Red Hat OpenStack Platform 16.2

Red Hat OpenStack Platform に Red Hat OpenShift Container Platform をデプロイするためのリファレンスアーキテクチャー

検証済みプライベートクラウドソリューションのガイドライン

Last Updated: 2024-04-04

Red Hat OpenStack Platform 16.2 Red Hat OpenStack Platform に Red Hat OpenShift Container Platform をデプロイするためのリファレンスアーキテクチャー

検証済みプライベートクラウドソリューションのガイドライン

August Simonelli
asimonel@redhat.com

法律上の通知

概要

このドキュメントの目的は、Red Hat OpenShift Container Platform 4.12 を Red Hat OpenStack Platform 16.2 にデプロイする際のガイドラインと考慮事項を提供することです。

目次

多様性を受け入れるオープンソースの強化

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

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

Jira でドキュメントのフィードバックを提供する

ドキュメントに関するフィードバックを提供するには、Create Issue フォームを使用します。Red Hat OpenStack Platform Jira プロジェクトで Jira Issue が作成され、フィードバックの進行状況を追跡できます。

1. Jira にログインしていることを確認してください。Jira アカウントをお持ちでない場合は、アカウントを作成してフィードバックを送信してください。
2. Create Issue をクリックして、**Create Issue** ページを開きます。
3. **Summary** フィールドと **Description** フィールドに入力します。**Description** フィールドに、ドキュメントの URL、章またはセクション番号、および問題の詳しい説明を入力します。フォーム内の他のフィールドは変更しないでください。
4. **Create** をクリックします。

第1章 このドキュメントの目的

このドキュメントでは、Red Hat OpenStack Platform 16.2 で Red Hat OpenShift Container Platform 4.12 を運用するための検証済みのプライベートクラウドソリューションを提案します。

Red Hat OpenShift Container Platform、Red Hat OpenStack Platform、および Red Hat Ceph Storage は、当社のソリューションの主要なアーキテクチャーコンポーネントです。

このドキュメントは実装ガイドではなく、製品ドキュメントの補足です。関連ドキュメントのリンク セクションに、3 つのアーキテクチャーコンポーネントの各製品ドキュメントが記載されています。いずれも十分にテスト済みで、サポート対象の完全なドキュメントです。

1.1. RED HAT のテスト済みソリューション

このドキュメントでは、OpenShift on OpenStack ソリューションの推奨プラクティスを紹介します。各コンポーネントを分析し、オプションを説明します。次に、実装のために、その選択を重点的に説明します。

推奨事項はすべて、Red Hat の OpenShift on OpenStack Quality Engineering (QE) チームによってラボ環境でテスト済みです。

このドキュメントは、実際に機能するテスト済みソリューションを反映しています。

特定の実装手順と手順については、Red Hat 製品のドキュメントに、デプロイメントを成功させるために必要な詳細が記載されています。このソリューションドキュメントに基づく評価とテストによる保証と組み合わせることで、OpenShift on OpenStack 実装の基盤を形成する包括的で実用的なソリューションを実現できます。

1.1.1. スケーリングとパフォーマンスのテスト

実際のシナリオに照らしてソリューションを評価するよう、最大限の努力を尽くしましたが、あらゆる可能性を考慮してテストすることはできません。このソリューションはデプロイメントのベースラインとしてご参考ください。特定のワークロードと使用パターンに合わせて、包括的なパフォーマンステストを実行することを推奨します。

コントロールプレーンノードのサイズは、予想されるクラスターサイズと増加に応じて慎重に決定してください。ストレージソリューションはそれに応じて実装および調整する必要があります。

このドキュメントは、Red Hat のグローバルエンジニアリングチームおよびサービスチームの最適なソリューションを組み合わせて、可能な限り、実際の状況下でのカスタマーエクスペリエンスを反映して更新されています。

詳細は、Recommended infrastructure practices を参照するか、Red Hat アカウントチームにお問い合わせください。

1.2. OPENSHIFT ON OPENSTACK によるクラウドネイティブ導入の簡略化

OpenShift on OpenStack を実装するための Red Hat のテスト済みソリューションは、クラウドネイティブの導入を簡略化するうえで重要な要素です。Red Hat は、信頼性と相互運用性を確保するために、OpenShift と OpenStack の両方でソリューションを設計し、十分にテストしました。両方のプラットフォームをサポートし、それらを確実に連携させることで、モダナイゼーションの導入からその先までお客様の組織を支える、実稼働環境レベルの柔軟な基盤を提供します。OpenShift on OpenStack を運用すると、以下を容易に行うことができます。

- **段階的にワークロードを移行する。** Red Hat OpenStack Platform 上で Red Hat OpenShift を実

行すると、サポートされている統合された同じ基盤上で仮想化アプリケーションとコンテナーアプリケーションを並行して実行できる柔軟性が得られ、移行および変換プロジェクトが容易になります。

- **クラウドネイティブスキルを開発し、DevOps プラクティスを加速する。**コンテナーと仮想マシンを一貫したソリューションに統合することで、IT 運用スタッフと開発者が 1 つのプラットフォームで作業できます。また、チームがクラウドネイティブのプロセスと方法論を共同で拡大、開発できるように支援します。

- **両方のプラットフォームの確かな専門知識とサポートにアクセスする。**Red Hat は両方のプラットフォームをサポートしているため、1 つの問い合わせ窓口で容易かつ効率的に問題を解決できます。さらに、Red Hat は既存アプリケーションのモダナイゼーションの効率化や新しいクラウドネイティブアプリケーションの構築に必要なプラクティス、ツール、文化の開発を支援するサービスとトレーニングを提供します。

第2章 テクノロジー概要

このドキュメントでは、Red Hat OpenStack Platform 16 で Red Hat OpenShift Container Platform 4.12 を運用するためのソリューションを提案します。このソリューションでは、Red Hat OpenShift Container Platform 4.12 を Red Hat OpenStack Platform 16.2 を実行する物理サーバーにデプロイします。Red Hat OpenStack Platform の Director を使用して、OpenStack の初期インストールと Day 2 運用を実行します。

2.1. OPENSHIFT と OPENSTACK の関係

OpenStack と OpenShift の関係は補完的です。

- OpenStack はアプリケーションプログラミングインターフェイス (API) を通じてリソースを公開します。OpenShift はそのリソースを要求します。
- OpenStack は、コンピュート、ストレージ、ネットワークインフラストラクチャーに加えて、セルフサービスロードバランサーや暗号化などの追加リソースを OpenShift に提供します。
- OpenShift は、OpenStack によってプロビジョニングされたインフラストラクチャー上でコンテナー化されたアプリケーションを実行します。

製品は緊密に統合されています。OpenShift は、ユーザーの介入なしに、オンデマンドで OpenStack リソースを消費できます。

2.1.1. Red Hat Enterprise Linux CoreOS (RHCOS)

OpenShift 4 以降、OpenShift ノードは Red Hat Enterprise Linux (RHEL) CoreOS (RHCOS) で実行されるようになりました。RHEL CoreOS は、Container Linux (旧称: CoreOS) からの簡単な無線更新と Red Hat Enterprise Linux カーネルを組み合わせて、より安全で管理しやすいコンテナーホストを提供します。

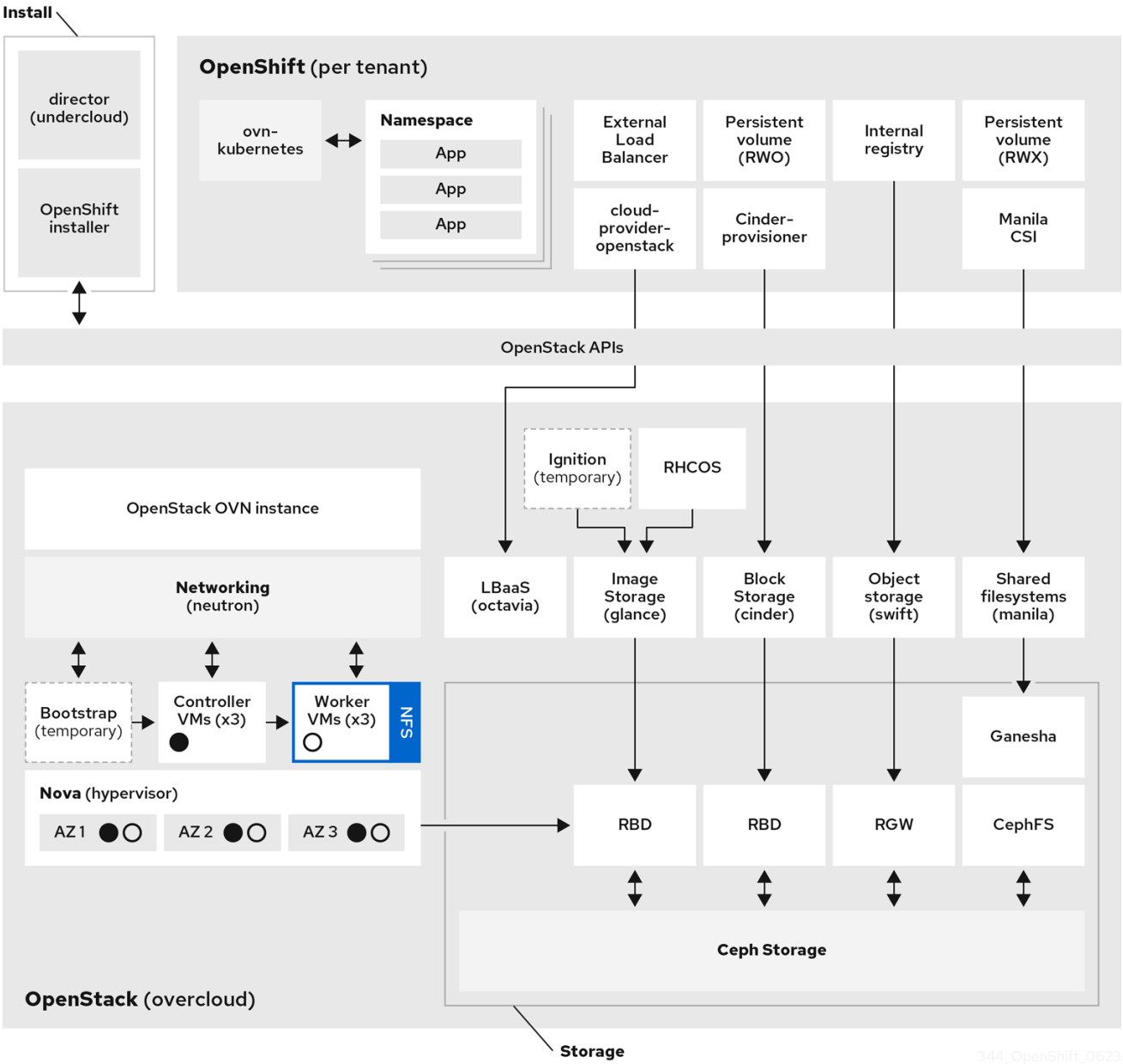
インストーラープロビジョニングインフラストラクチャーベースのデプロイメントでは、RHCOS がすべての OpenShift Container Platform ノードでサポートされるオペレーティングシステムであり、ワーカーおよびコントローラーにデフォルトで使用されます。コントローラーノードが RHCOS を実行することも、OpenShift の要件です。現在、RHCOS は OpenShift でのみ使用されており、独立したオペレーティングシステムとして使用するためには提供されていません。

詳細は、Red Hat Enterprise Linux (RHEL) CoreOS を参照してください。

2.2. ソリューション概要

OpenShift on OpenStack を配置するために利用できるオプションは多数ありますが、明確性、シンプルさ、およびサポート性を確保するために、検証済みのソリューションを 1 つ提供します。Red Hat のテスト済みソリューションは、このソリューションのコンポーネントと統合を表しており、QE によってテストされており、すべてのエンタープライズデプロイメントの出発点です。

図2.1 Red Hat ソリューションの図



Red Hat ソリューションの図 に示すインストールとセットアップを実施する際に、次の重要な選択を行いました。

インストール

- OpenStack は、Director を使用してインストールします。
- OpenStack は、外部 TLS 暗号化を使用してインストールします。
- OpenShift は、インストーラープロビジョニングインフラストラクチャー (IPI) を使用してインストールします。
- OpenShift は、権限のない OpenStack テナントを使用して、Director ホストからインストールします。

ストレージ

- OpenStack は、RWX コンテナーワークロードで使用できる Fileshare-as-a-Service (manila) をデプロイします。
- OpenStack は、RWO コンテナーワークロードで使用できる Block Storage サービス (Cinder) をデプロイします。
- OpenStack は、コンピュート (Nova) 一時ストレージにローカルストレージを使用します。
- OpenStack は、イメージ (Glance)、ブロックストレージ (Cinder)、オブジェクト (Swift)、およびオプションのコンピュート (Nova) に Red Hat Ceph Storage (RHCS) を使用します。
- OpenStack は、Fileshare-as-a-Service (manila) に Ganesha で RHCS を使用します。
- OpenShift は、Container Storage Interface (CSI) ドライバーを使用して、manila へのアクセスを提供します。
- OpenShift は、内部レジストリーにオブジェクトストレージを使用します。

Compute

- OpenShift のコントロールプレーン仮想マシンとワーカー仮想マシンは、Nova アベイラビリティーゾーンを使用してデプロイされ、高可用性を提供します。

ネットワーク

- OpenStack は、その SDN に Open Virtual Network (OVN) を使用します。
- OpenShift のネットワークは、OVN-Kubernetes で管理します。
- OpenStack は、OpenShift の負荷分散のために Load-Balancing-as-a-Service (Octavia) をデプロイします。
- OpenShift は、Octavia 用の Amphora ドライバーを使用して負荷分散を提供します。

第3章 設計上の考慮事項

このセクションでは、統合ソリューションの設計上の考慮事項に、Red Hat のソリューションがどのように対処しているかを説明します。OpenShift on OpenStack 実装のアーキテクチャー計画では、各セクションを検討する必要があります。各セクションでは、主要な統合、それらのさまざまなオプション、利点、懸念事項、および全体的な要件について説明します。次に、ソリューションで選択した設計上の考慮事項を説明して、各セクションを締めくくります。

このソリューションの推奨事項は、Red Hat 品質エンジニアリング、フィールドコンサルタント、エンジニア、および製品チームと協力して作成されました。

3.1. サポートライフサイクル

Red Hat OpenShift Container Platform 4.12 のリリースは、OpenShift の延長更新サポート (EUS) リリースです。お客様は、OpenStack の長期サポートオプションを OpenShift の EUS と組み合わせることができるようになりました。この統合ソリューションは、スタック全体でエンタープライズに不可欠なサポートタイムフレームを提供します。

3.1.1. Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform Red Hat の 4.12 リリースは、延長更新サポート (EUS) リリースとして指定されています。OpenShift 4.12 以降、EUS リリースでは、マイナーバージョンに対して 24 カ月のサポート期間がお客様に提供されます。

EUS リリースのサポートは、次の 3 つの異なるフェーズで提供されます。

- フルサポート (最初の 6 カ月間)
- メンテナンスサポート (その後の 12 カ月間)
- 延長更新サポート (さらにその後の 6 カ月間)

今後、Red Hat はすべての偶数番号のマイナーリリース (4.12、4.14、4.16) を、24 カ月のサポート期間を提供する延長更新サポートリリースと表記します。ライフサイクル更新や対象コンポーネントのリストなど、Red Hat OpenShift Container Platform の EUS の詳細は、Red Hat OpenShift Container Platform のライフサイクルポリシー を参照してください。

3.1.2. Red Hat OpenStack Platform

Red Hat OpenStack Platform の 16.2 リリースでは、2 年半の製品サポートと、1 年間の延長ライフサイクルサポート (ELS) を購入するオプションが提供されます。サポートは、次の 3 つの異なるフェーズで提供されます。

- フルサポート (初回リリース後、最初の 18 カ月間)
- メンテナンスサポート (その後の 12 カ月間)
- 延長ライフサポート (さらにその後の 1 年間。追加のサブスクリプションコストが発生)

Red Hat OpenStack Platform のライフサイクルとさまざまなサポートタイプの詳細は、Red Hat OpenStack Platform のライフサイクル を参照してください。

3.1.3. Red Hat のテスト済みソリューション: サポートライフサイクル

このソリューションでは、Red Hat OpenStack Platform 16.2 上で Red Hat OpenShift Platform 4.12 を使用します。いずれのプラットフォームにも長期サポートソリューションがあります。

3.2. 接続性

OpenShift は、さまざまなオンプレミスアーキテクチャーにデプロイできます。OpenShift は、接続済みインストールで、プロキシー接続を介して、制限された、またはエアギャップされたネットワーク環境で実行できます。

詳細は、Supported installation methods for different platforms を参照してください。

3.2.1. 接続インストール

OpenStack と OpenShift のインストーラーはどちらも、インターネットへの直接アクセスと Red Hat 製品の有効なサブスクリプションを利用できます。インストールプロセスは自己完結型ではなく、次のコアアセットを取得するには、外部ソースへのアクセスと DNS 解決が必要です。

- OpenStack サービス用のコンテナー
- OpenShift のコントロールおよびワーカーノードコンテナー
- OpenShift の RHCOS イメージ

3.2.2. プロキシーによるインストール

OpenShift デプロイメントは、HTTP または HTTPS クラスター全体のプロキシーの実装をサポートします。クラスター全体のプロキシー設定の詳細は、ドキュメントの Configuring the cluster-wide proxy セクションを参照してください。

3.2.3. 制限付きネットワークでのデプロイ (エアギャップ)

Red Hat OpenStack Platform 16.2 上の Red Hat OpenShift Platform 4 は、制限付きネットワークでのデプロイを完全にサポートします。制限付きネットワークでデプロイを実行する前に、インストールのブートストラップに必要な基本コンポーネントを準備する必要があります。

制限付きネットワークでのインストールの詳細は、ドキュメントの Installing a cluster on OpenStack in a restricted network セクションを参照してください。

3.2.4. Red Hat のテスト済みソリューション: 接続インストール

このソリューションでは、インストールに接続インストール方式を使用します。Director とすべての OpenShift ノードの完全な DNS 解決を含め、インターネットへの直接接続を使用しました。

プロキシーおよび制限付き (エアギャップ) 環境でのデプロイは、Red Hat QE によってテストされており、完全にサポートされています。ただし、Red Hat は、接続されたインフラストラクチャーと同じテストをすべて実行したわけではありません。詳細は、Red Hat 担当者にお問い合わせください。デプロイメントへの最も簡単なパスを確保し、ラボの要件を満たすために、このソリューションでは、それらを使用しないことを選択しました。

3.3. インストール方法とツール

最良の結果を得るには、Red Hat が推奨およびサポートするツールを使用して、Red Hat 製品および製品統合をインストールします。

3.3.1. Red Hat OpenStack Platform director

Red Hat OpenStack Platform の Director は、OpenStack TripleO プロジェクトに基づく管理およびインストールツールです。TripleO は OpenStack On OpenStack の略です。 このソリューションは、Director を使用して、Red Hat OpenStack Platform のライフサイクルをインストールおよび管理します。

Red Hat OpenStack Platform の Director の背後にある基本的な概念は、**アンダークラウド** と **オーバークラウド** の 2 つのクラウドがあるということです。アンダークラウドは、別のクラウドを管理することのみを目的とするスタンドアロンの OpenStack デプロイメントです。1 つの物理サーバーまたは仮想マシンにデプロイできます。管理者は、アンダークラウドの OpenStack サービスを使用して、実稼働の OpenStack クラウドを定義およびデプロイします。Director は、ソフトウェア更新の適用や OpenStack バージョン間のアップグレードなど、Day 2 管理運用にも使用されます。

オーバークラウド と呼ばれる第 2 のクラウドは、アンダークラウドによってデプロイされるフル機能の実稼働環境です。オーバークラウドは、さまざまなロールを持つ物理サーバーで設定されています。

1. **コントローラーノード**は、OpenStack API エンドポイントを実行します。また、OpenStack のステートフル設定データベースとメッセージングキューも保存します。

2. **コンピュートノード** は仮想マシンハイパーバイザーを実行します。ユーザーワークロードに割り当てられたコンピュートリソースをホストします。

3. **ストレージノード** は、ユーザーワークロード用のブロック、オブジェクト、またはソフトウェア定義のストレージを提供します。

OpenShift Container Platform は、オーバークラウド上のプロジェクトまたはテナント内で実行されます。各テナントの OpenShift クラスターは相互に分離されています。

Director は、すべての実稼働の OpenStack Platform デプロイメントに必要です。Director のデプロイメントツールに組み込まれた設定は、Red Hat エンジニアリングチームによってテストおよび検証されたものです。

3.3.2. Red Hat OpenShift Container Platform 4 のインストールパターン

OpenShift 4 インストーラーは、Red Hat OpenShift Container Platform インストールの 3 つの基本タイプを通じて柔軟性を提供します。

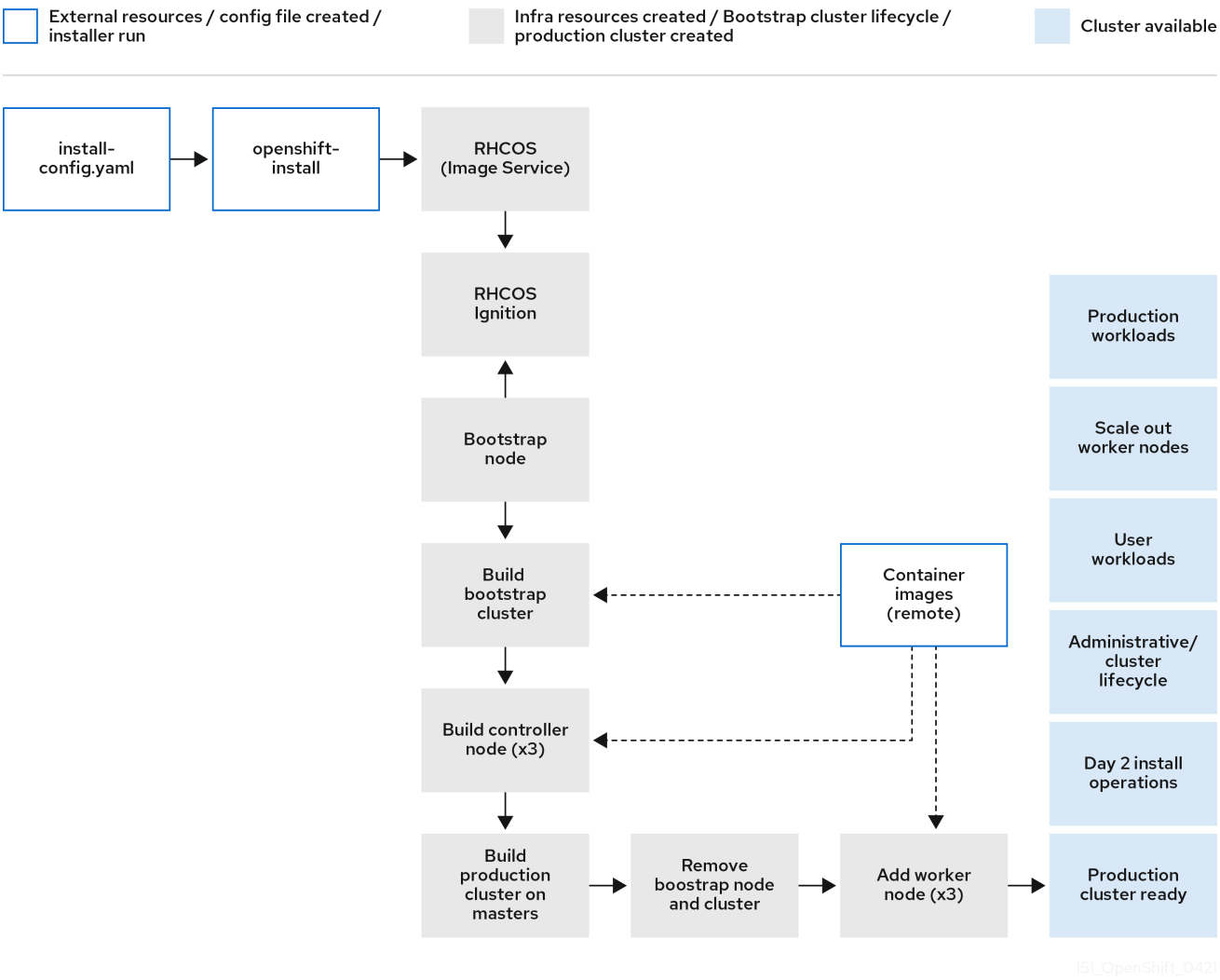
- **インストーラープロビジョニングインフラストラクチャー** (IPI) またはフルスタック自動化
- **ユーザープロビジョニングインフラストラクチャー** (UPI) または既存のインフラストラクチャー
- 以前はベアメタルインストールと呼ばれていた **プラットフォーム非依存**。

3.3.2.1. インストーラープロビジョニングインフラストラクチャー

インストーラープロビジョニングインフラストラクチャー (IPI) では、インストーラーがインストールのあらゆる側面を管理します。これには、OpenShift の独自のベストプラクティスのデプロイを伴うインフラストラクチャーのプロビジョニングも含まれます。インストーラーは、そのインフラストラクチャーの知識を持つインフラストラクチャー対応プロバイダーを介して、基盤となるインフラストラクチャーを直接要求、デプロイ、および管理できます。IPI を使用すると、最小限の設定で実稼働環境に対応したインストールを実現し、マネージドサービスにより自己管理型クラスターをデプロイできます。

IPI では、既存のネットワークインフラストラクチャーを使用することもできます。

図3.1 OpenShift IPI のワークフロー



3.3.2.1.1. 設定ファイルを生成するためのガイド付きインストール

IPI インストールは、インストール設定ファイルを作成するためのシンプルなガイド付きワークフローです。ガイド付きインストールでは、OpenShift のインストールに必要な値の大部分をユーザーに要求します。これにより、インストールの最終状態を記述する単純な YAML ファイルが作成されます。また、YAML ファイルを使用すると、Operator によって IPI フットプリント内のインストールにさらなるカスタマイズを追加できます。

3.3.2.2. ユーザープロビジョニングインフラストラクチャー

ユーザープロビジョニングインフラストラクチャー (UPI) インストール方式 では、管理者は基盤となる独自のインフラストラクチャーをインストール前に作成および管理する必要があります。インストーラーはインフラストラクチャー対応プロバイダーを使用しますが、管理者は事前にコンポーネントを準備する必要があります。その後、OpenShift インストーラーはインフラストラクチャープロバイダーを使用して、準備されたインフラストラクチャーとやりとりできます。OpenShift on OpenStack の場合、Red Hat は、この目的のためにサンプルの Ansible スクリプトを提供しています。

3.3.2.3. プラットフォーム非依存 (旧称: ベアメタルインストール)

プラットフォーム非依存のインストール は、インフラストラクチャープロバイダーを使用しないため、どのような基盤インフラストラクチャーでも使用できます。インストーラーは、基盤となるインフラストラクチャーの操作を制御することはできません。つまり、管理者はインフラストラクチャーを準備

し、独自のカスタムメソッドを使用して、インストールを調整する必要があります。このタイプのインストールは最も柔軟ですが、クラウド統合が行われないため、追加のカスタム自動化が必要です。

インストーラープロビジョニングインフラストラクチャー方式、ユーザープロビジョニングインフラストラクチャー方式、およびプラットフォーム非依存のインストール方式は、Red Hat によって完全にサポートされています。

インストールパターンとプラットフォームの包括的な概要については、公式ドキュメントの クラスターインストール方法の選択とユーザー向けの準備 を参照してください。

3.3.3. Red Hat のテスト済みソリューション: インストーラープロビジョニングインフラストラクチャー (IPI)

このソリューションでは、IPI を使用して Red Hat OpenShift Platform を Red Hat OpenStack Platform にインストールします。

- テストと柔軟性のために、OpenShift インストール設定ファイルを手動で作成します。
- このファイルを使用して、インフラストラクチャー対応の OpenShift プロバイダーを利用しながら、ネットワーク、マシン、およびオペレーティングシステムのすべての必要な部分を OpenStack API を介してプログラムで作成するように、インストーラーに指示します。
- これにより、可用性が高く、完全にテストされ、実稼働環境に適した完全にサポートされたソリューションであるアーキテクチャーが実現します。

注記

フルスタック自動化とインストーラープロビジョニングインフラストラクチャー (IPI) は、同義の用語です。このドキュメントでは主に、**インストーラープロビジョニングインフラストラクチャー** (IPI) という用語を使用します。

OpenShift の IPI 方式は、ベストプラクティスに基づくインストールを実行する、非常に規範的なインストールパターンです。このインストール方法により、手作業による差異を最小限に抑えることができます。IPI デプロイメントの場合は、原則として、インフラストラクチャーの変更をデプロイ後に手動でカスタマイズしないでください。すべての変更は、基盤インフラストラクチャーおよび API との直接的なやりとりを通じて、インストーラーによって実装する必要があります。マシンのスケールアウトなどの Day 2 運用は可能であり、サポートされていますが、インストールの低レベルの結果は完全にインストーラー主導である必要があります。

3.4. 高可用性

高可用性は、すべての実稼働環境の要件です。Red Hat のソリューションは、OpenStack、OpenShift、および Ceph レイヤーで高い可用性を備えています。

3.4.1. OpenStack HA

Red Hat OpenStack Platform は、以下のアクションによって OpenStack コントロールプレーンの高可用性を保証します。

- OpenStack Platform の Director は、3 つのコントローラーすべてで OpenStack サービスと API の複数のインスタンスを同時に実行する 3 つのコントローラーノードをデプロイします。
- HAproxy は、コントローラー API エンドポイント間で接続を負荷分散して、サービスの可用性を確保します。

- Galera クラスターは、OpenStack 状態データベースを保護します。
- メッセージバスを保護するために、すべてのノードで RabbitMQ キューが複製されます。

OpenStack の詳細は、高可用性のデプロイメントと使用 ガイドを参照してください。

3.4.2. OpenShift HA

OpenShift IPI インストーラーは、デフォルトで 3 つのコントロールプレーンノードをデプロイすることにより、OpenShift コントロールプレーンの高可用性を確保します。これにより、etcd 状態データベースが規定の最小 HA 要件を満たし、少なくとも 3 つの別々のノードに配置されます。

3.4.2.1. Node HA

OpenShift on OpenStack をデプロイする場合は、OpenShift ノードの配置を慎重に検討する必要があります。

- コントロールプレーンの HA を確保するには、コントロールプレーンノードを、基礎となる OpenStack コンピュートノードに別々に配置します。
- コンピュートインフラストラクチャー全体でワーカーノードを均等に配置します。

3.4.2.2. コントロールプレーンノードの配置

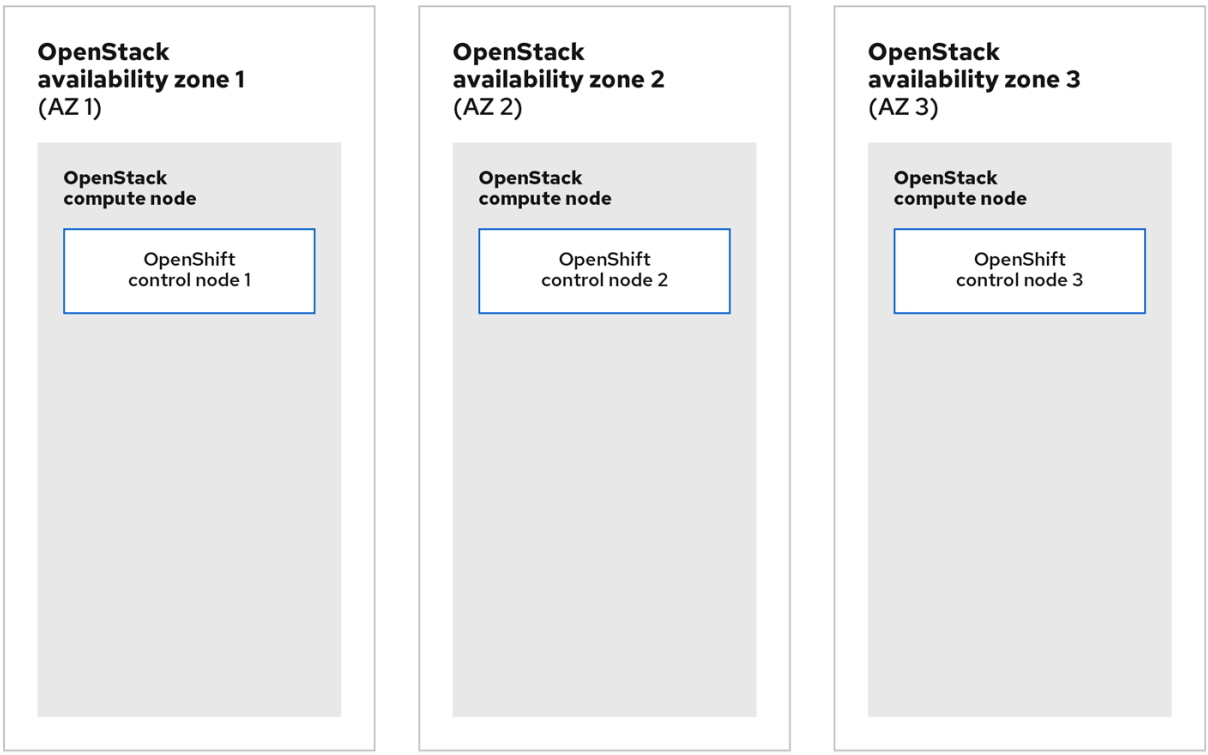
Red Hat は、OpenShift on OpenStack コントロールプレーン仮想マシンの高可用性を管理するためのサポートされている方法をいくつか提供しています。

- Nova アベイラビリティーゾーン
- OpenStack 非アフィニティーポリシー

3.4.2.3. Nova アベイラビリティーゾーン

OpenStack アベイラビリティーゾーンを使用することは、コントロールプレーンノードが HA に対して適切に管理されるようにするための推奨される方法です。

図3.2 Nova アベイラビリティーゾーンを使用してコンピュートホストに OpenShift コントロールプレーンノードを配置する



OpenStack では、アベイラビリティーゾーンの概念により、クラウドを論理的に分割し、クラウドテナントが簡単に使用できる物理インフラストラクチャーのシンプルな抽象化を提供できます。アベイラビリティーゾーンは、OpenStack では非常に細かく設定されています。Nova、Cinder、Neutron などの一部の OpenStack サービスには、独自のアベイラビリティーゾーンの実装があります。

HA OpenShift コントロールプレーンの場合、OpenStack クラウド管理者は、3 つ以上のアベイラビリティーゾーンを提供し、提供されるコンピュートリソースが適切に設定されていることを確認する必要があります。

OpenShift 4.12 では、コントロールプレーンノードが同じ L2 ネットワークに属している必要があります。コントロールプレーンノードを 3 つのアベイラビリティーゾーンに分散する場合、通常はそれらをストレッチ L2 プロバイダーネットワークに接続します。通常、VLAN または VXLAN でセグメント化された小規模な専用 L2 ネットワークを使用するか、このネットワークを相互接続できる障害ドメインファブリックを備えた Geneve プロトコルを使用して、インフラストラクチャーを設定する必要があります。ゾーン間の遅延は 100 ミリ秒未満である必要があります。

コンピュートアベイラビリティーゾーンを使用する場合は、同じ名前の対応するボリュームアベイラビリティーゾーンが必要です。これは、OpenShift 4.12 で有効な CSI トポロジー認識を使用する場合の要件です。

IPI インストールの場合、インストール前にコントロールプレーンのアベイラビリティーゾーンを **install-config.yaml** ファイルに手動で追加する必要があります。ガイド付きインストールでは、ゾーンの入力を求めるプロンプトが表示されないためです。ゾーンは、設定ファイルに YAML 配列として追加します。

OpenShift インストール設定ファイルでの Nova アベイラビリティーゾーンの設定。

```
controlPlane:
  name: master
```

```
platform:
  openstack:
    zones: ['AZ0', 'AZ1', 'AZ2']
replicas: 3
```

インストール時、OpenShift インストーラーは仮想マシンの作成時に各コントロールプレーンノードを異なる AZ に割り当てます。

3.4.2.4. OpenStack 非アフィニティーポリシー

OpenStack は、アフィニティーポリシーを通じてインスタンスを決定論的に配置する機能をサポートしています。アフィニティールールは、管理者が配置を制御できるように、詳細なポリシーにグループ化されます。これらのポリシーと OpenStack サーバーグループを使用することにより、Nova アベイラビリティーゾーンのないデプロイメントでも HA を維持できます。

OpenShift on OpenStack インストーラーは、コントロールプレーンとコンピュートノードの両方のサーバーグループを作成します。これらのサーバーグループは、インスタンスを別々の OpenStack コンピュートホストに配置することを要求するソフト非アフィニティーポリシーとともに使用されます。

3.4.2.5. ワーカーノードの配置

ワーカーノードのアベイラビリティーゾーンを使用するには、次の 2 つの方法があります。

- インストールファイルによるインストール時
- Day 2 運用中、**availabilityZone** の providerSpec 値を使用して手動で作成した machineSet を使用する。

ワーカーノードの場合、インストーラーには次の配置ルールがあります。

- インストーラーは、指定されたアベイラビリティーゾーンごとに MachineSet を作成します。
- インストーラーはノードを配置し、レプリカをゾーン間で可能なかぎり均等に分散します。
- インストーラーは、ワーカーノードを別のハイパーバイザーに配置するように要求するために、非アフィニティーを持つサーバーグループを作成します。

次の図は、ノードをアベイラビリティーゾーンに設定する 1 つの方法を示しています。

図3.3 3 つの Nova アベイラビリティーゾーンにデプロイされた OpenShift コントロールプレーンとワーカーノードの例。

ワーカーノードはコントロールプレーンノードとは異なるタイプのワークロードを実行するため、Ceph を基盤とするルートボリュームを利用できます。

これを行うには、OpenShift の **install-config.yaml** インストールファイルのコンピュートセクションで、ワーカーの **rootVolume** を明示的に設定します。これにより、コントロールプレーンノードのように一時ディスクではなく、Ceph を使用するようにインストーラーに指示します。

さらに、ボリュームゾーン名がコンピュートゾーン名と一致する必要があります。

OpenShift インストール設定ファイルでワーカーのマシンプールを設定します。

```
compute:
- name: worker
  platform:
    openstack:
      zones: ['AZ0', 'AZ1', 'AZ2']
      rootVolume:
        size: 100
        zones: ['AZ0', 'AZ1', 'AZ2']
  replicas: 3
```

3.4.2.6. ストレージレイヤーの HA

デフォルトでは、Red Hat Ceph Storage (RHCS) デプロイメントは高可用性です。このソリューションでは、Director を使用して、RHCS をデプロイします。この方法で RHCS をデプロイすると、コンピュート、イメージ、ブロックストレージ、およびオブジェクトストレージサービスに HA ストレージを自動的に統合して提供できます。

3.4.2.6.1. OpenShift コントロールプレーンインスタンス

OpenShift コントロールプレーンノードでは、Ceph を使用して OpenStack Nova サービスを設定することは推奨されません。代わりに、OpenShift コントロールプレーンインスタンスでは、etcd に最高のパフォーマンスを提供するために、ローカルストレージを Nova 一時ドライブに使用することを推奨します。

さらに、etcd はコントロールプレーンノード間でデータを複製するため、Ceph が提供する別の冗長性レイヤーは必要ありません。

3.4.3. Red Hat のテスト済みソリューション: 高可用性

3.4.3.1. OpenStack

OpenStack 向けのソリューションに高可用性を実装するには、複数のコンポーネントを使用します。

- Red Hat OpenStack Platform Director は 3 つの OpenStack コントローラーをデプロイします。
- ストレージについては、ハイパーコンバージドインフラストラクチャー (HCI) デプロイメントパターン を使用し、コンピュートサービスとストレージサービスを 3 つのハイパーコンバージドノードに配置しています。
- OpenStack サービス (イメージ、ブロックストレージ、オブジェクトストレージなど) は、Ceph にデプロイされ、ストレージレイヤーの耐障害性を提供します。
- OpenStack クラウドは、各ゾーンに 1 つ以上のコンピューティングホストを持つ 3 つのコンピュートアベイラビリティーゾーンでデプロイされます。
- OpenStack クラウドは 3 つのボリュームアベイラビリティーゾーンでデプロイされます。このゾーンの名前はコンピュートアベイラビリティーゾーンと同じです。
- ローカルの一時ストレージは OpenShift コントロールプレーンインスタンスに使用されます。HA は etcd のネイティブの耐障害性機能を通じて提供されます。
- Ceph は OpenShift コンピュートインスタンスに使用されます。

重要

OpenShift コントロールプレーンを適切に分離できるようにするには、3 つ以上の OpenStack コンピューティングが使用可能である必要があります。また、障害が発生したコンピュートホストを迅速に交換できるようにする必要があります。そうしないと、OpenStack コンピュートホストに障害が発生した場合、コントロールプレーンが HA として動作しなくなります。

OpenShift アプリケーションに 外部負荷分散 を実装するために、OpenStack は Octavia ロードバランシングソリューション を提供します。Octavia は、Red Hat OpenStack Platform 16.2 の Load-Balancing-as-a-Service コンポーネントで、完全にサポートおよびテストされています。

3.4.3.2. OpenShift

Red Hat は、Nova アベイラビリティーゾーンと非アフィニティーポリシーを使用して、OpenShift ソリューションの高可用性を実装しています。各コントローラーノードを別々のアベイラビリティーゾーンに配置して、それらが同じ物理サーバー上にないようにします。

ワーカーノードは、同じアベイラビリティーゾーン全体で可能なかぎり均等に分散されます。これにより、物理インフラストラクチャー全体で均等に分散されます。非アフィニティーポリシーを使用して、ワーカーノードを別々の物理サーバーに配置するよう要求します。ワーカーノードは、障害ドメインごとに異なるタイプのルートボリュームを使用します。Cinder アベイラビリティーゾーン名は、Nova アベイラビリティーゾーン名と同じです。

3.5. ストレージ

複雑さを最小限に抑えながら、OpenShift アプリケーションのニーズを満たすストレージバックエンドを選択します。OpenStack Platform と OpenShift Container Platform の両方には、独立および成熟したストレージソリューションがあります。ただし、計画を立てずに、各プラットフォームのソリューションを組み合わせると、複雑さが増し、不要なパフォーマンスオーバーヘッドが生じる可能性があります。

ストレージコンポーネントの重複を避ける場合は、ワークロードを考慮してください。

3.5.1. 全体的なストレージバックエンド

Red Hat Ceph Storage は、OpenStack 向けの優先されるスケーラブルな統合およびサポートされているクラウドストレージを提供します。Red Hat Ceph Storage は、完全な Red Hat クラウドソフトウェアスタックと緊密に統合されています。ブロックおよびオブジェクトストレージ機能を提供します。

Ceph クラスターサーバーは、モニターとオブジェクトストレージデバイス (OSD) ノードに分割されます。

- Ceph モニターは、クラスタートポロジーのメインコピーを保持するモニターデーモンを実行します。
- クライアントは、データの保存方法と取得方法を決定するアルゴリズムを介してデータの場所を計算します (CRUSH マップと呼ばれます)。
- クライアントは、OSD との間でデータを直接読み書きします。
- モニターは、クラスターの正常性、状態、およびトポロジーを維持します。
- クライアントは OSD と直接やりとりします。クライアントは、モニターのみをチェックして、CRUSH マップが最新であることを確認します。
- データは、ノード内の物理ディスク全体に複製されます。

3.5.1.1. Ceph バックエンド

Red Hat OpenStack Platform 16.2 は、Director 主導のストレージソリューションの一部として Red Hat Ceph Storage (RHCS) 4 を使用します。BlueStore バックエンドがデフォルトで使用されます。RHCS 4 以降のインストールでは、よりパフォーマンスの高い BlueStore バックエンドの使用のみがサポートされます。

RHCS でサポートされるストレージバックエンドの詳細は、「アーキテクチャーガイド」の Ceph ObjectStore を参照してください。

RHCS の設定の詳細は、サポート対象の設定 を参照してください。

3.5.2. Object Storage (OpenStack Swift)

OpenStack は、OpenStack Object Store サービス (Swift) を介した Object Storage へのアクセスをサポートしています。

デフォルトでは、Director を使用して、Red Hat OpenStack Platform をインストールすると、シンプルなオブジェクトストレージデプロイメントがコントローラーに配置されます。このオブジェクトストレージは、Glance のバックエンドなどの最小限のワークロードの実稼働環境でサポートされています。

3.5.2.1. OpenShift レジストリー

OpenShift インストーラーは、スケーリングされたレジストリーの推奨プラクティス に従い、OpenStack のオブジェクトストレージサービスを、内部イメージレジストリーバックエンドの場所として優先的に使用します。これを行うために、インストーラーはアクセス可能なオブジェクトストレージの場所をチェックし、見つかった場合は、それを使用します。見つからない場合は、スケーリングなしのレジストリーの推奨ベストプラクティス を使用し、ReadWriteOnce (RWO) Cinder ボリュームを作成します。

3.5.3. Image Storage (OpenStack Glance)

Director を使用して Red Hat Ceph Storage (RHCS) をデプロイする場合、OpenStack Image サービス (Glance) のデフォルトのインストールでは RHCS をバックエンドとして使用します。これにより、イメージを冗長な高速ストレージに保存できるようになり、コピーオンライト (CoW) のクローン作成によって、より高速な起動と最適なストレージが実現します。

ただしこの場合、QCOW2 形式のイメージの使用は推奨されません。詳細は、RAW 形式へのイメージの変換 を参照してください。一時的なバックエンドまたはボリュームからインスタンスを起動するには、RAW イメージを使用する必要があります。

RHOCP インストールプログラムは、公開されている QCOW2 形式のイメージを自動的にダウンロードして使用します。これを変更するには、**clusterOSImage** インストール変数を、RAW 形式の外部イメージの URL、または OpenStack Image サービスにすでに保存されている既存のデプロイ済み RAW 形式イメージの名前に設定します。OpenShift 4.12 の場合は、公式ソース から OpenStack イメージを取得します。

ガイド付きインストールでは、**clusterOSImage** 変数を使用できません。この変数を **install-config.yaml** ファイルに手動で追加する必要があります。

OpenShift インストーラーは、次の 2 つの目的で Image サービスを使用します。

- ブートストラップクラスターを起動する Openshift Ignition ファイルを保存します。
- Red Hat Enterprise Linux CoreOS イメージを保存します。

3.5.4. OpenShift の永続ストレージ

Kubernetes の永続ボリューム (PV) フレームワークにより、OpenShift ユーザーは、ストレージアーキテクチャーの知識がなくても、基盤となるストレージサブシステムからストレージを要求できます。

OpenShift は、ストレージバックエンドへのアクセスを提供するための多くの方法をサポートしています。OpenStack 16.2 上の OpenShift 4.12 の場合は、可能な限り、Container Storage Interface (CSI) 標準をサポートするストレージバックエンドを使用することを推奨します。CSI は、コンテナー化されたワークロードに基盤となるブロックおよびファイルシステムへのアクセスを提供する標準化された方法を提供します。

CSI の詳細は、Kubernetes Container Storage Interface の「Kubernetes CSI Developer」セクションを参照してください。

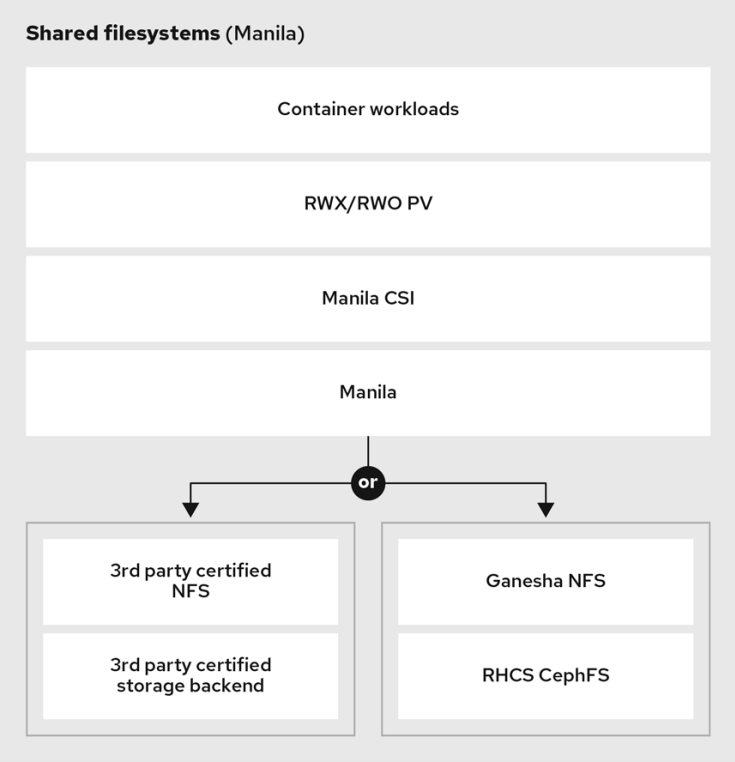
3.5.4.1. RWX PV の Manila-CSI

OpenShift は、OpenStack Manila CSI Driver Operator をサポートしているため、OpenStack の Shared File System サービス (manila) と対話して、リモートの共有可能なファイルシステムから PV をプロビジョニングできます。

Shared File Systems サービスを使用してリモートの共有ファイルシステムにアクセスするということは、コンテナーワークロードが、コンピュートインスタンス、ベアメタルノード、およびコンテナーへの同時アクセスを実現するストレージバックエンドを使用できることを意味します。OpenShift PV では、アクセスモードとして認識されるこのようなアクセスは、ReadWriteMany (RWX) と呼ばれており、複雑なコンテナーのユースケースに必要です。

OpenShift ユーザーは、Manila-CSI を使用すると、OpenStack API および Shared File System サービスを介して、リモートの共有ファイルシステムを簡単に使用できます。

図3.4 コンテナーワークロードに Manila を使用する



OpenShift インストーラーは、OpenShift が基盤となる OpenStack クラウドで Manila を検出すると、ストレージクラスを自動的に作成します。インストーラーは、そのストレージクラスをデフォルトとして設定しません。

インストーラープロビジョニングインフラストラクチャー方式で作成されるストレージクラス

```
$ oc get sc
NAME                 PROVISIONER                RECLAIMPOLICY   VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION   AGE
csi-manila-default   manila.csi.openstack.org   Delete          Immediate           false                  37d
standard-csi (default) cinder.csi.openstack.org   Delete          WaitForFirstConsumer   true
37d
```

デフォルトの StorageClass を設定する方法の詳細は、デフォルトストレージクラスの変更 を参照してください。

3.5.4.2. RWO PV の Cinder およびブロックストレージ

Openshift は、ボリュームをプロビジョニングするための Cinder CSI ドライバーをサポートしています。このドライバーでプロビジョニングされた PV は動的に割り当てられます。このボリュームは通常、ReadWriteOnce (RWO) アクセスモードで使用されます。ただし、マルチアタッチボリュームタイプ を使用する場合は、ボリュームを複数のインスタンスに同時に接続できます。一般に、このボリュームはブロックストレージや低遅延を必要とするアプリケーションに適しています。

このドキュメントではこれを考慮し、目的に応じて最適なオプションを使用します。

3.5.5. OpenShift ノードのストレージ

OpenShift ノードへのディスクの提供に関しては、一定の要件と選択肢があります。これは、特に etcd を実行するノードの場合に注意する必要があります。このセクションでは、選択肢について説明します。詳細は Red Hat 担当者にお問い合わせください。

OpenShift をインストールする場合、高速ストレージを提供する必要があります。

3.5.5.1. etcd の最小ディスク要件

コントロールプレーン仮想マシンは、etcd キー値ストアを実行するための既知のリソース要件を満たす必要があります。etcd の安定性とパフォーマンスを確保するために、書き込みレイテンシーが低いことが要件として求められます。Red Hat では、安定性を確保し、サポート性を保証するために、すべてのコントローラーが高速ディスク (SSD 以上) にアクセスできる必要があります。OSD が使用するディスクテクノロジーに関係なく、OpenShift コントロールプレーンで Ceph を使用する場合は、その他にも考慮事項があります。コントロールプレーンインスタンスに Ceph が必要な場合は、Red Hat サポートにガイダンスとサポート対象範囲の詳細をお問い合わせください。

etcd ディスク要件の詳細は、etcd のドキュメント を参照してください。

3.5.5.2. ディスクを OpenShift ノードに提供するオプション

クラスターにストレージを提供するには、複数のオプションがあります。決定に役立つように、次の 3 つのオプションを取り上げます。

- コンピュート上の一時ストレージ
- Ceph がバックアップするエフェメラル
- Cinder が提供するボリューム

3.5.5.2.1. コンピュート上の一時ストレージ

コンピュートノードでローカルにホストされているディスクを使用して仮想マシンを実行することは、コントロールプレーンにストレージを提供する最も簡単な方法です。Nova インスタンスのボリュームは、コンピュートハイパーバイザーのディスクに直接保存されます。これを行うには、ディスクが SSD 以上である必要があります。この方法を使用する場合は、物理的な OpenStack コンピュートノードが失われると、このハイパーバイザー上の一時ストレージを使用する仮想マシンインスタンスが破棄されることに注意してください。非アフィニティーを持つサーバーグループを使用して、OpenShift コントロールプレーンノードを別々のハイパーバイザーにプロビジョニングすることが重要です。

このソリューションでは、etcd に最高のパフォーマンスを提供するために、コントロールプレーンノードでこの方法を使用します。

3.5.5.2.2. Ceph がバックアップするエフェメラル

これは、Director でデプロイされた Ceph 環境のデフォルト設定です。この設定方法は、変数 **NovaEnableRBDBackend** を **True** に設定します。この設定は、多くの利点を持つ Ceph RBD を使用するように、すべての Nova バックエンドディスクに指示します。

- ボリュームはエフェメラルですが、Ceph クラスターから供給されます。
- Ceph Storage クラスター自体が耐障害性を提供します。

Ceph クラスターはより詳細なカスタマイズが可能です。Ceph は、ストレージディスクの使用方法を定義する 階層化オプションを実装 できます。

Ceph バックエンドを使用する etcd ベースのワークロードの環境の安定性と信頼性に関する重要な情報については、OpenStack での Ceph の使用に関するこちらの記事 を参照してください。

Director を使用して、Ceph をデプロイする場合、この設定方法はデフォルトで簡単に実装されます。

Director を使用して、Ceph をデプロイする場合に設定されるデフォルト値

```
# cat /usr/share/openstack-tripleo-heat-templates/environments/ceph-ansible/ceph-ansible.yaml
...
CinderEnableIscsiBackend: false
CinderEnableRbdBackend: true
CinderBackupBackend: ceph
NovaEnableRbdBackend: true
GlanceBackend: rbd
...
```

3.5.5.2.3. Cinder が提供するボリューム

Cinder は、Ceph を含む複数のバックエンドストレージプロバイダーに API フロントエンドを提供する OpenStack ブロックストレージサービスです。

Nova インスタンスは、Cinder API からボリュームを要求できます。次に、Cinder は、バックエンドプラグインを介して、ストレージをインスタンスで使用するように要求します。

Ceph を使用した OpenStack デプロイメントでは、OpenShift インストーラーで使用できるデフォルトのセットアップが作成され、Ceph がバックアップする Cinder ボリュームが OpenShift ノードに提供されます。この方法により、次のアクションを実行できる非常に緻密なソリューションが提供されます。

- Ceph の高速ストレージプールから Ceph がバックアップする Cinder ボリュームをプロビジョニングして、etcd のパフォーマンスを確保します。
- ストレージの複雑で洗練された層 (クラス) を作成して、さまざまなノードに提示します。
- 異なるバックエンドストレージ層からコントロールプレーンとワーカーノードにボリュームを割り当てます。

このソリューションは、柔軟性と耐障害性を提供するために、ソリューション内のワーカーノードに使用します。

3.5.6. Red Hat のテスト済みソリューション: ストレージ

このソリューションでは、ceph-ansible が提供するビルトインの Director テンプレートを使用して、Director で Red Hat Ceph Storage (RHCS) 4 をデプロイします。Nova 一時ドライブにローカルストレージを使用するために、Director で **NovaEnableRbdBackend** プロパティーを **false** に設定します。

デプロイメントは、ハイパーコンバージドインフラストラクチャー (HCI) モード を使用してテストされています。HCI デプロイメントパターンでは、コンピュートサービスとストレージサービスが同じノード (ハイパーコンバージドノード) に配置され、リソースの使用に合わせて最適化されます。これは、専用のストレージノードを用意することと機能的には同等ですが、必要な物理インフラストラクチャーが少なくなります。独自のワークロードに合わせて、要件に最も適したストレージレイアウトのタイプを選択できます。Ceph Monitor は OpenStack コントローラー上の同じ場所に配置されています。

この Director デプロイメントパターンでは、ローカルストレージ (一時ストレージ) をコントロールプレーンノードのバッキングストアとして設定し、ワーカーノード、Cinder ボリューム、Glance イメージ、およびオブジェクトストレージに RHCS を設定します。このような設定により、OpenShift コントロールプレーン用に高速ストレージを確保し、etcd のビルトインの耐障害性機能を利用して HA を実現するとともに、RHCS によってワーカーノードと OpenStack サービスに冗長性を提供します。

Director を使用して Shared File Systems サービス (manila) をインストールし、NFS に Ganesha を使用する RHCS バックエンドでサービスを設定します。CephFS NFS および Manila CSI の詳細は、Red Hat OSP 16.x の推奨事項 を参照してください。

サポートおよび認定されているサードパーティーの Manila CSI ストレージドライバーおよびベンダーについては、Red Hat Partner Ecosystem Catalog を参照してください。

このソリューションでは、Red Hat Ceph Storage は以下のサービスをサポートしています。

- ブロック (Cinder)
- Image (Glance)
- Ceph RADOS オブジェクトストアのゲートウェイである radosgw (RGW) 経由のオブジェクト (Swift)。

このソリューションでは、OpenShift は、さまざまな目的でストレージを消費します。

- コンテナーワークロード用のストレージは、Cinder (RWO) と manila (RWX) によって提供されます。
- OpenShift レジストリーのストレージは、Ceph RGW (オブジェクトストレージ) によって提供され、OpenStack テナントには Swift オペレーターロールが付与されます。
- 各テナントのクラスターの RHCOS イメージファイルとブートストラップ Ignition ファイルは、OpenStack のイメージサービスである Glance に保存されます。各 RHCOS イメージはクラスターが削除されるまで残ります。Ignition イメージは一時的なもので、インストールプロセスの最後にインストーラーによって削除されます。

3.6. ネットワーク

Red Hat OpenStack Platform と Red Hat OpenShift Container Platform には、独立および成熟したネットワークソリューションがあります。

適切なソリューションの選択は、ワークロードの詳細によって異なります。OpenShift アプリケーションのニーズを満たし、複雑さを最小限に抑えるネットワークソリューションを選択してください。

これらのソリューションは階層化できますが、ワークロードの要件を考慮して、ネットワーク機能が重複しないようにしてください。

3.6.1. OpenStack ネットワーク

OpenStack Neutron API は、さまざまなバックエンドネットワークプラグインに共通の抽象化レイヤーを提供します。OpenStack Platform は複数のバックエンドネットワークプラグインをサポートしています。

- ML2 OVS
- OVN
- 商用 SDN

プロバイダーネットワークは、外部からアクセス可能な一連のフローティング IP アドレスをテナントインスタンスに割り当てます。リモートクライアントは、フローティング IP アドレスを使用して、次のサービスにアクセスします。

- OpenShift アプリケーション
- OpenShift API エンドポイント
- Web コンソール

OpenStack ネットワークの詳細は、ネットワークガイド を参照してください。

3.6.1.1. OpenStack ネットワーク (Neutron) バックエンドプラグイン

他の多くの OpenStack コンポーネントと同様、ネットワークサブシステム (Neutron) はプラグイン可能であるため、ビジネス要件とテクノロジー要件に最適なソリューションを選択できます。Red Hat OpenStack Platform の場合は、サポートされているオプションから選択できます。

Red Hat OpenStack Platform 16.2 によって実装されるデフォルトの Neutron バックエンドは OVN です。

3.6.1.1.1. Open vSwitch (OVS)

OVS は、仮想化されたサーバー環境や複数の Red Hat 製品で仮想スイッチとして使用するように設計されたオープンソースのマルチレイヤーソフトウェアスイッチです。OVS は、複数のリリースで OSP のデフォルトの Neutron バックエンドであり、完全にテストおよびサポートされています。

3.6.1.1.2. Open Virtual Network (OVN)

Red Hat OpenStack Platform 16.2 は、デフォルトで Open vSwitch の Open Virtual Network (OVN) コンポーネントを使用します。OVN は、Open vSwitch (OVS) プロジェクトに組み込まれているネットワーク仮想化ソリューションです。OVN は、Neutron API をサポートし、ブリッジング、ルーティング、セキュリティーグループ、NAT、フローティング IP などの最も一般的なネットワーク機能のクリーンで分散された実装を提供します。

OVN を Neutron のバックエンドとして使用する場合は、Generic Network Virtualization Encapsulation (GENEVE) がネットワークのカプセル化に使用されます。OVN と GENEVE を使用すると、次の利点があります。

- 拡張可能なヘッダー形式による柔軟性の向上
- デフォルトによる L3 トラフィックの分散。

Red Hat OpenStack Platform 16.1 より前のバージョンでは、デフォルトで OVS の ML2 プラグインが使用されていました。

3.6.1.1.3. サードパーティーの SDN ソリューション

複数のネットワークベンダーが独自の Neutron 用プラグインをサポートしています。OpenShift on OpenStack の今後のイテレーションでは、追加のサードパーティーソリューションをサポートするためにパートナーシップとテストが強化される可能性があります。

3.6.1.2. Neutron バックエンドと Red Hat OpenStack Platform

固有の状況に合った Neutron バックエンドを選択する必要があります。

OpenStack 16.2 で OpenShift 4.12 を運用する予定の場合は、OVN のデフォルトバックエンドの使用を強く推奨します。これは、最もテストされた設定であり、必要に応じて LoadBalancer サービスに Octavia OVN プロバイダーを使用できるためです。

OpenShift on OpenStack をデプロイする際に Neutron バックエンドを選択するには、次の点を考慮してください。

- OVN は、OpenShift on OpenStack インストールで完全にテストされ、サポートされています。
- OVN は、Red Hat OpenStack Platform 16.1 以降のデフォルトの Neutron バックエンドです。

3.6.1.3. OpenStack 負荷分散

Red Hat OpenStack Platform は、Load Balancing as a Service (LBaaS) の実装に Octavia プロジェクトを使用します。OVN とともにデプロイされた OSP 16.2 では、デフォルトの Amphora と OVN プロバイダーという 2 つの Octavia バックエンドが利用可能です。主な違いとしては、Amphora は仮想マシンを作成し、その上に HAProxy インスタンスを設定して各ロードバランサーをデプロイする点が挙げられます。一方、OVN プロバイダーは OpenFlow ルールを使用してロードバランサーを実装します。

デフォルトでは、OpenShift は Octavia バックエンドとして Amphora を使用します。

制限を含む OVN Octavia プロバイダーの詳細については、アップストリームのドキュメント を確認し、Red Hat ソリューションアーキテクトまたはサポートアソシエイトにお問い合わせください。

3.6.1.4. Red Hat のテスト済みソリューション: OpenStack ネットワーク

このソリューションでは、Red Hat OpenStack Platform 16.2 のデフォルトの Neutron ネットワークバックエンドである OVN を使用します。

3.6.1.4.1. OpenShift ネットワーク

OpenShift on OpenStack の以前のリリースでは、ネットワークに Kuryr SDN プラグインを使用していました。Kuryr は、パフォーマンスが高く信頼性の高いソリューションであることが証明されており、多くの実稼働環境で問題なく使用されています。

OpenStack と OpenShift は、過去数回のリリースを通じて、利用可能な SDN ソリューションの要件と機能が劇的に進化しました。これにより、製品自体をコンパクト化しながら、機能の拡充と拡張性および信頼性の向上を実現しています。

現在、両製品は SDN ソリューションのデフォルトの選択肢として Open Virtual Network (OVN) を提供し、これをデフォルトで使用しています。

これらの変更により、SDN レイヤーでの二重カプセル化によって引き起こされる潜在的なパフォーマンスオーバーヘッドなど、以前は Kuryr を使用して対処していたユースケースの多くに対処しています。さらに、OVN への移行により、Kuryr で見られたスケーラビリティーと機能の制限の一部が解消さ

れました。

OVN-Kubernetes Container Network Interface (CNI) プラグインは、OpenShift ネットワーク用の次世代 SDN ソリューションです。OVN-Kubernetes は、OVN (Open Virtual Network) を利用するように構築されており、ベンダーに依存しない大規模なアップストリームコミュニティーによってサポートされています。OVN-Kubernetes には、次の特徴があります。

- OVN を使用して、ネットワークトラフィックフローを管理する
- Kubernetes ネットワークポリシーのサポートを実装する
- ノード間のオーバーレイネットワーク用の GENEVE (Generic Network Virtualization Encapsulation) プロトコルを導入します。

これにより、ネットワークレイヤーをより緻密に統合できるため、複雑さを軽減し、より予測可能な結果を確保できます。

ネットワークと OpenShift の詳細は、公式ドキュメントの Understanding networking を参照してください。

3.6.1.5. LoadBalancer サービス

OpenShift の LoadBalancer サービスは、cloud-provider-openstack および OpenStack Octavia によって処理されます。**LoadBalancer** タイプの各サービスに対して、Octavia にロードバランサーが作成されます。デフォルトでは、OpenShift は、より機能が豊富なソリューションである Amphora バックエンドを使用します。欠点としては、リソースの消費量が増加する点があります。Amphora はロードバランサーごとに仮想マシンを作成するためです。

ロードバランサーサービスは比較的静的である傾向があるため、Amphora はほとんどのユースケースに適していることが判明しました。

また、Amphora は、Pod が受信するトラフィックの送信元 IP を、常にロードバランサーの IP に設定することに注意してください。そのため、**.spec.externalTrafficPolicy** を **Local** に設定しても、PROXY プロトコルなどの追加機能を使用しない限り、元のソース IP を取得することはできません。

リソースの消費量が懸念される場合は、OVN Octavia プロバイダーを使用するように Openshift クラスター を設定できます。Octavia プロバイダーを変更する前に、次の制限事項を考慮する必要があります。

- OSP 16.2 の OVN Octavia プロバイダーはヘルスモニターをサポートしません。そのため、**.spec.externalTrafficPolicy** を **Local** に設定してサービスを実行することはサポートされていません。
- OVN Octavia Provider を使用する場合、**.spec.loadBalancerSourceRanges** オプションは無視され、トラフィックは常に無制限になります。

OSP 16.2 および OpenShift 4.12 では、cloud-provider-openstack で使用される Octavia バックエンドとしてデフォルトの Amphora を使用することを推奨します。

3.6.1.6. Red Hat のテスト済みソリューション: OpenShift ネットワーク

このソリューションでは、OpenShift ネットワークに OVN-Kubernetes を、Octavia バックエンドとして Amphora を使用しています。

3.7. DNS

OpenShift 4 インストーラーは、以前の OpenShift on OpenStack インストールで見られた DNS 要件を大幅に簡素化します。クラスターのすべての内部 DNS 解決、証明書の検証、およびブートストラップは、インストーラー制御のセルフホスト型ソリューションを通じて提供されます。

このソリューションでは、コントロールプレーンノード、ワーカーノード、およびブートストラップノードの IP を手動または動的にパブリック DNS の形式に追加する必要はありません。それは完全に自己完結型です。

詳細については、OpenStack インストーラープロビジョニングインフラストラクチャーのネットワークインフラストラクチャー ページを参照してください。

このソリューションでは、セルフホステッドネームサーバーを実行する必要はなく、Designate DNSaaS プロジェクトなどの外部ソリューションも必要ありません。

3.7.1. DNS に関する考慮事項

インストールを完了するには、次の DNS 要件を満たす必要があります。

- インストールホストは OpenShift API アドレスを解決する必要があります。
- コンテナーイメージをダウンロードするには、ノードがコンテナーレジストリーのアドレスを解決できる必要があります。

インストールが完了したら、次の DNS 要件を満たす必要があります。

- ワイルドカードドメインは Ingress ポートに解決する必要があります。

3.7.1.1. API DNS

クラスターをインストールする前に、到達可能な IP アドレスが DNS に存在している必要があります。次のアドレス空間を解決する必要があります。

```
api.<cluster name>.<base domain>
```

インストーラーは、**install-config.yaml** の OpenStack プラットフォームセクションの **apiFloatingIP** 値により、既存の Floating IP を API ポートに自動的に関連付けることができます。

OpenShift の API アドレスを特定のフローティング IP 値に割り当てる方法の例。

```
platform:
  openstack:
...
    apiFloatingIP:     "10.46.43.176"
```

3.7.1.2. アプリケーション DNS

このドメインは、OpenShift で実行されているアプリケーションへのアクセスに使用されます。DNS で、次の名前構造を解決するワイルドカードエントリーを作成します。

```
*.apps.<cluster name>.<base domain>.
```

install-config.yaml の **ingressFloatingIP** 値を使用して、この OpenShift アプリケーションの IP をクラスターに自動的に関連付けることができます。**ingressVIP** と **ingressFloatingIP** を混同しないでく

ださい。**ingressVIP** は Machine ネットワーク用であり、 **ingressFloatingIP** は外部ネットワーク用です。

3.7.1.3. ブートストラップノード

ブートストラップノードは、ブートストラップクラスターと実稼働クラスターを起動するための基本的なリソースを取得します。そのため、レジストリードメイン名を直接解決できる必要があります。

3.7.1.4. 追加の DNS 機能

3.7.1.4.1. externalDNS

OpenStack テナントは、そのクラウドがテナントのサブネットに名前解決を自動的に提供しない場合、OpenShift のインストール時に、**install-config.yaml** のオプションの **externalDNS** 値を使用してこの名前解決を設定できます。**externalDNS** は、作成する OpenShift サブネットに DNS IP を追加するように、インストーラーに指示します。この値は配列であるため、複数のエントリーを含めることができます。

インストーラーが作成した各サブネットの DNS 値を設定する方法の例。

```
externalDNS:    ["203.0.113.1", ”203.0.113.2”]
```

注記

インストールファイルに **externalDNS** を手動で追加する必要があります。ガイド付きインストールでは、**externalDNS** を使用できません。

3.7.2. Red Hat のテスト済みソリューション: DNS

このソリューションでは、DNS で 2 つの Floating IP を事前割り当ておよび事前設定し、API アドレスとアプリケーションアドレスを提供します。

また、**externalDNS** パラメーターを使用して、インストーラーが構築したサブネットが外部 DNS 解決をインスタンスに提供できるようにします。

3.8. セキュリティーと認証

OpenShift と OpenStack は、いずれもロールベースのアクセス制御、および既存のユーザー認証システムと統合するための柔軟なオプションをサポートしています。また、いずれも、SELinux など、Red Hat Enterprise Linux のネイティブなセキュリティー機能を継承します。

3.8.1. 認証

OpenStack アイデンティティーサービスは、次の 2 つの方法でユーザー認証情報を保存します。

- ローカルの状態データベース。
- 外部の LDAP 準拠のディレクトリーサーバー内。

OpenShift コントローラーノードは、トークンを発行して、API でユーザーリクエストを認証します。OpenShift は、HTPassword や LDAP など、さまざまな ID プロバイダーをサポートしています。

OpenShift と OpenStack のアイデンティティープロバイダー間の詳細な統合があります。管理者は、

OpenStack Keystone サービスを使用して、OpenShift キーストーンアイデンティティープロバイダーを設定できます。この設定により、ユーザーは Keystone 認証情報を使用して OpenShift Container Platform にログインできます。詳細は、OpenShift ドキュメントの Configuring a Keystone identity provider セクションを参照してください。

3.8.2. セキュリティー

セキュリティーのベストプラクティスの多くは、デフォルトの OpenStack Platform デプロイメントに組み込まれています。Red Hat OpenStack Platform は、ANSSI や FedRamp など、さまざまなレベルの標準セキュリティーコンプライアンスを満たしています。ここに記載されているこのソリューションは、OpenStack を保護するための包括的なリソースではなく、比較的基本的ですが、実稼働環境でサポートされているレベルのセキュリティーを前提としています。すべてのエンタープライズ要件に適しているとはかぎりません。

OpenStack を保護するためのベストプラクティスの詳細は、Red Hat OpenStack Platform 16 セキュリティーおよび強化ガイド を参照してください。

3.8.2.1. OpenStack セキュリティーグループ

OpenShift on OpenStack をインストールする場合、インストーラーは機能的なインストールに必要な OpenStack セキュリティーグループを作成します。インストーラーは、コントロールプレーンノードとワーカーノードのセキュリティーグループを作成します。

セキュリティーグループを確認して、内部のセキュリティー要件にどのように影響するかを明確に理解する必要があります。

作成されるルールの詳細を確認するには、コントロールプレーンノード と ワーカーノード のアップストリームのコードを確認してください。

これらのグループの使用方法の詳細については、OpenStack のアップストリームクラスター API ドキュメントの セキュリティーグループルール を参照してください。

インストール時にセキュリティーグループを直接追加する方法については、インストールガイドの Optional RHOSP configuration parameters セクションを参照してください。

3.8.3. Red Hat のテスト済みソリューション: セキュリティーと認証

インストーラープロビジョニングインフラストラクチャーメソッドは、必要なすべての OpenStack セキュリティーグループとルールを作成および管理します。これらのグループは、テナントの OpenShift インストールをクラウド上の他のグループから完全に分離します。

Transport Layer Security (TLS) を使用して、OpenStack パブリック API エンドポイントを暗号化します。

- インストールホストで自己署名証明書とローカル認証局を使用します。
- OpenStack TLS パブリックエンドポイント暗号化を有効にした後、エンドポイントにコマンドを発行するホストに証明書をインポートします。
- Director ホストを使用して、インストーラーを実行するため、暗号化されたエンドポイントとのやりとりは、すべて Director ホストから行われます。

OpenStack でのこの手順の詳細は、SSL/TLS 証明書署名要求の作成 を参照してください。

第4章 概要

Red Hat OpenShift Platform 4、Red Hat OpenStack Platform 16.2、および Red Hat Ceph Storage 4 を使用すると、オンプレミスのコンテナーインフラストラクチャーを包括的かつ規範的な方法でインストールできます。

この Red Hat のテスト済みソリューションは、コンテナー化されたインフラストラクチャー、仮想マシン (VM)、および関連するアプリケーションとインフラストラクチャーサービスの迅速なプロビジョニングとライフサイクル管理を提供する Red Hat の規範的な検証済みのプライベートクラウドソリューションを紹介しています。

Red Hat Quality Engineering (QE) チームは、このソリューションで提示されている実装をテストおよび検証しました。このソリューションを迅速に運用しようとしているお客様は、提示されたすべてのオプションが完全にテストされており、Red Hat によって完全にサポートされていることが保証されます。

Red Hat OpenShift Container Platform、Red Hat OpenStack Platform、および Red Hat Ceph Storage は、このソリューションの主要なアーキテクチャーコンポーネントです。これらを統合することは、ハイブリッドおよびマルチクラウドソリューションにおける重要な要素です。OpenShift Container Platform は、デプロイメントフットプリント全体の共通のコンテナーおよびプラットフォームとして機能します。

第5章 追加のリンクとリファレンス

5.1. 以前のリファレンスアーキテクチャー

- Red Hat OpenStack Platform 13 への Red Hat OpenShift Container Platform 3.11 のデプロイ
- Red Hat OpenStack Platform 13 および 16 への Red Hat OpenShift Container Platform 4.4 のデプロイ
- Red Hat Openstack Platform 16.1 への Red Hat Openshift Container Platform 4.6 および 4.7 のデプロイ

5.2. 関連ドキュメントへのリンク

5.2.1. Red Hat OpenShift Container Platform

- 公式の製品ドキュメント
- OpenStack クラスターへの OpenShift Container Platform のインストール
- アップストリームインストーラーのコードとドキュメント
- 4.12 のインストール設定ファイルのパラメーター
- OpenShift Container Platform の詳細

5.2.2. Red Hat OpenStack Platform

- 公式の製品ドキュメント
- コンテナー化された Red Hat Ceph を持つオーバークラウドのデプロイ

5.2.3. Red Hat Ceph Storage

- 公式の製品ドキュメント

付録A コントリビューター

Martin André、 Matthew Booth、 Jon Uriarte、 Itzik Brown、 Emilien Macchi、 Eric Duen、 Gil Rosenberg、 Michal Dulko、 Roger Heslop、 August Simonelli、 Ramón Lobillo