OpenShift Container Platform 4.12

RHV へのインストール

Red Hat Virtualization への OpenShift Container Platform のインストール

Last Updated: 2025-11-12

OpenShift Container Platform 4.12 RHV へのインストール

Red Hat Virtualization への OpenShift Container Platform のインストール

Legal Notice

Abstract

本書では、Red Hat Virtualization に OpenShift Container Platform をインストールする方法を説明します。

Table of Contents

第1章 RED HAT VIRTUALIZATION (RHV) へのインストールの準備

1.1. 前提条件

- OpenShift Container Platform のインストールおよび更新 プロセスの詳細を確認している。
- Support Matrix for OpenShift Container Platform on Red Hat Virtualization (RHV) に記載のあるサポートされるバージョンの組み合わせを使用できる。
- クラスターインストール方法の選択およびそのユーザー向けの準備 を確認している。

1.2. RHV に OPENSHIFT CONTAINER PLATFORM をインストールする方法の選択

OpenShift Container Platform をインストーラーまたはユーザーによってプロビジョニングされるインフラストラクチャーにインストールすることができます。デフォルトのインストールタイプは、installer-provisioned infrastructure を使用します。この場合、インストールプログラムがクラスターの基礎となるインフラストラクチャーをプロビジョニングします。OpenShift Container Platform は、ユーザーがプロビジョニングするインフラストラクチャーにインストールすることもできます。インストールプログラムがプロビジョニングするインフラストラクチャーを使用しない場合は、クラスターリソースをユーザー自身で管理し、維持する必要があります。

インストーラーおよびユーザーによってプロビジョニングされる インストールプロセス の詳細は、インストールプロセス を参照してください。

1.2.1. インストーラーでプロビジョニングされるインフラストラクチャーへのクラスターのインストール

以下の方法のいずれかを使用して、OpenShift Container Platform インストールプログラムでプロビジョニングされる Red Hat Virtualization (RHV) 仮想マシンに、クラスターをインストールできます。

- **クラスターの RHV へのクイックインストール**: OpenShift Container Platform インストールプログラムがプロビジョニングする RHV 仮想マシンに OpenShift Container Platform をクイックインストールできます。
- **カスタマイズによる RHV へのクラスター**のインストール：RHV のインストーラーによってプロビジョニングされたゲストに、カスタマイズされた OpenShift Container Platform クラスターをインストールできます。インストールプログラムは、インストールの段階で一部のカスタマイズを適用できるようにします。その他の数多くのカスタマイズオプションは、インストール後 に利用できます。

1.2.2. ユーザーによってプロビジョニングされるインフラストラクチャーへのクラスターのインストール

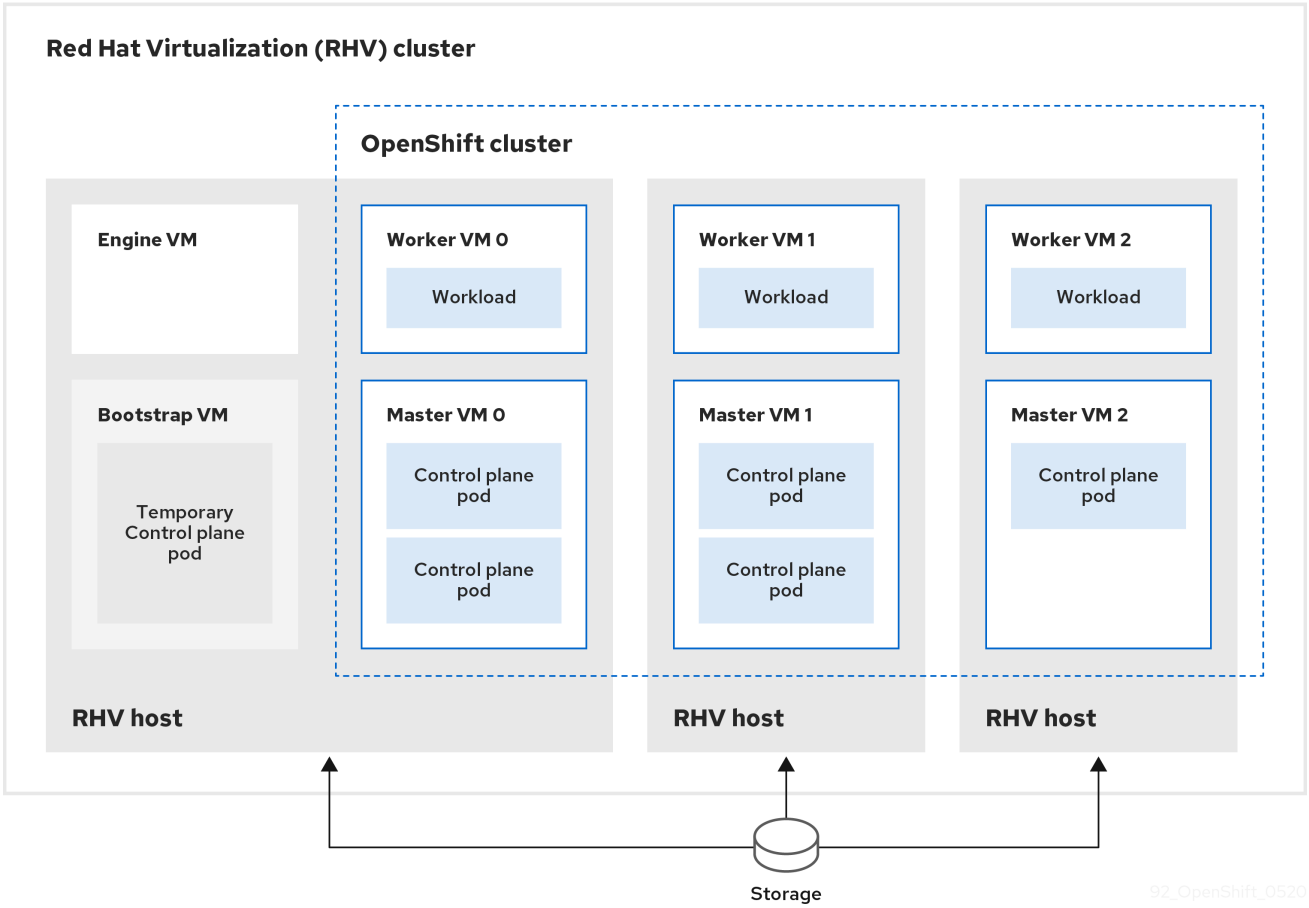
以下の方法のいずれかを使用して、独自にプロビジョニングする RHV 仮想マシンにクラスターをインストールできます。

- **ユーザーによってプロビジョニングされるインフラストラクチャーでの RHV へのクラスター**のインストール：プロビジョニングする RHV 仮想マシンに OpenShift Container Platform をインストールできます。提供される Ansible Playbook を使用してインストールを支援することができます。

- **ネットワークが制限された環境での RHV へのクラスター**のインストール：インストールリリースコンテンツの内部ミラーを作成して、OpenShift Container Platform をネットワークが制限された環境で RHV にインストールできます。この方法を使用して、ソフトウェアコンポーネントを取得するためにアクティブなインターネット接続を必要としないユーザーによってプロビジョニングされるクラスターをインストールできます。また、このインストール方法を使用して、クラスターが外部コンテンツに対する組織の制御の条件を満たすコンテナーイメージのみを使用するようにすることもできます。

第2章 RHV へのクラスターのクイックインストール

以下の図に示されるように、デフォルトの、カスタマイズされていない OpenShift Container Platform クラスターを Red Hat Virtualization (RHV) クラスターにすばやくインストールできます。



インストールプログラムは、インストーラーでプロビジョニングされるインフラストラクチャーを使用してクラスターの作成およびデプロイを自動化します。

デフォルトのクラスターをインストールするには、環境を準備し、インストールプログラムを実行してプロンプトに応答します。次に、インストールプログラムは OpenShift Container Platform クラスターを作成します。

デフォルトクラスターの代替インストール方法については、カスタマイズによるクラスターのインストール を 参照してください。

注記

このインストールプログラムは、Linux および macOS でのみ利用できます。

2.1. 前提条件

- OpenShift Container Platform のインストールおよび更新 プロセスの詳細を確認している。
- Support Matrix for OpenShift Container Platform on Red Hat Virtualization (RHV) に記載のあるサポートされるバージョンの組み合わせを使用できる。
- クラスターインストール方法の選択およびそのユーザー向けの準備 を確認している。

- ファイアウォールを使用する場合は、クラスターがアクセスを必要とするサイトを許可するようにファイアウォールを設定する必要がある。

2.2. OPENSHIFT CONTAINER PLATFORM のインターネットアクセス

OpenShift Container Platform 4.12 では、クラスターをインストールするためにインターネットアクセスが必要になります。

インターネットへのアクセスは以下を実行するために必要です。

- OpenShift Cluster Manager Hybrid Cloud Console にアクセスし、インストールプログラムをダウンロードし、サブスクリプション管理を実行します。クラスターにインターネットアクセスがあり、Telemetry を無効にしない場合、そのサービスは有効なサブスクリプションでクラスターを自動的に使用します。
- クラスターのインストールに必要なパッケージを取得するために Quay.io にアクセスします。
- クラスターの更新を実行するために必要なパッケージを取得します。

重要

クラスターでインターネットに直接アクセスできない場合、プロビジョニングする一部のタイプのインフラストラクチャーでネットワークが制限されたインストールを実行できます。このプロセスで、必要なコンテンツをダウンロードし、これを使用してミラーレジストリーにインストールパッケージを設定します。インストールタイプに応じて、クラスターのインストール環境でインターネットアクセスが不要となる場合があります。クラスターを更新する前に、ミラーレジストリーのコンテンツを更新します。

2.3. RHV 環境の要件

OpenShift Container Platform バージョン 4.12 クラスターをインストールし、実行するには、RHV 環境が以下の要件を満たしている必要があります。

これらの要件を満たさないと、インストールまたはプロセスが失敗する可能性があります。さらに、これらの要件を満たしていないと、OpenShift Container Platform クラスターはインストールしてから数日または数週間後に失敗する可能性があります。

CPU、メモリー、ストレージリソースについての以下の要件は、インストールプログラムが作成する仮想マシンのデフォルト数で乗算した **デフォルト** 値に基づいています。これらのリソースは、RHV 環境が OpenShift Container Platform 以外の操作に使用するものに **加え**、利用可能でなければなりません。

デフォルトでは、インストールプログラムは 7 つの仮想マシンをインストールプロセスで作成します。まず、ブートストラップ仮想マシンを作成し、OpenShift Container Platform クラスターの残りの部分を作成する間に一時サービスとコントロールプレーンを提供します。インストールプログラムがクラスターの作成を終了すると、ブートストラップマシンが削除され、そのリソースが解放されます。

RHV 環境の仮想マシン数を増やす場合は、リソースを適宜増やす必要があります。

要件

- RHV のバージョンは 4.4 である。
- RHV 環境に **Up** 状態のデータセンターが 1 つあること。
- RHV データセンターに RHV クラスターが含まれていること。

- RHV クラスターに OpenShift Container Platform クラスター専用の以下のリソースがあること。
  - 最小 28 vCPU: インストール時に作成される 7 仮想マシンのそれぞれに 4 vCPU。
  - 以下を含む 112 GiB 以上の RAM。
    - 一時的なコントロールプレーンを提供するブートストラップマシン用に 16 GiB 以上。
    - コントロールプレーンを提供する 3 つのコントロールプレーンマシンのそれぞれに 16 GiB 以上。
    - アプリケーションワークロードを実行する 3 つのコンピュートマシンのそれぞれに 16 GiB 以上。
- RHV ストレージドメインは、これらの etcd バックエンドのパフォーマンス要件 を満たす必要があります。
- アフィニティーグループのサポートの場合: RHV クラスター内の 3 つ以上のホスト。必要に応じて、アフィニティーグループを無効にすることができます。詳細は、**カスタマイズによる RHV へのクラスターのインストール**の**実稼働以外のラボセットアップのすべてのアフィニティーグループを削除する例**を参照してください。
- 実稼働環境では、各仮想マシンに 120 GiB 以上が必要です。そのため、ストレージドメインはデフォルトの OpenShift Container Platform クラスターに 840 GiB 以上を提供する必要があります。リソースに制約のある環境または非実稼働環境では、各仮想マシンに 32 GiB 以上を指定する必要があるため、ストレージドメインにはデフォルトの OpenShift Container Platform クラスター用に 230 GiB 以上が必要になります。
- インストールおよび更新中に Red Hat Ecosystem Catalog からイメージをダウンロードするには、RHV クラスターがインターネット接続にアクセスできる必要があります。また、サブスクリプションおよびエンタイトルメントプロセスを単純化するために Telemetry サービスにもインターネット接続が必要です。
- RHV クラスターには、RHV Manager の REST API にアクセスできる仮想ネットワークが必要です。インストーラーが作成する仮想マシンが DHCP を使用して IP アドレスを取得するため、DHCP がこのネットワークで有効にされていることを確認します。
- ターゲット RHV クラスターに OpenShift Container Platform クラスターをインストールし、管理するための以下の最小限の権限を持つユーザーアカウントおよびグループ。
  - **DiskOperator**
  - **DiskCreator**
  - **UserTemplateBasedVm**
  - **TemplateOwner**
  - **TemplateCreator**
  - ターゲットクラスターの **ClusterAdmin**

警告

最小権限の原則を適用します。インストールプロセスで RHV で **SuperUser** 権限を持つ管理者アカウントを使用することを避けます。インストールプログラムは、ユーザーが指定する認証情報を、危険にさらされる可能性のある一時的な **ovirt-config.yaml** ファイルに保存します。

関連情報

- 実稼働以外のラボセットアップのすべてのアフィニティーグループを削除する例。

2.4. RHV 環境の要件の確認

RHV 環境が OpenShift Container Platform クラスターをインストールし、実行するための要件を満たしていることを確認します。これらの要件を満たさないと、エラーが発生する可能性があります。

重要

これらの要件は、インストールプログラムがコントロールプレーンおよびコンピュートマシンの作成に使用するデフォルトのリソースに基づいています。これらのリソースには、vCPU、メモリー、およびストレージが含まれます。これらのリソースを変更するか、OpenShift Container Platform マシンの数を増やす場合は、これらの要件を適宜調整します。

手順

1. RHV バージョンが OpenShift Container Platform バージョン 4.12 のインストールをサポートしていることを確認します。
   a. RHV Administration Portal の右上にある ? ヘルプアイコンをクリックし、**About** を選択します。
   b. 開かれるウィンドウで、**RHV ソフトウェアのバージョン** をメモします。
   c. RHV のバージョンが 4.4 であることを確認します。サポートされるバージョンの組み合わせについての詳細は、Support Matrix for OpenShift Container Platform on RHV を参照してください。
2. データセンター、クラスター、およびストレージを検査します。
   a. RHV 管理ポータルで、**Compute → Data Centers** をクリックします。
   b. OpenShift Container Platform をインストールする予定のデータセンターにアクセスできることを確認します。
   c. そのデータセンターの名前をクリックします。
   d. データセンターの詳細の **Storage** タブで、OpenShift Container Platform をインストールする予定のストレージドメインが **Active** であることを確認します。
   e. 後で使用できるように **ドメイン名** を記録します。

f. **空き領域** に 230 GiB 以上あることを確認します。

g. ストレージドメインが これらの etcd バックエンドのパフォーマンス要件 を満たしていることを確認します。これは、fio パフォーマンスベンチマークツールを使用して測定できます。

h. データセンターの詳細で、**Clusters** タブをクリックします。

i. OpenShift Container Platform をインストールする予定の RHV クラスターを見つけます。後で使用できるようにクラスター名を記録します。

3. RHV ホストリソースを確認します。

a. RHV 管理ポータルで、**Compute** > **Clusters** をクリックします。

b. OpenShift Container Platform をインストールする予定のクラスターをクリックします。

c. クラスターの詳細で、**Hosts** タブをクリックします。

d. ホストを検査し、それらに OpenShift Container Platform クラスター **専用** として利用可能な **論理 CPU コア** の合計が 28 つ以上であることを確認します。

e. 後で使用できるように、利用可能な **論理 CPU コア** の数を記録します。

f. これらの CPU コアが分散され、インストール時に作成された 7 つの仮想マシンのそれぞれに 4 つのコアを持たせることができることを確認します。

g. ホストには、以下の OpenShift Container Platform マシンのそれぞれの要件を満たすように **新規仮想マシンをスケジュールするための最大空きメモリー** として 112 GiB があることを確認します。

- ブートストラップマシンに 16 GiB が必要です。
- 3 つのコントロールプレーンマシンのそれぞれに 16 GiB が必要です。
- 3 つのコンピュートマシンのそれぞれに 16 GiB が必要です。

h. 後で使用できるように **新規仮想マシンをスケジュールするための最大空きメモリー** の量を記録します。

4. OpenShift Container Platform をインストールするための仮想ネットワークが RHV Manager の REST API にアクセスできることを確認します。このネットワーク上の仮想マシンから、RHV Manager の REST API に到達するために curl を使用します。

```
$ curl -k -u <username>@<profile>:<password> \ 1
https://<engine-fqdn>/ovirt-engine/api 2
```

1 **<username>** については、RHV で OpenShift Container Platform クラスターを作成および管理する権限を持つ RHV アカウントのユーザー名を指定します。**<profile>** には、ログインプロファイルを指定します。ログインプロファイルは、RHV Administration Portal ログインページに移動し、 **Profile** ドロップダウンリストで確認できます。**<password>** に、そのユーザー名のパスワードを指定します。

2 **<engine-fqdn>** に、RHV 環境の完全修飾ドメイン名を指定します。

以下に例を示します。

```
$ curl -k -u ocpadmin@internal:pw123 \
https://rhv-env.virtlab.example.com/ovirt-engine/api
```

2.5. RHV でのネットワーク環境の準備

OpenShift Container Platform クラスターの 2 つの静的 IP アドレスを設定し、これらのアドレスを使用して DNS エントリーを作成します。

手順

1. 2 つの静的 IP アドレスを予約します。
   a. OpenShift Container Platform をインストールするネットワークで、DHCP リースプール外にある 2 つの静的 IP アドレスを特定します。
   b. このネットワーク上のホストに接続し、それぞれの IP アドレスが使用されていないことを確認します。たとえば、Address Resolution Protocol (ARP) を使用して、IP アドレスのいずれにもエントリーがないことを確認します。

      ```
      $ arp 10.35.1.19
      ```

      出力例

      ```
      10.35.1.19 (10.35.1.19) -- no entry
      ```

   c. ネットワーク環境の標準的な方法に従って、2 つの静的 IP アドレスを予約します。
   d. 今後の参照用にこれらの IP アドレスを記録します。
2. 以下の形式を使用して、OpenShift Container Platform REST API およびアプリケーションドメイン名の DNS エントリーを作成します。

   ```
   api.<cluster-name>.<base-domain>   <ip-address> 1
   *.apps.<cluster-name>.<base-domain>   <ip-address> 2
   ```

   1 **<cluster-name>**、**<base-domain>**、および **<ip-address>** には、クラスター名、ベースドメイン、および OpenShift Container Platform API の静的 IP アドレスを指定します。

   2 Ingress およびロードバランサー用に OpenShift Container Platform アプリケーションのクラスター名、ベースドメイン、および静的 IP アドレスを指定します。

   以下に例を示します。

   ```
   api.my-cluster.virtlab.example.com 10.35.1.19
   *.apps.my-cluster.virtlab.example.com 10.35.1.20
   ```

2.6. OPENSHIFT CONTAINER PLATFORM OPENSTACK クラスターの RHV への非セキュアモードでのインストール

デフォルトで、インストーラーは CA 証明書を作成し、確認を求めるプロンプトを出し、インストール時に使用する証明書を保存します。これは、手動で作成したりインストールしたりする必要はありません。

推奨されていませんが、OpenShift Container Platform を RHV に **非セキュアモード** でインストールして、この機能を上書きし、証明書の検証なしに OpenShift Container Platform をインストールすることができます。

警告

非セキュア モードでのインストールは推奨されていません。これにより、攻撃者が中間者 (Man-in-the-Middle) 攻撃を実行し、ネットワーク上の機密の認証情報を取得できる可能性が生じるためです。

手順

1. **~/.ovirt/ovirt-config.yaml** という名前のファイルを作成します。
2. 以下の内容を **ovirt-config.yaml** に追加します。

```
ovirt_url: https://ovirt.example.com/ovirt-engine/api (1)
ovirt_fqdn: ovirt.example.com (2)
ovirt_pem_url: ""
ovirt_username: ocpadmin@internal
ovirt_password: super-secret-password (3)
ovirt_insecure: true
```

(1) oVirt エンジンのホスト名またはアドレスを指定します。

(2) oVirt エンジンの完全修飾ドメイン名を指定します。

(3) oVirt エンジンの管理者パスワードを指定します。

3. インストーラーを実行します。

2.7. クラスターノードの SSH アクセス用のキーペアの生成

OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定できます。キーは、Ignition 設定ファイルを介して Red Hat Enterprise Linux CoreOS (RHCOS) ノードに渡され、ノードへの SSH アクセスを認証するために使用されます。このキーは各ノードの **core** ユーザーの **~/.ssh/authorized_keys** リストに追加され、パスワードなしの認証が可能になります。

キーがノードに渡されると、キーペアを使用して RHCOS ノードにユーザー **core** として SSH を実行できます。SSH 経由でノードにアクセスするには、秘密鍵のアイデンティティーをローカルユーザーの SSH で管理する必要があります。

インストールのデバッグまたは障害復旧を実行するためにクラスターノードに対して SSH を実行する場合は、インストールプロセスの間に SSH 公開鍵を指定する必要があります。**./openshift-install gather** コマンドでは、SSH 公開鍵がクラスターノードに配置されている必要もあります。

重要

障害復旧およびデバッグが必要な実稼働環境では、この手順を省略しないでください。

手順

1. クラスターノードへの認証に使用するローカルマシンに既存の SSH キーペアがない場合は、これを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ ssh-keygen -t ed25519 -N '' -f <path>/<file_name> 1
```

1 新しい SSH キーのパスとファイル名 (~/**.ssh/id_ed25519** など) を指定します。既存のキーペアがある場合は、公開鍵が ~/**.ssh** ディレクトリーにあることを確認します。

注記

FIPS で検証済みまたは進行中のモジュール (Modules in Process) 暗号ライブラリーを使用する OpenShift Container Platform クラスターを **x86_64**、**ppc64le**、および **s390x** アーキテクチャーにインストールする予定の場合は、**ed25519** アルゴリズムを使用するキーは作成しないでください。代わりに、**rsa** アルゴリズムまたは **ecdsa** アルゴリズムを使用するキーを作成します。

2. 公開 SSH キーを表示します。

```
$ cat <path>/<file_name>.pub
```

たとえば、次のコマンドを実行して ~/**.ssh/id_ed25519.pub** 公開鍵を表示します。

```
$ cat ~/.ssh/id_ed25519.pub
```

3. ローカルユーザーの SSH エージェントに SSH 秘密鍵 ID が追加されていない場合は、それを追加します。キーの SSH エージェント管理は、クラスターノードへのパスワードなしの SSH 認証、または **./openshift-install gather** コマンドを使用する場合は必要になります。

注記

一部のディストリビューションでは、~/**.ssh/id_rsa** および ~/**.ssh/id_dsa** などのデフォルトの SSH 秘密鍵のアイデンティティーは自動的に管理されます。

a. **ssh-agent** プロセスがローカルユーザーに対して実行されていない場合は、バックグラウンドタスクとして開始します。

```
$ eval "$(ssh-agent -s)"
```

出力例

```
Agent pid 31874
```

注記

クラスターが FIPS モードにある場合は、FIPS 準拠のアルゴリズムのみを使用して SSH キーを生成します。鍵は RSA または ECDSA のいずれかである必要があります。

4. SSH プライベートキーを **ssh-agent** に追加します。

```
$ ssh-add <path>/<file_name> (1)
```

(1) ~/**.ssh/id_ed25519** などの、SSH プライベートキーのパスおよびファイル名を指定します。

出力例

```
Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

次のステップ

- OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定します。

2.8. インストールプログラムの取得

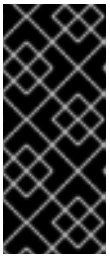
OpenShift Container Platform をインストールする前に、インストールに使用しているホストにインストールファイルをダウンロードします。

前提条件

- 500 MB のローカルディスク領域がある Linux または macOS を実行するコンピューターが必要です。

手順

1. OpenShift Cluster Manager サイトの インフラストラクチャープロバイダー ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使用してログインします。アカウントがない場合はこれを作成します。
2. インフラストラクチャープロバイダーを選択します。
3. インストールタイプのページに移動し、ホストオペレーティングシステムとアーキテクチャーに対応するインストールプログラムをダウンロードして、インストール設定ファイルを保存するディレクトリーにファイルを配置します。



重要

インストールプログラムは、クラスターのインストールに使用するコンピューターにいくつかのファイルを作成します。クラスターのインストール完了後は、インストールプログラムおよびインストールプログラムが作成するファイルを保持する必要があります。ファイルはいずれもクラスターを削除するために必要になります。

重要

インストールプログラムで作成されたファイルを削除しても、クラスターがインストール時に失敗した場合でもクラスターは削除されません。クラスターを削除するには、特定のクラウドプロバイダー用の OpenShift Container Platform のアンインストール手順を実行します。

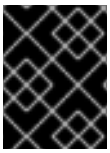
4. インストールプログラムを展開します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ tar -xvf openshift-install-linux.tar.gz
```

5. Red Hat OpenShift Cluster Manager からインストールプルシークレット をダウンロードします。このプルシークレットを使用し、OpenShift Container Platform コンポーネントのコンテナーイメージを提供する Quay.io など、組み込まれた各種の認証局によって提供されるサービスで認証できます。

2.9. クラスターのデプロイ

互換性のあるクラウドプラットフォームに OpenShift Container Platform をインストールできます。



重要

インストールプログラムの **create cluster** コマンドは、初期インストール時に 1 回だけ実行できます。

前提条件

- インストーラーを実行するマシンから **ovirt-imageio** ポートを Manager へのポートを開放する。デフォルトでは、ポートは **54322** です。
- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得する。
- ホスト上のクラウドプロバイダーアカウントに、クラスターをデプロイするための適切な権限があることを確認してください。アカウントの権限が正しくないと、インストールプロセスが失敗し、不足している権限を示すエラーメッセージが表示されます。

手順

1. インストールプログラムが含まれるディレクトリーに切り替え、クラスターのデプロイメントを初期化します。

```
$ ./openshift-install create cluster --dir <installation_directory> \ 1
    --log-level=info 2
```

1 **<installation_directory>** の場合、インストールプログラムが作成するファイルを保存するためにディレクトリー名を指定します。

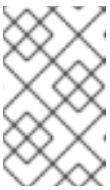
2 異なるインストールの詳細情報を表示するには、**info** ではなく、**warn**、**debug**、または **error** を指定します。

ディレクトリーを指定する場合:

- ディレクトリーに **execute** 権限があることを確認します。この権限は、インストールディレクトリーで Terraform バイナリーを実行するために必要です。
- 空のディレクトリーを使用します。ブートストラップ X.509 証明書などの一部のインストールアセットは有効期限が短いため、インストールディレクトリーを再利用しないでください。別のクラスターインストールの個別のファイルを再利用する必要がある場合は、それらをディレクトリーにコピーすることができます。ただし、インストールアセットのファイル名はリリース間で変更される可能性があります。インストールファイルを以前のバージョンの OpenShift Container Platform からコピーする場合は注意してコピーを行ってください。

2. インストールプログラムのプロンプトに対応します。

   a. オプション: **SSH Public Key** には、 パスワードなしのパブリックキー (例: **~/.ssh/id_rsa.pub**) を選択します。このキーは、新規 OpenShift Container Platform クラスターとの接続を認証します。

   注記

   インストールのデバッグまたは障害復旧を実行する必要のある実稼働用の OpenShift Container Platform クラスターには、**ssh-agent** プロセスが使用する SSH キーを選択します。

   b. **Platform** には、**ovirt** を選択します。

   c. **Engine FQDN[:PORT]** に、RHV 環境の完全修飾ドメイン名 (FQDN) を入力します。以下に例を示します。

   ```
   rhv-env.virtlab.example.com:443
   ```

   d. インストールプログラムは CA 証明書を自動的に生成します。**Would you like to use the above certificate to connect to the Manager?** では、**y** または **N** のいずれかで回答します。**N** と回答する場合は、OpenShift Container Platform を非セキュアモードでインストールする必要があります。

   e. **Engine username** には、この形式を使用して RHV 管理者のユーザー名およびプロファイルを入力します。

   ```
   <username>@<profile> 1
   ```

   1 **<username>** に、RHV 管理者のユーザー名を指定します。**<profile>** には、ログインプロファイルを指定します。ログインプロファイルは、RHV Administration Portal ログインページに移動し、 **Profile** ドロップダウンリストで確認できます。例: **admin@internal**

f. **Engine password** に、RHV 管理者パスワードを入力します。

g. **Cluster** には、OpenShift Container Platform をインストールするための RHV クラスターを選択します。

h. **Storage domain** には、OpenShift Container Platform をインストールするためのストレージドメインを選択します。

i. **Network** には、RHV Manager REST API へのアクセスのある仮想ネットワークを選択します。

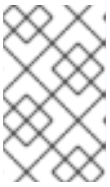
j. **Internal API Virtual IP** に、クラスターの REST API とは別の静的 IP アドレスを入力します。

k. **Ingress virtual IP** に、ワイルドカードアプリドメイン用に予約した静的 IP アドレスを入力します。

l. **Base Domain** に、OpenShift Container Platform クラスターのベースドメインを入力します。このクラスターが外部に公開される場合、これは DNS インフラストラクチャーが認識する有効なドメインである必要があります。たとえば、**virtlab.example.com** を入力します。

m. **Cluster Name** に、クラスターの名前を入力します。例: **my-cluster**OpenShift Container Platform REST API およびアプリケーションドメイン名向けに作成した外部登録/解決可能な DNS エントリーのクラスター名を使用します。インストールプログラムは、この名前を RHV 環境のクラスターにも指定します。

n. **Pull secret** には、先にダウンロードした **pull-secret.txt** ファイルからプルシークレットをコピーし、ここに貼り付けます。Red Hat OpenShift Cluster Manager から同じプルシークレット のコピーを取得することもできます。



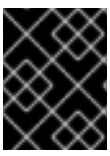
注記

ホストに設定したクラウドプロバイダーアカウントにクラスターをデプロイするための十分なパーミッションがない場合、インストールプロセスは停止し、不足しているパーミッションが表示されます。

検証

クラスターのデプロイが正常に完了すると、次のようになります。

- ターミナルには、Web コンソールへのリンクや **kubeadmin** ユーザーの認証情報など、クラスターにアクセスするための指示が表示されます。
- 認証情報は **<installation_directory>/.openshift_install.log** にも出力されます。



重要

インストールプログラム、またはインストールプログラムが作成するファイルを削除することはできません。これらはいずれもクラスターを削除するために必要になります。

出力例

```
...
INFO Install complete!
```

```
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/myuser/install_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.mycluster.example.com
INFO Login to the console with user: "kubeadmin", and password: "password"
INFO Time elapsed: 36m22s
```

重要

- インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになり、その後に更新される証明書が含まれます。証明書を更新する前にクラスターが停止し、24 時間経過した後にクラスターを再起動すると、クラスターは期限切れの証明書を自動的に復元します。例外として、kubelet 証明書を回復するために保留状態の **node-bootstrapper** 証明書署名要求 (CSR) を手動で承認する必要があります。詳細は、**コントロールプレーン証明書の期限切れの状態からのリカバリー** に関するドキュメントを参照してください。
- 24 時間証明書はクラスターのインストール後 16 時間から 22 時間にローテーションするため、Ignition 設定ファイルは、生成後 12 時間以内に使用することを推奨します。12 時間以内に Ignition 設定ファイルを使用することにより、インストール中に証明書の更新が実行された場合のインストールの失敗を回避できます。

重要

クラスターのインストールに必要な手順を完了している必要があります。残りの手順では、クラスターを検証し、インストールのトラブルシューティングを行う方法を説明します。

2.10. バイナリーのダウンロードによる OPENSHIFT CLI のインストール

コマンドラインインターフェイスを使用して OpenShift Container Platform と対話するために CLI (**oc**) をインストールすることができます。**oc** は Linux、Windows、または macOS にインストールできます。

重要

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.12 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールします。

Linux への OpenShift CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Linux にインストールできます。

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **Product Variant** ドロップダウンリストからアーキテクチャーを選択します。
3. **バージョン** ドロップダウンリストから適切なバージョンを選択します。

4. **OpenShift v4.12 Linux Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。
5. アーカイブを展開します。

```
$ tar xvf <file>
```

6. **oc** バイナリーを、**PATH** にあるディレクトリーに配置します。
PATH を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

検証

- OpenShift CLI のインストール後に、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

Windows への OpenShift CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Windows にインストールできます。

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
3. **OpenShift v4.12 Windows Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。
4. ZIP プログラムでアーカイブを解凍します。
5. **oc** バイナリーを、**PATH** にあるディレクトリーに移動します。
PATH を確認するには、コマンドプロンプトを開いて以下のコマンドを実行します。

```
C:\> path
```

検証

- OpenShift CLI のインストール後に、**oc** コマンドを使用して利用できます。

```
C:\> oc <command>
```

macOS への OpenShift CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを macOS にインストールできます。

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。

2. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
3. **OpenShift v4.12 macOS Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。

注記

macOS arm64 の場合は、**OpenShift v4.12 macOS arm64 Client** エントリーを選択します。

4. アーカイブを展開し、解凍します。
5. **oc** バイナリーをパスにあるディレクトリーに移動します。
PATH を確認するには、ターミナルを開き、以下のコマンドを実行します。

```
$ echo $PATH
```

検証

- OpenShift CLI のインストール後に、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

詳細は、OpenShift CLI の使用を開始 する を参照してください。

2.11. CLI の使用によるクラスターへのログイン

クラスター **kubeconfig** ファイルをエクスポートし、デフォルトシステムユーザーとしてクラスターにログインできます。**kubeconfig** ファイルには、クライアントを正しいクラスターおよび API サーバーに接続するために CLI で使用されるクラスターに関する情報が含まれます。このファイルはクラスターに固有のファイルであり、OpenShift Container Platform のインストール時に作成されます。

前提条件

- OpenShift Container Platform クラスターをデプロイしていること。
- **oc** CLI がインストールされている。

手順

1. **kubeadmin** 認証情報をエクスポートします。

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

1 **<installation_directory>** には、インストールファイルを保存したディレクトリーへのパスを指定します。

2. エクスポートされた設定を使用して、**oc** コマンドを正常に実行できることを確認します。

```
$ oc whoami
```

出力例

```
system:admin
```

関連情報

- OpenShift Container Platform Web コンソールへ のアクセスと理解の詳細については、Web コンソールへのアクセス を参照してください。

2.12. クラスターステータスの確認

インストール時またはインストール後に OpenShift Container Platform クラスターのステータスを確認することができます。

手順

1. クラスター環境で、管理者の kubeconfig ファイルをエクスポートします。

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

1 **<installation_directory>** には、インストールファイルを保存したディレクトリーへのパスを指定します。

kubeconfig ファイルには、クライアントを正しいクラスターおよび API サーバーに接続するために CLI で使用されるクラスターに関する情報が含まれます。

2. デプロイメント後に作成されたコントロールプレーンおよびコンピュートマシンを表示します。

```
$ oc get nodes
```

3. クラスターのバージョンを表示します。

```
$ oc get clusterversion
```

4. Operator のステータスを表示します。

```
$ oc get clusteroperator
```

5. クラスター内のすべての実行中の Pod を表示します。

```
$ oc get pods -A
```

トラブルシューティング

インストールが失敗すると、インストールプログラムがタイムアウトし、エラーメッセージが表示されます。詳細は、インストール問題のトラブルシューティング を参照してください。

2.13. RHV での OPENSHIFT CONTAINER PLATFORM WEB コンソールへのアクセス

OpenShift Container Platform クラスターの初期化後に、OpenShift Container Platform Web コンソールにログインできます。

手順

1. オプション: Red Hat Virtualization (RHV) Administration Portal で、**Compute → Cluster** を開きます。
2. インストールプログラムが仮想マシンを作成することを確認します。
3. インストールプログラムが実行されているコマンドラインに戻ります。インストールプログラムが完了すると、OpenShift Container Platform Web コンソールにログインするためのユーザー名およびパスワードの一時パスワードが表示されます。
4. ブラウザーから OpenShift Container Platform の Web コンソールの URL を開きます。URL は以下の形式を使用します。

```
console-openshift-console.apps.<clustername>.<basedomain> 1
```

1 **<clustername>.<basedomain>** に、クラスター名およびベースドメインを指定します。

以下に例を示します。

```
console-openshift-console.apps.my-cluster.virtlab.example.com
```

2.14. OPENSHIFT CONTAINER PLATFORM の TELEMETRY アクセス

OpenShift Container Platform 4.12 では、クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは OpenShift Cluster Manager Hybrid Cloud Console に登録されます。

OpenShift Cluster Manager インベントリーが正常である (Telemetry によって自動的に維持、または OpenShift Cluster Manager Hybrid Cloud Console を使用して手動で維持) ことを確認した後に、subscription watch を使用 して、アカウントまたはマルチクラスターレベルで OpenShift Container Platform サブスクリプションを追跡します。

関連情報

- Telemetry サービスの詳細は、リモートヘルスモニタリング を参照して ください。

2.15. RED HAT VIRTUALIZATION (RHV) へのインストールに関するよくある問題のトラブルシューティング

以下に、一般的な問題およびそれらについて考えられる原因および解決策を記載します。

2.15.1. CPU 負荷が増大し、ノードが Not Ready 状態になる

- **現象**: CPU 負荷が大幅に増大し、ノードが **Not Ready** 状態に切り替わり始める。
- **原因**: ストレージドメインのレイテンシーが高すぎる可能性があります (特にコントロールプレーンノードの場合)。

- **解決策**:
  Kubelet サービスを再起動して、ノードを再度 Ready 状態にします。

  ```
  $ systemctl restart kubelet
  ```

  OpenShift Container Platform メトリックサービスを検査します。これは、etcd ディスクの同期期間などの有用なデータを収集し、これについて報告します。クラスターが機能している場合は、このデータを使用して、ストレージのレイテンシーまたはスループットが根本的な問題かどうかを判断します。その場合、レイテンシーが短く、スループットの高いストレージリソースの使用を検討してください。

  未加工メトリックを取得するには、kubeadmin または cluster-admin 権限を持つユーザーで以下のコマンドを実行します。

  ```
  $ oc get --insecure-skip-tls-verify --server=https://localhost:<port> --raw=/metrics
  ```

  詳細は、Exploring Application Endpoints for the purposes of Debugging with OpenShift 4.x を参照してください。

2.15.2. OpenShift Container Platform クラスター API に接続できない

- **現象**: インストールプログラムは完了するが、OpenShift Container Platform クラスター API は利用できない。ブートストラップの仮想マシンは、ブートストラッププロセスの完了後も起動した状態になります。以下のコマンドを入力すると、応答がタイムアウトします。

  ```
  $ oc login -u kubeadmin -p *** <apiurl>
  ```

- **原因**: ブートストラップ仮想マシンがインストールプログラムによって削除されず、クラスターの API IP アドレスをリリースしない。

- **解決策**: **wait-for** サブコマンドを使用して、ブートストラッププロセスの完了時に通知を受信する。

  ```
  $ ./openshift-install wait-for bootstrap-complete
  ```

  ブートストラッププロセスが完了したら、ブートストラップ仮想マシンを削除します。

  ```
  $ ./openshift-install destroy bootstrap
  ```

2.16. インストール後のタスク

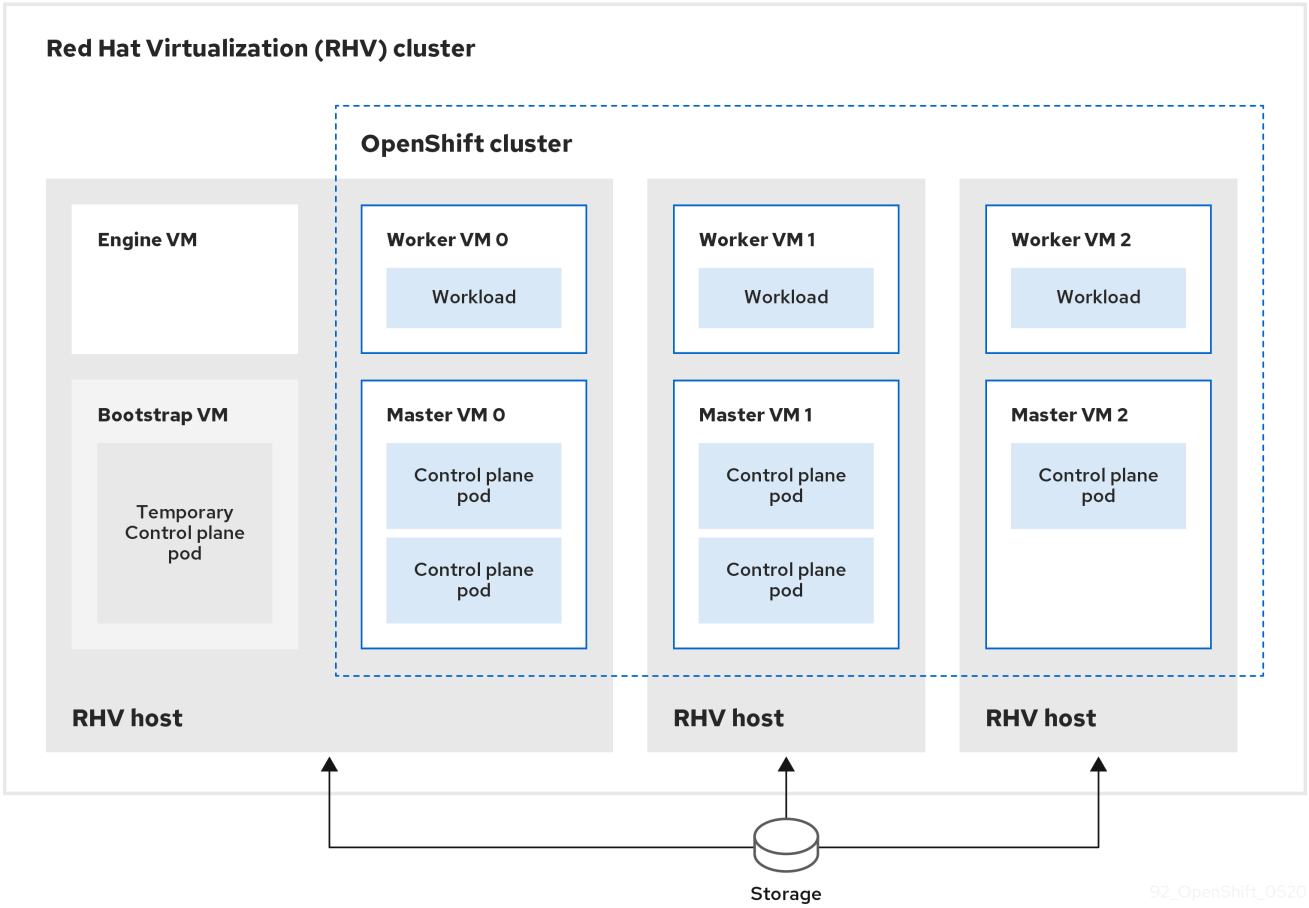
OpenShift Container Platform クラスターの初期化後に、以下のタスクを実行できます。

- オプション: デプロイメント後に、OpenShift Container Platform で Machine Config Operator (MCO) を使用して SSH キーを追加するか、置き換えます。

- オプション: **kubeadmin** ユーザーを削除します。代わりに、認証プロバイダーを使用して cluster-admin 権限を持つユーザーを作成します。

第3章 カスタマイズによる RHV へのクラスターのインストール

以下の図に示されるように、OpenShift Container Platform クラスターを Red Hat Virtualization (RHV) でカスタマイズし、インストールすることができます。



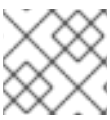
インストールプログラムは、インストーラーでプロビジョニングされるインフラストラクチャーを使用してクラスターの作成およびデプロイを自動化します。

カスタマイズされたクラスターをインストールするには、環境を準備し、以下の手順を実行します。

1. インストールプログラムを実行し、そのプロンプトに応答して、インストール設定ファイル **install-config.yaml** ファイルを作成します。
2. **install-config.yaml** ファイルでパラメーターを検査し、変更します。
3. **install-config.yaml** ファイルの作業用コピーを作成します。
4. **install-config.yaml** ファイルのコピーを使用してインストールプログラムを実行します。

次に、インストールプログラムは OpenShift Container Platform クラスターを作成します。

カスタマイズされたクラスターをインストールする代替方法については、デフォルトのクラスターのインストール を 参照してください。



注記

このインストールプログラムは、Linux および macOS でのみ利用できます。

3.1. 前提条件

- OpenShift Container Platform のインストールおよび更新 プロセスの詳細を確認している。
- Support Matrix for OpenShift Container Platform on Red Hat Virtualization (RHV) に記載のあるサポートされるバージョンの組み合わせを使用できる。
- クラスターインストール方法の選択およびそのユーザー向けの準備 を確認している。
- ファイアウォールを使用する場合は、クラスターがアクセスを必要とするサイトを許可するようにファイアウォールを設定する必要がある。

3.2. OPENSHIFT CONTAINER PLATFORM のインターネットアクセス

OpenShift Container Platform 4.12 では、クラスターをインストールするためにインターネットアクセスが必要になります。

インターネットへのアクセスは以下を実行するために必要です。

- OpenShift Cluster Manager Hybrid Cloud Console にアクセスし、インストールプログラムをダウンロードし、サブスクリプション管理を実行します。クラスターにインターネットアクセスがあり、Telemetry を無効にしない場合、そのサービスは有効なサブスクリプションでクラスターを自動的に使用します。
- クラスターのインストールに必要なパッケージを取得するために Quay.io にアクセスします。
- クラスターの更新を実行するために必要なパッケージを取得します。

重要

クラスターでインターネットに直接アクセスできない場合、プロビジョニングする一部のタイプのインフラストラクチャーでネットワークが制限されたインストールを実行できます。このプロセスで、必要なコンテンツをダウンロードし、これを使用してミラーレジストリーにインストールパッケージを設定します。インストールタイプに応じて、クラスターのインストール環境でインターネットアクセスが不要となる場合があります。クラスターを更新する前に、ミラーレジストリーのコンテンツを更新します。

3.3. RHV 環境の要件

OpenShift Container Platform バージョン 4.12 クラスターをインストールし、実行するには、RHV 環境が以下の要件を満たしている必要があります。

これらの要件を満たさないと、インストールまたはプロセスが失敗する可能性があります。さらに、これらの要件を満たしていないと、OpenShift Container Platform クラスターはインストールしてから数日または数週間後に失敗する可能性があります。

CPU、メモリー、ストレージリソースについての以下の要件は、インストールプログラムが作成する仮想マシンのデフォルト数で乗算した **デフォルト** 値に基づいています。これらのリソースは、RHV 環境が OpenShift Container Platform 以外の操作に使用するものに **加え**、利用可能でなければなりません。

デフォルトでは、インストールプログラムは 7 つの仮想マシンをインストールプロセスで作成します。まず、ブートストラップ仮想マシンを作成し、OpenShift Container Platform クラスターの残りの部分を作成する間に一時サービスとコントロールプレーンを提供します。インストールプログラムがクラスターの作成を終了すると、ブートストラップマシンが削除され、そのリソースが解放されます。

RHV 環境の仮想マシン数を増やす場合は、リソースを適宜増やす必要があります。

要件

- RHV のバージョンは 4.4 である。
- RHV 環境に **Up** 状態のデータセンターが 1 つあること。
- RHV データセンターに RHV クラスターが含まれていること。
- RHV クラスターに OpenShift Container Platform クラスター専用の以下のリソースがあること。
    - 最小 28 vCPU: インストール時に作成される 7 仮想マシンのそれぞれに 4 vCPU。
    - 以下を含む 112 GiB 以上の RAM。
        - 一時的なコントロールプレーンを提供するブートストラップマシン用に 16 GiB 以上。
        - コントロールプレーンを提供する 3 つのコントロールプレーンマシンのそれぞれに 16 GiB 以上。
        - アプリケーションワークロードを実行する 3 つのコンピュートマシンのそれぞれに 16 GiB 以上。
- RHV ストレージドメインは、これらの etcd バックエンドのパフォーマンス要件 を満たす必要があります。
- アフィニティーグループのサポートの場合:
  ワーカーまたはコントロールプレーンごとに 1 台の物理マシン。ワーカーとコントロールプレーンは、同じ物理マシン上に置くことができます。たとえば、3 つのワーカーと 3 つのコントロールプレーンがある場合、3 台の物理マシンが必要です。4 つのワーカーと 3 つのコントロールプレーンがある場合は、4 台の物理マシンが必要です。
    - 強い非アフィニティーの場合 (デフォルト): 最低 3 台の物理マシン。3 つを超えるワーカーノードの場合、ワーカーまたはコントロールプレーンごとに 1 台の物理マシン。ワーカーとコントロールプレーンは、同じ物理マシン上に置くことができます。
    - カスタムアフィニティーグループの場合: リソースが、定義するアフィニティーグループルールに適していることを確認します。
- 実稼働環境では、各仮想マシンに 120 GiB 以上が必要です。そのため、ストレージドメインはデフォルトの OpenShift Container Platform クラスターに 840 GiB 以上を提供する必要があります。リソースに制約のある環境または非実稼働環境では、各仮想マシンに 32 GiB 以上を指定する必要があるため、ストレージドメインにはデフォルトの OpenShift Container Platform クラスター用に 230 GiB 以上が必要になります。
- インストールおよび更新中に Red Hat Ecosystem Catalog からイメージをダウンロードするには、RHV クラスターがインターネット接続にアクセスできる必要があります。また、サブスクリプションおよびエンタイトルメントプロセスを単純化するために Telemetry サービスにもインターネット接続が必要です。
- RHV クラスターには、RHV Manager の REST API にアクセスできる仮想ネットワークが必要です。インストーラーが作成する仮想マシンが DHCP を使用して IP アドレスを取得するため、DHCP がこのネットワークで有効にされていることを確認します。

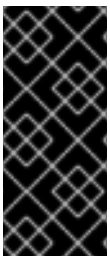
- ターゲット RHV クラスターに OpenShift Container Platform クラスターをインストールし、管理するための以下の最小限の権限を持つユーザーアカウントおよびグループ。
  - **DiskOperator**
  - **DiskCreator**
  - **UserTemplateBasedVm**
  - **TemplateOwner**
  - **TemplateCreator**
  - ターゲットクラスターの **ClusterAdmin**

警告

最小権限の原則を適用します。インストールプロセスで RHV で **SuperUser** 権限を持つ管理者アカウントを使用することを避けます。インストールプログラムは、ユーザーが指定する認証情報を、危険にさらされる可能性のある一時的な **ovirt-config.yaml** ファイルに保存します。

3.4. RHV 環境の要件の確認

RHV 環境が OpenShift Container Platform クラスターをインストールし、実行するための要件を満たしていることを確認します。これらの要件を満たさないと、エラーが発生する可能性があります。



重要

これらの要件は、インストールプログラムがコントロールプレーンおよびコンピュートマシンの作成に使用するデフォルトのリソースに基づいています。これらのリソースには、vCPU、メモリー、およびストレージが含まれます。これらのリソースを変更するか、OpenShift Container Platform マシンの数を増やす場合は、これらの要件を適宜調整します。

手順

1. RHV バージョンが OpenShift Container Platform バージョン 4.12 のインストールをサポートしていることを確認します。
   a. RHV Administration Portal の右上にある ? ヘルプアイコンをクリックし、**About** を選択します。
   b. 開かれるウィンドウで、**RHV ソフトウェアのバージョン** をメモします。
   c. RHV のバージョンが 4.4 であることを確認します。サポートされるバージョンの組み合わせについての詳細は、Support Matrix for OpenShift Container Platform on RHV を参照してください。
2. データセンター、クラスター、およびストレージを検査します。

a. RHV 管理ポータルで、**Compute → Data Centers** をクリックします。

b. OpenShift Container Platform をインストールする予定のデータセンターにアクセスできることを確認します。

c. そのデータセンターの名前をクリックします。

d. データセンターの詳細の **Storage** タブで、OpenShift Container Platform をインストールする予定のストレージドメインが **Active** であることを確認します。

e. 後で使用できるように **ドメイン名** を記録します。

f. **空き領域** に 230 GiB 以上あることを確認します。

g. ストレージドメインが これらの etcd バックエンドのパフォーマンス要件 を満たしていることを確認します。これは、fio パフォーマンスベンチマークツールを使用して測定できます。

h. データセンターの詳細で、**Clusters** タブをクリックします。

i. OpenShift Container Platform をインストールする予定の RHV クラスターを見つけます。後で使用できるようにクラスター名を記録します。

3. RHV ホストリソースを確認します。

a. RHV 管理ポータルで、**Compute > Clusters** をクリックします。

b. OpenShift Container Platform をインストールする予定のクラスターをクリックします。

c. クラスターの詳細で、**Hosts** タブをクリックします。

d. ホストを検査し、それらに OpenShift Container Platform クラスター **専用** として利用可能な **論理 CPU コア** の合計が 28 つ以上であることを確認します。

e. 後で使用できるように、利用可能な **論理 CPU コア** の数を記録します。

f. これらの CPU コアが分散され、インストール時に作成された 7 つの仮想マシンのそれぞれに 4 つのコアを持たせることができることを確認します。

g. ホストには、以下の OpenShift Container Platform マシンのそれぞれの要件を満たすように **新規仮想マシンをスケジュールするための最大空きメモリー** として 112 GiB があることを確認します。

- ブートストラップマシンに 16 GiB が必要です。
- 3 つのコントロールプレーンマシンのそれぞれに 16 GiB が必要です。
- 3 つのコンピュートマシンのそれぞれに 16 GiB が必要です。

h. 後で使用できるように **新規仮想マシンをスケジュールするための最大空きメモリー** の量を記録します。

4. OpenShift Container Platform をインストールするための仮想ネットワークが RHV Manager の REST API にアクセスできることを確認します。このネットワーク上の仮想マシンから、RHV Manager の REST API に到達するために curl を使用します。

```
$ curl -k -u <username>@<profile>:<password> \ 1
https://<engine-fqdn>/ovirt-engine/api 2
```

■

1. **<username>** については、RHV で OpenShift Container Platform クラスターを作成および管理する権限を持つ RHV アカウントのユーザー名を指定します。**<profile>** には、ログインプロファイルを指定します。ログインプロファイルは、RHV Administration Portal ログインページに移動し、 **Profile** ドロップダウンリストで確認できます。**<password>** に、そのユーザー名のパスワードを指定します。

2. **<engine-fqdn>** に、RHV 環境の完全修飾ドメイン名を指定します。

以下に例を示します。

```
$ curl -k -u ocpadmin@internal:pw123 \
https://rhv-env.virtlab.example.com/ovirt-engine/api
```

3.5. RHV でのネットワーク環境の準備

OpenShift Container Platform クラスターの 2 つの静的 IP アドレスを設定し、これらのアドレスを使用して DNS エントリーを作成します。

手順

1. 2 つの静的 IP アドレスを予約します。
   a. OpenShift Container Platform をインストールするネットワークで、DHCP リースプール外にある 2 つの静的 IP アドレスを特定します。
   b. このネットワーク上のホストに接続し、それぞれの IP アドレスが使用されていないことを確認します。たとえば、Address Resolution Protocol (ARP) を使用して、IP アドレスのいずれにもエントリーがないことを確認します。

      ```
      $ arp 10.35.1.19
      ```

      出力例

      ```
      10.35.1.19 (10.35.1.19) -- no entry
      ```

   c. ネットワーク環境の標準的な方法に従って、2 つの静的 IP アドレスを予約します。
   d. 今後の参照用にこれらの IP アドレスを記録します。
2. 以下の形式を使用して、OpenShift Container Platform REST API およびアプリケーションドメイン名の DNS エントリーを作成します。

```
api.<cluster-name>.<base-domain>   <ip-address> 1
*.apps.<cluster-name>.<base-domain>   <ip-address> 2
```

1. **<cluster-name>**、**<base-domain>**、および **<ip-address>** には、クラスター名、ベースドメイン、および OpenShift Container Platform API の静的 IP アドレスを指定します。

2. Ingress およびロードバランサー用に OpenShift Container Platform アプリケーションのクラスター名、ベースドメイン、および静的 IP アドレスを指定します。

以下に例を示します。

```
api.my-cluster.virtlab.example.com 10.35.1.19
*.apps.my-cluster.virtlab.example.com 10.35.1.20
```

3.6. OPENSHIFT CONTAINER PLATFORM OPENSTACK クラスターの RHV への非セキュアモードでのインストール

デフォルトで、インストーラーは CA 証明書を作成し、確認を求めるプロンプトを出し、インストール時に使用する証明書を保存します。これは、手動で作成したりインストールしたりする必要はありません。

推奨されていませんが、OpenShift Container Platform を RHV に **非セキュアモード** でインストールして、この機能を上書きし、証明書の検証なしに OpenShift Container Platform をインストールすることができます。

警告

非セキュア モードでのインストールは推奨されていません。これにより、攻撃者が中間者 (Man-in-the-Middle) 攻撃を実行し、ネットワーク上の機密の認証情報を取得できる可能性が生じるためです。

手順

1. **~/.ovirt/ovirt-config.yaml** という名前のファイルを作成します。
2. 以下の内容を **ovirt-config.yaml** に追加します。

```
ovirt_url: https://ovirt.example.com/ovirt-engine/api (1)
ovirt_fqdn: ovirt.example.com (2)
ovirt_pem_url: ""
ovirt_username: ocpadmin@internal
ovirt_password: super-secret-password (3)
ovirt_insecure: true
```

(1) oVirt エンジンのホスト名またはアドレスを指定します。

(2) oVirt エンジンの完全修飾ドメイン名を指定します。

(3) oVirt エンジンの管理者パスワードを指定します。

3. インストーラーを実行します。

3.7. クラスターノードの SSH アクセス用のキーペアの生成

OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定できます。キーは、Ignition 設定ファイルを介して Red Hat Enterprise Linux CoreOS

(RHCOS) ノードに渡され、ノードへの SSH アクセスを認証するために使用されます。このキーは各ノードの **core** ユーザーの **~/.ssh/authorized_keys** リストに追加され、パスワードなしの認証が可能になります。

キーがノードに渡されると、キーペアを使用して RHCOS ノードにユーザー **core** として SSH を実行できます。SSH 経由でノードにアクセスするには、秘密鍵のアイデンティティーをローカルユーザーの SSH で管理する必要があります。

インストールのデバッグまたは障害復旧を実行するためにクラスターノードに対して SSH を実行する場合は、インストールプロセスの間に SSH 公開鍵を指定する必要があります。**./openshift-install gather** コマンドでは、SSH 公開鍵がクラスターノードに配置されている必要もあります。

重要

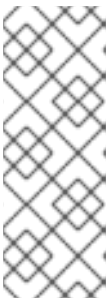
障害復旧およびデバッグが必要な実稼働環境では、この手順を省略しないでください。

手順

1. クラスターノードへの認証に使用するローカルマシンに既存の SSH キーペアがない場合は、これを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ ssh-keygen -t ed25519 -N '' -f <path>/<file_name> 1
```

1 新しい SSH キーのパスとファイル名 (**~/.ssh/id_ed25519** など) を指定します。既存のキーペアがある場合は、公開鍵が **~/.ssh** ディレクトリーにあることを確認します。



注記

FIPS で検証済みまたは進行中のモジュール (Modules in Process) 暗号ライブラリーを使用する OpenShift Container Platform クラスターを **x86_64**、**ppc64le**、および **s390x** アーキテクチャーにインストールする予定の場合は、**ed25519** アルゴリズムを使用するキーは作成しないでください。代わりに、**rsa** アルゴリズムまたは **ecdsa** アルゴリズムを使用するキーを作成します。

2. 公開 SSH キーを表示します。

```
$ cat <path>/<file_name>.pub
```

たとえば、次のコマンドを実行して **~/.ssh/id_ed25519.pub** 公開鍵を表示します。

```
$ cat ~/.ssh/id_ed25519.pub
```

3. ローカルユーザーの SSH エージェントに SSH 秘密鍵 ID が追加されていない場合は、それを追加します。キーの SSH エージェント管理は、クラスターノードへのパスワードなしの SSH 認証、または **./openshift-install gather** コマンドを使用する場合は必要になります。

注記

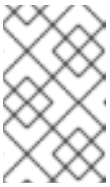
一部のディストリビューションでは、~/**.ssh/id_rsa** および ~/**.ssh/id_dsa** などのデフォルトの SSH 秘密鍵のアイデンティティーは自動的に管理されます。

a. **ssh-agent** プロセスがローカルユーザーに対して実行されていない場合は、バックグラウンドタスクとして開始します。

```
$ eval "$(ssh-agent -s)"
```

出力例

```
Agent pid 31874
```



注記

クラスターが FIPS モードにある場合は、FIPS 準拠のアルゴリズムのみを使用して SSH キーを生成します。鍵は RSA または ECDSA のいずれかである必要があります。

4. SSH プライベートキーを **ssh-agent** に追加します。

```
$ ssh-add <path>/<file_name> 1
```

1 ~/**.ssh/id_ed25519** などの、SSH プライベートキーのパスおよびファイル名を指定します。

出力例

```
Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

次のステップ

- OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定します。

3.8. インストールプログラムの取得

OpenShift Container Platform をインストールする前に、インストールに使用しているホストにインストールファイルをダウンロードします。

前提条件

- 500 MB のローカルディスク領域がある Linux または macOS を実行するコンピューターが必要です。

手順

1. OpenShift Cluster Manager サイトの インフラストラクチャープロバイダー ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使用してログインします。アカウントがない場合はこれを作成します。
2. インフラストラクチャープロバイダーを選択します。
3. インストールタイプのページに移動し、ホストオペレーティングシステムとアーキテクチャーに対応するインストールプログラムをダウンロードして、インストール設定ファイルを保存するディレクトリーにファイルを配置します。

重要

インストールプログラムは、クラスターのインストールに使用するコンピューターにいくつかのファイルを作成します。クラスターのインストール完了後は、インストールプログラムおよびインストールプログラムが作成するファイルを保持する必要があります。ファイルはいずれもクラスターを削除するために必要になります。

重要

インストールプログラムで作成されたファイルを削除しても、クラスターがインストール時に失敗した場合でもクラスターは削除されません。クラスターを削除するには、特定のクラウドプロバイダー用の OpenShift Container Platform のアンインストール手順を実行します。

4. インストールプログラムを展開します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ tar -xvf openshift-install-linux.tar.gz
```

5. Red Hat OpenShift Cluster Manager からインストールプルシークレット をダウンロードします。このプルシークレットを使用し、OpenShift Container Platform コンポーネントのコンテナーイメージを提供する Quay.io など、組み込まれた各種の認証局によって提供されるサービスで認証できます。

3.9. インストール設定ファイルの作成

Red Hat Virtualization (RHV) にインストールする OpenShift Container Platform クラスターをカスタマイズできます。

前提条件

- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得する。
- サブスクリプションレベルでサービスプリンシパルのパーミッションを取得する。

手順

1. **install-config.yaml** ファイルを作成します。
   a. インストールプログラムが含まれるディレクトリーに切り替え、以下のコマンドを実行します。

```
$ ./openshift-install create install-config --dir <installation_directory> 1
```

1 **<installation_directory>** の場合、インストールプログラムが作成するファイルを保存するためにディレクトリー名を指定します。

ディレクトリーを指定する場合:

- ディレクトリーに **execute** 権限があることを確認します。この権限は、インストールディレクトリーで Terraform バイナリーを実行するために必要です。
- 空のディレクトリーを使用します。ブートストラップ X.509 証明書などの一部のインストールアセットは有効期限が短いため、インストールディレクトリーを再利用しないでください。別のクラスターインストールの個別のファイルを再利用する必要がある場合は、それらをディレクトリーにコピーすることができます。ただし、インストールアセットのファイル名はリリース間で変更される可能性があります。インストールファイルを以前のバージョンの OpenShift Container Platform からコピーする場合は注意してコピーを行ってください。

b. インストールプログラムのプロンプトに対応します。

i. **SSH Public Key** では、パスワードなしのパブリックキー (例: **~/.ssh/id_rsa.pub**) を選択します。このキーは、新規 OpenShift Container Platform クラスターとの接続を認証します。

注記

インストールのデバッグまたは障害復旧を実行する必要のある実稼働用の OpenShift Container Platform クラスターには、**ssh-agent** プロセスが使用する SSH キーを選択します。

ii. **Platform** には、**ovirt** を選択します。

iii. **Enter oVirt's API endpoint URL** に、この形式を使用して RHV API の URL を入力します。

```
https://<engine-fqdn>/ovirt-engine/api 1
```

1 **<engine-fqdn>** に、RHV 環境の完全修飾ドメイン名を指定します。

以下に例を示します。

```
$ curl -k -u ocpadmin@internal:pw123 \
https://rhv-env.virtlab.example.com/ovirt-engine/api
```

iv. **Is the oVirt CA trusted locally?** には、CA 証明書がすでに設定されているため **Yes** を入力します。そうでない場合は、**No** と入力します。

v. **oVirt's CA bundle** には、前の質問で **Yes** を入力している場合には、 **/etc/pki/ca-trust/source/anchors/ca.pem** の内容をコピーし、ここに貼り付けます。その後、**Enter** を 2 回押します。そうでない場合、つまり、前の質問で **No** と入力している場合は、この質問は表示されません。

vi. **oVirt engine username** には、この形式を使用して RHV 管理者のユーザー名およびプロファイルを入力します。

```
<username>@<profile> 1
```

1 **<username>** に、RHV 管理者のユーザー名を指定します。**<profile>** には、ログインプロファイルを指定します。ログインプロファイルは、RHV Administration Portal ログインページに移動し、 **Profile** ドロップダウンリストで確認できます。ユーザー名とプロファイルは以下のようになります。

```
ocpadmin@internal
```

vii. **oVirt engine password** に、RHV 管理者パスワードを入力します。

viii. **oVirt cluster** には、OpenShift Container Platform をインストールするためのクラスターを選択します。

ix. **oVirt storage domain** には、OpenShift Container Platform をインストールするためのストレージドメインを選択します。

x. **oVirt network** には、RHV Manager REST API へのアクセスのある仮想ネットワークを選択します。

xi. **Internal API Virtual IP** に、クラスターの REST API とは別の静的 IP アドレスを入力します。

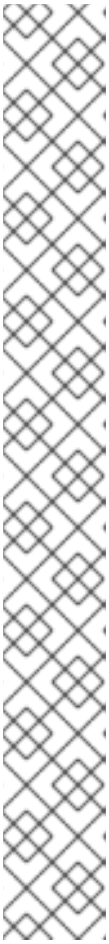
xii. **Ingress virtual IP** に、ワイルドカードアプリドメイン用に予約した静的 IP アドレスを入力します。

xiii. **Base Domain** に、OpenShift Container Platform クラスターのベースドメインを入力します。このクラスターが外部に公開される場合、これは DNS インフラストラクチャーが認識する有効なドメインである必要があります。たとえば、**virtlab.example.com** を入力します。

xiv. **Cluster Name** に、クラスターの名前を入力します。例: **my-cluster**OpenShift Container Platform REST API およびアプリケーションドメイン名向けに作成した外部登録/解決可能な DNS エントリーのクラスター名を使用します。インストールプログラムは、この名前を RHV 環境のクラスターにも指定します。

xv. **Pull secret** には、先にダウンロードした **pull-secret.txt** ファイルからプルシークレットをコピーし、ここに貼り付けます。Red Hat OpenShift Cluster Manager から同じプルシークレット のコピーを取得することもできます。

2. **install-config.yaml** ファイルを変更します。利用可能なパラメーターの詳細は、インストール設定パラメーターのセクションを参照してください。



注記

Manager に中間 CA 証明書がある場合は、証明書が **ovirt-config.yaml** ファイルおよび **install-config.yaml** ファイルに表示されることを確認します。表示されない場合は、以下のように追加します。

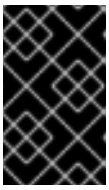
1. ~/**.ovirt**/**ovirt-config.yaml** ファイルの場合:

```
[ovirt_ca_bundle]: |
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA>
    -----END CERTIFICATE-----
    -----BEGIN CERTIFICATE-----
    <INTERMEDIATE_CA>
    -----END CERTIFICATE-----
```

2. **install-config.yaml** ファイルの場合:

```
[additionalTrustBundle]: |
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA>
    -----END CERTIFICATE-----
    -----BEGIN CERTIFICATE-----
    <INTERMEDIATE_CA>
    -----END CERTIFICATE-----
```

3. **install-config.yaml** ファイルをバックアップし、複数のクラスターをインストールするのに使用できるようにします。



重要

install-config.yaml ファイルはインストールプロセス時に使用されます。このファイルを再利用する必要がある場合は、この段階でこれをバックアップしてください。

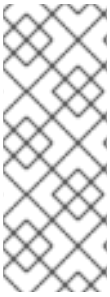
3.9.1. Red Hat Virtualization (RHV) のサンプル **install-config.yaml** ファイル

install-config.yaml ファイルのパラメーターおよびパラメーター値を変更して、インストールプログラムが作成する OpenShift Container Platform クラスターをカスタマイズできます。

以下は、RHV への OpenShift Container Platform のインストールに固有の例です。

install-config.yaml は、以下のコマンドを実行した際に指定した **<installation_directory>** にあります。

```
$ ./openshift-install create install-config --dir <installation_directory>
```



注記

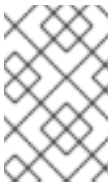
- これらのサンプルファイルは参照用にのみ提供されます。インストールプログラムを使用して **install-config.yaml** ファイルを取得する必要があります。
- **install-config.yaml** ファイルを変更すると、クラスターに必要なリソースを増やすことができます。RHV 環境にそれらの追加リソースがあることを確認します。これらがない場合は、インストールまたはクラスターが失敗します。

デフォルトの **install-config.yaml** ファイルの例

```
apiVersion: v1
baseDomain: example.com
compute:
- architecture: amd64
  hyperthreading: Enabled
  name: worker
  platform:
    ovirt:
      sparse: false 1
      format: raw 2
  replicas: 3
controlPlane:
  architecture: amd64
  hyperthreading: Enabled
  name: master
  platform:
      ovirt:
        sparse: false 3
        format: raw 4
  replicas: 3
metadata:
  creationTimestamp: null
  name: my-cluster
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  networkType: OVNKubernetes 5
  serviceNetwork:
  - 172.30.0.0/16
platform:
  ovirt:
    api_vips:
      - 10.0.0.10
    ingress_vips:
      - 10.0.0.11
    ovirt_cluster_id: 68833f9f-e89c-4891-b768-e2ba0815b76b
    ovirt_storage_domain_id: ed7b0f4e-0e96-492a-8fff-279213ee1468
    ovirt_network_name: ovirtmgmt
    vnicProfileID: 3fa86930-0be5-4052-b667-b79f0a729692
```

```
publish: External
pullSecret: '{"auths": ...}'
sshKey: ssh-ed12345 AAAA...
```

1 3 このオプションを **false** に設定すると、ディスクの事前割り当てが有効になります。デフォルトは **true** です。**format** を **raw** に設定して **sparse** を **true** に設定することは、ブロックストレージドメインでは使用できません。**raw** 形式は、仮想ディスク全体を基盤となる物理ディスクに書き込みます。

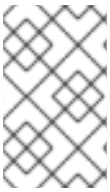


注記

ファイルストレージドメインにディスクを事前に割り当てると、ファイルにゼロが書き込まれます。基盤となるストレージによっては、実際にはディスクが事前に割り当てられない場合があります。

2 4 **cow** または **raw** に設定できます。デフォルトは **cow** です。**cow** のフォーマットは仮想マシン用に最適化されています。

5 インストールするクラスターネットワークプラグイン。サポートされている値は **OVNKubernetes** と **OpenShiftSDN** です。デフォルトの値は **OVNkubernetes** です。



注記

OpenShift Container Platform 4.12 以降では、**api_vip** および **ingress_vip** 設定は非推奨です。代わりに、リスト形式を使用して、**api_vips** および **ingress_vips** 設定に値を入力します。

最小の **install-config.yaml** ファイルの例

```
apiVersion: v1
baseDomain: example.com
metadata:
  name: test-cluster
platform:
  ovirt:
    api_vips:
      - 10.46.8.230
    ingress_vips:
      - 10.46.8.232
    ovirt_cluster_id: 68833f9f-e89c-4891-b768-e2ba0815b76b
    ovirt_storage_domain_id: ed7b0f4e-0e96-492a-8fff-279213ee1468
    ovirt_network_name: ovirtmgmt
    vnicProfileID: 3fa86930-0be5-4052-b667-b79f0a729692
pullSecret: '{"auths": ...}'
sshKey: ssh-ed12345 AAAA...
```

注記

OpenShift Container Platform 4.12 以降では、**api_vip** および **ingress_vip** 設定は非推奨です。代わりに、リスト形式を使用して、**api_vips** および **ingress_vips** 設定に値を入力します。

install-config.yaml ファイルのカスタムマシンプールの例

```
apiVersion: v1
baseDomain: example.com
controlPlane:
  name: master
  platform:
    ovirt:
      cpu:
        cores: 4
        sockets: 2
      memoryMB: 65536
      osDisk:
        sizeGB: 100
      vmType: server
  replicas: 3
compute:
- name: worker
  platform:
    ovirt:
      cpu:
        cores: 4
        sockets: 4
      memoryMB: 65536
      osDisk:
        sizeGB: 200
      vmType: server
  replicas: 5
metadata:
  name: test-cluster
platform:
  ovirt:
    api_vips:
      - 10.46.8.230
    ingress_vips:
      - 10.46.8.232
    ovirt_cluster_id: 68833f9f-e89c-4891-b768-e2ba0815b76b
    ovirt_storage_domain_id: ed7b0f4e-0e96-492a-8fff-279213ee1468
    ovirt_network_name: ovirtmgmt
    vnicProfileID: 3fa86930-0be5-4052-b667-b79f0a729692
pullSecret: '{"auths": ...}'
sshKey: ssh-ed25519 AAAA...
```

注記

OpenShift Container Platform 4.12 以降では、**api_vip** および **ingress_vip** 設定は非推奨です。代わりに、リスト形式を使用して、**api_vips** および **ingress_vips** 設定に値を入力します。

Enforcing 以外のアフィニティーグループの例
可能であれば、できるだけ多くのクラスターを使用するために、コントロールプレーンとワーカーを分散するために、enforcing 以外のアフィニティーグループを追加することを推奨します。

```
platform:
```

```
    ovirt:
      affinityGroups:
      - description: AffinityGroup to place each compute machine on a separate host
        enforcing: true
        name: compute
        priority: 3
      - description: AffinityGroup to place each control plane machine on a separate host
        enforcing: true
        name: controlplane
        priority: 5
      - description: AffinityGroup to place worker nodes and control plane nodes on separate hosts
        enforcing: false
        name: openshift
        priority: 5
  compute:
  - architecture: amd64
    hyperthreading: Enabled
    name: worker
    platform:
      ovirt:
        affinityGroupsNames:
        - compute
        - openshift
    replicas: 3
  controlPlane:
    architecture: amd64
    hyperthreading: Enabled
    name: master
    platform:
      ovirt:
        affinityGroupsNames:
        - controlplane
        - openshift
    replicas: 3
```

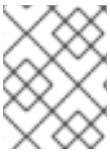
実稼働以外のラボセットアップのすべてのアフィニティーグループを削除する例
実稼働以外のラボセットアップでは、すべてのアフィニティーグループを削除して、OpenShift Container Platform クラスターをいくつかのホストに集中させる必要があります。

```
  platform:
    ovirt:
      affinityGroups: []
  compute:
  - architecture: amd64
    hyperthreading: Enabled
    name: worker
    platform:
      ovirt:
        affinityGroupsNames: []
    replicas: 3
  controlPlane:
    architecture: amd64
    hyperthreading: Enabled
    name: master
    platform:
```

```
    ovirt:
      affinityGroupsNames: []
  replicas: 3
```

3.9.2. インストール設定パラメーター

OpenShift Container Platform クラスターをデプロイする前に、クラスターをホストするクラウドプラットフォームでアカウントを記述し、クラスターのプラットフォームをオプションでカスタマイズするためにパラメーターの値を指定します。**install-config.yaml** インストール設定ファイルを作成する際に、コマンドラインで必要なパラメーターの値を指定します。クラスターをカスタマイズする場合、**install-config.yaml** ファイルを変更して、プラットフォームについての詳細情報を指定できます。



注記

インストール後は、これらのパラメーターを **install-config.yaml** ファイルで変更することはできません。

3.9.2.1. 必須設定パラメーター

必須のインストール設定パラメーターは、以下の表で説明されています。

表3.1 必須パラメーター

パラメーター	説明	値
apiVersion	**install-config.yaml** コンテンツの API バージョン。現在のバージョンは **v1** です。インストールプログラムは、古い API バージョンもサポートしている場合があります。	文字列
baseDomain	クラウドプロバイダーのベースドメイン。ベースドメインは、OpenShift Container Platform クラスターコンポーネントへのルートを作成するために使用されます。クラスターの完全な DNS 名は、**baseDomain** と **<metadata.name>.<baseDomain>** 形式を使用する **metadata.name** パラメーターの値の組み合わせです。	**example.com** などの完全修飾ドメインまたはサブドメイン名。
metadata	Kubernetes リソース **ObjectMeta**。ここからは **name** パラメーターのみが消費されます。	オブジェクト

パラメーター	説明	値
metadata.name	クラスターの名前。クラスターの DNS レコードはすべて **{{.metadata.name}}.{{.baseDomain}}** のサブドメインです。	**dev** などの小文字、ハイフン (**-**)、およびピリオド (**.**) が含まれる文字列。
platform	インストールを実行する特定のプラットフォームの設定: **alibabacloud**、**aws**、**baremetal**、**azure**、**gcp**、**ibmcloud**、**nutanix**、**openstack**、**ovirt**、**vsphere**、または **{}**。**platform.<platform>** パラメーターに関する追加情報は、以下の表で特定のプラットフォームを参照してください。	オブジェクト

pullSecret	Red Hat OpenShift Cluster Manager からプルシークレット を取得して、Quay.io などのサービスから OpenShift Container Platform コンポーネントのコンテナーイメージをダウンロードすることを認証します。	<pre>{
		  "auths":{
		  "cloud.openshift.com":{
		  "auth":"b3Blb=",
		  "email":"you@example.com"
		  },
		  "quay.io":{
		  "auth":"b3Blb=",
		  "email":"you@example.com"
		  }
		  }
		}</pre>

3.9.2.2. ネットワーク設定パラメーター

既存のネットワークインフラストラクチャーの要件に基づいて、インストール設定をカスタマイズできます。たとえば、クラスターネットワークの IP アドレスブロックを拡張するか、デフォルトとは異なる IP アドレスブロックを指定できます。

IPv4 アドレスのみがサポートされます。

注記

Globalnet は、Red Hat OpenShift Data Foundation ディザスターリカバリーソリューションではサポートされていません。局地的なディザスターリカバリーのシナリオでは、各クラスター内のクラスターとサービスネットワークに重複しない範囲のプライベート IP アドレスを使用するようにしてください。

表3.2 ネットワークパラメーター

パラメーター	説明	値
networking	クラスターのネットワークの設定。	オブジェクト **注記** インストール後に **networking** オブジェクトで指定したパラメーターを変更することはできません。
networking.networkType	インストールする Red Hat OpenShift Networking ネットワークプラグイン。	**OpenShiftSDN** または **OVNKubernetes** のいずれか。**OpenShiftSDN** は、全 Linux ネットワーク用の CNI プラグインです。**OVNKubernetes** は、Linux ネットワークと、Linux サーバーと Windows サーバーの両方を含む Linux ネットワークおよびハイブリッドネットワーク用の CNI プラグインです。デフォルトの値は **OVNkubernetes** です。
networking.clusterNetwork	Pod の IP アドレスブロック。 デフォルト値は **10.128.0.0/14** で、ホストの接頭辞は **/23** です。 複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。	オブジェクトの配列。以下に例を示します。 networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23
networking.clusterNetwork.cidr	**networking.clusterNetwork** を使用する場合に必須です。IP アドレスブロック。 IPv4 ネットワーク	CIDR (Classless Inter-Domain Routing) 表記の IP アドレスブロック。IPv4 ブロックの接頭辞長は **0** から **32** の間になります。
networking.clusterNetwork.hostPrefix	それぞれの個別ノードに割り当てるサブネット接頭辞長。たとえば、**hostPrefix** が **23** に設定される場合、各ノードに指定の **cidr** から **/23** サブネットが割り当てられます。**hostPrefix** 値の **23** は、510 (2^(32 - 23) - 2) Pod IP アドレスを提供します。	サブネット接頭辞。 デフォルト値は **23** です。

パラメーター	説明	値
networking.serviceNetwork	サービスの IP アドレスブロック。デフォルト値は **172.30.0.0/16** です。 OpenShift SDN および OVN-Kubernetes ネットワークプラグインは、サービスネットワークの単一 IP アドレスブロックのみをサポートします。	CIDR 形式の IP アドレスブロックを持つ配列。以下に例を示します。 networking: serviceNetwork: - 172.30.0.0/16
networking.machineNetwork	マシンの IP アドレスブロック。 複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。	オブジェクトの配列。以下に例を示します。 networking: machineNetwork: - cidr: 10.0.0.0/16
networking.machineNetwork.cidr	**networking.machineNetwork** を使用する場合に必須です。IP アドレスブロック。libvirt 以外のすべてのプラットフォームでは、デフォルト値は **10.0.0.0/16** です。libvirt の場合、デフォルト値は **192.168.126.0/24** です。	CIDR 表記の IP ネットワークブロック。 例: **10.0.0.0/16** **注記** 優先される NIC が置かれている CIDR に一致する **networking.machineNetwork** を設定します。

3.9.2.3. オプションの設定パラメーター

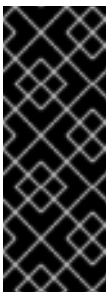
オプションのインストール設定パラメーターは、以下の表で説明されています。

表3.3 オプションのパラメーター

パラメーター	説明	値
additionalTrustBundle	ノードの信頼済み証明書ストアに追加される PEM でエンコードされた X.509 証明書バンドル。この信頼バンドルは、プロキシーが設定される際にも使用できます。	文字列

パラメーター	説明	値
capabilities	オプションのコアクラスターコンポーネントのインストールを制御します。オプションのコンポーネントを無効にすることで、OpenShift Container Platform クラスターのフットプリントを削減できます。詳細は、**インストール** の「クラスター機能ページ」を参照してください。	文字列配列
capabilities.baselineCapabilitySet	有効にするオプション機能の初期セットを選択します。有効な値は **None**、**v4.11**、**v4.12**、**vCurrent** です。デフォルト値は **vCurrent** です。	文字列
capabilities.additionalEnabledCapabilities	オプションの機能のセットを、**baselineCapabilitySet** で指定したものを超えて拡張します。このパラメーターで複数の機能を指定できます。	文字列配列
compute	コンピュートノードを設定するマシンの設定。	**MachinePool** オブジェクトの配列。詳細は、マシンプールの追加の RHV パラメーターの表を参照してください。
compute.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現在、さまざまなアーキテクチャーのクラスターはサポートされていません。すべてのプールは同じアーキテクチャーを指定する必要があります。有効な値は **amd64** (デフォルト) です。	文字列
compute.hyperthreading	コンピュートマシンで同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。  **重要** 同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。	**Enabled** または **Disabled**

パラメーター	説明	値
compute.name	**compute** を使用する場合に必須です。マシンプールの名前。	**worker**
compute.platform	**compute** を使用する場合に必須です。このパラメーターを使用して、ワーカーマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は **controlPlane.platform** パラメーターの値に一致する必要があります。	**alibabacloud**、**aws**、**azure**, **gcp**、**ibmcloud**、**nutanix**、**openstack**、**ovirt**、**vsphere**、または **{}**
compute.replicas	プロビジョニングするコンピュートマシン (ワーカーマシンとしても知られる) の数。	**2** 以上の正の整数。デフォルト値は **3** です。
featureSet	機能セットのクラスターを有効にします。機能セットは、デフォルトで有効にされない OpenShift Container Platform 機能のコレクションです。インストール中に機能セットを有効にする方法の詳細は、「機能ゲートの使用による各種機能の有効化」を参照してください。	文字列。**TechPreviewNoUpgrade** など、有効にする機能セットの名前。
controlPlane	コントロールプレーンを設定するマシンの設定。	**MachinePool** オブジェクトの配列。詳細は、マシンプールの追加の RHV パラメーターの表を参照してください。
controlPlane.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現在、さまざまなアーキテクチャーのクラスターはサポートされていません。すべてのプールは同じアーキテクチャーを指定する必要があります。有効な値は **amd64** (デフォルト) です。	文字列

パラメーター	説明	値
controlPlane.hyperthreading	コントロールプレーンマシンで同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。  **重要** 同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。	**Enabled** または **Disabled**
controlPlane.name	**controlPlane** を使用する場合に必須です。マシンプールの名前。	**master**
controlPlane.platform	**controlPlane** を使用する場合に必須です。このパラメーターを使用して、コントロールプレーンマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は **compute.platform** パラメーターの値に一致する必要があります。	**alibabacloud**、**aws**、**azure**, **gcp**、**ibmcloud**、**nutanix**、**openstack**、**ovirt**、**vsphere**、または **{}**
controlPlane.replicas	プロビジョニングするコントロールプレーンマシンの数。	サポートされる値は **3** のみです (これはデフォルト値です)。

パラメーター	説明	値
credentialsMode	Cloud Credential Operator (CCO) モード。モードを指定しないと、CCO は指定された認証情報の機能を動的に判別しようとします。この場合、複数のモードがサポートされるプラットフォームで Mint モードが優先されます。  **注記** すべてのクラウドプロバイダーですべての CCO モードがサポートされているわけではありません。CCO モードの詳細は、**Cluster Operators リファレンス** の **Cloud Credential Operator** を参照してください。  **注記** AWS アカウントでサービスコントロールポリシー (SCP) が有効になっている場合は、**credentialsMode** パラメーターを **Mint**、**Passthrough** または **Manual** に設定する必要があります。	**Mint**、**Passthrough**、**Manual**、または空の文字列 (**""**)。

パラメーター	説明	値
fips	FIPS モードを有効または無効にします。デフォルトは **false** (無効) です。FIPS モードが有効にされている場合、OpenShift Container Platform が実行される Red Hat Enterprise Linux CoreOS (RHCOS) マシンがデフォルトの Kubernetes 暗号スイートをバイパスし、代わりに RHCOS で提供される暗号モジュールを使用します。  **重要** クラスターで FIPS モードを有効にするには、FIPS モードで動作するように設定された Red Hat Enterprise Linux (RHEL) コンピューターからインストールプログラムを実行する必要があります。RHEL での FIPS モードの設定の詳細は、FIPS モードでのシステムのインストール を参照してください。FIPS 検証済み/Modules In Process 暗号ライブラリーの使用は、**x86_64**、**ppc64le**、および **s390x** アーキテクチャー上の OpenShift Container Platform デプロイメントでのみサポートされます。  **注記** Azure File ストレージを使用している場合、FIPS モードを有効にすることはできません。	**false** または **true**

パラメーター	説明	値
imageContentSources	release-image コンテンツのソースおよびリポジトリー。	オブジェクトの配列。この表の以下の行で説明されているように、**source** およびオプションで **mirrors** が含まれます。
imageContentSources.source	**imageContentSources** を使用する場合に必須です。ユーザーが参照するリポジトリーを指定します (例: イメージプル仕様)。	文字列
imageContentSources.mirrors	同じイメージが含まれる可能性のあるリポジトリーを 1 つ以上指定します。	文字列の配列。
publish	Kubernetes API、OpenShift ルートなどのクラスターのユーザーに表示されるエンドポイントをパブリッシュまたは公開する方法。	**Internal** または **External**。デフォルト値は **External** です。 このパラメーターを **Internal** に設定することは、クラウド以外のプラットフォームではサポートされません。  **重要** フィールドの値が **Internal** に設定されている場合、クラスターは機能しなくなります。詳細は、BZ#1953035 を参照してください。

パラメーター	説明	値
sshKey	クラスターマシンへのアクセスを認証するための SSH キー。  **注記** インストールのデバッグまたは障害復旧を実行する必要のある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。	たとえば、**sshKey: ssh-ed25519 AAAA..** です。

3.9.2.4. 追加の Red Hat Virtualization (RHV) 設定パラメーター

追加の RHV 設定パラメーターは以下の表で説明されています。

表3.4 クラスターの追加の Red Hat Virtualization (RHV) パラメーター

パラメーター	説明	値
platform.ovirt.ovirt_cluster_id	必須。仮想マシンが作成されるクラスター。	文字列。例: **68833f9f-e89c-4891-b768-e2ba0815b76b**
platform.ovirt.ovirt_storage_domain_id	必須。仮想マシンディスクが作成されるストレージドメイン ID。	文字列。例: **ed7b0f4e-0e96-492a-8fff-279213ee1468**
platform.ovirt.ovirt_network_name	必須。仮想マシン NIC が作成されるネットワーク名。	文字列。例: **ocpcluster**
platform.ovirt.vnicProfileID	必須。仮想マシンネットワークインターフェイスの vNIC プロファイル ID。これは、クラスターネットワークに単一のプロファイルがある場合に示唆されます。	文字列。例: **3fa86930-0be5-4052-b667-b79f0a729692**

パラメーター	説明	値
platform.ovirt.api_vips	必須。API 仮想 IP (VIP) に割り当てられるマシンネットワークの IP アドレス。このエンドポイントで OpenShift API にアクセスできます。デュアルスタックネットワークの場合、最大 2 つの IP アドレスを割り当てます。プライマリー IP アドレスは IPv4 ネットワークからのものである必要があります。  **注記** OpenShift Container Platform 4.12 以降では、**api_vip** 設定は非推奨です。代わりに、リスト形式を使用して **api_vips** 設定に値を入力してください。リストの順序は、各サービスのプライマリーおよびセカンダリー VIP アドレスを示しています。	文字列。例: **10.46.8.230**
platform.ovirt.ingress_vips	必須。Ingress 仮想 IP (VIP) に割り当てられるマシンネットワークの IP アドレス。デュアルスタックネットワークの場合、最大 2 つの IP アドレスを割り当てます。プライマリー IP アドレスは IPv4 ネットワークからのものである必要があります。  **注記** OpenShift Container Platform 4.12 以降では、**ingress_vip** 設定は非推奨です。代わりに、リスト形式を使用して **ingress_vips** 設定に値を入力してください。リストの順序は、各サービスのプライマリーおよびセカンダリー VIP アドレスを示しています。	文字列。例: **10.46.8.232**
platform.ovirt.affinityGroups	オプション。インストールプロセス中に作成するアフィニティーグループのリスト。	オブジェクトのリスト

パラメーター	説明	値
platform.ovirt.affinityGroups.description	**platform.ovirt.affinityGroups** を含める場合は必須です。アフィニティーグループのの説明	文字列。例: **AffinityGroup for spreading each compute machine to a different host**
platform.ovirt.affinityGroups.enforcing	**platform.ovirt.affinityGroups** を含める場合は必須です。**true** に設定すると、十分なハードウェアノードが使用できない場合、RHV はマシンをプロビジョニングしません。**false** に設定すると、十分なハードウェアノードが使用できない場合でも、RHV はマシンをプロビジョニングするため、複数の仮想マシンが同じ物理マシンでホストされます。	文字列。例: **true**
platform.ovirt.affinityGroups.name	**platform.ovirt.affinityGroups** を含める場合は必須です。アフィニティーグループの名前。	文字列。例: **compute**
platform.ovirt.affinityGroups.priority	**platform.ovirt.affinityGroups** を含める場合は必須です。**platform.ovirt.affinityGroups.enforcing = false** の場合に、アフィニティーグループに与えられる優先度。RHV は、優先順位の高い順にアフィニティーグループを適用します。この場合、小さい番号よりも大きい番号が優先されます。複数のアフィニティーグループの優先度が同じである場合、それらが適用される順序は保証されません。	integer例: **3**

3.9.2.5. マシンプールの追加 RHV パラメーター

マシンプールの追加の RHV 設定パラメーターは以下の表で説明されています。

表3.5 マシンプールの追加 RHV パラメーター

パラメーター	説明	値
<machine-pool>.platform.ovirt.cpu	オプション。仮想マシンの CPU を定義します。	オブジェクト

パラメーター	説明	値
<machine-pool>.platform.ovirt.cpu.cores	**<machine-pool>.platform.ovirt.cpu** を使用する場合に必須です。コア数。仮想 CPU (vCPU) の合計はコア * ソケットです。	整数
<machine-pool>.platform.ovirt.cpu.sockets	**<machine-pool>.platform.ovirt.cpu** を使用する場合に必須です。コアあたりのソケット数。仮想 CPU (vCPU) の合計はコア * ソケットです。	整数
<machine-pool>.platform.ovirt.memoryMB	オプション。仮想マシンのメモリー (MiB 単位)。	整数
<machine-pool>.platform.ovirt.osDisk	オプション。仮想マシンの起動可能な初回の、および起動可能なディスクを定義します。	文字列
<machine-pool>.platform.ovirt.osDisk.sizeGB	**<machine-pool>.platform.ovirt.osDisk** を使用する場合に必須です。ディスクのサイズ (GiB 単位)。	数字

パラメーター	説明	値
<machine-pool>.platform.ovirt.vmType	オプション。**high-performance**、**server**、 または **desktop** などの仮想マシンワークロードタイプ。デフォルトでは、コントロールプレーンノードは **high performance** を使用し、ワーカーノードは **server** を使用します。詳細は、**仮想マシン管理ガイド**の 仮想マシンの一般設定に関する説明 および ハイパフォーマンス仮想マシン、テンプレート、およびプールの設定 を参照してください。  **注記** **high_performance** により、仮想マシンのパフォーマンスが向上しますが、制限があります。たとえば、グラフィカルコンソールを使用して仮想マシンにはアクセスできません。詳細は、**Virtual Machine Management Guide**の ハイパフォーマンス仮想マシン、テンプレート、およびプールの設定 を参照してください。	文字列

パラメーター	説明	値

<machine-pool>.platform.ovirt.affinityGroupsNames	オプション。仮想マシンに適用する必要があるアフィニティーグループ名のリスト。アフィニティーグループは RHV に存在するか、このトピックの**クラスターの追加 RHV パラメーター**で説明されているように、インストール中に作成する必要があります。このエントリーは空にすることができます。
	
	2 つのアフィニティーグループの例
	
	この例では、**compute** および **clusterWideNonEnforcing** という名前の 2 つのアフィニティーグループを定義します。
	
	`<machine-pool>:`
	`  platform:`
	`  ovirt:`
	`  affinityGroupNames:`
	`  - compute`
	`  - clusterWideNonEnforcing`
	
	この例では、アフィニティーグループを定義していません。
	
	`<machine-pool>:`
	`  platform:`
	`  ovirt:`
	`  affinityGroupNames: []`
		文字列

<machine-pool>.platform.ovirt.AutoPinningPolicy	オプション。AutoPinningPolicy は、インスタンスのホストへのピニングを含む、CPU と NUMA 設定を自動的に設定するポリシーを定義します。フィールドを省略すると、デフォルトは **none** です。サポートされる値は、**none**、**resize_and_pin** です。詳細は、**Virtual Machine Management Guide**の Setting NUMA Nodes を参照してください。	文字列
<machine-pool>.platform.ovirt.hugepages	オプション。hugepages は、仮想マシンで hugepage を定義するためのサイズ (KiB) です。対応している値は **2048** および **1048576** です。詳細は、**Virtual Machine Management Guide**の Configuring Huge Pages を参照してください。	整数

注記

<machine-pool> を **controlPlane** または **compute** に置き換えることができます。

3.10. クラスターのデプロイ

互換性のあるクラウドプラットフォームに OpenShift Container Platform をインストールできます。

重要

インストールプログラムの **create cluster** コマンドは、初期インストール時に 1 回だけ実行できます。

前提条件

- インストーラーを実行するマシンから **ovirt-imageio** ポートを Manager へのポートを開放する。デフォルトでは、ポートは **54322** です。
- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得する。
- ホスト上のクラウドプロバイダーアカウントに、クラスターをデプロイするための適切な権限があることを確認してください。アカウントの権限が正しくないと、インストールプロセスが失敗し、不足している権限を示すエラーメッセージが表示されます。

手順

- インストールプログラムが含まれるディレクトリーに切り替え、クラスターのデプロイメントを初期化します。

```
$ ./openshift-install create cluster --dir <installation_directory> \ 1
    --log-level=info 2
```

1. **<installation_directory>** については、カスタマイズした **./install-config.yaml** ファイルの場所を指定します。
2. 異なるインストールの詳細情報を表示するには、**info** ではなく、**warn**、**debug**、または **error** を指定します。

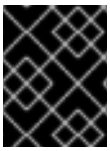
注記

ホストに設定したクラウドプロバイダーアカウントにクラスターをデプロイするための十分なパーミッションがない場合、インストールプロセスは停止し、不足しているパーミッションが表示されます。

検証

クラスターのデプロイが正常に完了すると、次のようになります。

- ターミナルには、Web コンソールへのリンクや **kubeadmin** ユーザーの認証情報など、クラスターにアクセスするための指示が表示されます。
- 認証情報は **<installation_directory>/.openshift_install.log** にも出力されます。

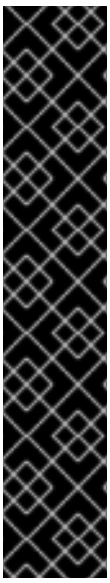


重要

インストールプログラム、またはインストールプログラムが作成するファイルを削除することはできません。これらはいずれもクラスターを削除するために必要になります。

出力例

```
...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/myuser/install_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.mycluster.example.com
INFO Login to the console with user: "kubeadmin", and password: "password"
INFO Time elapsed: 36m22s
```



重要

- インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになり、その後に更新される証明書が含まれます。証明書を更新する前にクラスターが停止し、24 時間経過した後にクラスターを再起動すると、クラスターは期限切れの証明書を自動的に復元します。例外として、kubelet 証明書を回復するために保留状態の **node-bootstrapper** 証明書署名要求 (CSR) を手動で承認する必要があります。詳細は、**コントロールプレーン証明書の期限切れの状態からのリカバリー** に関するドキュメントを参照してください。
- 24 時間証明書はクラスターのインストール後 16 時間から 22 時間にローテーションするため、Ignition 設定ファイルは、生成後 12 時間以内に使用することを推奨します。12 時間以内に Ignition 設定ファイルを使用することにより、インストール中に証明書の更新が実行された場合のインストールの失敗を回避できます。

重要

クラスターのインストールに必要な手順を完了している必要があります。残りの手順では、クラスターを検証し、インストールのトラブルシューティングを行う方法を説明します。

3.11. バイナリーのダウンロードによる OPENSHIFT CLI のインストール

コマンドラインインターフェイスを使用して OpenShift Container Platform と対話するために CLI (**oc**) をインストールすることができます。**oc** は Linux、Windows、または macOS にインストールできます。

重要

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.12 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールします。

Linux への OpenShift CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Linux にインストールできます。

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **Product Variant** ドロップダウンリストからアーキテクチャーを選択します。
3. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
4. **OpenShift v4.12 Linux Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。
5. アーカイブを展開します。

```
$ tar xvf <file>
```

6. **oc** バイナリーを、**PATH** にあるディレクトリーに配置します。
PATH を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

検証

- OpenShift CLI のインストール後に、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

Windows への OpenShift CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを Windows にインストールできます。

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
3. **OpenShift v4.12 Windows Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。
4. ZIP プログラムでアーカイブを解凍します。
5. **oc** バイナリーを、**PATH** にあるディレクトリーに移動します。
PATH を確認するには、コマンドプロンプトを開いて以下のコマンドを実行します。

```
C:\> path
```

検証

- OpenShift CLI のインストール後に、**oc** コマンドを使用して利用できます。

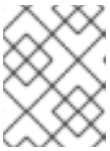
```
C:\> oc <command>
```

macOS への OpenShift CLI のインストール

以下の手順を使用して、OpenShift CLI (**oc**) バイナリーを macOS にインストールできます。

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
3. **OpenShift v4.12 macOS Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。



注記

macOS arm64 の場合は、**OpenShift v4.12 macOS arm64 Client** エントリーを選択します。

4. アーカイブを展開し、解凍します。
5. **oc** バイナリーをパスにあるディレクトリーに移動します。
PATH を確認するには、ターミナルを開き、以下のコマンドを実行します。

```
$ echo $PATH
```

検証

- OpenShift CLI のインストール後に、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

3.12. CLI の使用によるクラスターへのログイン

クラスター **kubeconfig** ファイルをエクスポートし、デフォルトシステムユーザーとしてクラスターにログインできます。**kubeconfig** ファイルには、クライアントを正しいクラスターおよび API サーバーに接続するために CLI で使用されるクラスターに関する情報が含まれます。このファイルはクラスターに固有のファイルであり、OpenShift Container Platform のインストール時に作成されます。

前提条件

- OpenShift Container Platform クラスターをデプロイしていること。
- **oc** CLI がインストールされている。

手順

1. **kubeadmin** 認証情報をエクスポートします。

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

1. **<installation_directory>** には、インストールファイルを保存したディレクトリーへのパスを指定します。

2. エクスポートされた設定を使用して、**oc** コマンドを正常に実行できることを確認します。

```
$ oc whoami
```

出力例

```
system:admin
```

詳細は、OpenShift CLI の使用を開始 する を参照してください。

3.13. クラスターステータスの確認

インストール時またはインストール後に OpenShift Container Platform クラスターのステータスを確認することができます。

手順

1. クラスター環境で、管理者の kubeconfig ファイルをエクスポートします。

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

1 **<installation_directory>** には、インストールファイルを保存したディレクトリーへのパスを指定します。

kubeconfig ファイルには、クライアントを正しいクラスターおよび API サーバーに接続するために CLI で使用されるクラスターに関する情報が含まれます。

2. デプロイメント後に作成されたコントロールプレーンおよびコンピュートマシンを表示します。

```
$ oc get nodes
```

3. クラスターのバージョンを表示します。

```
$ oc get clusterversion
```

4. Operator のステータスを表示します。

```
$ oc get clusteroperator
```

5. クラスター内のすべての実行中の Pod を表示します。

```
$ oc get pods -A
```

トラブルシューティング

インストールが失敗すると、インストールプログラムがタイムアウトし、エラーメッセージが表示されます。詳細は、インストール問題のトラブルシューティング を参照してください。

3.14. RHV での OPENSHIFT CONTAINER PLATFORM WEB コンソールへのアクセス

OpenShift Container Platform クラスターの初期化後に、OpenShift Container Platform Web コンソールにログインできます。

手順

1. オプション: Red Hat Virtualization (RHV) Administration Portal で、**Compute → Cluster** を開きます。
2. インストールプログラムが仮想マシンを作成することを確認します。
3. インストールプログラムが実行されているコマンドラインに戻ります。インストールプログラムが完了すると、OpenShift Container Platform Web コンソールにログインするためのユーザー名およびパスワードの一時パスワードが表示されます。
4. ブラウザーから OpenShift Container Platform の Web コンソールの URL を開きます。URL は以下の形式を使用します。

```
console-openshift-console.apps.<clustername>.<basedomain> 1
```

1 **<clustername>.<basedomain>** に、クラスター名およびベースドメインを指定します。

以下に例を示します。

```
console-openshift-console.apps.my-cluster.virtlab.example.com
```

3.15. OPENSHIFT CONTAINER PLATFORM の TELEMETRY アクセス

OpenShift Container Platform 4.12 では、クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは OpenShift Cluster Manager Hybrid Cloud Console に登録されます。

OpenShift Cluster Manager インベントリーが正常である (Telemetry によって自動的に維持、または OpenShift Cluster Manager Hybrid Cloud Console を使用して手動で維持) ことを確認した後に、subscription watch を使用 して、アカウントまたはマルチクラスターレベルで OpenShift Container Platform サブスクリプションを追跡します。

関連情報

- Telemetry サービスの詳細は、リモートヘルスモニタリング を参照して ください。

3.16. RED HAT VIRTUALIZATION (RHV) へのインストールに関するよくある問題のトラブルシューティング

以下に、一般的な問題およびそれらについて考えられる原因および解決策を記載します。

3.16.1. CPU 負荷が増大し、ノードが Not Ready 状態になる

- **現象**: CPU 負荷が大幅に増大し、ノードが **Not Ready** 状態に切り替わり始める。

- **原因**: ストレージドメインのレイテンシーが高すぎる可能性があります (特にコントロールプレーンノードの場合)。
- **解決策**:
  Kubelet サービスを再起動して、ノードを再度 Ready 状態にします。

```
$ systemctl restart kubelet
```

  OpenShift Container Platform メトリックサービスを検査します。これは、etcd ディスクの同期期間などの有用なデータを収集し、これについて報告します。クラスターが機能している場合は、このデータを使用して、ストレージのレイテンシーまたはスループットが根本的な問題かどうかを判断します。その場合、レイテンシーが短く、スループットの高いストレージリソースの使用を検討してください。

  未加工メトリックを取得するには、kubeadmin または cluster-admin 権限を持つユーザーで以下のコマンドを実行します。

```
$ oc get --insecure-skip-tls-verify --server=https://localhost:<port> --raw=/metrics
```

  詳細は、Exploring Application Endpoints for the purposes of Debugging with OpenShift 4.x を参照してください。

3.16.2. OpenShift Container Platform クラスター API に接続できない

- **現象**: インストールプログラムは完了するが、OpenShift Container Platform クラスター API は利用できない。ブートストラップの仮想マシンは、ブートストラッププロセスの完了後も起動した状態になります。以下のコマンドを入力すると、応答がタイムアウトします。

```
$ oc login -u kubeadmin -p *** <apiurl>
```

- **原因**: ブートストラップ仮想マシンがインストールプログラムによって削除されず、クラスターの API IP アドレスをリリースしない。
- **解決策**: **wait-for** サブコマンドを使用して、ブートストラッププロセスの完了時に通知を受信する。

```
$ ./openshift-install wait-for bootstrap-complete
```

  ブートストラッププロセスが完了したら、ブートストラップ仮想マシンを削除します。

```
$ ./openshift-install destroy bootstrap
```

3.17. インストール後のタスク

OpenShift Container Platform クラスターの初期化後に、以下のタスクを実行できます。

- オプション: デプロイメント後に、OpenShift Container Platform で Machine Config Operator (MCO) を使用して SSH キーを追加するか、置き換えます。
- オプション: **kubeadmin** ユーザーを削除します。代わりに、認証プロバイダーを使用して cluster-admin 権限を持つユーザーを作成します。

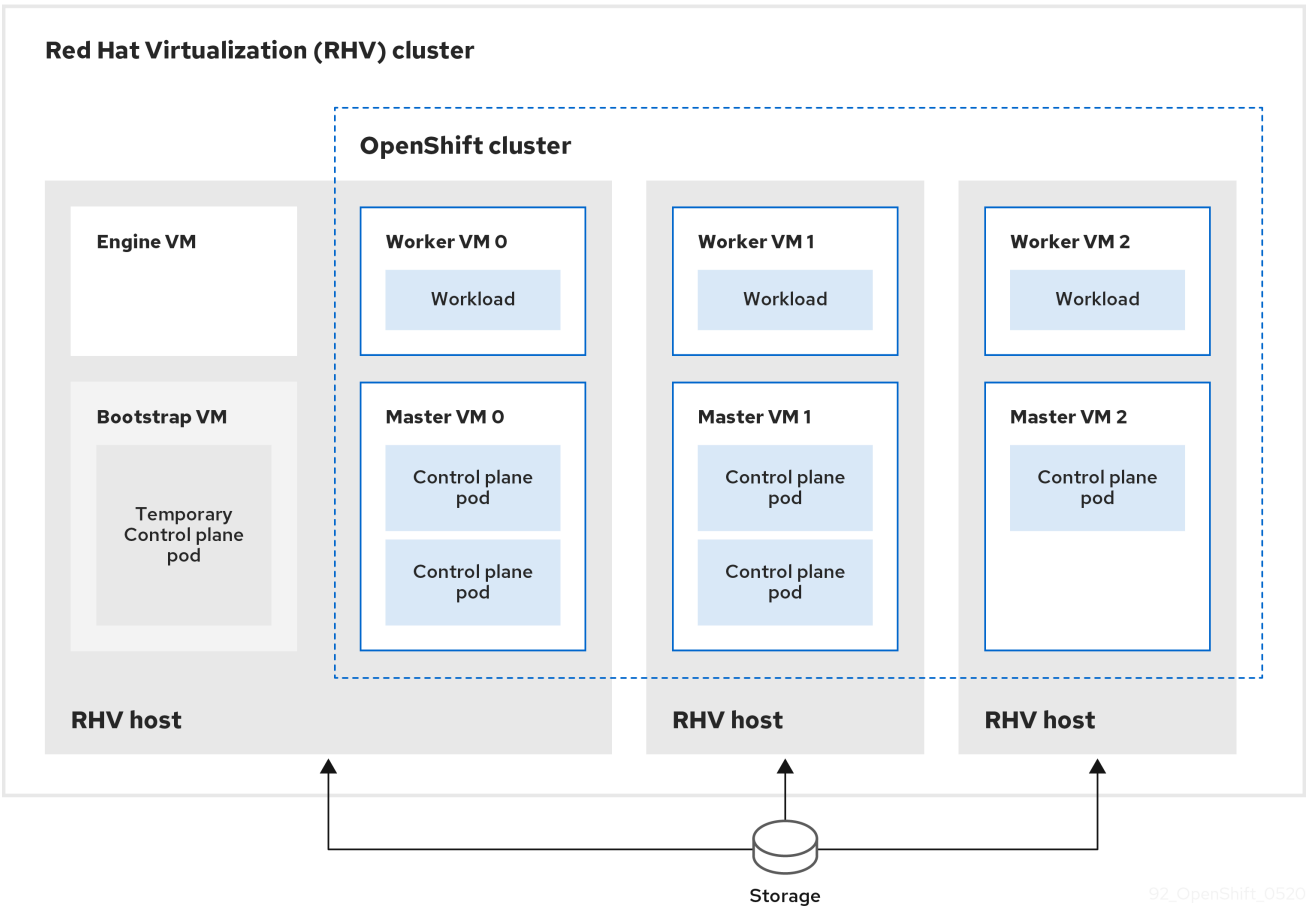
3.18. 次のステップ

- クラスターをカスタマイズします。
- リモートヘルスレポート

第4章 ユーザーによってプロビジョニングされるインフラストラクチャーを使用した RHV へのクラスターのインストール

OpenShift Container Platform バージョン 4.12 では、Red Hat Virtualization (RHV) および独自に提供する他のインフラストラクチャーに、カスタマイズされた OpenShift Container Platform クラスターをインストールできます。OpenShift Container Platform ドキュメントでは、**ユーザーによってプロビジョニングされるインフラストラクチャー** という用語を使用して、このインフラストラクチャータイプに言及しています。

以下の図は、RHV クラスターで実行される可能性のある OpenShift Container Platform クラスターの例を示しています。



RHV ホストは、コントロールプレーンとコンピュート Pod の両方が含まれる仮想マシンを実行します。ホストのいずれかが Manage 仮想マシンと、一時的なコントロールプレーン Pod を含むブートストラップ仮想マシンも実行します。

4.1. 前提条件

OpenShift Container Platform クラスターを RHV 環境にインストールするには、以下の要件を満たしている必要があります。

- OpenShift Container Platform のインストールおよび更新 プロセスの詳細を確認している。
- クラスターインストール方法の選択およびそのユーザー向けの準備 を確認している。
- Support Matrix for OpenShift Container Platform on Red Hat Virtualization (RHV) に記載のあるサポートされるバージョンの組み合わせを使用できる。

4.2. OPENSHIFT CONTAINER PLATFORM のインターネットアクセス

OpenShift Container Platform 4.12 では、クラスターをインストールするためにインターネットアクセスが必要になります。

インターネットへのアクセスは以下を実行するために必要です。

- OpenShift Cluster Manager Hybrid Cloud Console にアクセスし、インストールプログラムをダウンロードし、サブスクリプション管理を実行します。クラスターにインターネットアクセスがあり、Telemetry を無効にしない場合、そのサービスは有効なサブスクリプションでクラスターを自動的に使用します。
- クラスターのインストールに必要なパッケージを取得するために Quay.io にアクセスします。
- クラスターの更新を実行するために必要なパッケージを取得します。

重要

クラスターでインターネットに直接アクセスできない場合、プロビジョニングする一部のタイプのインフラストラクチャーでネットワークが制限されたインストールを実行できます。このプロセスで、必要なコンテンツをダウンロードし、これを使用してミラーレジストリーにインストールパッケージを設定します。インストールタイプに応じて、クラスターのインストール環境でインターネットアクセスが不要となる場合があります。クラスターを更新する前に、ミラーレジストリーのコンテンツを更新します。

4.3. RHV 環境の要件

OpenShift Container Platform バージョン 4.12 クラスターをインストールし、実行するには、RHV 環境が以下の要件を満たしている必要があります。

これらの要件を満たさないと、インストールまたはプロセスが失敗する可能性があります。さらに、これらの要件を満たしていないと、OpenShift Container Platform クラスターはインストールしてから数日または数週間後に失敗する可能性があります。

CPU、メモリー、ストレージリソースについての以下の要件は、インストールプログラムが作成する仮想マシンのデフォルト数で乗算した **デフォルト** 値に基づいています。これらのリソースは、RHV 環境が OpenShift Container Platform 以外の操作に使用するものに **加え**、利用可能でなければなりません。

デフォルトでは、インストールプログラムは 7 つの仮想マシンをインストールプロセスで作成します。まず、ブートストラップ仮想マシンを作成し、OpenShift Container Platform クラスターの残りの部分を作成する間に一時サービスとコントロールプレーンを提供します。インストールプログラムがクラスターの作成を終了すると、ブートストラップマシンが削除され、そのリソースが解放されます。

RHV 環境の仮想マシン数を増やす場合は、リソースを適宜増やす必要があります。

要件

- RHV のバージョンは 4.4 である。
- RHV 環境に **Up** 状態のデータセンターが 1 つあること。
- RHV データセンターに RHV クラスターが含まれていること。
- RHV クラスターに OpenShift Container Platform クラスター専用の以下のリソースがあること。
  - 最小 28 GiB インストール時に作成される 7 仮想マシンのそれぞれに 4 GiB

- 最小 28 vCPU: インストール時に作成される 7 仮想マシンのそれぞれに 4 vCPU。
- 以下を含む 112 GiB 以上の RAM。
  - 一時的なコントロールプレーンを提供するブートストラップマシン用に 16 GiB 以上。
  - コントロールプレーンを提供する 3 つのコントロールプレーンマシンのそれぞれに 16 GiB 以上。
  - アプリケーションワークロードを実行する 3 つのコンピュートマシンのそれぞれに 16 GiB 以上。

- RHV ストレージドメインは、これらの etcd バックエンドのパフォーマンス要件 を満たす必要があります。
- 実稼働環境では、各仮想マシンに 120 GiB 以上が必要です。そのため、ストレージドメインはデフォルトの OpenShift Container Platform クラスターに 840 GiB 以上を提供する必要があります。リソースに制約のある環境または非実稼働環境では、各仮想マシンに 32 GiB 以上を指定する必要があるため、ストレージドメインにはデフォルトの OpenShift Container Platform クラスター用に 230 GiB 以上が必要になります。
- インストールおよび更新中に Red Hat Ecosystem Catalog からイメージをダウンロードするには、RHV クラスターがインターネット接続にアクセスできる必要があります。また、サブスクリプションおよびエンタイトルメントプロセスを単純化するために Telemetry サービスにもインターネット接続が必要です。
- RHV クラスターには、RHV Manager の REST API にアクセスできる仮想ネットワークが必要です。インストーラーが作成する仮想マシンが DHCP を使用して IP アドレスを取得するため、DHCP がこのネットワークで有効にされていることを確認します。
- ターゲット RHV クラスターに OpenShift Container Platform クラスターをインストールし、管理するための以下の最小限の権限を持つユーザーアカウントおよびグループ。
  - **DiskOperator**
  - **DiskCreator**
  - **UserTemplateBasedVm**
  - **TemplateOwner**
  - **TemplateCreator**
  - ターゲットクラスターの **ClusterAdmin**

警告

最小権限の原則を適用します。インストールプロセスで RHV で **SuperUser** 権限を持つ管理者アカウントを使用することを避けます。インストールプログラムは、ユーザーが指定する認証情報を、危険にさらされる可能性のある一時的な **ovirt-config.yaml** ファイルに保存します。

4.4. RHV 環境の要件の確認

RHV 環境が OpenShift Container Platform クラスターをインストールし、実行するための要件を満たしていることを確認します。これらの要件を満たさないと、エラーが発生する可能性があります。

重要

これらの要件は、インストールプログラムがコントロールプレーンおよびコンピュートマシンの作成に使用するデフォルトのリソースに基づいています。これらのリソースには、vCPU、メモリー、およびストレージが含まれます。これらのリソースを変更するか、OpenShift Container Platform マシンの数を増やす場合は、これらの要件を適宜調整します。

手順

1. RHV バージョンが OpenShift Container Platform バージョン 4.12 のインストールをサポートしていることを確認します。
   a. RHV Administration Portal の右上にある ? ヘルプアイコンをクリックし、**About** を選択します。
   b. 開かれるウィンドウで、**RHV ソフトウェアのバージョン** をメモします。
   c. RHV のバージョンが 4.4 であることを確認します。サポートされるバージョンの組み合わせについての詳細は、Support Matrix for OpenShift Container Platform on RHV を参照してください。
2. データセンター、クラスター、およびストレージを検査します。
   a. RHV 管理ポータルで、**Compute → Data Centers** をクリックします。
   b. OpenShift Container Platform をインストールする予定のデータセンターにアクセスできることを確認します。
   c. そのデータセンターの名前をクリックします。
   d. データセンターの詳細の **Storage** タブで、OpenShift Container Platform をインストールする予定のストレージドメインが **Active** であることを確認します。
   e. 後で使用できるように **ドメイン名** を記録します。
   f. **空き領域** に 230 GiB 以上あることを確認します。
   g. ストレージドメインが これらの etcd バックエンドのパフォーマンス要件 を満たしていることを確認します。これは、fio パフォーマンスベンチマークツールを使用して測定できます。
   h. データセンターの詳細で、**Clusters** タブをクリックします。
   i. OpenShift Container Platform をインストールする予定の RHV クラスターを見つけます。後で使用できるようにクラスター名を記録します。
3. RHV ホストリソースを確認します。
   a. RHV 管理ポータルで、**Compute > Clusters** をクリックします。
   b. OpenShift Container Platform をインストールする予定のクラスターをクリックします。

c. クラスターの詳細で、**Hosts** タブをクリックします。

d. ホストを検査し、それらに OpenShift Container Platform クラスター **専用** として利用可能な **論理 CPU コア** の合計が 28 つ以上であることを確認します。

e. 後で使用できるように、利用可能な **論理 CPU コア** の数を記録します。

f. これらの CPU コアが分散され、インストール時に作成された 7 つの仮想マシンのそれぞれに 4 つのコアを持たせることができることを確認します。

g. ホストには、以下の OpenShift Container Platform マシンのそれぞれの要件を満たすように **新規仮想マシンをスケジュールするための最大空きメモリー** として 112 GiB があることを確認します。

- ブートストラップマシンに 16 GiB が必要です。
- 3 つのコントロールプレーンマシンのそれぞれに 16 GiB が必要です。
- 3 つのコンピュートマシンのそれぞれに 16 GiB が必要です。

h. 後で使用できるように **新規仮想マシンをスケジュールするための最大空きメモリー** の量を記録します。

4. OpenShift Container Platform をインストールするための仮想ネットワークが RHV Manager の REST API にアクセスできることを確認します。このネットワーク上の仮想マシンから、RHV Manager の REST API に到達するために curl を使用します。

```
$ curl -k -u <username>@<profile>:<password> \ 1
https://<engine-fqdn>/ovirt-engine/api 2
```

1 **<username>** については、RHV で OpenShift Container Platform クラスターを作成および管理する権限を持つ RHV アカウントのユーザー名を指定します。**<profile>** には、ログインプロファイルを指定します。ログインプロファイルは、RHV Administration Portal ログインページに移動し、 **Profile** ドロップダウンリストで確認できます。**<password>** に、そのユーザー名のパスワードを指定します。

2 **<engine-fqdn>** に、RHV 環境の完全修飾ドメイン名を指定します。

以下に例を示します。

```
$ curl -k -u ocpadmin@internal:pw123 \
https://rhv-env.virtlab.example.com/ovirt-engine/api
```

4.5. ユーザーによってプロビジョニングされるインフラストラクチャーのネットワーク要件

すべての Red Hat Enterprise Linux CoreOS (RHCOS) マシンでは、起動時に **initramfs** でネットワークを設定し、Ignition 設定ファイルをフェッチする必要があります。

初回の起動時に、マシンには DHCP サーバーを使用して設定される IP アドレス設定、または必要な起動オプションを指定して静的に設定される IP アドレス設定が必要です。ネットワーク設定の確立後に、マシンは HTTP または HTTPS サーバーから Ignition 設定ファイルをダウンロードします。その

後、Ignition 設定ファイルは各マシンの正確な状態を設定するために使用されます。Machine Config Operator はインストール後に、新しい証明書やキーの適用など、マシンへの追加の変更を完了します。

クラスターマシンの長期管理に DHCP サーバーを使用することが推奨されます。DHCP サーバーが永続 IP アドレス、DNS サーバー情報、およびホスト名をクラスターマシンに提供するように設定されていることを確認します。

注記

DHCP サービスが user-provisioned infrastructure で利用できない場合は、IP ネットワーク設定および DNS サーバーのアドレスを RHCOS のインストール時にノードに提供することができます。ISO イメージからインストールしている場合は、ブート引数として渡すことができます。静的 IP プロビジョニングと高度なネットワークオプションの詳細は、**RHCOS のインストールと OpenShift Container Platform ブートストラッププロセスの開始** のセクションを参照してください。

Kubernetes API サーバーはクラスターマシンのノード名を解決できる必要があります。API サーバーおよびワーカーノードが異なるゾーンに置かれている場合、デフォルトの DNS 検索ゾーンを、API サーバーでノード名を解決できるように設定することができます。もう 1 つの実行可能な方法として、ノードオブジェクトとすべての DNS 要求の両方において、ホストを完全修飾ドメイン名で常に参照します。

ファイアウォール

クラスターが必要なサイトにアクセスできるようにファイアウォールを設定します。

以下も参照してください。

- Red Hat Virtualization Manager ファイアウォールの要件
- ホストのファイアウォール要件

ロードバランサー

レイヤー 4 のロードバランサーを 1 つまたは 2 つ (推奨) 設定します。

- コントロールプレーンおよびブートストラップマシンのポート **6443** および **22623** に対して負荷分散を行います。ポート **6443** は Kubernetes API サーバーへのアクセスを提供し、内外で到達可能である必要があります。ポート **22623** はクラスター内のノードからアクセスできる必要があります。

- Ingress ルーターを実行するマシン (通常はデフォルト設定のコンピュートノード) 向けに、ポート **443** および **80** に対する負荷分散を行います。いずれのポートもクラスター内外でアクセスできる必要があります。

DNS

インフラストラクチャーで提供される DNS を設定して、主要なコンポーネントとサービスの正しい解決を許可します。1 つのロードバランサーのみを使用する場合、これらの DNS レコードは同じ IP アドレスを参照できます。

- **api.<cluster_name>.<base_domain>** (内部および外部解決) と、コントロールプレーンマシンのロードバランサーを参照する **api-int.<cluster_name>.<base_domain>** (内部解決) の DNS レコードを作成します。

- Ingress ルーターのロードバランサーを参照する ***.apps.<cluster_name>.<base_domain>** の DNS レコードを作成します。たとえば、コンピュートマシンのポート **443** および **80** などが含まれます。

4.5.1. DHCP を使用したクラスターノードのホスト名の設定

Red Hat Enterprise Linux CoreOS (RHCOS) マシンでは、ホスト名は NetworkManager 経由で設定されます。デフォルトでは、マシンは DHCP 経由でホスト名を取得します。ホスト名が DHCP によって提供されない場合、カーネル引数を介して静的に設定される場合、または別の方法でホスト名が取得される場合は、逆引き DNS ルックアップによって取得されます。逆引き DNS ルックアップは、ネットワークがノードで初期化された後に発生し、解決に時間がかかる場合があります。その他のシステムサービスは、これより前に起動し、ホスト名を **localhost** または同様のものとして検出できます。これを回避するには、DHCP を使用して各クラスターノードのホスト名を指定できます。

また、DHCP を介してホスト名を設定すると、DNS スプリットホライズンが実装されている環境での手動の DNS レコード名設定エラーを回避できます。

4.5.2. ネットワーク接続の要件

OpenShift Container Platform クラスターのコンポーネントが通信できるように、マシン間のネットワーク接続を設定する必要があります。すべてのマシンではクラスターの他のすべてのマシンのホスト名を解決できる必要があります。

このセクションでは、必要なポートの詳細を説明します。

重要

インターネットに接続された OpenShift Container Platform 環境では、プラットフォームコンテナーのイメージをプルし、Red Hat にテレメトリーデータを提供するために、すべてのノードがインターネットにアクセスできる必要があります。

表4.1 すべてのマシンからすべてのマシンへの通信に使用されるポート

プロトコル	ポート	説明
ICMP	該当なし	ネットワーク到達性のテスト
TCP	**1936**	メトリクス
	9000-**9999**	ホストレベルのサービス。ポート **9100**-**9101** のノードエクスポーター、ポート **9099** の Cluster Version Operator が含まれます。
	10250-**10259**	Kubernetes が予約するデフォルトポート
	10256	openshift-sdn
UDP	**4789**	VXLAN
	6081	Geneve

プロトコル	ポート	説明
	9000-**9999**	ポート **9100**-**9101** のノードエクスポーターを含む、ホストレベルのサービス。
	500	IPsec IKE パケット
	4500	IPsec NAT-T パケット
	123	UDP ポート **123** のネットワークタイムプロトコル (NTP) 外部 NTP タイムサーバーが設定されている場合は、UDP ポート **123** を開く必要があります。
TCP/UDP	**30000**-**32767**	Kubernetes ノードポート
ESP	該当なし	IPsec Encapsulating Security Payload (ESP)

表4.2 すべてのマシンからコントロールプレーンへの通信に使用されるポート

プロトコル	ポート	説明
TCP	**6443**	Kubernetes API

表4.3 コントロールプレーンマシンからコントロールプレーンマシンへの通信に使用されるポート

プロトコル	ポート	説明
TCP	**2379**-**2380**	etcd サーバーおよびピアポート

user-provisioned infrastructure の NTP 設定

OpenShift Container Platform クラスターは、デフォルトでパブリック Network Time Protocol (NTP) サーバーを使用するように設定されます。ローカルのエンタープライズ NTP サーバーを使用する必要があるか、クラスターが切断されたネットワークにデプロイされている場合は、特定のタイムサーバーを使用するようにクラスターを設定できます。詳細は、**chrony タイムサービスの設定** のドキュメントを参照してください。

DHCP サーバーが NTP サーバー情報を提供する場合、Red Hat Enterprise Linux CoreOS (RHCOS) マシンの chrony タイムサービスは情報を読み取り、NTP サーバーとクロックを同期できます。

4.6. インストールマシンの設定

バイナリー **openshift-install** インストールプログラムおよび Ansible スクリプトを実行するには、Manager 上の RHV 環境および REST API にネットワークでアクセスできるように、RHV Manager または Red Hat Enterprise Linux (RHEL) を設定します。

手順

1. Python3 および Ansible を更新またはインストールします。以下に例を示します。

```
# dnf update python3 ansible
```

2. **python3-ovirt-engine-sdk4** パッケージをインストール して、Python Software Development Kit を取得します。

3. **ovirt.image-template** Ansible ロールをインストールします。RHV Manager およびその他の Red Hat Enterprise Linux (RHEL) マシンでは、このロールは **ovirt-ansible-image-template** パッケージとして提供されます。たとえば、 以下を入力します。

```
# dnf install ovirt-ansible-image-template
```

4. **ovirt.vm-infra** Ansible ロールをインストールします。RHV Manager およびその他の RHEL マシンでは、このロールは **ovirt-ansible-vm-infra** パッケージとして提供されます。

```
# dnf install ovirt-ansible-vm-infra
```

5. 環境変数を作成し、その環境変数に絶対パスまたは相対パスを割り当てます。たとえば、 以下を入力します。

```
$ export ASSETS_DIR=./wrk
```

注記

インストールプログラムはこの変数を使用して、重要なインストール関連のファイルを保存するディレクトリーを作成します。その後、インストールプロセスはこの変数を再利用して、これらのアセットファイルを見つけます。このアセットディレクトリーを削除しないでください。これは、クラスターのアンインストールに必要になります。

4.7. OPENSHIFT CONTAINER PLATFORM OPENSTACK クラスターの RHV への非セキュアモードでのインストール

デフォルトで、インストーラーは CA 証明書を作成し、確認を求めるプロンプトを出し、インストール時に使用する証明書を保存します。これは、手動で作成したりインストールしたりする必要はありません。

推奨されていませんが、OpenShift Container Platform を RHV に **非セキュアモード** でインストールして、この機能を上書きし、証明書の検証なしに OpenShift Container Platform をインストールすることができます。

警告

非セキュア モードでのインストールは推奨されていません。これにより、攻撃者が中間者 (Man-in-the-Middle) 攻撃を実行し、ネットワーク上の機密の認証情報を取得できる可能性が生じるためです。

手順

1. **~/.ovirt/ovirt-config.yaml** という名前のファイルを作成します。
2. 以下の内容を **ovirt-config.yaml** に追加します。

```
ovirt_url: https://ovirt.example.com/ovirt-engine/api (1)
ovirt_fqdn: ovirt.example.com (2)
ovirt_pem_url: ""
ovirt_username: ocpadmin@internal
ovirt_password: super-secret-password (3)
ovirt_insecure: true
```

(1) oVirt エンジンのホスト名またはアドレスを指定します。

(2) oVirt エンジンの完全修飾ドメイン名を指定します。

(3) oVirt エンジンの管理者パスワードを指定します。

3. インストーラーを実行します。

4.8. クラスターノードの SSH アクセス用のキーペアの生成

OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定できます。キーは、Ignition 設定ファイルを介して Red Hat Enterprise Linux CoreOS (RHCOS) ノードに渡され、ノードへの SSH アクセスを認証するために使用されます。このキーは各ノードの **core** ユーザーの **~/.ssh/authorized_keys** リストに追加され、パスワードなしの認証が可能になります。

キーがノードに渡されると、キーペアを使用して RHCOS ノードにユーザー **core** として SSH を実行できます。SSH 経由でノードにアクセスするには、秘密鍵のアイデンティティーをローカルユーザーの SSH で管理する必要があります。

インストールのデバッグまたは障害復旧を実行するためにクラスターノードに対して SSH を実行する場合は、インストールプロセスの間に SSH 公開鍵を指定する必要があります。**./openshift-install gather** コマンドでは、SSH 公開鍵がクラスターノードに配置されている必要もあります。

重要

障害復旧およびデバッグが必要な実稼働環境では、この手順を省略しないでください。

注記

AWS キーペア などのプラットフォームに固有の方法で設定したキーではなく、ローカルキーを使用する必要があります。

手順

1. クラスターノードへの認証に使用するローカルマシンに既存の SSH キーペアがない場合は、これを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ ssh-keygen -t ed25519 -N '' -f <path>/<file_name> (1)
```

1 新しい SSH キーのパスとファイル名 (~/**.ssh/id_ed25519** など) を指定します。既存のキーペアがある場合は、公開鍵が ~/**.ssh** ディレクトリーにあることを確認します。

注記

FIPS で検証済みまたは進行中のモジュール (Modules in Process) 暗号ライブラリーを使用する OpenShift Container Platform クラスターを **x86_64**、**ppc64le**、および **s390x** アーキテクチャーにインストールする予定の場合は、**ed25519** アルゴリズムを使用するキーは作成しないでください。代わりに、**rsa** アルゴリズムまたは **ecdsa** アルゴリズムを使用するキーを作成します。

2. 公開 SSH キーを表示します。

```
$ cat <path>/<file_name>.pub
```

たとえば、次のコマンドを実行して ~/**.ssh/id_ed25519.pub** 公開鍵を表示します。

```
$ cat ~/.ssh/id_ed25519.pub
```

3. ローカルユーザーの SSH エージェントに SSH 秘密鍵 ID が追加されていない場合は、それを追加します。キーの SSH エージェント管理は、クラスターノードへのパスワードなしの SSH 認証、または **./openshift-install gather** コマンドを使用する場合は必要になります。

注記

一部のディストリビューションでは、~/**.ssh/id_rsa** および ~/**.ssh/id_dsa** などのデフォルトの SSH 秘密鍵のアイデンティティーは自動的に管理されます。

a. **ssh-agent** プロセスがローカルユーザーに対して実行されていない場合は、バックグラウンドタスクとして開始します。

```
$ eval "$(ssh-agent -s)"
```

出力例

```
Agent pid 31874
```

注記

クラスターが FIPS モードにある場合は、FIPS 準拠のアルゴリズムのみを使用して SSH キーを生成します。鍵は RSA または ECDSA のいずれかである必要があります。

4. SSH プライベートキーを **ssh-agent** に追加します。

```
$ ssh-add <path>/<file_name> 1
```

1 ~/**.ssh/id_ed25519** などの、SSH プライベートキーのパスおよびファイル名を指定します。

出力例

```
Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

次のステップ

- OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定します。

4.9. インストールプログラムの取得

OpenShift Container Platform をインストールする前に、インストールに使用しているホストにインストールファイルをダウンロードします。

前提条件

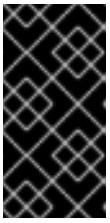
- 500 MB のローカルディスク領域がある Linux または macOS を実行するコンピューターが必要です。

手順

1. OpenShift Cluster Manager サイトの インフラストラクチャープロバイダー ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使用してログインします。アカウントがない場合はこれを作成します。
2. インフラストラクチャープロバイダーを選択します。
3. インストールタイプのページに移動し、ホストオペレーティングシステムとアーキテクチャーに対応するインストールプログラムをダウンロードして、インストール設定ファイルを保存するディレクトリーにファイルを配置します。

重要

インストールプログラムは、クラスターのインストールに使用するコンピューターにいくつかのファイルを作成します。クラスターのインストール完了後は、インストールプログラムおよびインストールプログラムが作成するファイルを保持する必要があります。ファイルはいずれもクラスターを削除するために必要になります。



重要

インストールプログラムで作成されたファイルを削除しても、クラスターがインストール時に失敗した場合でもクラスターは削除されません。クラスターを削除するには、特定のクラウドプロバイダー用の OpenShift Container Platform のアンインストール手順を実行します。

4. インストールプログラムを展開します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ tar -xvf openshift-install-linux.tar.gz
```

5. Red Hat OpenShift Cluster Manager からインストールプルシークレット をダウンロードします。このプルシークレットを使用し、OpenShift Container Platform コンポーネントのコンテ

ナーイメージを提供する Quay.io など、組み込まれた各種の認証局によって提供されるサービスで認証できます。

4.10. ANSIBLE PLAYBOOK のダウンロード

RHV に OpenShift Container Platform バージョン 4.12 をインストールするために Ansible Playbook をダウンロードします。

手順

- インストールマシンで、以下のコマンドを実行します。

```
$ mkdir playbooks
```

```
$ cd playbooks
```

```
$  xargs -n 1 curl -O <<< '
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/bootstrap.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/common-auth.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/create-templates-and-vms.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/inventory.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/masters.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/retire-bootstrap.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/retire-masters.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/retire-workers.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/workers.yml'
```

次のステップ

- これらの Ansible Playbook をダウンロードしたら、インストールプログラムを実行してインストール設定ファイルを作成する前に、アセットディレクトリーの環境変数を作成し、**inventory.yml** ファイルをカスタマイズする必要もあります。

4.11. INVENTORY.YML ファイル

inventory.yml ファイルを使用して、インストールする OpenShift Container Platform クラスターの各種の要素を定義し、作成します。これには、Red Hat Enterprise Linux CoreOS(RHCOS) イメージ、仮想マシンテンプレート、ブートストラップマシン、コントロールプレーンノード、ワーカーノードなどの要素が含まれます。また、**inventory.yml** を使用してクラスターを破棄します。

以下の **inventory.yml** の例は、パラメーターとそれらのデフォルト値を示しています。これらのデフォルト値の量と数は、RHV 環境で実稼働用の OpenShift Container Platform クラスターを実行するための要件を満たしています。

inventory.yml ファイルの例

```
---
all:
```

```
  vars:

    ovirt_cluster: "Default"
    ocp:
      assets_dir: "{{ lookup('env', 'ASSETS_DIR') }}"
      ovirt_config_path: "{{ lookup('env', 'HOME') }}/.ovirt/ovirt-config.yaml"

    # ---
    # {op-system} section
    # ---
    rhcos:
      image_url: "https://mirror.openshift.com/pub/openshift-v4/dependencies/rhcos/4.12/latest/rhcos-
openstack.x86_64.qcow2.gz"
      local_cmp_image_path: "/tmp/rhcos.qcow2.gz"
      local_image_path: "/tmp/rhcos.qcow2"

    # ---
    # Profiles section
    # ---
    control_plane:
      cluster: "{{ ovirt_cluster }}"
      memory: 16GiB
      sockets: 4
      cores: 1
      template: rhcos_tpl
      operating_system: "rhcos_x64"
      type: high_performance
      graphical_console:
        headless_mode: false
        protocol:
        - spice
        - vnc
      disks:
      - size: 120GiB
        name: os
        interface: virtio_scsi
        storage_domain: depot_nvme
      nics:
      - name: nic1
        network: lab
        profile: lab

    compute:
      cluster: "{{ ovirt_cluster }}"
      memory: 16GiB
      sockets: 4
      cores: 1
      template: worker_rhcos_tpl
      operating_system: "rhcos_x64"
      type: high_performance
      graphical_console:
        headless_mode: false
        protocol:
        - spice
        - vnc
      disks:
```

```
    - size: 120GiB
      name: os
      interface: virtio_scsi
      storage_domain: depot_nvme
    nics:
    - name: nic1
      network: lab
      profile: lab

  # ---
  # Virtual machines section
  # ---
  vms:
  - name: "{{ metadata.infraID }}-bootstrap"
    ocp_type: bootstrap
    profile: "{{ control_plane }}"
    type: server
  - name: "{{ metadata.infraID }}-master0"
    ocp_type: master
    profile: "{{ control_plane }}"
  - name: "{{ metadata.infraID }}-master1"
    ocp_type: master
    profile: "{{ control_plane }}"
  - name: "{{ metadata.infraID }}-master2"
    ocp_type: master
    profile: "{{ control_plane }}"
  - name: "{{ metadata.infraID }}-worker0"
    ocp_type: worker
    profile: "{{ compute }}"
  - name: "{{ metadata.infraID }}-worker1"
    ocp_type: worker
    profile: "{{ compute }}"
  - name: "{{ metadata.infraID }}-worker2"
    ocp_type: worker
    profile: "{{ compute }}"
```

重要

Enter から始まる説明のあるパラメーターの値を入力します。それ以外の場合は、デフォルト値を使用するか、新しい値に置き換えることができます。

General セクション

- **ovirt_cluster**: OpenShift Container Platform クラスターをインストールする既存の RHV クラスターの名前を入力します。
- **ocp.assets_dir**: **openshift-install** インストールプログラムが生成するファイルを保存するために作成するディレクトリーのパス。
- **ocp.ovirt_config_path**: インストールプログラムが生成する **ovirt-config.yaml** ファイルのパス (**./wrk/install-config.yaml** など)。このファイルには、Manager の REST API との対話に必要な認証情報が含まれます。

Red Hat Enterprise Linux CoreOS (RHCOS) セクション

- **image_url**: ダウンロード用に指定した RHCOS イメージの URL を入力します。
- **local_cmp_image_path**: 圧縮された RHCOS イメージのローカルダウンロードディレクトリーのパス。
- **local_image_path**: デプロイメントした RHCOS イメージのローカルディレクトリーのパス。

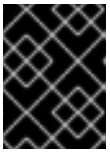
Profiles セクション

このセクションは、2 つのプロファイルで設定されます。

- **control_plane**: ブートストラップおよびコントロールプレーンノードのプロファイル。
- **compute**: コンピュートプレーン内のワーカーノードのプロファイル。

これらのプロファイルには以下のパラメーターが含まれます。パラメーターのデフォルト値は、実稼働クラスターを実行するために必要な最小要件を満たします。これらの値は、ワークロードの要件に応じて増減したり、カスタマイズしたりできます。

- **cluster**: 値は、General セクションの **ovirt_cluster** からクラスター名を取得します。
- **memory**: 仮想マシンに必要なメモリーの量 (GB)。
- **sockets**: 仮想マシンのソケット数。
- **cores**: 仮想マシンのコア数。
- **template**: 仮想マシンテンプレートの名前。複数のクラスターをインストールする計画があり、これらのクラスターが異なる仕様が含まれるテンプレートを使用する場合には、テンプレート名の先頭にクラスターの ID を付けます。
- **operating_system**: 仮想マシンのゲストオペレーティングシステムのタイプ。oVirt/RHV バージョン 4.4 では、**Ignition script** の値を仮想マシンに渡すことができるようにするために、この値を **rhcos_x64** にする必要があります。
- **type**: 仮想マシンのタイプとして **server** を入力します。



重要

type パラメーターの値を **high_performance** から **server** に変更する必要があります。

- **disks**: ディスクの仕様。**control_plane** と **compute** ノードには、異なるストレージドメインを設定できます。
- **size**: ディスクの最小サイズ。
- **name**: RHV のターゲットクラスターに接続されたディスクの名前を入力します。
- **interface**: 指定したディスクのインターフェイスタイプを入力します。
- **storage_domain**: 指定したディスクのストレージドメインを入力します。

- **nics**: 仮想マシンが使用する **name** および **network** を入力します。仮想ネットワークインターフェイスプロファイルを指定することもできます。デフォルトでは、NIC は oVirt/RHV MAC プールから MAC アドレスを取得します。

仮想マシンセクション

この最後のセクション **vms** は、クラスターで作成およびデプロイする予定の仮想マシンを定義します。デフォルトで、実稼働環境用の最小数のコントロールプレーンおよびワーカーノードが提供されます。

vms には 3 つの必須要素が含まれます。

- **name**: 仮想マシンの名前。この場合、**metadata.infraID** は、仮想マシン名の先頭に **metadata.yml** ファイルのインフラストラクチャー ID を付けます。
- **ocp_type**: OpenShift Container Platform クラスター内の仮想マシンのロール。使用できる値は **bootstrap**、**master**、**worker** です。
- **profile**: それぞれの仮想マシンが仕様を継承するプロファイルの名前。この例で使用可能な値は **control_plane** または **compute** です。
  仮想マシンがプロファイルから継承する値を上書きできます。これを実行するには、**inventory.yml** の仮想マシンに profile 属性の名前を追加し、これに上書きする値を割り当てます。この例を確認するには、直前の **inventory.yml** の例の **name: "{{ metadata.infraID }}-bootstrap"** 仮想マシンを検査します。これには値が **server** の **type** 属性があり、この仮想マシンがそれ以外の場合に **control_plane** プロファイルから継承する **type** 属性の値を上書きします。

メタデータ変数

仮想マシンの場合、**metadata.infraID** は、仮想マシンの名前の先頭に、Ignition ファイルのビルド時に作成する **metadata.json** ファイルのインフラストラクチャー ID を付けます。

Playbook は以下のコードを使用して、**ocp.assets_dir** にある特定のファイルから **infraID** を読み取ります。

```
---
- name: include metadata.json vars
  include_vars:
    file: "{{ ocp.assets_dir }}/metadata.json"
    name: metadata


  ...
```

4.12. RHCOS イメージ設定の指定

inventory.yml ファイルの Red Hat Enterprise Linux CoreOS (RHCOS) イメージ設定を更新します。後にこのファイルを Playbook のいずれかとして実行すると、圧縮された Red Hat Enterprise Linux CoreOS (RHCOS) イメージが **image_url** URL から **local_cmp_image_path** ディレクトリーにダウンロードされます。次に Playbook はイメージを **local_image_path** ディレクトリーにデプロイメントし、これを使用して oVirt/RHV テンプレートを作成します。

手順

1. インストールする OpenShift Container Platform バージョンの RHCOS イメージダウンロードページを見つけます (例: /pub/openshift-v4/dependencies/rhcos/latest/latest のインデックス)。

2. そのダウンロードページから、**https://mirror.openshift.com/pub/openshift-v4/dependencies/rhcos/4.12/latest/rhcos-openstack.x86_64.qcow2.gz** などの OpenStack **qcow2** イメージの URL をコピーします。

3. 先のステップでダウンロードした **inventory.yml** Playbook を編集します。この中で、URL を **image_url** の値として貼り付けます。以下に例を示します。

```
rhcos:
  "https://mirror.openshift.com/pub/openshift-v4/dependencies/rhcos/4.12/latest/rhcos-openstack.x86_64.qcow2.gz"
```

4.13. インストール設定ファイルの作成

インストールプログラム **openshift-install** を実行し、先に指定または収集した情報でプロンプトに応答し、インストール設定ファイルを作成します。

プロンプトに応答すると、インストールプログラムは、以前に指定したアセットディレクトリーの **install-config.yaml** ファイルの初期バージョンを作成します (例: **./wrk/install-config.yaml**)。

インストールプログラムは、Manager に到達して REST API を使用するために必要なすべての接続パラメーターが含まれる **$HOME/.ovirt/ovirt-config.yaml** ファイルも作成します。

注: インストールプロセスでは、**Internal API virtual IP** および **Ingress virtual IP** などの一部のパラメーターに指定する値を使用しません。それらの値はインフラストラクチャー DNS にすでに設定されているためです。

また、**oVirt cluster**、 **oVirt storage**、および **oVirt network** などの値のような **inventory.yml** のパラメーターに指定する値を使用します。また、スクリプトを使用して **install-config.yaml** の同じ値を削除するか、これを前述の **virtual IPs** に置き換えます。

手順

1. インストールプログラムを実行します。

```
$ openshift-install create install-config --dir $ASSETS_DIR
```

2. インストールプログラムのプロンプトに応答し、システムに関する情報を提供します。

出力例

```
? SSH Public Key /home/user/.ssh/id_dsa.pub
? Platform <ovirt>
? Engine FQDN[:PORT] [? for help] <engine.fqdn>
? Enter ovirt-engine username <ocpadmin@internal>
? Enter password <******>
? oVirt cluster <cluster>
? oVirt storage <storage>
? oVirt network <net>
? Internal API virtual IP <172.16.0.252>
? Ingress virtual IP <172.16.0.251>
```

```
? Base Domain <example.org>
? Cluster Name <ocp4>
? Pull Secret [? for help] <********>
```

```
? SSH Public Key /home/user/.ssh/id_dsa.pub
? Platform <ovirt>
? Engine FQDN[:PORT] [? for help] <engine.fqdn>
? Enter ovirt-engine username <ocpadmin@internal>
? Enter password <******>
? oVirt cluster <cluster>
? oVirt storage <storage>
? oVirt network <net>
? Internal API virtual IP <172.16.0.252>
? Ingress virtual IP <172.16.0.251>
? Base Domain <example.org>
? Cluster Name <ocp4>
? Pull Secret [? for help] <********>
```

Internal API virtual IP および **Ingress virtual IP** について、DNS サービスの設定時に指定した IP アドレスを指定します。

さらに、**oVirt cluster** および **Base Domain** プロンプトに対して入力する値は REST API および作成するアプリケーションの URL の一部を設定します (例: **https://api.ocp4.example.org:6443/** and **https://console-openshift-console.apps.ocp4.example.org**)。

Red Hat OpenShift Cluster Manager からプルシークレット を取得できます。

4.14. INSTALL-CONFIG.YAML のカスタマイズ

ここでは、3 つの python スクリプトを使用して、インストールプログラムのデフォルト動作の一部を上書きします。

- デフォルトでは、インストールプログラムはマシン API を使用してノードを作成します。このデフォルトの動作を上書きするには、コンピュートノードの数をゼロ (0) レプリカに設定します。後に Ansible Playbook を使用してコンピュートノードを作成します。
- デフォルトでは、インストールプログラムはノードのマシンネットワークの IP 範囲を設定します。このデフォルトの動作を上書きするには、インフラストラクチャーに一致するように IP 範囲を設定します。
- デフォルトでは、インストールプログラムはプラットフォームを **ovirt** に設定します。ただし、ユーザーによってプロビジョニングされるインフラストラクチャーにクラスターをインストールすることは、ベアメタルにクラスターをインストールすることに似ています。したがって、ovirt プラットフォームセクションを **install-config.yaml** から削除し、プラットフォームを **none** に変更します。代わりに、**inventory.yml** を使用して、必要な設定をすべて指定します。

注記

これらのスニペットは Python 3 および Python 2 で動作します。

手順

1. コンピュートノードの数をゼロ (0) レプリカに設定します。

```
$ python3 -c 'import os, yaml
path = "%s/install-config.yaml" % os.environ["ASSETS_DIR"]
conf = yaml.safe_load(open(path))
conf["compute"][0]["replicas"] = 0
open(path, "w").write(yaml.dump(conf, default_flow_style=False))'
```

2. マシンネットワークの IP 範囲を設定します。たとえば、範囲を **172.16.0.0/16** に設定するには、以下を実行します。

```
$ python3 -c 'import os, yaml
path = "%s/install-config.yaml" % os.environ["ASSETS_DIR"]
conf = yaml.safe_load(open(path))
conf["networking"]["machineNetwork"][0]["cidr"] = "172.16.0.0/16"
open(path, "w").write(yaml.dump(conf, default_flow_style=False))'
```

3. **ovirt** セクションを削除し、プラットフォームを **none** に変更します。

```
$ python3 -c 'import os, yaml
path = "%s/install-config.yaml" % os.environ["ASSETS_DIR"]
conf = yaml.safe_load(open(path))
platform = conf["platform"]
del platform["ovirt"]
platform["none"] = {}
open(path, "w").write(yaml.dump(conf, default_flow_style=False))'
```

警告

Red Hat Virtualization は現在、oVirt プラットフォーム上にあるユーザーによってプロビジョニングされるインフラストラクチャーでのインストールをサポートしていません。そのため、プラットフォームを **none** に設定し、OpenShift Container Platform が各ノードをベアメタルノードとして、およびクラスターをベアメタルクラスターとして識別できるようにします。これは、任意のプラットフォームにクラスターをインストールするのと同じであり、次の制限があります。

1. クラスタープロバイダーがないため、各マシンを手動で追加する必要があり、ノードスケーリング機能はありません。
2. oVirt CSI ドライバーはインストールされず、CSI 機能はありません。

4.15. マニフェストファイルの生成

インストールプログラムを使用して、アセットディレクトリーにマニフェストファイルのセットを生成します。

マニフェストファイルを生成するコマンドにより、**install-config.yaml** ファイルを使用する前に警告メッセージが表示されます。

install-config.yaml ファイルを再利用する予定の場合には、マニフェストファイルを生成する前にバックアップしてからバックアップコピーを作成してください。

手順

1. オプション: **install-config.yaml** ファイルのバックアップコピーを作成します。

```
$ cp install-config.yaml install-config.yaml.backup
```

2. アセットディレクトリーにマニフェストのセットを生成します。

```
$ openshift-install create manifests --dir $ASSETS_DIR
```

このコマンドにより、以下の情報が表示されます。

出力例

```
INFO Consuming Install Config from target directory
WARNING Making control-plane schedulable by setting MastersSchedulable to true for
Scheduler cluster settings
```

このコマンドにより、以下のマニフェストファイルが生成されます。

出力例

```
$ tree
.
└── wrk
    ├── manifests
    │   ├── 04-openshift-machine-config-operator.yaml
    │   ├── cluster-config.yaml
    │   ├── cluster-dns-02-config.yml
    │   ├── cluster-infrastructure-02-config.yml
    │   ├── cluster-ingress-02-config.yml
    │   ├── cluster-network-01-crd.yml
    │   ├── cluster-network-02-config.yml
    │   ├── cluster-proxy-01-config.yaml
    │   ├── cluster-scheduler-02-config.yml
    │   ├── cvo-overrides.yaml
    │   ├── etcd-ca-bundle-configmap.yaml
    │   ├── etcd-client-secret.yaml
    │   ├── etcd-host-service-endpoints.yaml
    │   ├── etcd-host-service.yaml
    │   ├── etcd-metric-client-secret.yaml
    │   ├── etcd-metric-serving-ca-configmap.yaml
    │   ├── etcd-metric-signer-secret.yaml
    │   ├── etcd-namespace.yaml
    │   ├── etcd-service.yaml
    │   ├── etcd-serving-ca-configmap.yaml
    │   ├── etcd-signer-secret.yaml
    │   ├── kube-cloud-config.yaml
    │   ├── kube-system-configmap-root-ca.yaml
    │   ├── machine-config-server-tls-secret.yaml
    │   └── openshift-config-secret-pull-secret.yaml
```

```
└──── openshift
    ├──── 99_kubeadmin-password-secret.yaml
    ├──── 99_openshift-cluster-api_master-user-data-secret.yaml
    ├──── 99_openshift-cluster-api_worker-user-data-secret.yaml
    ├──── 99_openshift-machineconfig_99-master-ssh.yaml
    ├──── 99_openshift-machineconfig_99-worker-ssh.yaml
    └──── openshift-install-manifests.yaml
```

次のステップ

- コントロールプレーンノードをスケジュール対象外にします。

4.16. コントロールプレーンノードのスケジュール対象外の設定

コントロールプレーンマシンを手動で作成し、デプロイしているので、コントロールプレーンノードをスケジュール対象外にするようにマニフェストファイルを設定する必要があります。

手順

1. コントロールプレーンノードをスケジュール対象外にするには、以下を入力します。

```
$ python3 -c 'import os, yaml
path = "%s/manifests/cluster-scheduler-02-config.yml" % os.environ["ASSETS_DIR"]
data = yaml.safe_load(open(path))
data["spec"]["mastersSchedulable"] = False
open(path, "w").write(yaml.dump(data, default_flow_style=False))'
```

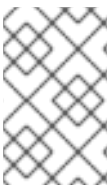
4.17. IGNITION ファイルのビルド

生成および変更したマニフェストファイルから Ignition ファイルを作成するには、インストールプログラムを実行します。このアクションにより、Ignition ファイルをフェッチし、ノードを作成するために必要な設定を実行する Red Hat Enterprise Linux CoreOS (RHCOS) マシン **initramfs** が作成されます。

Ignition ファイルのほかに、インストールプログラムは以下を生成します。

- **oc** および **kubectl** ユーティリティーを使用してクラスターに接続するための管理者認証情報が含まれる **auth** ディレクトリー。
- OpenShift Container Platform クラスター名、クラスター ID、および現行インストールのインフラストラクチャー ID などの情報を含む **metadata.json** ファイル。

このインストールプロセスの Ansible Playbook は、**infraID** の値を、作成する仮想マシンの接頭辞として使用します。これにより、同じ oVirt/RHV クラスターに複数のインストールがある場合の命名の競合が回避されます。



注記

Ignition 設定ファイルの証明書は 24 時間後に有効期限が切れます。最初の証明書のローテーションが終了するように、クラスターのインストールを完了し、クラスターを動作が低下していない状態で 24 時間実行し続ける必要があります。

手順

1. Ignition ファイルをビルドするには、以下を入力します。

```
$ openshift-install create ignition-configs --dir $ASSETS_DIR
```

出力例

```
$ tree
.
└── wrk
    ├── auth
    │   ├── kubeadmin-password
    │   └── kubeconfig
    ├── bootstrap.ign
    ├── master.ign
    ├── metadata.json
    └── worker.ign
```

4.18. テンプレートおよび仮想マシンの作成

inventory.yml の変数を確認した後に、最初の Ansible プロビジョニング Playbook **create-templates-and-vms.yml** を実行します。

この Playbook は、**$HOME/.ovirt/ovirt-config.yaml** から RHV Manager の接続パラメーターを使用し、アセットディレクトリーで **metadata.json** を読み取ります。

ローカルの Red Hat Enterprise Linux CoreOS (RHCOS) イメージが存在しない場合、Playbook は **inventory.yml** の **image_url** に指定した URL からダウンロードします。これはイメージをデプロイメントし、これを RHV にアップロードしてテンプレートを作成します。

Playbook は、**inventory.yml** ファイルの **control_plane** と **compute** プロファイルに基づいてテンプレートを作成します。これらのプロファイルの名前が異なる場合、2 つのテンプレートが作成されます。

Playbook が完了すると、作成される仮想マシンは停止します。他のインフラストラクチャー要素の設定に役立つ情報を取得できます。たとえば、仮想マシンの MAC アドレスを取得して、仮想マシンに永続的な IP アドレスを割り当てるように DHCP を設定できます。

手順

1. **inventory.yml** の **control_plane** および **compute** 変数で、**type: high_performance** の 両方のインスタンスを **type: server** に変更します。

2. オプション: 同じクラスターに複数のインストールを実行する予定の場合には、OpenShift Container Platform インストールごとに異なるテンプレートを作成します。**inventory.yml** ファイルで、**template** の値の先頭に **infraID** を付けます。以下に例を示します。

```
control_plane:
  cluster: "{{ ovirt_cluster }}"
  memory: 16GiB
  sockets: 4
  cores: 1
  template: "{{ metadata.infraID }}-rhcos_tpl"
  operating_system: "rhcos_x64"
  ...
```

3. テンプレートおよび仮想マシンを作成します。

```
$ ansible-playbook -i inventory.yml create-templates-and-vms.yml
```

4.19. ブートストラップマシンの作成

bootstrap.yml Playbook を実行してブートストラップマシンを作成します。この Playbook はブートストラップ仮想マシンを起動し、これをアセットディレクトリーから **bootstrap.ign** Ignition ファイルに渡します。ブートストラップノードは、Ignition ファイルをコントロールプレーンノードに送信できるように設定します。

ブートストラッププロセスをモニターするには、RHV 管理ポータルでコンソールを使用するか、SSH を使用して仮想マシンに接続します。

手順

1. ブートストラップマシンを作成します。

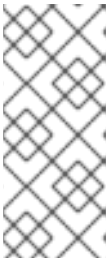
```
$ ansible-playbook -i inventory.yml bootstrap.yml
```

2. 管理ポータルまたは SSH のコンソールを使用してブートストラップマシンに接続します。**<bootstrap_ip>** をブートストラップノードの IP アドレスに置き換えます。SSH を使用するには、以下を入力します。

```
$ ssh core@<boostrap.ip>
```

3. ブートストラップノードからリリースイメージサービスについての **bootkube.service** journald ユニットログを収集します。

```
[core@ocp4-lk6b4-bootstrap ~]$ journalctl -b -f -u release-image.service -u bootkube.service
```



注記

ブートストラップノードの **bootkube.service** のログは etcd の **connection refused** エラーを出力し、ブートストラップサーバーがコントロールプレーンノードの etcd に接続できないことを示します。etcd が各コントロールプレーンノードで起動し、ノードがクラスターに参加した後には、エラーは発生しなくなるはずです。

4.20. コントロールプレーンノードの作成

masters.yml Playbook を実行してコントロールプレーンノードを作成します。この Playbook は **master.ign** Ignition ファイルをそれぞれの仮想マシンに渡します。Ignition ファイルには、**https://api-int.ocp4.example.org:22623/config/master** などの URL から Ignition を取得するためのコントロールプレーンノードのディレクティブが含まれます。この URL のポート番号はロードバランサーによって管理され、クラスター内でのみアクセスできます。

手順

1. コントロールプレーンノードを作成します。

```
$ ansible-playbook -i inventory.yml masters.yml
```

2. Playbook がコントロールプレーンを作成する間に、ブートストラッププロセスをモニターします。

```
$ openshift-install wait-for bootstrap-complete --dir $ASSETS_DIR
```

出力例

```
INFO API v1.25.0 up
INFO Waiting up to 40m0s for bootstrapping to complete...
```

3. コントロールプレーンノードおよび etcd のすべての Pod が実行されている場合、インストールプログラムは以下の出力を表示します。

出力例

```
INFO It is now safe to remove the bootstrap resources
```

4.21. クラスターステータスの確認

インストール時またはインストール後に OpenShift Container Platform クラスターのステータスを確認することができます。

手順

1. クラスター環境で、管理者の kubeconfig ファイルをエクスポートします。

```
$ export KUBECONFIG=$ASSETS_DIR/auth/kubeconfig
```

kubeconfig ファイルには、クライアントを正しいクラスターおよび API サーバーに接続するために CLI で使用されるクラスターについての情報が含まれます。

2. デプロイメント後に作成されたコントロールプレーンおよびコンピュートマシンを表示します。

```
$ oc get nodes
```

3. クラスターのバージョンを表示します。

```
$ oc get clusterversion
```

4. Operator のステータスを表示します。

```
$ oc get clusteroperator
```

5. クラスター内のすべての実行中の Pod を表示します。

```
$ oc get pods -A
```

4.22. ブートストラップマシンの削除

wait-for コマンドがブートストラッププロセスが完了したことを示していることを確認したら、ブートストラップ仮想マシンを削除してコンピュート、メモリー、およびストレージリソースを解放する必要があります。また、ロードバランサーディレクティブからブートストラップマシンの設定を削除します。

手順

1. クラスターからブートストラップマシンを削除するには、以下を実行します。

```
$ ansible-playbook -i inventory.yml retire-bootstrap.yml
```

2. ロードバランサーディレクティブからブートストラップマシンの設定を削除します。

4.23. ワーカーノードの作成およびインストールの完了

ワーカーノードの作成は、コントロールプレーンノードの作成と同様です。ただし、ワーカーノードはクラスターに自動的に参加しません。これらをクラスターに追加するには、ワーカーの保留状態の CSR(証明書署名要求) を確認し、承認します。

最初の要求の承認後に、ワーカーノードがすべて承認されるまで CSR の承認を継続します。このプロセスが完了すると、ワーカーノードは **Ready** になり、Pod がそれらで実行されるようにスケジュールできます。

最後に、コマンドラインを監視し、インストールプロセスが完了するタイミングを確認します。

手順

1. ワーカーノードを作成します。

```
$ ansible-playbook -i inventory.yml workers.yml
```

2. すべての CSR をリスト表示するには、以下を入力します。

```
$ oc get csr -A
```

最終的に、このコマンドはノードごとに 1 つの CSR を表示します。以下に例を示します。

出力例

```
NAME        AGE    SIGNERNAME                                    REQUESTOR
CONDITION
csr-2lnxd   63m    kubernetes.io/kubelet-serving                 system:node:ocp4-lk6b4-
master0.ocp4.example.org                                         Approved,Issued
csr-hff4q   64m    kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper
Approved,Issued
csr-hsn96   60m    kubernetes.io/kubelet-serving                 system:node:ocp4-lk6b4-
master2.ocp4.example.org                                         Approved,Issued
csr-m724n   6m2s   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper    Pending
csr-p4dz2   60m    kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper
```

```
Approved,Issued
csr-t9vfj   60m    kubernetes.io/kubelet-serving                 system:node:ocp4-lk6b4-
master1.ocp4.example.org                                 Approved,Issued
csr-tggtr   61m    kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper
Approved,Issued
csr-wcbrf   7m6s   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
```

3. リストをフィルターし、保留中の CSR のみを表示するには、以下を実行します。

```
$ watch "oc get csr -A | grep pending -i"
```

このコマンドは 2 秒ごとに出力を更新し、保留中の CSR のみを表示します。以下に例を示します。

出力例

```
Every 2.0s: oc get csr -A | grep pending -i

csr-m724n   10m   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
csr-wcbrf   11m   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
```

4. 保留中のそれぞれの要求を検査します。以下に例を示します。

出力例

```
$ oc describe csr csr-m724n
```

出力例

```
Name:               csr-m724n
Labels:             <none>
Annotations:        <none>
CreationTimestamp:  Sun, 19 Jul 2020 15:59:37 +0200
Requesting User:    system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper
Signer:             kubernetes.io/kube-apiserver-client-kubelet
Status:             Pending
Subject:
        Common Name:    system:node:ocp4-lk6b4-worker1.ocp4.example.org
        Serial Number:
        Organization:   system:nodes
Events:  <none>
```

5. CSR 情報が正しい場合は、要求を承認します。

```
$ oc adm certificate approve csr-m724n
```

6. インストールプロセスが完了するまで待機します。

```
$ openshift-install wait-for install-complete --dir $ASSETS_DIR --log-level debug
```

インストールが完了すると、コマンドラインには OpenShift Container Platform Web コンソールの URL と、管理者のユーザー名およびパスワードが表示されます。

4.24. OPENSHIFT CONTAINER PLATFORM の TELEMETRY アクセス

OpenShift Container Platform 4.12 では、クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは OpenShift Cluster Manager Hybrid Cloud Console に登録されます。

OpenShift Cluster Manager インベントリーが正常である (Telemetry によって自動的に維持、または OpenShift Cluster Manager Hybrid Cloud Console を使用して手動で維持) ことを確認した後に、subscription watch を使用 して、アカウントまたはマルチクラスターレベルで OpenShift Container Platform サブスクリプションを追跡します。

関連情報

- Telemetry サービスの詳細は、リモートヘルスモニタリング を参照してください。

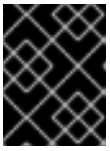
第5章 ネットワークが制限された環境での RHV へのクラスターのインストール

OpenShift Container Platform バージョン 4.12 では、インストールリリースコンテンツの内部ミラーを作成して、ネットワークが制限された環境の Red Hat Virtualization (RHV) にカスタマイズされた OpenShift Container Platform クラスターをインストールできます。

5.1. 前提条件

OpenShift Container Platform クラスターを RHV 環境にインストールするには、以下の要件を満たしている必要があります。

- OpenShift Container Platform のインストールおよび更新 プロセスの詳細を確認した。
- クラスターインストール方法の選択およびそのユーザー向けの準備 を確認した。
- Support Matrix for OpenShift Container Platform on RHV に記載のサポートされるバージョンの組み合わせを使用できる。
- ミラーホストでレジストリーを作成 しており、使用しているバージョンの OpenShift Container Platform の **imageContentSources** データを取得している。



重要

インストールメディアはミラーホストにあるため、そのコンピューターを使用してすべてのインストール手順を完了することができます。

- クラスターの 永続ストレージ をプロビジョニングした。プライベートイメージレジストリーをデプロイするには、ストレージで ReadWriteMany アクセスモードを指定する必要があります。
- クラスターがアクセスを必要とする サイトを許可するようにファイアウォールを設定 している (ファイアウォールを使用し、Telemetry サービスを使用する予定の場合)。

注記

プロキシーを設定する場合は、このサイトリストも確認してください。

5.2. ネットワークが制限された環境でのインストールについて

OpenShift Container Platform 4.12 では、ソフトウェアコンポーネントを取得するためにインターネットへのアクティブな接続を必要としないインストールを実行できます。ネットワークが制限された環境のインストールは、クラスターのインストール先となるクラウドプラットフォームに応じて、インストーラーでプロビジョニングされるインフラストラクチャーまたはユーザーによってプロビジョニングされるインフラストラクチャーを使用して実行できます。

クラウドプラットフォーム上でネットワークが制限されたインストールの実行を選択した場合でも、そのクラウド API へのアクセスが必要になります。Amazon Web Service の Route 53 DNS や IAM サービスなどの一部のクラウド機能には、インターネットアクセスが必要です。ネットワークによっては、ベアメタルハードウェア、Nutanix、または VMware vSphere へのインストールに必要なインターネットアクセスが少なくて済む場合があります。

ネットワークが制限されたインストールを完了するには、OpenShift イメージレジストリーのコンテンツをミラーリングし、インストールメディアを含むレジストリーを作成する必要があります。このミ

ラーは、インターネットと制限されたネットワークの両方にアクセスできるミラーホストで、または制限に対応する他の方法を使用して作成できます。

5.2.1. その他の制限

ネットワークが制限された環境のクラスターには、以下の追加の制限および制約があります。

- **ClusterVersion** ステータスには **Unable to retrieve available updates** エラーが含まれます。
- デフォルトで、開発者カタログのコンテンツは、必要とされるイメージストリームタグにアクセスできないために使用できません。

5.3. OPENSHIFT CONTAINER PLATFORM のインターネットアクセス

OpenShift Container Platform 4.12 では、クラスターのインストールに必要なイメージを取得するために、インターネットにアクセスする必要があります。

インターネットへのアクセスは以下を実行するために必要です。

- OpenShift Cluster Manager Hybrid Cloud Console にアクセスし、インストールプログラムをダウンロードし、サブスクリプション管理を実行します。クラスターにインターネットアクセスがあり、Telemetry を無効にしない場合、そのサービスは有効なサブスクリプションでクラスターを自動的に使用します。
- クラスターのインストールに必要なパッケージを取得するために Quay.io にアクセスします。
- クラスターの更新を実行するために必要なパッケージを取得します。

重要

クラスターでインターネットに直接アクセスできない場合、プロビジョニングする一部のタイプのインフラストラクチャーでネットワークが制限されたインストールを実行できます。このプロセスで、必要なコンテンツをダウンロードし、これを使用してミラーレジストリーにインストールパッケージを設定します。インストールタイプによっては、クラスターのインストール環境でインターネットアクセスが不要となる場合があります。クラスターを更新する前に、ミラーレジストリーのコンテンツを更新します。

5.4. RHV 環境の要件

OpenShift Container Platform バージョン 4.12 クラスターをインストールし、実行するには、RHV 環境が以下の要件を満たしている必要があります。

これらの要件を満たさないと、インストールまたはプロセスが失敗する可能性があります。さらに、これらの要件を満たしていないと、OpenShift Container Platform クラスターはインストールしてから数日または数週間後に失敗する可能性があります。

CPU、メモリー、ストレージリソースについての以下の要件は、インストールプログラムが作成する仮想マシンのデフォルト数で乗算した **デフォルト** 値に基づいています。これらのリソースは、RHV 環境が OpenShift Container Platform 以外の操作に使用するものに **加え**、利用可能でなければなりません。

デフォルトでは、インストールプログラムは 7 つの仮想マシンをインストールプロセスで作成します。まず、ブートストラップ仮想マシンを作成し、OpenShift Container Platform クラスターの残りの部分を作成する間に一時サービスとコントロールプレーンを提供します。インストールプログラムがクラスターの作成を終了すると、ブートストラップマシンが削除され、そのリソースが解放されます。

RHV 環境の仮想マシン数を増やす場合は、リソースを適宜増やす必要があります。

要件

- RHV のバージョンは 4.4 である。
- RHV 環境に **Up** 状態のデータセンターが 1 つあること。
- RHV データセンターに RHV クラスターが含まれていること。
- RHV クラスターに OpenShift Container Platform クラスター専用の以下のリソースがあること。
  - 最小 28 vCPU: インストール時に作成される 7 仮想マシンのそれぞれに 4 vCPU。
  - 以下を含む 112 GiB 以上の RAM。
    - 一時的なコントロールプレーンを提供するブートストラップマシン用に 16 GiB 以上。
    - コントロールプレーンを提供する 3 つのコントロールプレーンマシンのそれぞれに 16 GiB 以上。
    - アプリケーションワークロードを実行する 3 つのコンピュートマシンのそれぞれに 16 GiB 以上。
- RHV ストレージドメインは、これらの etcd バックエンドのパフォーマンス要件 を満たす必要があります。
- 実稼働環境では、各仮想マシンに 120 GiB 以上が必要です。そのため、ストレージドメインはデフォルトの OpenShift Container Platform クラスターに 840 GiB 以上を提供する必要があります。リソースに制約のある環境または非実稼働環境では、各仮想マシンに 32 GiB 以上を指定する必要があるため、ストレージドメインにはデフォルトの OpenShift Container Platform クラスター用に 230 GiB 以上が必要になります。
- インストールおよび更新中に Red Hat Ecosystem Catalog からイメージをダウンロードするには、RHV クラスターがインターネット接続にアクセスできる必要があります。また、サブスクリプションおよびエンタイトルメントプロセスを単純化するために Telemetry サービスにもインターネット接続が必要です。
- RHV クラスターには、RHV Manager の REST API にアクセスできる仮想ネットワークが必要です。インストーラーが作成する仮想マシンが DHCP を使用して IP アドレスを取得するため、DHCP がこのネットワークで有効にされていることを確認します。
- ターゲット RHV クラスターに OpenShift Container Platform クラスターをインストールし、管理するための以下の最小限の権限を持つユーザーアカウントおよびグループ。
  - **DiskOperator**
  - **DiskCreator**
  - **UserTemplateBasedVm**
  - **TemplateOwner**
  - **TemplateCreator**
  - ターゲットクラスターの **ClusterAdmin**

警告

最小権限の原則を適用します。インストールプロセスで RHV で **SuperUser** 権限を持つ管理者アカウントを使用することを避けます。インストールプログラムは、ユーザーが指定する認証情報を、危険にさらされる可能性のある一時的な **ovirt-config.yaml** ファイルに保存します。

5.5. RHV 環境の要件の確認

RHV 環境が OpenShift Container Platform クラスターをインストールし、実行するための要件を満たしていることを確認します。これらの要件を満たさないと、エラーが発生する可能性があります。

重要

これらの要件は、インストールプログラムがコントロールプレーンおよびコンピュートマシンの作成に使用するデフォルトのリソースに基づいています。これらのリソースには、vCPU、メモリー、およびストレージが含まれます。これらのリソースを変更するか、OpenShift Container Platform マシンの数を増やす場合は、これらの要件を適宜調整します。

手順

1. RHV バージョンが OpenShift Container Platform バージョン 4.12 のインストールをサポートしていることを確認します。
   a. RHV Administration Portal の右上にある ? ヘルプアイコンをクリックし、**About** を選択します。
   b. 開かれるウィンドウで、**RHV ソフトウェアのバージョン** をメモします。
   c. RHV のバージョンが 4.4 であることを確認します。サポートされるバージョンの組み合わせについての詳細は、Support Matrix for OpenShift Container Platform on RHV を参照してください。
2. データセンター、クラスター、およびストレージを検査します。
   a. RHV 管理ポータルで、**Compute → Data Centers** をクリックします。
   b. OpenShift Container Platform をインストールする予定のデータセンターにアクセスできることを確認します。
   c. そのデータセンターの名前をクリックします。
   d. データセンターの詳細の **Storage** タブで、OpenShift Container Platform をインストールする予定のストレージドメインが **Active** であることを確認します。
   e. 後で使用できるように **ドメイン名** を記録します。
   f. **空き領域** に 230 GiB 以上あることを確認します。

g. ストレージドメインが これらの etcd バックエンドのパフォーマンス要件 を満たしていることを確認します。これは、fio パフォーマンスベンチマークツールを使用して測定できます。

h. データセンターの詳細で、**Clusters** タブをクリックします。

i. OpenShift Container Platform をインストールする予定の RHV クラスターを見つけます。後で使用できるようにクラスター名を記録します。

3. RHV ホストリソースを確認します。

a. RHV 管理ポータルで、**Compute** > **Clusters** をクリックします。

b. OpenShift Container Platform をインストールする予定のクラスターをクリックします。

c. クラスターの詳細で、**Hosts** タブをクリックします。

d. ホストを検査し、それらに OpenShift Container Platform クラスター **専用** として利用可能な **論理 CPU コア** の合計が 28 つ以上であることを確認します。

e. 後で使用できるように、利用可能な **論理 CPU コア** の数を記録します。

f. これらの CPU コアが分散され、インストール時に作成された 7 つの仮想マシンのそれぞれに 4 つのコアを持たせることができることを確認します。

g. ホストには、以下の OpenShift Container Platform マシンのそれぞれの要件を満たすように **新規仮想マシンをスケジュールするための最大空きメモリー** として 112 GiB があることを確認します。

- ブートストラップマシンに 16 GiB が必要です。
- 3 つのコントロールプレーンマシンのそれぞれに 16 GiB が必要です。
- 3 つのコンピュートマシンのそれぞれに 16 GiB が必要です。

h. 後で使用できるように **新規仮想マシンをスケジュールするための最大空きメモリー** の量を記録します。

4. OpenShift Container Platform をインストールするための仮想ネットワークが RHV Manager の REST API にアクセスできることを確認します。このネットワーク上の仮想マシンから、RHV Manager の REST API に到達するために curl を使用します。

```
$ curl -k -u <username>@<profile>:<password> \ 1
https://<engine-fqdn>/ovirt-engine/api 2
```

1 **<username>** については、RHV で OpenShift Container Platform クラスターを作成および管理する権限を持つ RHV アカウントのユーザー名を指定します。**<profile>** には、ログインプロファイルを指定します。ログインプロファイルは、RHV Administration Portal ログインページに移動し、 **Profile** ドロップダウンリストで確認できます。**<password>** に、そのユーザー名のパスワードを指定します。

2 **<engine-fqdn>** に、RHV 環境の完全修飾ドメイン名を指定します。

以下に例を示します。

```
$ curl -k -u ocpadmin@internal:pw123 \
https://rhv-env.virtlab.example.com/ovirt-engine/api
```

5.6. ユーザーによってプロビジョニングされるインフラストラクチャーのネットワーク要件

すべての Red Hat Enterprise Linux CoreOS (RHCOS) マシンでは、起動時に **initramfs** でネットワークを設定し、Ignition 設定ファイルをフェッチする必要があります。

初回の起動時に、マシンには DHCP サーバーを使用して設定される IP アドレス設定、または必要な起動オプションを指定して静的に設定される IP アドレス設定が必要です。ネットワーク設定の確立後に、マシンは HTTP または HTTPS サーバーから Ignition 設定ファイルをダウンロードします。その後、Ignition 設定ファイルは各マシンの正確な状態を設定するために使用されます。Machine Config Operator はインストール後に、新しい証明書やキーの適用など、マシンへの追加の変更を完了します。

クラスターマシンの長期管理に DHCP サーバーを使用することが推奨されます。DHCP サーバーが永続 IP アドレス、DNS サーバー情報、およびホスト名をクラスターマシンに提供するように設定されていることを確認します。

注記

DHCP サービスが user-provisioned infrastructure で利用できない場合は、IP ネットワーク設定および DNS サーバーのアドレスを RHCOS のインストール時にノードに提供することができます。ISO イメージからインストールしている場合は、ブート引数として渡すことができます。静的 IP プロビジョニングと高度なネットワークオプションの詳細は、**RHCOS のインストールと OpenShift Container Platform ブートストラッププロセスの開始**のセクションを参照してください。

Kubernetes API サーバーはクラスターマシンのノード名を解決できる必要があります。API サーバーおよびワーカーノードが異なるゾーンに置かれている場合、デフォルトの DNS 検索ゾーンを、API サーバーでノード名を解決できるように設定することができます。もう 1 つの実行可能な方法として、ノードオブジェクトとすべての DNS 要求の両方において、ホストを完全修飾ドメイン名で常に参照します。

ファイアウォール

クラスターが必要なサイトにアクセスできるようにファイアウォールを設定します。

以下も参照してください。

- Red Hat Virtualization Manager ファイアウォールの要件
- ホストのファイアウォール要件

DNS

インフラストラクチャーで提供される DNS を設定して、主要なコンポーネントとサービスの正しい解決を許可します。1 つのロードバランサーのみを使用する場合、これらの DNS レコードは同じ IP アドレスを参照できます。

- **api.<cluster_name>.<base_domain>** (内部および外部解決) と、コントロールプレーンマシンのロードバランサーを参照する **api-int.<cluster_name>.<base_domain>** (内部解決) の DNS レコードを作成します。

- Ingress ルーターのロードバランサーを参照する ***.apps.<cluster_name>.<base_domain>** の DNS レコードを作成します。たとえば、コンピュートマシンのポート **443** および **80** などが含まれます。

5.6.1. DHCP を使用したクラスターノードのホスト名の設定

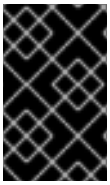
Red Hat Enterprise Linux CoreOS (RHCOS) マシンでは、ホスト名は NetworkManager 経由で設定されます。デフォルトでは、マシンは DHCP 経由でホスト名を取得します。ホスト名が DHCP によって提供されない場合、カーネル引数を介して静的に設定される場合、または別の方法でホスト名が取得される場合は、逆引き DNS ルックアップによって取得されます。逆引き DNS ルックアップは、ネットワークがノードで初期化された後に発生し、解決に時間がかかる場合があります。その他のシステムサービスは、これより前に起動し、ホスト名を **localhost** または同様のものとして検出できます。これを回避するには、DHCP を使用して各クラスターノードのホスト名を指定できます。

また、DHCP を介してホスト名を設定すると、DNS スプリットホライズンが実装されている環境での手動の DNS レコード名設定エラーを回避できます。

5.6.2. ネットワーク接続の要件

OpenShift Container Platform クラスターのコンポーネントが通信できるように、マシン間のネットワーク接続を設定する必要があります。すべてのマシンではクラスターの他のすべてのマシンのホスト名を解決できる必要があります。

このセクションでは、必要なポートの詳細を説明します。



重要

インターネットに接続された OpenShift Container Platform 環境では、プラットフォームコンテナーのイメージをプルし、Red Hat にテレメトリーデータを提供するために、すべてのノードがインターネットにアクセスできる必要があります。

表5.1 すべてのマシンからすべてのマシンへの通信に使用されるポート

プロトコル	ポート	説明
ICMP	該当なし	ネットワーク到達性のテスト
TCP	**1936**	メトリクス
	9000-**9999**	ホストレベルのサービス。ポート **9100**-**9101** のノードエクスポーター、ポート **9099** の Cluster Version Operator が含まれます。
	10250-**10259**	Kubernetes が予約するデフォルトポート
	10256	openshift-sdn
UDP	**4789**	VXLAN
	6081	Geneve

プロトコル	ポート	説明
	9000-**9999**	ポート **9100**-**9101** のノードエクスポーターを含む、ホストレベルのサービス。
	500	IPsec IKE パケット
	4500	IPsec NAT-T パケット
	123	UDP ポート **123** のネットワークタイムプロトコル (NTP) 外部 NTP タイムサーバーが設定されている場合は、UDP ポート **123** を開く必要があります。
TCP/UDP	**30000**-**32767**	Kubernetes ノードポート
ESP	該当なし	IPsec Encapsulating Security Payload (ESP)

表5.2 すべてのマシンからコントロールプレーンへの通信に使用されるポート

プロトコル	ポート	説明
TCP	**6443**	Kubernetes API

表5.3 コントロールプレーンマシンからコントロールプレーンマシンへの通信に使用されるポート

プロトコル	ポート	説明
TCP	**2379**-**2380**	etcd サーバーおよびピアポート

user-provisioned infrastructure の NTP 設定

OpenShift Container Platform クラスターは、デフォルトでパブリック Network Time Protocol (NTP) サーバーを使用するように設定されます。ローカルのエンタープライズ NTP サーバーを使用する必要があるか、クラスターが切断されたネットワークにデプロイされている場合は、特定のタイムサーバーを使用するようにクラスターを設定できます。詳細は、**chrony タイムサービスの設定** のドキュメントを参照してください。

DHCP サーバーが NTP サーバー情報を提供する場合、Red Hat Enterprise Linux CoreOS (RHCOS) マシンの chrony タイムサービスは情報を読み取り、NTP サーバーとクロックを同期できます。

5.7. ユーザーによってプロビジョニングされる DNS 要件

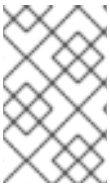
OpenShift Container Platform のデプロイメントでは、以下のコンポーネントに DNS 名前解決が必要です。

- The Kubernetes API
- OpenShift Container Platform のアプリケーションワイルドカード

- ブートストラップ、コントロールプレーンおよびコンピュートマシン

また、Kubernetes API、ブートストラップマシン、コントロールプレーンマシン、およびコンピュートマシンに逆引き DNS 解決も必要です。

DNS A/AAAA または CNAME レコードは名前解決に使用され、PTR レコードは逆引き名前解決に使用されます。ホスト名が DHCP によって提供されていない場合は、Red Hat Enterprise Linux CoreOS (RHCOS) は逆引きレコードを使用してすべてのノードのホスト名を設定するため、逆引きレコードは重要です。さらに、逆引きレコードは、OpenShift Container Platform が動作するために必要な証明書署名要求 (CSR) を生成するために使用されます。

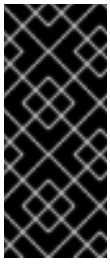


注記

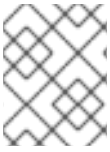
各クラスターノードにホスト名を提供するために DHCP サーバーを使用することが推奨されます。詳細は、**user-provisioned infrastructure に関する DHCP の推奨事項**のセクションを参照してください。

以下の DNS レコードは、user-provisioned OpenShift Container Platform クラスターに必要で、これはインストール前に設定されている必要があります。各レコードで、**<cluster_name>** はクラスター名で、**<base_domain>** は、**install-config.yaml** ファイルに指定するベースドメインです。完全な DNS レコードは **<component>.<cluster_name>.<base_domain>.** の形式を取ります。

表5.4 必要な DNS レコード

コンポーネント	レコード	説明
Kubernetes API	**api.<cluster_name>.<base_domain>.**	API ロードバランサーを特定するための DNS A/AAAA または CNAME レコード、および DNS PTR レコード。これらのレコードは、クラスター外のクライアントおよびクラスター内のすべてのノードで解決できる必要があります。
	api-int.<cluster_name>.<base_domain>.	API ロードバランサーを内部的に識別するための DNS A/AAAA または CNAME レコード、および DNS PTR レコード。これらのレコードは、クラスター内のすべてのノードで解決できる必要があります。  **重要** API サーバーは、Kubernetes に記録されるホスト名でワーカーノードを解決できる必要があります。API サーバーがノード名を解決できない場合、プロキシーされる API 呼び出しが失敗し、Pod からログを取得できなくなる可能性があります。

コンポーネント	レコード	説明
ルート	***.apps.<cluster_name>.<base_domain>.**	アプリケーション Ingress ロードバランサーを参照するワイルドカード DNS A/AAAA または CNAME レコード。アプリケーション Ingress ロードバランサーは、Ingress コントローラー Pod を実行するマシンをターゲットにします。Ingress コントローラー Pod はデフォルトでコンピュートマシンで実行されます。これらのレコードは、クラスター外のクライアントおよびクラスター内のすべてのノードで解決できる必要があります。 たとえば、**console-openshift-console.apps.<cluster_name>.<base_domain>** は、OpenShift Container Platform コンソールへのワイルドカードルートとして使用されます。
ブートストラップマシン	**bootstrap.<cluster_name>.<base_domain>.**	ブートストラップマシンを識別するための DNS A / AAAA または CNAME レコード、および DNS PTR レコード。これらのレコードは、クラスター内のノードで解決できる必要があります。
コントロールプレーンマシン	**<control_plane><n>.<cluster_name>.<base_domain>.**	コントロールプレーンノードの各マシンを特定するための DNS A/AAAA または CNAME レコードおよび DNS PTR レコード。これらのレコードは、クラスター内のノードで解決できる必要があります。
コンピュートマシン	**<compute><n>.<cluster_name>.<base_domain>.**	ワーカーノードの各マシンを特定するための DNS A/AAAA または CNAME レコード、および DNS PTR レコード。これらのレコードは、クラスター内のノードで解決できる必要があります。



注記

OpenShift Container Platform 4.4 以降では、DNS 設定で etcd ホストおよび SRV レコードを指定する必要はありません。

ヒント

dig コマンドを使用して、名前および逆引き名前解決を確認することができます。検証手順の詳細は、**user-provisioned infrastructure の DNS 解決の検証**のセクションを参照してください。

5.7.1. user-provisioned クラスターの DNS 設定の例

このセクションでは、user-provisioned infrastructure に OpenShift Container Platform をデプロイするための DNS 要件を満たす A および PTR レコード設定サンプルを提供します。サンプルは、特定の DNS ソリューションを選択するためのアドバイスを提供することを目的としていません。

この例では、クラスター名は **ocp4** で、ベースドメインは **example.com** です。

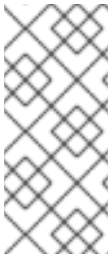
user-provisioned クラスターの DNS A レコードの設定例

BIND ゾーンファイルの以下の例は、user-provisioned クラスターの名前解決の A レコードの例を示しています。

例5.1 DNS ゾーンデータベースのサンプル

```
$TTL 1W
@ IN SOA ns1.example.com. root (
  2019070700 ; serial
  3H  ; refresh (3 hours)
  30M  ; retry (30 minutes)
  2W  ; expiry (2 weeks)
  1W )  ; minimum (1 week)
 IN NS ns1.example.com.
 IN MX 10 smtp.example.com.
;
;
ns1.example.com.  IN A 192.168.1.5
smtp.example.com.  IN A 192.168.1.5
;
helper.example.com.  IN A 192.168.1.5
helper.ocp4.example.com. IN A 192.168.1.5
;
api.ocp4.example.com.  IN A 192.168.1.5 1
api-int.ocp4.example.com. IN A 192.168.1.5 2
;
*.apps.ocp4.example.com. IN A 192.168.1.5 3
;
bootstrap.ocp4.example.com. IN A 192.168.1.96 4
;
control-plane0.ocp4.example.com. IN A 192.168.1.97 5
control-plane1.ocp4.example.com. IN A 192.168.1.98 6
control-plane2.ocp4.example.com. IN A 192.168.1.99 7
;
compute0.ocp4.example.com. IN A 192.168.1.11 8
compute1.ocp4.example.com. IN A 192.168.1.7 9
;
;EOF
```

1. Kubernetes API の名前解決を提供します。レコードは API ロードバランサーの IP アドレスを参照します。
2. Kubernetes API の名前解決を提供します。レコードは API ロードバランサーの IP アドレスを参照し、内部クラスター通信に使用されます。
3. ワイルドカードルートの名前解決を提供します。レコードは、アプリケーション Ingress ロードバランサーの IP アドレスを参照します。アプリケーション Ingress ロードバランサーは、Ingress コントローラー Pod を実行するマシンをターゲットにします。Ingress コントローラー Pod はデフォルトでコンピュートマシンで実行されます。



注記

この例では、同じロードバランサーが Kubernetes API およびアプリケーションの Ingress トラフィックに使用されます。実稼働のシナリオでは、API およびアプリケーション Ingress ロードバランサーを個別にデプロイし、それぞれのロードバランサーインフラストラクチャーを分離してスケーリングすることができます。

4 ブートストラップマシンの名前解決を提供します。

5 6 7 コントロールプレーンマシンの名前解決を提供します。

8 9 コンピュートマシンの名前解決を提供します。

user-provisioned クラスターの DNS PTR レコードの設定例

以下の BIND ゾーンファイルの例では、user-provisioned クラスターの逆引き名前解決の PTR レコードの例を示しています。

例5.2 逆引きレコードの DNS ゾーンデータベースの例

```
$TTL 1W
@ IN SOA ns1.example.com. root (
  2019070700 ; serial
  3H  ; refresh (3 hours)
  30M  ; retry (30 minutes)
  2W  ; expiry (2 weeks)
  1W )  ; minimum (1 week)
 IN NS ns1.example.com.
;
5.1.168.192.in-addr.arpa. IN PTR api.ocp4.example.com. 1
5.1.168.192.in-addr.arpa. IN PTR api-int.ocp4.example.com. 2
;
96.1.168.192.in-addr.arpa. IN PTR bootstrap.ocp4.example.com. 3
;
97.1.168.192.in-addr.arpa. IN PTR control-plane0.ocp4.example.com. 4
98.1.168.192.in-addr.arpa. IN PTR control-plane1.ocp4.example.com. 5
99.1.168.192.in-addr.arpa. IN PTR control-plane2.ocp4.example.com. 6
;
11.1.168.192.in-addr.arpa. IN PTR compute0.ocp4.example.com. 7
7.1.168.192.in-addr.arpa. IN PTR compute1.ocp4.example.com. 8
;
;EOF
```

1 Kubernetes API の逆引き DNS 解決を提供します。PTR レコードは、API ロードバランサーのレコード名を参照します。

2 Kubernetes API の逆引き DNS 解決を提供します。PTR レコードは、API ロードバランサーのレコード名を参照し、内部クラスター通信に使用されます。

3 ブートストラップマシンの逆引き DNS 解決を提供します。

4 5 6 コントロールプレーンマシンの逆引き DNS 解決を提供します。

7 8 コンピュートマシンの逆引き DNS 解決を提供します。

注記

PTR レコードは、OpenShift Container Platform アプリケーションのワイルドカードには必要ありません。

5.7.2. ユーザーによってプロビジョニングされるインフラストラクチャーの負荷分散要件

OpenShift Container Platform をインストールする前に、API およびアプリケーションの Ingress 負荷分散インフラストラクチャーをプロビジョニングする必要があります。実稼働のシナリオでは、API およびアプリケーション Ingress ロードバランサーを個別にデプロイし、それぞれのロードバランサーインフラストラクチャーを分離してスケーリングすることができます。

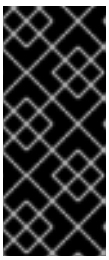
注記

Red Hat Enterprise Linux (RHEL) インスタンスを使用して API およびアプリケーションイングレスロードバランサーをデプロイする場合は、RHEL サブスクリプションを別途購入する必要があります。

負荷分散インフラストラクチャーは以下の要件を満たす必要があります。

1. **API ロードバランサー**: プラットフォームと対話およびプラットフォームを設定するためのユーザー向けの共通のエンドポイントを提供します。以下の条件を設定します。

   - Layer 4 の負荷分散のみ。これは、Raw TCP または SSL パススルーモードと呼ばれます。
   - ステートレス負荷分散アルゴリズム。オプションは、ロードバランサーの実装によって異なります。



重要

API ロードバランサーのセッションの永続性は設定しないでください。Kubernetes API サーバーのセッション永続性を設定すると、OpenShift Container Platform クラスターとクラスター内で実行される Kubernetes API の過剰なアプリケーショントラフィックによりパフォーマンスの問題が発生する可能性があります。

ロードバランサーのフロントとバックの両方で以下のポートを設定します。

表5.5 API ロードバランサー

ポート	バックエンドマシン (プールメンバー)	内部	外部	説明

ポート	バックエンドマシン (プールメンバー)	内部	外部	説明
6443	ブートストラップおよびコントロールプレーン。ブートストラップマシンがクラスターのコントロールプレーンを初期化した後に、ブートストラップマシンをロードバランサーから削除します。API サーバーのヘルスチェックプローブの **/readyz** エンドポイントを設定する必要があります。	X	X	Kubernetes API サーバー
22623	ブートストラップおよびコントロールプレーン。ブートストラップマシンがクラスターのコントロールプレーンを初期化した後に、ブートストラップマシンをロードバランサーから削除します。	X		マシン設定サーバー

注記

ロードバランサーは、API サーバーが **/readyz** エンドポイントをオフにしてからプールから API サーバーインスタンスを削除するまで最大 30 秒かかるように設定する必要があります。**/readyz** の後の時間枠内でエラーが返されたり、正常になったりする場合は、エンドポイントが削除または追加されているはずです。5 秒または 10 秒ごとのプロービングで、2 回連続成功すると正常、3 回連続失敗すると異常と判断する設定は、十分にテストされた値です。

2. **Application Ingress ロードバランサー**: クラスター外から送られるアプリケーショントラフィックの Ingress ポイントを提供します。Ingress ルーターの作業用の設定が OpenShift Container Platform クラスターに必要です。
以下の条件を設定します。

   - Layer 4 の負荷分散のみ。これは、Raw TCP または SSL パススルーモードと呼ばれます。
   - 選択可能なオプションやプラットフォーム上でホストされるアプリケーションの種類に基づいて、接続ベースの永続化またはセッションベースの永続化が推奨されます。

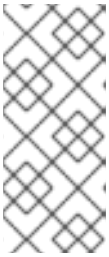
ヒント

クライアントの実際の IP アドレスがアプリケーション Ingress ロードバランサーによって確認できる場合、ソースの IP ベースのセッション永続化を有効にすると、エンドツーエンドの TLS 暗号化を使用するアプリケーションのパフォーマンスを強化できます。

ロードバランサーのフロントとバックの両方で以下のポートを設定します。

表5.6 アプリケーション Ingress ロードバランサー

ポート	バックエンドマシン (プールメンバー)	内部	外部	説明
443	デフォルトで Ingress コントローラー Pod、コンピュート、またはワーカーを実行するマシン。	X	X	HTTPS トラフィック
80	デフォルトで Ingress コントローラー Pod、コンピュート、またはワーカーを実行するマシン。	X	X	HTTP トラフィック



注記

ゼロ (0) コンピュートノードで 3 ノードクラスターをデプロイする場合、Ingress コントローラー Pod はコントロールプレーンノードで実行されます。3 ノードクラスターデプロイメントでは、HTTP および HTTPS トラフィックをコントロールプレーンノードにルーティングするようにアプリケーション Ingress ロードバランサーを設定する必要があります。

5.7.2.1. ユーザーによってプロビジョニングされるクラスターのロードバランサーの設定例

このセクションでは、ユーザーによってプロビジョニングされるクラスターの負荷分散要件を満たす API およびアプリケーション Ingress ロードバランサーの設定例を説明します。この例は、HAProxy ロードバランサーの **/etc/haproxy/haproxy.cfg** 設定です。この例では、特定の負荷分散ソリューションを選択するためのアドバイスを提供することを目的としていません。

この例では、同じロードバランサーが Kubernetes API およびアプリケーションの Ingress トラフィックに使用されます。実稼働のシナリオでは、API およびアプリケーション Ingress ロードバランサーを個別にデプロイし、それぞれのロードバランサーインフラストラクチャーを分離してスケーリングすることができます。

注記

HAProxy をロードバランサーとして使用し、SELinux が **enforcing** に設定されている場合は、**setsebool -P haproxy_connect_any=1** を実行して、HAProxy サービスが設定済みの TCP ポートにバインドできることを確認する必要があります。

例5.3 API およびアプリケーション Ingress ロードバランサーの設定例

```
global
  log         127.0.0.1 local2
  pidfile     /var/run/haproxy.pid
  maxconn     4000
  daemon
defaults
  mode                    http
  log                     global
  option                  dontlognull
  option http-server-close
  option                  redispatch
  retries                 3
  timeout http-request    10s
```

```
    timeout queue           1m
    timeout connect         10s
    timeout client          1m
    timeout server          1m
    timeout http-keep-alive 10s
    timeout check           10s
    maxconn                 3000
  listen api-server-6443 (1)
    bind *:6443
    mode tcp
    option  httpchk GET /readyz HTTP/1.0
    option  log-health-checks
    balance roundrobin
    server bootstrap bootstrap.ocp4.example.com:6443 verify none check check-ssl inter 10s fall 2
  rise 3 backup (2)
    server master0 master0.ocp4.example.com:6443 weight 1 verify none check check-ssl inter 10s
  fall 2 rise 3
    server master1 master1.ocp4.example.com:6443 weight 1 verify none check check-ssl inter 10s
  fall 2 rise 3
    server master2 master2.ocp4.example.com:6443 weight 1 verify none check check-ssl inter 10s
  fall 2 rise 3
  listen machine-config-server-22623 (3)
    bind *:22623
    mode tcp
    server bootstrap bootstrap.ocp4.example.com:22623 check inter 1s backup (4)
    server master0 master0.ocp4.example.com:22623 check inter 1s
    server master1 master1.ocp4.example.com:22623 check inter 1s
    server master2 master2.ocp4.example.com:22623 check inter 1s
  listen ingress-router-443 (5)
    bind *:443
    mode tcp
    balance source
    server worker0 worker0.ocp4.example.com:443 check inter 1s
    server worker1 worker1.ocp4.example.com:443 check inter 1s
  listen ingress-router-80 (6)
    bind *:80
    mode tcp
    balance source
    server worker0 worker0.ocp4.example.com:80 check inter 1s
    server worker1 worker1.ocp4.example.com:80 check inter 1s
```

(1) ポート **6443** は Kubernetes API トラフィックを処理し、コントロールプレーンマシンを参照します。

(2)(4) ブートストラップエントリーは、OpenShift Container Platform クラスターのインストール前に有効にし、ブートストラッププロセスの完了後にそれらを削除する必要があります。

(3) ポート **22623** はマシン設定サーバートラフィックを処理し、コントロールプレーンマシンを参照します。

(5) ポート **443** は HTTPS トラフィックを処理し、Ingress コントローラー Pod を実行するマシンを参照します。Ingress コントローラー Pod はデフォルトでコンピュートマシンで実行されます。

(6) ポート **80** は HTTP トラフィックを処理し、Ingress コントローラー Pod を実行するマシンを参

注記

ゼロ (0) コンピュートノードで 3 ノードクラスターをデプロイする場合、Ingress コントローラー Pod はコントロールプレーンノードで実行されます。3 ノードクラスターデプロイメントでは、HTTP および HTTPS トラフィックをコントロールプレーンノードにルーティングするようにアプリケーション Ingress ロードバランサーを設定する必要があります。

ヒント

HAProxy をロードバランサーとして使用する場合は、HAProxy ノードで **netstat -nltupe** を実行して、ポート **6443**、**22623**、**443**、および **80** で **haproxy** プロセスがリッスンしていることを確認することができます。

5.8. インストールマシンの設定

バイナリー **openshift-install** インストールプログラムおよび Ansible スクリプトを実行するには、Manager 上の RHV 環境および REST API にネットワークでアクセスできるように、RHV Manager または Red Hat Enterprise Linux (RHEL) を設定します。

手順

1. Python3 および Ansible を更新またはインストールします。以下に例を示します。

```
# dnf update python3 ansible
```

2. **python3-ovirt-engine-sdk4** パッケージをインストール して、Python Software Development Kit を取得します。

3. **ovirt.image-template** Ansible ロールをインストールします。RHV Manager およびその他の Red Hat Enterprise Linux (RHEL) マシンでは、このロールは **ovirt-ansible-image-template** パッケージとして提供されます。たとえば、 以下を入力します。

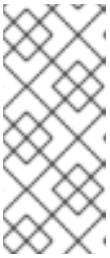
```
# dnf install ovirt-ansible-image-template
```

4. **ovirt.vm-infra** Ansible ロールをインストールします。RHV Manager およびその他の RHEL マシンでは、このロールは **ovirt-ansible-vm-infra** パッケージとして提供されます。

```
# dnf install ovirt-ansible-vm-infra
```

5. 環境変数を作成し、その環境変数に絶対パスまたは相対パスを割り当てます。たとえば、 以下を入力します。

```
$ export ASSETS_DIR=./wrk
```



注記

インストールプログラムはこの変数を使用して、重要なインストール関連のファイルを保存するディレクトリーを作成します。その後、インストールプロセスはこの変数を再利用して、これらのアセットファイルを見つけます。このアセットディレクトリーを削除しないでください。これは、クラスターのアンインストールに必要になります。

5.9. RHV 用の CA 証明書の設定

Red Hat Virtualization (RHV) Manager から CA 証明書をダウンロードし、インストールマシンにこれを設定します。

RHV Manager からの Web サイトまたは **curl** コマンドを使用して、証明書をダウンロードできます。

その後、インストールプログラムに証明書を提供します。

手順

1. 以下の 2 つの方法のいずれかを使用して CA 証明書をダウンロードします。
   - Manager の Web ページ (**https://<engine-fqdn>/ovirt-engine/**) に移動します。次に、**Downloads** で **CA Certificate** のリンクをクリックします。
   - 以下のコマンドを実行します。

     ```
     $ curl -k 'https://<engine-fqdn>/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA' -o /tmp/ca.pem
     ```
     (1)

     (1) **<engine-fqdn>** には、RHV Manager の完全修飾ドメイン名 (例: **rhv-env.virtlab.example.com**) を指定します。

2. ルートレスユーザーに Manager へのアクセスを付与するように CA ファイルを設定します。CA ファイルのパーミッションを 8 進数の **0644** に設定します (シンボリック値: **-rw-r—r--**):

   ```
   $ sudo chmod 0644 /tmp/ca.pem
   ```

3. Linux の場合は、サーバー証明書のディレクトリーに CA 証明書をコピーします。**-p** を使用してパーミッションを保存します。

   ```
   $ sudo cp -p /tmp/ca.pem /etc/pki/ca-trust/source/anchors/ca.pem
   ```

4. オペレーティングシステム用の証明書マネージャーに証明書を追加します。
   - MacOS の場合は、証明書ファイルをダブルクリックして、**Keychain Access** ユーティリティーを使用してファイルを **System** キーチェーンに追加します。
   - Linux の場合は、CA 信頼を更新します。

     ```
     $ sudo update-ca-trust
     ```

注記

独自の認証局を使用する場合は、システムがこれを信頼することを確認します。

関連情報

- 詳細は、RHV ドキュメントの Authentication and Security を参照してください。

5.10. クラスターノードの SSH アクセス用のキーペアの生成

OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定できます。キーは、Ignition 設定ファイルを介して Red Hat Enterprise Linux CoreOS (RHCOS) ノードに渡され、ノードへの SSH アクセスを認証するために使用されます。このキーは各ノードの **core** ユーザーの ~/**.ssh/authorized_keys** リストに追加され、パスワードなしの認証が可能になります。

キーがノードに渡されると、キーペアを使用して RHCOS ノードにユーザー **core** として SSH を実行できます。SSH 経由でノードにアクセスするには、秘密鍵のアイデンティティーをローカルユーザーの SSH で管理する必要があります。

インストールのデバッグまたは障害復旧を実行するためにクラスターノードに対して SSH を実行する場合は、インストールプロセスの間に SSH 公開鍵を指定する必要があります。**./openshift-install gather** コマンドでは、SSH 公開鍵がクラスターノードに配置されている必要もあります。

重要

障害復旧およびデバッグが必要な実稼働環境では、この手順を省略しないでください。

注記

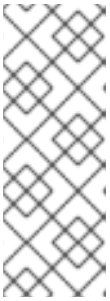
AWS キーペア などのプラットフォームに固有の方法で設定したキーではなく、ローカルキーを使用する必要があります。

手順

1. クラスターノードへの認証に使用するローカルマシンに既存の SSH キーペアがない場合は、これを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ ssh-keygen -t ed25519 -N '' -f <path>/<file_name> 1
```

1 新しい SSH キーのパスとファイル名 (~/**.ssh/id_ed25519** など) を指定します。既存のキーペアがある場合は、公開鍵が ~/**.ssh** ディレクトリーにあることを確認します。



注記

FIPS で検証済みまたは進行中のモジュール (Modules in Process) 暗号ライブラリーを使用する OpenShift Container Platform クラスターを **x86_64**、**ppc64le**、および **s390x** アーキテクチャーにインストールする予定の場合は、**ed25519** アルゴリズムを使用するキーは作成しないでください。代わりに、**rsa** アルゴリズムまたは **ecdsa** アルゴリズムを使用するキーを作成します。

2. 公開 SSH キーを表示します。

```
$ cat <path>/<file_name>.pub
```

たとえば、次のコマンドを実行して ~/**.ssh/id_ed25519.pub** 公開鍵を表示します。

```
$ cat ~/.ssh/id_ed25519.pub
```

3. ローカルユーザーの SSH エージェントに SSH 秘密鍵 ID が追加されていない場合は、それを追加します。キーの SSH エージェント管理は、クラスターノードへのパスワードなしの SSH 認証、または **./openshift-install gather** コマンドを使用する場合は必要になります。

注記

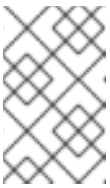
一部のディストリビューションでは、~/**.ssh/id_rsa** および ~/**.ssh/id_dsa** などのデフォルトの SSH 秘密鍵のアイデンティティーは自動的に管理されます。

a. **ssh-agent** プロセスがローカルユーザーに対して実行されていない場合は、バックグラウンドタスクとして開始します。

```
$ eval "$(ssh-agent -s)"
```

出力例

```
Agent pid 31874
```



注記

クラスターが FIPS モードにある場合は、FIPS 準拠のアルゴリズムのみを使用して SSH キーを生成します。鍵は RSA または ECDSA のいずれかである必要があります。

4. SSH プライベートキーを **ssh-agent** に追加します。

```
$ ssh-add <path>/<file_name> 1
```

1 ~/**.ssh/id_ed25519** などの、SSH プライベートキーのパスおよびファイル名を指定します。

出力例

```
Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

次のステップ

- OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定します。

5.11. ANSIBLE PLAYBOOK のダウンロード

RHV に OpenShift Container Platform バージョン 4.12 をインストールするために Ansible Playbook をダウンロードします。

手順

- インストールマシンで、以下のコマンドを実行します。

```
$ mkdir playbooks
```

```
$ cd playbooks
```

```
$  xargs -n 1 curl -O <<< '
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/bootstrap.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/common-auth.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/create-templates-and-vms.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/inventory.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/masters.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/retire-bootstrap.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/retire-masters.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/retire-workers.yml
      https://raw.githubusercontent.com/openshift/installer/release-4.12/upi/ovirt/workers.yml'
```

次のステップ

- これらの Ansible Playbook をダウンロードしたら、インストールプログラムを実行してインストール設定ファイルを作成する前に、アセットディレクトリーの環境変数を作成し、**inventory.yml** ファイルをカスタマイズする必要もあります。

5.12. INVENTORY.YML ファイル

inventory.yml ファイルを使用して、インストールする OpenShift Container Platform クラスターの各種の要素を定義し、作成します。これには、Red Hat Enterprise Linux CoreOS(RHCOS) イメージ、仮想マシンテンプレート、ブートストラップマシン、コントロールプレーンノード、ワーカーノードなどの要素が含まれます。また、**inventory.yml** を使用してクラスターを破棄します。

以下の **inventory.yml** の例は、パラメーターとそれらのデフォルト値を示しています。これらのデフォルト値の量と数は、RHV 環境で実稼働用の OpenShift Container Platform クラスターを実行するための要件を満たしています。

inventory.yml ファイルの例

```
---
all:
  vars:

    ovirt_cluster: "Default"
    ocp:
      assets_dir: "{{ lookup('env', 'ASSETS_DIR') }}"
      ovirt_config_path: "{{ lookup('env', 'HOME') }}/.ovirt/ovirt-config.yaml"

    # ---
    # {op-system} section
    # ---
    rhcos:
      image_url: "https://mirror.openshift.com/pub/openshift-v4/dependencies/rhcos/4.12/latest/rhcos-
openstack.x86_64.qcow2.gz"
      local_cmp_image_path: "/tmp/rhcos.qcow2.gz"
      local_image_path: "/tmp/rhcos.qcow2"

    # ---
    # Profiles section
    # ---
    control_plane:
      cluster: "{{ ovirt_cluster }}"
      memory: 16GiB
      sockets: 4
      cores: 1
      template: rhcos_tpl
      operating_system: "rhcos_x64"
      type: high_performance
      graphical_console:
        headless_mode: false
        protocol:
        - spice
        - vnc
      disks:
      - size: 120GiB
        name: os
        interface: virtio_scsi
        storage_domain: depot_nvme
      nics:
      - name: nic1
        network: lab
        profile: lab

    compute:
      cluster: "{{ ovirt_cluster }}"
      memory: 16GiB
      sockets: 4
      cores: 1
      template: worker_rhcos_tpl
      operating_system: "rhcos_x64"
      type: high_performance
      graphical_console:
        headless_mode: false
```

```
    protocol:
    - spice
    - vnc
  disks:
  - size: 120GiB
    name: os
    interface: virtio_scsi
    storage_domain: depot_nvme
  nics:
  - name: nic1
    network: lab
    profile: lab

# ---
# Virtual machines section
# ---
vms:
- name: "{{ metadata.infraID }}-bootstrap"
  ocp_type: bootstrap
  profile: "{{ control_plane }}"
  type: server
- name: "{{ metadata.infraID }}-master0"
  ocp_type: master
  profile: "{{ control_plane }}"
- name: "{{ metadata.infraID }}-master1"
  ocp_type: master
  profile: "{{ control_plane }}"
- name: "{{ metadata.infraID }}-master2"
  ocp_type: master
  profile: "{{ control_plane }}"
- name: "{{ metadata.infraID }}-worker0"
  ocp_type: worker
  profile: "{{ compute }}"
- name: "{{ metadata.infraID }}-worker1"
  ocp_type: worker
  profile: "{{ compute }}"
- name: "{{ metadata.infraID }}-worker2"
  ocp_type: worker
  profile: "{{ compute }}"
```

重要

Enter から始まる説明のあるパラメーターの値を入力します。それ以外の場合は、デフォルト値を使用するか、新しい値に置き換えることができます。

General セクション

- **ovirt_cluster**: OpenShift Container Platform クラスターをインストールする既存の RHV クラスターの名前を入力します。
- **ocp.assets_dir**: **openshift-install** インストールプログラムが生成するファイルを保存するために作成するディレクトリーのパス。

- **ocp.ovirt_config_path**: インストールプログラムが生成する **ovirt-config.yaml** ファイルのパス (**./wrk/install-config.yaml** など)。このファイルには、Manager の REST API との対話に必要な認証情報が含まれます。

Red Hat Enterprise Linux CoreOS (RHCOS) セクション

- **image_url**: ダウンロード用に指定した RHCOS イメージの URL を入力します。
- **local_cmp_image_path**: 圧縮された RHCOS イメージのローカルダウンロードディレクトリーのパス。
- **local_image_path**: デプロイメントした RHCOS イメージのローカルディレクトリーのパス。

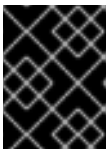
Profiles セクション

このセクションは、2 つのプロファイルで設定されます。

- **control_plane**: ブートストラップおよびコントロールプレーンノードのプロファイル。
- **compute**: コンピュートプレーン内のワーカーノードのプロファイル。

これらのプロファイルには以下のパラメーターが含まれます。パラメーターのデフォルト値は、実稼働クラスターを実行するために必要な最小要件を満たします。これらの値は、ワークロードの要件に応じて増減したり、カスタマイズしたりできます。

- **cluster**: 値は、General セクションの **ovirt_cluster** からクラスター名を取得します。
- **memory**: 仮想マシンに必要なメモリーの量 (GB)。
- **sockets**: 仮想マシンのソケット数。
- **cores**: 仮想マシンのコア数。
- **template**: 仮想マシンテンプレートの名前。複数のクラスターをインストールする計画があり、これらのクラスターが異なる仕様が含まれるテンプレートを使用する場合には、テンプレート名の先頭にクラスターの ID を付けます。
- **operating_system**: 仮想マシンのゲストオペレーティングシステムのタイプ。oVirt/RHV バージョン 4.4 では、**Ignition script** の値を仮想マシンに渡すことができるようにするために、この値を **rhcos_x64** にする必要があります。
- **type**: 仮想マシンのタイプとして **server** を入力します。



重要

type パラメーターの値を **high_performance** から **server** に変更する必要があります。

- **disks**: ディスクの仕様。**control_plane** と **compute** ノードには、異なるストレージドメインを設定できます。
- **size**: ディスクの最小サイズ。
- **name**: RHV のターゲットクラスターに接続されたディスクの名前を入力します。

- **interface**: 指定したディスクのインターフェイスタイプを入力します。
- **storage_domain**: 指定したディスクのストレージドメインを入力します。
- **nics**: 仮想マシンが使用する **name** および **network** を入力します。仮想ネットワークインターフェイスプロファイルを指定することもできます。デフォルトでは、NIC は oVirt/RHV MAC プールから MAC アドレスを取得します。

仮想マシンセクション

この最後のセクション **vms** は、クラスターで作成およびデプロイする予定の仮想マシンを定義します。デフォルトで、実稼働環境用の最小数のコントロールプレーンおよびワーカーノードが提供されます。

vms には 3 つの必須要素が含まれます。

- **name**: 仮想マシンの名前。この場合、**metadata.infraID** は、仮想マシン名の先頭に **metadata.yml** ファイルのインフラストラクチャー ID を付けます。
- **ocp_type**: OpenShift Container Platform クラスター内の仮想マシンのロール。使用できる値は **bootstrap**、**master**、**worker** です。
- **profile**: それぞれの仮想マシンが仕様を継承するプロファイルの名前。この例で使用可能な値は **control_plane** または **compute** です。
  仮想マシンがプロファイルから継承する値を上書きできます。これを実行するには、**inventory.yml** の仮想マシンに profile 属性の名前を追加し、これに上書きする値を割り当てます。この例を確認するには、直前の **inventory.yml** の例の **name: "{{ metadata.infraID }}-bootstrap"** 仮想マシンを検査します。これには値が **server** の **type** 属性があり、この仮想マシンがそれ以外の場合に **control_plane** プロファイルから継承する **type** 属性の値を上書きします。

メタデータ変数

仮想マシンの場合、**metadata.infraID** は、仮想マシンの名前の先頭に、Ignition ファイルのビルド時に作成する **metadata.json** ファイルのインフラストラクチャー ID を付けます。

Playbook は以下のコードを使用して、**ocp.assets_dir** にある特定のファイルから **infraID** を読み取ります。

```
---
- name: include metadata.json vars
  include_vars:
    file: "{{ ocp.assets_dir }}/metadata.json"
    name: metadata

  ...
```

5.13. RHCOS イメージ設定の指定

inventory.yml ファイルの Red Hat Enterprise Linux CoreOS (RHCOS) イメージ設定を更新します。後にこのファイルを Playbook のいずれかとして実行すると、圧縮された Red Hat Enterprise Linux CoreOS (RHCOS) イメージが **image_url** URL から **local_cmp_image_path** ディレクトリーにダウンロードされます。次に Playbook はイメージを **local_image_path** ディレクトリーにデプロイメントし、これを使用して oVirt/RHV テンプレートを作成します。

手順

1. インストールする OpenShift Container Platform バージョンの RHCOS イメージダウンロードページを見つけます (例: /pub/openshift-v4/dependencies/rhcos/latest/latest のインデックス)。
2. そのダウンロードページから、**https://mirror.openshift.com/pub/openshift-v4/dependencies/rhcos/4.12/latest/rhcos-openstack.x86_64.qcow2.gz** などの OpenStack **qcow2** イメージの URL をコピーします。
3. 先のステップでダウンロードした **inventory.yml** Playbook を編集します。この中で、URL を **image_url** の値として貼り付けます。以下に例を示します。

```
rhcos:
  "https://mirror.openshift.com/pub/openshift-v4/dependencies/rhcos/4.12/latest/rhcos-openstack.x86_64.qcow2.gz"
```

5.14. インストール設定ファイルの作成

インストールプログラム **openshift-install** を実行し、先に指定または収集した情報でプロンプトに応答し、インストール設定ファイルを作成します。

プロンプトに応答すると、インストールプログラムは、以前に指定したアセットディレクトリーの **install-config.yaml** ファイルの初期バージョンを作成します (例: **./wrk/install-config.yaml**)。

インストールプログラムは、Manager に到達して REST API を使用するために必要なすべての接続パラメーターが含まれる **$HOME/.ovirt/ovirt-config.yaml** ファイルも作成します。

注: インストールプロセスでは、**Internal API virtual IP** および **Ingress virtual IP** などの一部のパラメーターに指定する値を使用しません。それらの値はインフラストラクチャー DNS にすでに設定されているためです。

また、**oVirt cluster**、 **oVirt storage**、および **oVirt network** などの値のような **inventory.yml** のパラメーターに指定する値を使用します。また、スクリプトを使用して **install-config.yaml** の同じ値を削除するか、これを前述の **virtual IPs** に置き換えます。

手順

1. インストールプログラムを実行します。

```
$ openshift-install create install-config --dir $ASSETS_DIR
```

2. インストールプログラムのプロンプトに応答し、システムに関する情報を提供します。

出力例

```
? SSH Public Key /home/user/.ssh/id_dsa.pub
? Platform <ovirt>
? Engine FQDN[:PORT] [? for help] <engine.fqdn>
? Enter ovirt-engine username <ocpadmin@internal>
? Enter password <******>
? oVirt cluster <cluster>
? oVirt storage <storage>
? oVirt network <net>
```

```
? Internal API virtual IP <172.16.0.252>
? Ingress virtual IP <172.16.0.251>
? Base Domain <example.org>
? Cluster Name <ocp4>
? Pull Secret [? for help] <********>
```

```
? SSH Public Key /home/user/.ssh/id_dsa.pub
? Platform <ovirt>
? Engine FQDN[:PORT] [? for help] <engine.fqdn>
? Enter ovirt-engine username <ocpadmin@internal>
? Enter password <******>
? oVirt cluster <cluster>
? oVirt storage <storage>
? oVirt network <net>
? Internal API virtual IP <172.16.0.252>
? Ingress virtual IP <172.16.0.251>
? Base Domain <example.org>
? Cluster Name <ocp4>
? Pull Secret [? for help] <********>
```

Internal API virtual IP および **Ingress virtual IP** について、DNS サービスの設定時に指定した IP アドレスを指定します。

さらに、**oVirt cluster** および **Base Domain** プロンプトに対して入力する値は REST API および作成するアプリケーションの URL の一部を設定します (例: **https://api.ocp4.example.org:6443/** and **https://console-openshift-console.apps.ocp4.example.org**)。

Red Hat OpenShift Cluster Manager からプルシークレット を取得できます。

5.15. RHV のサンプル INSTALL-CONFIG.YAML ファイル

install-config.yaml ファイルをカスタマイズして、OpenShift Container Platform クラスターのプラットフォームについての詳細を指定するか、必要なパラメーターの値を変更することができます。

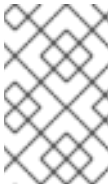
```
apiVersion: v1
baseDomain: example.com (1)
compute: (2)
- hyperthreading: Enabled (3)
  name: worker
  replicas: 0 (4)
controlPlane: (5)
  hyperthreading: Enabled (6)
  name: master
  replicas: 3 (7)
metadata:
  name: test (8)
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14 (9)
    hostPrefix: 23 (10)
  networkType: OVNKubernetes (11)
  serviceNetwork: (12)
  - 172.30.0.0/16
```

```
platform:
  none: {} 13
fips: false 14
pullSecret: '{"auths": ...}' 15
sshKey: 'ssh-ed25519 AAAA...' 16
```

1 クラスターのベースドメイン。すべての DNS レコードはこのベースのサブドメインである必要があり、クラスター名が含まれる必要があります。

2 5 **controlPlane** セクションは単一マッピングですが、**compute** セクションはマッピングのシーケンスになります。複数の異なるデータ構造の要件を満たすには、**compute** セクションの最初の行はハイフン **-** で始め、**controlPlane** セクションの最初の行はハイフンで始めることができません。1 つのコントロールプレーンプールのみが使用されます。

3 6 同時マルチスレッド (SMT) またはハイパースレッディングを有効/無効にするかどうかを指定します。デフォルトでは、SMT はマシンのコアのパフォーマンスを上げるために有効にされます。パラメーター値を **Disabled** に設定するとこれを無効にすることができます。SMT を無効にする場合、これをすべてのクラスターマシンで無効にする必要があります。これにはコントロールプレーンとコンピュートマシンの両方が含まれます。



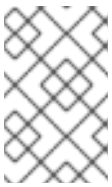
注記

同時マルチスレッド (SMT) はデフォルトで有効になっています。SMT が BIOS 設定で有効になっていない場合は、**hyperthreading** パラメーターは効果がありません。

重要

BIOS または **install-config.yaml** ファイルであるかに関係なく **hyperthreading** を無効にする場合、容量計画においてマシンのパフォーマンスの大幅な低下が考慮に入れられていることを確認します。

4 OpenShift Container Platform を user-provisioned infrastructure にインストールする場合は、この値を **0** に設定する必要があります。installer-provisioned installation では、パラメーターはクラスターが作成し、管理するコンピュートマシンの数を制御します。user-provisioned installation では、クラスターのインストールの終了前にコンピュートマシンを手動でデプロイする必要があります。



注記

3 ノードクラスターをインストールする場合は、Red Hat Enterprise Linux CoreOS (RHCOS) マシンをインストールする際にコンピュートマシンをデプロイしないでください。

7 クラスターに追加するコントロールプレーンマシンの数。クラスターをこれらの値をクラスターの etcd エンドポイント数として使用するため、値はデプロイするコントロールプレーンマシンの数に一致する必要があります。

8 DNS レコードに指定したクラスター名。

9 Pod IP アドレスの割り当てに使用する IP アドレスのブロック。このブロックは既存の物理ネットワークと重複できません。これらの IP アドレスは Pod ネットワークに使用されます。外部ネットワークから Pod にアクセスする必要がある場合、ロードバランサーおよびルーターを、トラ

フィックを管理するように設定する必要があります。

注記

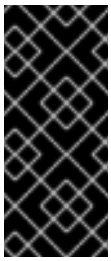
クラス E の CIDR 範囲は、将来の使用のために予約されています。クラス E CIDR 範囲を使用するには、ネットワーク環境がクラス E CIDR 範囲内の IP アドレスを受け入れるようにする必要があります。

10 それぞれの個別ノードに割り当てるサブネット接頭辞長。たとえば、**hostPrefix** が **23** に設定されている場合、各ノードに指定の **cidr** から **/23** サブネットが割り当てられます。これにより、510 (2^(32 - 23) - 2) Pod IP アドレスが許可されます。外部ネットワークからのノードへのアクセスを提供する必要がある場合には、ロードバランサーおよびルーターを、トラフィックを管理するように設定します。

11 インストールするクラスターネットワークプラグイン。サポートされている値は **OVNKubernetes** と **OpenShiftSDN** です。デフォルトの値は **OVNkubernetes** です。

12 サービス IP アドレスに使用する IP アドレスプール。1 つの IP アドレスプールのみを入力できます。このブロックは既存の物理ネットワークと重複できません。外部ネットワークからサービスにアクセスする必要がある場合、ロードバランサーおよびルーターを、トラフィックを管理するように設定します。

13 プラットフォームを **none** に設定する必要があります。RHV インフラストラクチャー用に追加のプラットフォーム設定変数を指定できません。



重要

プラットフォームタイプ **none** でインストールされたクラスターは、Machine API を使用したコンピューティングマシンの管理など、一部の機能を使用できません。この制限は、クラスターに接続されている計算マシンが、通常はこの機能をサポートするプラットフォームにインストールされている場合でも適用されます。このパラメーターは、インストール後に変更することはできません。

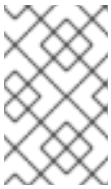
14 FIPS モードを有効または無効にするかどうか。デフォルトでは、FIPS モードは有効にされません。FIPS モードが有効にされている場合、OpenShift Container Platform が実行される Red Hat Enterprise Linux CoreOS (RHCOS) マシンがデフォルトの Kubernetes 暗号スイートをバイパスし、代わりに RHCOS で提供される暗号モジュールを使用します。

重要

クラスターで FIPS モードを有効にするには、FIPS モードで動作するように設定された Red Hat Enterprise Linux (RHEL) コンピューターからインストールプログラムを実行する必要があります。RHEL での FIPS モードの設定の詳細は、FIPS モードでのシステムのインストール を参照してください。FIPS 検証済み/Modules In Process 暗号ライブラリーの使用は、**x86_64**、**ppc64le**、および **s390x** アーキテクチャー上の OpenShift Container Platform デプロイメントでのみサポートされます。

15 Red Hat OpenShift Cluster Manager からのプルシークレット 。このプルシークレットを使用し、OpenShift Container Platform コンポーネントのコンテナーイメージを提供する Quay.io など、組み込まれた各種の認証局によって提供されるサービスで認証できます。

16 Red Hat Enterprise Linux CoreOS (RHCOS) の **core** ユーザーの SSH 公開鍵。



注記

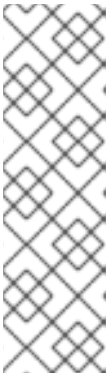
インストールのデバッグまたは障害復旧を実行する必要のある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

5.15.1. インストール時のクラスター全体のプロキシーの設定

実稼働環境では、インターネットへの直接アクセスを拒否し、代わりに HTTP または HTTPS プロキシーを使用することができます。プロキシー設定を **install-config.yaml** ファイルで行うことにより、新規の OpenShift Container Platform クラスターをプロキシーを使用するように設定できます。

前提条件

- 既存の **install-config.yaml** ファイルがある。
- クラスターがアクセスする必要のあるサイトを確認済みで、それらのいずれかがプロキシーをバイパスする必要があるかどうかを判別している。デフォルトで、すべてのクラスター Egress トラフィック (クラスターをホストするクラウドに関するクラウドプロバイダー API に対する呼び出しを含む) はプロキシーされます。プロキシーを必要に応じてバイパスするために、サイトを **Proxy** オブジェクトの **spec.noProxy** フィールドに追加している。



注記

Proxy オブジェクトの **status.noProxy** フィールドには、インストール設定の **networking.machineNetwork[].cidr**、**networking.clusterNetwork[].cidr**、および **networking.serviceNetwork[]** フィールドの値が設定されます。

Amazon Web Services (AWS)、Google Cloud、Microsoft Azure、および Red Hat OpenStack Platform (RHOSP)へのインストールの場合、**Proxy** オブジェクトの **status.noProxy** フィールドには、インスタンスメタデータのエンドポイント(**169.254.169.254**)も設定されます。

手順

1. **install-config.yaml** ファイルを編集し、プロキシー設定を追加します。以下に例を示します。

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

1. クラスター外の HTTP 接続を作成するために使用するプロキシー URL。URL スキームは **http** である必要があります。

2 クラスター外で HTTPS 接続を作成するために使用するプロキシー URL。

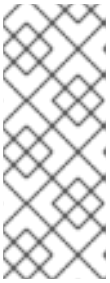
3 プロキシーから除外するための宛先ドメイン名、IP アドレス、または他のネットワーク CIDR のコンマ区切りのリスト。サブドメインのみと一致するように、ドメインの前に **.** を付けます。たとえば、**.y.com** は **x.y.com** に一致しますが、**y.com** には一致しません。***** を使用し、すべての宛先のプロキシーをバイパスします。

4 指定されている場合、インストールプログラムは HTTPS 接続のプロキシーに必要な 1 つ以上の追加の CA 証明書が含まれる **user-ca-bundle** という名前の設定マップを **openshift-config** namespace に生成します。次に Cluster Network Operator は、これらのコンテンツを Red Hat Enterprise Linux CoreOS (RHCOS) 信頼バンドルにマージする **trusted-ca-bundle** 設定マップを作成し、この設定マップは **Proxy** オブジェクトの **trustedCA** フィールドで参照されます。**additionalTrustBundle** フィールドは、プロキシーのアイデンティティー証明書が RHCOS 信頼バンドルからの認証局によって署名されない限り必要になります。

5 オプション: **trustedCA** フィールドの **user-ca-bundle** 設定マップを参照する **Proxy** オブジェクトの設定を決定するポリシー。許可される値は **Proxyonly** および **Always** です。**Proxyonly** を使用して、**http/https** プロキシーが設定されている場合にのみ **user-ca-bundle** 設定マップを参照します。**Always** を使用して、常に **user-ca-bundle** 設定マップを参照します。デフォルト値は **Proxyonly** です。

注記

インストールプログラムは、プロキシーの **readinessEndpoints** フィールドをサポートしません。



注記

インストーラーがタイムアウトした場合は、インストーラーの **wait-for** コマンドを使用してデプロイメントを再起動してからデプロイメントを完了します。以下に例を示します。

```
$ ./openshift-install wait-for install-complete --log-level debug
```

2. ファイルを保存し、OpenShift Container Platform のインストール時にこれを参照します。

インストールプログラムは、指定の **install-config.yaml** ファイルのプロキシー設定を使用する **cluster** という名前のクラスター全体のプロキシーを作成します。プロキシー設定が指定されていない場合、**cluster Proxy** オブジェクトが依然として作成されますが、これには **spec** がありません。

注記

cluster という名前の **Proxy** オブジェクトのみがサポートされ、追加のプロキシーを作成することはできません。

5.16. INSTALL-CONFIG.YAML のカスタマイズ

ここでは、3 つの python スクリプトを使用して、インストールプログラムのデフォルト動作の一部を上書きします。

- デフォルトでは、インストールプログラムはマシン API を使用してノードを作成します。このデフォルトの動作を上書きするには、コンピュートノードの数をゼロ (0) レプリカに設定します。後に Ansible Playbook を使用してコンピュートノードを作成します。
- デフォルトでは、インストールプログラムはノードのマシンネットワークの IP 範囲を設定します。このデフォルトの動作を上書きするには、インフラストラクチャーに一致するように IP 範囲を設定します。
- デフォルトでは、インストールプログラムはプラットフォームを **ovirt** に設定します。ただし、ユーザーによってプロビジョニングされるインフラストラクチャーにクラスターをインストールすることは、ベアメタルにクラスターをインストールすることに似ています。したがって、ovirt プラットフォームセクションを **install-config.yaml** から削除し、プラットフォームを **none** に変更します。代わりに、**inventory.yml** を使用して、必要な設定をすべて指定します。

注記

これらのスニペットは Python 3 および Python 2 で動作します。

手順

1. コンピュートノードの数をゼロ (0) レプリカに設定します。

```
$ python3 -c 'import os, yaml
path = "%s/install-config.yaml" % os.environ["ASSETS_DIR"]
conf = yaml.safe_load(open(path))
conf["compute"][0]["replicas"] = 0
open(path, "w").write(yaml.dump(conf, default_flow_style=False))'
```

2. マシンネットワークの IP 範囲を設定します。たとえば、範囲を **172.16.0.0/16** に設定するには、以下を実行します。

```
$ python3 -c 'import os, yaml
path = "%s/install-config.yaml" % os.environ["ASSETS_DIR"]
conf = yaml.safe_load(open(path))
conf["networking"]["machineNetwork"][0]["cidr"] = "172.16.0.0/16"
open(path, "w").write(yaml.dump(conf, default_flow_style=False))'
```

3. **ovirt** セクションを削除し、プラットフォームを **none** に変更します。

```
$ python3 -c 'import os, yaml
path = "%s/install-config.yaml" % os.environ["ASSETS_DIR"]
conf = yaml.safe_load(open(path))
platform = conf["platform"]
del platform["ovirt"]
platform["none"] = {}
open(path, "w").write(yaml.dump(conf, default_flow_style=False))'
```

警告

Red Hat Virtualization は現在、oVirt プラットフォーム上にあるユーザーによってプロビジョニングされるインフラストラクチャーでのインストールをサポートしていません。そのため、プラットフォームを **none** に設定し、OpenShift Container Platform が各ノードをベアメタルノードとして、およびクラスターをベアメタルクラスターとして識別できるようにします。これは、任意のプラットフォームにクラスターをインストール するのと同じであり、次の制限があります。

1. クラスタープロバイダーがないため、各マシンを手動で追加する必要があり、ノードスケーリング機能はありません。
2. oVirt CSI ドライバーはインストールされず、CSI 機能はありません。

5.17. マニフェストファイルの生成

インストールプログラムを使用して、アセットディレクトリーにマニフェストファイルのセットを生成します。

マニフェストファイルを生成するコマンドにより、**install-config.yaml** ファイルを使用する前に警告メッセージが表示されます。

install-config.yaml ファイルを再利用する予定の場合には、マニフェストファイルを生成する前にバックアップしてからバックアップコピーを作成してください。

手順

1. オプション: **install-config.yaml** ファイルのバックアップコピーを作成します。

```
$ cp install-config.yaml install-config.yaml.backup
```

2. アセットディレクトリーにマニフェストのセットを生成します。

```
$ openshift-install create manifests --dir $ASSETS_DIR
```

このコマンドにより、以下の情報が表示されます。

出力例

```
INFO Consuming Install Config from target directory
WARNING Making control-plane schedulable by setting MastersSchedulable to true for
Scheduler cluster settings
```

このコマンドにより、以下のマニフェストファイルが生成されます。

出力例

```
$ tree
.
```

```
└── wrk
    ├── manifests
    │   ├── 04-openshift-machine-config-operator.yaml
    │   ├── cluster-config.yaml
    │   ├── cluster-dns-02-config.yml
    │   ├── cluster-infrastructure-02-config.yml
    │   ├── cluster-ingress-02-config.yml
    │   ├── cluster-network-01-crd.yml
    │   ├── cluster-network-02-config.yml
    │   ├── cluster-proxy-01-config.yaml
    │   ├── cluster-scheduler-02-config.yml
    │   ├── cvo-overrides.yaml
    │   ├── etcd-ca-bundle-configmap.yaml
    │   ├── etcd-client-secret.yaml
    │   ├── etcd-host-service-endpoints.yaml
    │   ├── etcd-host-service.yaml
    │   ├── etcd-metric-client-secret.yaml
    │   ├── etcd-metric-serving-ca-configmap.yaml
    │   ├── etcd-metric-signer-secret.yaml
    │   ├── etcd-namespace.yaml
    │   ├── etcd-service.yaml
    │   ├── etcd-serving-ca-configmap.yaml
    │   ├── etcd-signer-secret.yaml
    │   ├── kube-cloud-config.yaml
    │   ├── kube-system-configmap-root-ca.yaml
    │   ├── machine-config-server-tls-secret.yaml
    │   └── openshift-config-secret-pull-secret.yaml
    └── openshift
        ├── 99_kubeadmin-password-secret.yaml
        ├── 99_openshift-cluster-api_master-user-data-secret.yaml
        ├── 99_openshift-cluster-api_worker-user-data-secret.yaml
        ├── 99_openshift-machineconfig_99-master-ssh.yaml
        ├── 99_openshift-machineconfig_99-worker-ssh.yaml
        └── openshift-install-manifests.yaml
```

次のステップ

- コントロールプレーンノードをスケジュール対象外にします。

5.18. コントロールプレーンノードのスケジュール対象外の設定

コントロールプレーンマシンを手動で作成し、デプロイしているので、コントロールプレーンノードをスケジュール対象外にするようにマニフェストファイルを設定する必要があります。

手順

1. コントロールプレーンノードをスケジュール対象外にするには、以下を入力します。

```
$ python3 -c 'import os, yaml
path = "%s/manifests/cluster-scheduler-02-config.yml" % os.environ["ASSETS_DIR"]
data = yaml.safe_load(open(path))
data["spec"]["mastersSchedulable"] = False
open(path, "w").write(yaml.dump(data, default_flow_style=False))'
```

5.19. IGNITION ファイルのビルド

生成および変更したマニフェストファイルから Ignition ファイルを作成するには、インストールプログラムを実行します。このアクションにより、Ignition ファイルをフェッチし、ノードを作成するために必要な設定を実行する Red Hat Enterprise Linux CoreOS (RHCOS) マシン **initramfs** が作成されます。

Ignition ファイルのほかに、インストールプログラムは以下を生成します。

- **oc** および **kubectl** ユーティリティーを使用してクラスターに接続するための管理者認証情報が含まれる **auth** ディレクトリー。
- OpenShift Container Platform クラスター名、クラスター ID、および現行インストールのインフラストラクチャー ID などの情報を含む **metadata.json** ファイル。

このインストールプロセスの Ansible Playbook は、**infraID** の値を、作成する仮想マシンの接頭辞として使用します。これにより、同じ oVirt/RHV クラスターに複数のインストールがある場合の命名の競合が回避されます。

注記

Ignition 設定ファイルの証明書は 24 時間後に有効期限が切れます。最初の証明書のローテーションが終了するように、クラスターのインストールを完了し、クラスターを動作が低下していない状態で 24 時間実行し続ける必要があります。

手順

1. Ignition ファイルをビルドするには、以下を入力します。

```
$ openshift-install create ignition-configs --dir $ASSETS_DIR
```

出力例

```
$ tree
.
└── wrk
    ├── auth
    │   ├── kubeadmin-password
    │   └── kubeconfig
    ├── bootstrap.ign
    ├── master.ign
    ├── metadata.json
    └── worker.ign
```

5.20. テンプレートおよび仮想マシンの作成

inventory.yml の変数を確認した後に、最初の Ansible プロビジョニング Playbook **create-templates-and-vms.yml** を実行します。

この Playbook は、**$HOME/.ovirt/ovirt-config.yaml** から RHV Manager の接続パラメーターを使用し、アセットディレクトリーで **metadata.json** を読み取ります。

ローカルの Red Hat Enterprise Linux CoreOS (RHCOS) イメージが存在しない場合、Playbook は **inventory.yml** の **image_url** に指定した URL からダウンロードします。これはイメージをデプロイメントし、これを RHV にアップロードしてテンプレートを作成します。

Playbook は、**inventory.yml** ファイルの **control_plane** と **compute** プロファイルに基づいてテンプレートを作成します。これらのプロファイルの名前が異なる場合、2 つのテンプレートが作成されます。

Playbook が完了すると、作成される仮想マシンは停止します。他のインフラストラクチャー要素の設定に役立つ情報を取得できます。たとえば、仮想マシンの MAC アドレスを取得して、仮想マシンに永続的な IP アドレスを割り当てるように DHCP を設定できます。

手順

1. **inventory.yml** の **control_plane** および **compute** 変数で、**type: high_performance** の 両方のインスタンスを **type: server** に変更します。

2. オプション: 同じクラスターに複数のインストールを実行する予定の場合には、OpenShift Container Platform インストールごとに異なるテンプレートを作成します。**inventory.yml** ファイルで、**template** の値の先頭に **infraID** を付けます。以下に例を示します。

```
control_plane:
  cluster: "{{ ovirt_cluster }}"
  memory: 16GiB
  sockets: 4
  cores: 1
  template: "{{ metadata.infraID }}-rhcos_tpl"
  operating_system: "rhcos_x64"
  ...
```

3. テンプレートおよび仮想マシンを作成します。

```
$ ansible-playbook -i inventory.yml create-templates-and-vms.yml
```

5.21. ブートストラップマシンの作成

bootstrap.yml Playbook を実行してブートストラップマシンを作成します。この Playbook はブートストラップ仮想マシンを起動し、これをアセットディレクトリーから **bootstrap.ign** Ignition ファイルに渡します。ブートストラップノードは、Ignition ファイルをコントロールプレーンノードに送信できるように設定します。

ブートストラッププロセスをモニターするには、RHV 管理ポータルでコンソールを使用するか、SSH を使用して仮想マシンに接続します。

手順

1. ブートストラップマシンを作成します。

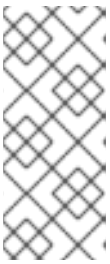
```
$ ansible-playbook -i inventory.yml bootstrap.yml
```

2. 管理ポータルまたは SSH のコンソールを使用してブートストラップマシンに接続します。**<bootstrap_ip>** をブートストラップノードの IP アドレスに置き換えます。SSH を使用するには、以下を入力します。

```
$ ssh core@<boostrap.ip>
```

3. ブートストラップノードからリリースイメージサービスについての **bootkube.service** journald ユニットログを収集します。

```
[core@ocp4-lk6b4-bootstrap ~]$ journalctl -b -f -u release-image.service -u bootkube.service
```



注記

ブートストラップノードの **bootkube.service** のログは etcd の **connection refused** エラーを出力し、ブートストラップサーバーがコントロールプレーンノードの etcd に接続できないことを示します。etcd が各コントロールプレーンノードで起動し、ノードがクラスターに参加した後には、エラーは発生しなくなるはずです。

5.22. コントロールプレーンノードの作成

masters.yml Playbook を実行してコントロールプレーンノードを作成します。この Playbook は **master.ign** Ignition ファイルをそれぞれの仮想マシンに渡します。Ignition ファイルには、**https://api-int.ocp4.example.org:22623/config/master** などの URL から Ignition を取得するためのコントロールプレーンノードのディレクティブが含まれます。この URL のポート番号はロードバランサーによって管理され、クラスター内でのみアクセスできます。

手順

1. コントロールプレーンノードを作成します。

```
$ ansible-playbook -i inventory.yml masters.yml
```

2. Playbook がコントロールプレーンを作成する間に、ブートストラッププロセスをモニターします。

```
$ openshift-install wait-for bootstrap-complete --dir $ASSETS_DIR
```

出力例

```
INFO API v1.25.0 up
INFO Waiting up to 40m0s for bootstrapping to complete...
```

3. コントロールプレーンノードおよび etcd のすべての Pod が実行されている場合、インストールプログラムは以下の出力を表示します。

出力例

```
INFO It is now safe to remove the bootstrap resources
```

5.23. クラスターステータスの確認

インストール時またはインストール後に OpenShift Container Platform クラスターのステータスを確認することができます。

手順

1. クラスター環境で、管理者の kubeconfig ファイルをエクスポートします。

```
$ export KUBECONFIG=$ASSETS_DIR/auth/kubeconfig
```

kubeconfig ファイルには、クライアントを正しいクラスターおよび API サーバーに接続するために CLI で使用されるクラスターについての情報が含まれます。

2. デプロイメント後に作成されたコントロールプレーンおよびコンピュートマシンを表示します。

```
$ oc get nodes
```

3. クラスターのバージョンを表示します。

```
$ oc get clusterversion
```

4. Operator のステータスを表示します。

```
$ oc get clusteroperator
```

5. クラスター内のすべての実行中の Pod を表示します。

```
$ oc get pods -A
```

5.24. ブートストラップマシンの削除

wait-for コマンドがブートストラッププロセスが完了したことを示していることを確認したら、ブートストラップ仮想マシンを削除してコンピュート、メモリー、およびストレージリソースを解放する必要があります。また、ロードバランサーディレクティブからブートストラップマシンの設定を削除します。

手順

1. クラスターからブートストラップマシンを削除するには、以下を実行します。

```
$ ansible-playbook -i inventory.yml retire-bootstrap.yml
```

2. ロードバランサーディレクティブからブートストラップマシンの設定を削除します。

5.25. ワーカーノードの作成およびインストールの完了

ワーカーノードの作成は、コントロールプレーンノードの作成と同様です。ただし、ワーカーノードはクラスターに自動的に参加しません。これらをクラスターに追加するには、ワーカーの保留状態の CSR(証明書署名要求) を確認し、承認します。

最初の要求の承認後に、ワーカーノードがすべて承認されるまで CSR の承認を継続します。このプロセスが完了すると、ワーカーノードは **Ready** になり、Pod がそれらで実行されるようにスケジュールできます。

最後に、コマンドラインを監視し、インストールプロセスが完了するタイミングを確認します。

手順

1. ワーカーノードを作成します。

```
$ ansible-playbook -i inventory.yml workers.yml
```

2. すべての CSR をリスト表示するには、以下を入力します。

```
$ oc get csr -A
```

最終的に、このコマンドはノードごとに 1 つの CSR を表示します。以下に例を示します。

出力例

```
NAME        AGE    SIGNERNAME                                    REQUESTOR
CONDITION
csr-2lnxd   63m    kubernetes.io/kubelet-serving                 system:node:ocp4-lk6b4-
master0.ocp4.example.org                                         Approved,Issued
csr-hff4q   64m    kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper
Approved,Issued
csr-hsn96   60m    kubernetes.io/kubelet-serving                 system:node:ocp4-lk6b4-
master2.ocp4.example.org                                         Approved,Issued
csr-m724n   6m2s   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
csr-p4dz2   60m    kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper
Approved,Issued
csr-t9vfj   60m    kubernetes.io/kubelet-serving                 system:node:ocp4-lk6b4-
master1.ocp4.example.org                                         Approved,Issued
csr-tggtr   61m    kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper
Approved,Issued
csr-wcbrf   7m6s   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
```

3. リストをフィルターし、保留中の CSR のみを表示するには、以下を実行します。

```
$ watch "oc get csr -A | grep pending -i"
```

このコマンドは 2 秒ごとに出力を更新し、保留中の CSR のみを表示します。以下に例を示します。

出力例

```
Every 2.0s: oc get csr -A | grep pending -i

csr-m724n   10m   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
csr-wcbrf   11m   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
```

4. 保留中のそれぞれの要求を検査します。以下に例を示します。

出力例

```
$ oc describe csr csr-m724n
```

出力例

```
Name:               csr-m724n
Labels:             <none>
Annotations:        <none>
CreationTimestamp:  Sun, 19 Jul 2020 15:59:37 +0200
Requesting User:    system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper
Signer:             kubernetes.io/kube-apiserver-client-kubelet
Status:             Pending
Subject:
        Common Name:    system:node:ocp4-lk6b4-worker1.ocp4.example.org
        Serial Number:
        Organization:   system:nodes
Events:  <none>
```

5. CSR 情報が正しい場合は、要求を承認します。

```
$ oc adm certificate approve csr-m724n
```

6. インストールプロセスが完了するまで待機します。

```
$ openshift-install wait-for install-complete --dir $ASSETS_DIR --log-level debug
```

インストールが完了すると、コマンドラインには OpenShift Container Platform Web コンソールの URL と、管理者のユーザー名およびパスワードが表示されます。

5.26. OPENSHIFT CONTAINER PLATFORM の TELEMETRY アクセス

OpenShift Container Platform 4.12 では、クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは OpenShift Cluster Manager Hybrid Cloud Console に登録されます。

OpenShift Cluster Manager インベントリーが正常である (Telemetry によって自動的に維持、または OpenShift Cluster Manager Hybrid Cloud Console を使用して手動で維持) ことを確認した後に、subscription watch を使用 して、アカウントまたはマルチクラスターレベルで OpenShift Container Platform サブスクリプションを追跡します。

関連情報

- Telemetry サービスの詳細は、リモートヘルスモニタリング を参照して ください。

5.27. デフォルトの OPERATORHUB カタログソースの無効化

Red Hat によって提供されるコンテンツを調達する Operator カタログおよびコミュニティープロジェクトは、OpenShift Container Platform のインストール時にデフォルトで OperatorHub に設定されます。ネットワークが制限された環境では、クラスター管理者としてデフォルトのカタログを無効にする必要があります。

手順

- **disableAllDefaultSources: true** を **OperatorHub** オブジェクトに追加して、デフォルトカタログのソースを無効にします。

```
$ oc patch OperatorHub cluster --type json \
    -p '[{"op": "add", "path": "/spec/disableAllDefaultSources", "value": true}]'
```

ヒント

または、Web コンソールを使用してカタログソースを管理できます。**Administration → Cluster Settings → Configuration → OperatorHub** ページから、**Sources** タブをクリックして、個別のソースを作成、更新、削除、無効化、有効化できます。

第6章 RHV でのクラスターのアンインストール

OpenShift Container Platform クラスターを Red Hat Virtualization (RHV) から削除することができます。

6.1. インストーラーでプロビジョニングされるインフラストラクチャーを使用するクラスターの削除

インストーラーでプロビジョニングされるインフラストラクチャーを使用するクラスターは、クラウドから削除できます。

注記

アンインストール後に、とくにユーザーによってプロビジョニングされるインフラストラクチャー (UPI) クラスターで適切に削除されていないリソースがあるかどうかについて、クラウドプロバイダーを確認します。インストールプログラムが作成しなかったリソース、またはインストールプログラムがアクセスできないリソースが存在する可能性があります。

前提条件

- クラスターをデプロイするために使用したインストールプログラムのコピーがある。
- クラスター作成時にインストールプログラムが生成したファイルがあります。

手順

1. クラスターのインストールに使用したコンピューターで、インストールプログラムを含むディレクトリーに移動し、次のコマンドを実行します。

```
$ ./openshift-install destroy cluster \
--dir <installation_directory> --log-level info 1 2
```

1. **<installation_directory>** には、インストールファイルを保存したディレクトリーへのパスを指定します。
2. 異なる詳細情報を表示するには、**info** ではなく、**warn**、**debug**、または **error** を指定します。

注記

クラスターのクラスター定義ファイルが含まれるディレクトリーを指定する必要があります。クラスターを削除するには、インストールプログラムでこのディレクトリーにある **metadata.json** ファイルが必要になります。

2. オプション: **<installation_directory>** ディレクトリーおよび OpenShift Container Platform インストールプログラムを削除します。

6.2. ユーザーによってプロビジョニングされるインフラストラクチャーを使用するクラスターの削除

クラスターの使用が完了したら、ユーザーによってプロビジョニングされるインフラストラクチャーを使用するクラスターをクラウドから削除できます。

前提条件

- クラスターのインストールに使用した元の Playbook ファイル、アセットディレクトリーおよびファイル、および **$ASSETS_DIR** 環境変数が含まれます。通常、クラスターのインストール時に使用したのと同じコンピューターを使用してこれを実行できます。

手順

1. クラスターを削除するには、以下を入力します。

```
$ ansible-playbook -i inventory.yml \
    retire-bootstrap.yml \
    retire-masters.yml   \
    retire-workers.yml
```

2. DNS、ロードバランサー、およびこのクラスターの他のインフラストラクチャーに追加した設定を削除します。