10.20.179.201,10.20.179.205,10.20.179.202
api_port = 5000
api_user = admin
api_password = admin
api_secure = true
```

この場合、次のようなサービス仕様ファイルiscsi.ymlを作成する必要があります。

```
service_type: iscsi
service_id: igw
placement:
  hosts:
    - ses-min5
spec:
  pool: iscsi-images
  trusted_ip_list: "10.20.179.203,10.20.179.201,10.20.179.205,10.20.179.202"
  api_port: 5000
  api_user: admin
  api_password: admin
  api_secure: true
  ssl_cert: |
    -----BEGIN CERTIFICATE-----
    MIIDtTCCAp2gAwIBAgIYMC4xNzc1NDQxNjEzMzc2MjMyXzxxvQ7EcMA0GCSqGSIb3
    DQEBChUAMG0xCzAJBgNVBAYTAlVTMQ0wCwYDVQQIDARVdGFoMRcwFQYDVQQHDA5T
    [...]
    -----END CERTIFICATE-----
  ssl_key: |
    -----BEGIN PRIVATE KEY-----
    MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQC5jdYbjtNTAKW4
```

```
/CwQr/7w0iLGzVxChn3mmCIF3DwbL/qvTFTX2d8bDf6LjGwLYloXHscRfxszX/4h  
[...]  
-----END PRIVATE KEY-----
```



## 注記

`pool`、`trusted_ip_list`、`api_port`、`api_user`、`api_password`、`api_secure`の設定は、`/etc/ceph/iscsi-gateway.cfg`ファイルの内容とまったく同じです。`ssl_cert`と`ssl_key`の値は、既存のSSL証明書とキーファイルからコピーできます。これらの設定が適切にインデントされていることを確認してください。また、`ssl_cert`:行と`ssl_key`:行の末尾に「パイプ文字」(`|`)があることを確認してください(上記の`iscsi.yml`ファイルの内容を参照してください)。

3. `ceph orch apply -i iscsi.yml`コマンドを実行して、サービス仕様を適用し、iSCSI Gatewayデーモンを起動します。
4. 古い `ceph-iscsi` パッケージを既存のiSCSI Gatewayノードからそれぞれ削除します。

```
cephuser@adm > zypper rm -u ceph-iscsi
```

## 7.8 アップグレード後のクリーンアップ

アップグレードの完了後に、以下のクリーンアップ手順を実行してください。

1. 現在のCephバージョンをチェックして、クラスタのアップグレードに成功しているかを確認します。

```
cephuser@adm > ceph versions
```

2. 古いOSDがクラスタに参加していないことを確認します。

```
cephuser@adm > ceph osd require-osd-release octopus
```

3. 自動拡張モジュールを有効化します。

```
cephuser@adm > ceph mgr module enable pg_autoscaler
```

## ！ 重要

SUSE Enterprise Storage 6のプールはpg\_autoscale\_modeがデフォルトでwarnに設定されています。そのため、配置グループ数が最適でない場合に警告メッセージが出ますが、警告のみで自動拡張は行われません。SUSE Enterprise Storage 7のデフォルト設定では、新しいプールのpg\_autoscale\_modeオプションはonに設定されるため、配置グループは自動拡張が実際に行われます。手順に従ってアップグレードを行っても、既存プールのpg\_autoscale\_modeは、自動では変更されません。設定をonに変更して自動拡張を活用したい場合は、『運用と管理ガイド』、第17章「保存データの管理」、17.4.12項「配置グループの自動拡張の有効化」の手順を参照してください。

詳細については、『運用と管理ガイド』、第17章「保存データの管理」、17.4.12項「配置グループの自動拡張の有効化」を参照してください。

### 4. Luminousより前のバージョンのクライアントを拒否します。

```
cephuser@adm > ceph osd set-require-min-compat-client luminous
```

### 5. バランサモジュールを有効化します。

```
cephuser@adm > ceph balancer mode upmap  
cephuser@adm > ceph balancer on
```

詳細については、『運用と管理ガイド』、第29章「Ceph Managerモジュール」、29.1項「バランサ」を参照してください。

### 6. 必要に応じてテレメトリモジュールを有効にします。

```
cephuser@adm > ceph mgr module enable telemetry  
cephuser@adm > ceph telemetry on
```

詳細については、『運用と管理ガイド』、第29章「Ceph Managerモジュール」、29.2項「テレメトリモジュールの有効化」を参照してください。

## A アップストリーム「Octopus」ポイントリリースに基づくCeph保守更新

SUSE Enterprise Storage 7のいくつかの主要パッケージは、CephのOctopusリリースシリーズに基づいています。Cephプロジェクト(<https://github.com/ceph/ceph>)がOctopusシリーズの新しいポイントリリースを公開した場合、SUSE Enterprise Storage 7は、アップストリームの最新のバグ修正や機能のバックポートのメリットを得られるように更新されます。この章には、製品に組み込み済みか、組み込みが予定されている各アップストリームポイントリリースに含まれる重要な変更点についての概要が記載されています。

### Octopus 15.2.5ポイントリリース

Octopusのポイントリリース15.2.5では、以下の修正とその他の変更が行われました。

- CephFS: 新しい拡張ディレクトリ属性である、分散型エフェメラルピンニングとランダムエフェメラルピンニングを使用して、自動静的サブツリーパーティショニングポリシーを自動で設定できるようになりました。詳細については、次のドキュメントを参照してください。 <https://docs.ceph.com/docs/master/cephfs/multimds/>
- Monitorに`mon_osd_warn_num_repaired`設定オプションが追加されました。デフォルトで10に設定されています。いずれかのOSDで、保存データのI/Oエラーを修復した数がこの値を超えた場合、`OSD_T00_MANY_REPAIRS`ヘルス警告が発生します。
- グローバルまたはプールごとに`no_scrub`フラグと`no_deep_scrub`フラグの両方またはいずれかが設定されていない場合、無効化されたタイプのスケジュール済みスクラブは中止されます。ユーザが開始したスクラブはすべて中断されません。
- 正常なクラスタでosdmapが最適化されない問題を修正しました。

### Octopus 15.2.4ポイントリリース

Octopusのポイントリリース15.2.4では、以下の修正とその他の変更が行われました。

- CVE-2020-10753: rgw: s3 CORSConfigurationのExposeHeaderで改行をサニタイズ
- Object Gateway: オーフアンを処理する`radosgw-admin`サブコマンド(`radosgw-admin orphans find`、`radosgw-admin orphans finish`、`radosgw-admin orphans list-jobs`)は非推奨となりました。これらのサブコマンドはあまり保守されておらず、またク

ラストに中間結果を保存するため空き容量の少ないクラスタを満杯にしてしまう可能性があります。そのため、現在、現在実験的と見なされている **rgw-orphan-list** ツールに置き換えられました。

- RBD: RBDのごみ箱を空にするスケジュールを保存するために使用される RBD プールオブジェクトの名前が `rbd_trash_trash_purge_schedule` から `rbd_trash_purge_schedule` に変更されました。ユーザが RBDのごみ箱を空にするスケジュール機能の使用を開始しており、プールごとまたはネームスペースごとにスケジュールを設定している場合は、`rbd_trash_trash_purge_schedule` オブジェクトを `rbd_trash_purge_schedule` にコピーしてからアップグレードして `rbd_trash_purge_schedule` を削除する必要があります。この削除には、ごみ箱を空にするスケジュールが設定されていたすべての RBD プールとネームスペースに対して次のコマンドを使用します。

```
rados -p pool-name [-N namespace] cp rbd_trash_trash_purge_schedule  
rbd_trash_purge_schedule  
rados -p pool-name [-N namespace] rm rbd_trash_trash_purge_schedule
```

もしくは、他の便利な方法を使用して、アップグレード後にスケジュールを復元してください。

## Octopus 15.2.3 ポイントリリース

- Octopus ポイントリリース 15.2.3  
は、`bluefs_preextend_wal_files` と `bluefs_buffered_io` を同時に有効にした場合に WAL の破損が見られる問題に対処するためのホットフィックスリリースでした。15.2.3 での修正はあくまで応急処置です (`bluefs_preextend_wal_files` のデフォルト値を `false` に変更)。恒久的な修正では `bluefs_preextend_wal_files` オプションを完全に削除することになります。この修正は 15.2.6 ポイントリリースに含まれる可能性が高いです。

## Octopus 15.2.2 ポイントリリース

Octopus のポイントリリース 15.2.2 では、1 つのセキュリティ脆弱性にパッチを適用しました。

- CVE-2020-10736: MONとMGRでの認証バイパスを修正

## Octopus 15.2.1ポイントリリース

Octopusのポイントリリース15.2.1では、Luminous (SES5.5)からNautilus (SES6)さらにOctopus (SES7)に素早くアップグレードした際に、OSDがクラッシュする問題を修正しました。また、最初のOctopus (15.2.0)リリースに存在した2つのセキュリティ脆弱性にパッチを適用しました。

- CVE-2020-1759: msgr V2セキュアモードでのナンスの再利用を修正
- CVE-2020-1760: RGW GetObjectヘッダー分割によるXSSを修正

## B マニュアルの更新

この章には、SUSE Enterprise Storageの現行リリースに適用される、マニュアルの更新内容の一覧を記載しています。

- Cephダッシュボードの章(『運用と管理ガイド』)を1つ上のレベルに移し、目次からトピックの詳細を直接検索できるようにしました。
- 『運用と管理ガイド』の構成を見直し、現行のガイドの範囲と一致させました。一部の章は、他のガイドに移動させました(jsc#SES-1397)。

# 用語集

## 一般

### Alertmanager

Prometheusサーバによって送信されるアラートを処理し、エンドユーザーに通知する単一のバイナリ。

### Ceph Manager

Ceph Manager (MGR)は、Cephの管理ソフトウェアです。クラスタ全体からすべての状態を一か所に収集します。

### Ceph Monitor

Ceph Monitor (MON)は、Cephのモニターソフトウェアです。

### Ceph Object Storage

オブジェクトストレージの「製品」、サービス、またはケーパビリティです。Ceph Storage ClusterとCeph Object Gatewayから構成されます。

### Ceph OSDデーモン

`ceph-osd`デーモンは、ローカルファイルシステムにオブジェクトを保存し、ネットワーク経由でそれらにアクセスできるようにするCephのコンポーネントです。

### Ceph Storage Cluster

ユーザのデータを保存するストレージソフトウェアのコアセット。1つのセットは複数のCeph Monitorと複数のOSDで構成されます。

### `ceph-salt`

Saltを使用して、`cephadm`に管理されるCephクラスタを展開するツールを提供します。

### `cephadm`

`cephadm`はCephクラスタを展開、管理します。その手段として、SSHを使用してマネージャデーモンからホストに接続し、Cephデーモンコンテナを追加、削除、更新します。

### CephFS

Cephのファイルシステム。

## CephX

Cephの認証プロトコル。CephXはKerberosのように動作しますが、単一障害点がありません。

## Cephクライアント

Ceph Storage Clusterにアクセスできる、Cephコンポーネントのコレクション。たとえば、Object Gateway、Ceph Block Device、CephFS、およびこれらに関連するライブラリ、カーネルモジュール、FUSEクライアントなどがあります。

## Cephダッシュボード

WebベースのCeph管理/監視用ビルトインアプリケーションで、クラスタの様々な側面とオブジェクトを管理します。このダッシュボードはCeph Managerモジュールとして実装されます。

## CRUSH、CRUSHマップ

「Controlled Replication Under Scalable Hashing」：データの保存場所を計算することによって、データの保存と取得の方法を決定するアルゴリズム。CRUSHは、クラスタ全体に均等に分散したデータを擬似ランダムにOSDで保存および取得するために、クラスタのマップを必要とします。

## CRUSHルール

特定のプール(単数または複数)に適用される、CRUSHのデータ配置ルール。

## DriveGroups

DriveGroupsは、物理ドライブにマッピングできる1つ以上のOSDレイアウトの宣言です。OSDレイアウトはCephが指定された基準を満たすようにOSDストレージをメディア上に物理的に割り当てる方法を定義します。

## Grafana

データベース分析および監視ソリューション。

## Metadata Server

メタデータサーバ(MDS)は、Cephのメタデータソフトウェアです。

## Multi-zone

## Object Gateway

Ceph Object Store用のS3/Swiftゲートウェイコンポーネント。RADOS Gateway (RGW)とも呼ばれます。

## OSD

「Object Storage Device」：物理ストレージユニットまたは論理ストレージユニット。

## **OSDノード**

データの保存、データレプリケーションの処理、回復、バックフィル、およびリバランスを実行し、他のCeph OSDデーモンを確認することによってCeph Monitorにモニタリング情報を提供するクラスタノード。

## **PG**

配置グループ: 「プール」を細分化したもので、パフォーマンスを調整するために使用します。

## **Prometheus**

システム監視およびアラートツールキット。

## **RADOS Block Device (RBD)**

Cephのブロックストレージコンポーネント。Cephブロックデバイスとも呼ばれます。

## **Reliable Autonomic Distributed Object Store (RADOS)**

ユーザのデータを保存するストレージソフトウェアのコアセット(MON+OSD)。

## **Samba**

Windowsとの統合ソフトウェア。

## **Sambaゲートウェイ**

SambaゲートウェイはWindowsドメインのActive Directoryに参加し、ユーザの認証と権限の付与を行います。

## **アーカイブ同期モジュール**

S3オブジェクトのバージョン履歴を保持するためのObject Gatewayゾーンを作成できるモジュール。

## **ゾーングループ**

## **ノード**

Cephクラスタ内の1つのマシンまたはサーバ。

## **バケット**

他のノードを物理的な場所の階層に集約するポイント。

## **プール**

ディスクイメージなどのオブジェクトを保存するための論理パーティション。

## **ポイントリリース**

バグ修正やセキュリティ上の修正だけを含む、応急措置的なリリース。

### **ルーティングツリー**

受信者が実行できるさまざまなルートを示す図に付けられる用語。

### **ルールセット**

プールのデータ配置を決定するためのルール。

### **管理ノード**

このホストからCeph関連のコマンドを実行して、クラスタのホストを管理します。