



OpenShift Container Platform 4.3

ベアメタルへのインストール

OpenShift Container Platform 4.3 ベアメタルクラスターのインストール

OpenShift Container Platform 4.3 ベアメタルへのインストール

OpenShift Container Platform 4.3 ベアメタルクラスタのインストール

法律上の通知

Copyright © 2020 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本書では、ベアメタルのインフラストラクチャーに OpenShift Container Platform 4.3 クラスターをインストールする方法について説明します。

目次

第1章 ベアメタルへのインストール	3
1.1. クラスターのベアメタルへのインストール	3
1.2. ネットワークのカスタマイズによるベアメタルへのクラスターのインストール	33
1.3. ネットワークが制限された環境でのクラスターのベアメタルへのインストール	63

第1章 ベアメタルへのインストール

1.1. クラスターのベアメタルへのインストール

OpenShift Container Platform version 4.3 では、クラスターをプロビジョニングするベアメタルのインフラストラクチャーにインストールできます。



重要

以下の手順に従って仮想化環境またはクラウド環境にクラスターをデプロイすることができますが、ベアメタルプラットフォーム以外の場合は追加の考慮事項に注意してください。このような環境で OpenShift Container Platform クラスターのインストールを試行する前に、「[Deploying OpenShift 4.x on non-tested platforms using the bare metal install method](#)」にある情報を確認してください。

1.1.1. 前提条件

- [OpenShift Container Platform のインストールおよび更新](#) プロセスについての詳細を確認します。
- ファイアウォールを使用する場合、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](#)する必要があります。



注記

プロキシを設定する場合は、この[サイト一覧](#)も確認してください。

1.1.2. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス

OpenShift Container Platform 4.3 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。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 [Red Hat OpenShift Cluster Manager \(OCM\)](#) に登録されます。

Red Hat OpenShift Cluster Manager インベントリーが Telemetry によって自動的に維持されるか、または OCM を手動で使用しているかのいずれによって正常である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。

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。

- [Red Hat OpenShift Cluster Manager](#) ページにアクセスし、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クラスターのインストールおよびインストールプログラムの生成に必要な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。

1.1.3. 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場合のクラスターのマシン要件

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含むクラスターの場合、必要なマシンすべてをデプロイする必要があります。

1.1.3.1. 必要なマシン

最小の OpenShift Container Platform クラスターでは以下のホストが必要です。

- 1つの一時的なブートストラップマシン
- 3つのコントロールプレーン、またはマスター、マシン
- 少なくとも2つのコンピュータマシン(ワーカーマシンとしても知られる)。



注記

クラスターでは、ブートストラップマシンが OpenShift Container Platform クラスターを3つのコントロールプレーンマシンにデプロイする必要があります。クラスターのインストール後にブートストラップマシンを削除できます。



重要

クラスターの高可用性を維持するには、これらのクラスターマシンについて別個の物理ホストを使用します。

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タマシンでは、Red Hat Enterprise Linux CoreOS (RHCOS) をオペレーティングシステムとして使用する必要があります。

RHCOS は Red Hat Enterprise Linux 8 をベースとしており、そのハードウェア認定および要件が継承されることに注意してください。「[Red Hat Enterprise Linux テクノロジーの機能と制限](#)」を参照してください。

1.1.3.2. ネットワーク接続の要件

すべての Red Hat Enterprise Linux CoreOS (RHCOS) マシンでは、起動時に **initramfs** のネットワークがマシン設定サーバーから 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、Ignition 設定ファイルをダウンロードできるようにネットワーク接続を確立するために、マシンには DHCP サーバーまたはその静的 IP アドレスが設定されている必要になります。

1.1.3.3. 最小リソース要件

それぞれのクラスターマシンは、以下の最小要件を満たしている必要があります。

マシン	Operating System	vCPU	仮想 RAM	ストレージ
ブートストラップ	RHCOS	4	16 GB	120 GB
コントロールプレーン	RHCOS	4	16 GB	120 GB
コンピューター	RHCOS または RHEL 7.6	2	8 GB	120 GB

1.1.3.4. 証明書署名要求の管理

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を使用する場合、クラスターの自動マシン管理へのアクセスは制限されるため、インストール後にクラスターの証明書署名要求 (CSR) のメカニズムを提供する必要があります。**kube-controller-manager** は kubelet クライアント CSR のみを承認します。**machine-approver** は、kubelet 認証情報を使用して要求される提供証明書の有効性を保証できません。適切なマシンがこの要求を発行したかどうかを確認できないためです。kubelet 提供証明書の要求の有効性を検証し、それらを承認する方法を判別し、実装する必要があります。

1.1.4. 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作成

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 OpenShift Container Platform クラスターをデプロイする前に、基礎となるインフラストラクチャーを作成する必要があります。

前提条件

- クラスターでサポートするインフラストラクチャーを作成する前に、「[OpenShift Container Platform 4.x のテスト済みインテグレーション](#)」ページを参照してください。

手順

1. 各ノードに DHCP を設定するか、または静的 IP アドレスを設定します。
2. 必要なロードバランサーをプロビジョニングします。
3. マシンのポートを設定します。
4. DNS を設定します。
5. ネットワーク接続を確認します。

1.1.4.1. 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** のネットワークがマシン設定サーバーから Ignition 設定をフェッチする必要があります。

初回の起動時に、Ignition 設定ファイルをダウンロードできるようにネットワーク接続を確立するために、マシンには DHCP サーバーまたはその静的 IP アドレスが設定されている必要があります。

クラスターのマシンを長期間管理するために DHCP サーバーを使用することが推奨されています。DHCP サーバーが永続 IP アドレス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。

クラスターの正常なインストール後に各マスターノードで実行される 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することができます。

マシン間のネットワーク接続を、クラスターのコンポーネントが通信できるように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。

表1.1 すべてのマシンに対応するすべてのマシン

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	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 および Geneve
	6081	VXLAN および Geneve
	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。
TCP/UDP	30000-32767	Kubernetes NodePort

表1.2 コントロールプレーンへのすべてのマシン

プロトコル	ポート	説明
TCP	2379-2380	etcd サーバー、ピア、およびメトリクスポート
	6443	Kubernetes API

ネットワークポロジリー要件

クラスター用にプロビジョニングするインフラストラクチャーは、ネットワークポロジリーの以下の要件を満たす必要があります。



重要

OpenShift Container Platform では、すべてのノードが、プラットフォームコンテナのイメージをプルし、Telemetry データを Red Hat に提供するためにインターネットへの直接のアクセスが必要です。

ロードバランサー

OpenShift Container Platform をインストールする前に、以下の要件を満たす 2 つのロードバランサーをプロビジョニングする必要があります。

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、SSL パススルー、または SSL ブリッジモードと呼ばれます。SSL ブリッジモードを使用する場合は、API ルートの Server Name Indication (SNI) を有効にする必要があります。
 - 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。



注記

API ロードバランサーが適切に機能するには、セッション永続性は必要ありません。

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1.3 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
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が、5 秒または 10 秒ごとにプローブし、2 つの正常な要求が正常な状態になり、3 つの要求が正常な状態になりません。これら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。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、SSL パススルー、または SSL ブリッジモードと呼ばれます。SSL ブリッジモードを使用する場合は、Ingress ルートの Server Name Indication (SNI) を有効にする必要があります。
- 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。

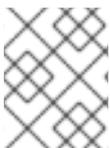
ロードバランサーのフロントとバックの両方で以下のポートを設定します。

表1.4 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 ルーター 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 ルーター 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

ヒント

クライアントの実際の IP アドレスが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。



注記

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。コントロールプレーンの初期化後に Ingress ルーターを設定する必要があります。

1.1.4.2. 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 DNS 要件

以下の DNS レコードは、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する OpenShift Container Platform クラスターに必要です。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クラスターの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1.5 必要な DNS レコード

コンポーネント	レコード	説明
---------	------	----

コンポーネント	レコード	説明
Kubernetes API	api.<cluster_name>.<base_domain>.	この DNS A/AAAA または CNAME レコードは、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必要があります。このレコードは、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>.	この DNS A/AAAA または CNAME レコードは、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必要があります。このレコードは、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。これがノード名を解決できない場合、プロキシされる 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。
ルート	*.apps.<cluster_name>.<base_domain>.	Ingress ルーター Pod を実行するマシンをターゲットにする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 レコードです。このレコードは、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。

コンポーネント	レコード	説明
etcd	etcd-<code><index></code>.<code><cluster_name></code>.<code><base_domain></code>.	<p>OpenShift Container Platform では、各 etcd インスタンスの DNS A/AAAA レコードがインスタンスをホストするコントロールプレーンマシンを参照する必要があります。etcd インスタンスは <code><index></code> 値によって差別化されます。この値は 0 で始まり、n-1 で終了します。ここで、n はクラスターのコントロールプレーンマシンの数です。DNS レコードはコントロールプレーンマシンのユニキャスト IPv4 アドレスに解決し、レコードはクラスター内のすべてのノードで解決可能である必要があります。</p>

コンポーネント	レコード	説明
	<code>_etcd-server-ssl._tcp.<cluster_name>.<base_domain>.</code>	<p>それぞれのコントロールプレーンマシンについて、OpenShift Container Platform では、そのマシンに優先度 0、重み 10 およびポート 2380 の etcd サーバーの SRV DNS レコードも必要になります。3 つのコントロールプレーンマシンを使用するクラスターには以下のレコードが必要です。</p> <pre># _service._proto.name. TTL class SRV priority weight port target. _etcd-server-ssl._tcp. <cluster_name>. <base_domain>. 86400 IN SRV 0 10 2380 etcd- 0.<cluster_name>. <base_domain> _etcd-server-ssl._tcp. <cluster_name>. <base_domain>. 86400 IN SRV 0 10 2380 etcd- 1.<cluster_name>. <base_domain> _etcd-server-ssl._tcp. <cluster_name>. <base_domain>. 86400 IN SRV 0 10 2380 etcd- 2.<cluster_name>. <base_domain></pre>

1.1.5. SSH プライベートキーの生成およびエージェントへの追加

クラスターで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がある場合、**ssh-agent** とインストールプログラムの両方に SSH キーを指定する必要があります。



注記

実稼働環境では、障害復旧およびデバッグが必要です。

このキーを使用して、ユーザー **core** としてマスターノードに対して SSH を実行できます。クラスターをデプロイする際に、キーは **core** ユーザーの `~/.ssh/authorized_keys` 一覧に追加されます。



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. パスワードなしの認証に設定されている SSH キーがコンピューター上にない場合は、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 rsa -b 4096 -N "" \
  -f <path>/<file_name> ①
```

- ① `~/.ssh/id_rsa` などの、SSH キーのパスおよびファイル名を指定します。既存の SSH キーは上書きされるため、指定しないでください。

このコマンドを実行すると、指定した場所にパスワードを必要としない SSH キーが生成されます。

2. **ssh-agent** プロセスをバックグラウンドタスクとして開始します。

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

Agent pid 31874
```

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

```
$ ssh-add <path>/<file_name> ①

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

- ① `~/.ssh/id_rsa` 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。

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。クラスターを独自に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場合は、このキーをクラスターのマシンに指定する必要があります。

1.1.6. 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、インストールファイルをローカルコンピューターにダウンロードします。

前提条件

- Linux または macOS を使用するコンピューターからクラスターをインストールする必要があります。
- インストールプログラムをダウンロードするには、500 MB のローカルディスク領域が必要です。

手順

1. Red Hat OpenShift Cluster Manager サイトの「[Infrastructure Provider](#)」ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使ってログインします。アカウントがない場合はこれを作成します。

2. 選択するインストールタイプのページに移動し、オペレーティングシステムのインストールプログラムをダウンロードし、ファイルを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。クラスターインストールの完了後は、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の両方を保持する必要があります。



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。特定のクラウドプロバイダー用に記載された OpenShift Container Platform のアンインストール手順を完了して、クラスターを完全に削除する必要があります。

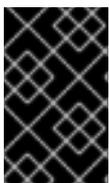
3. インストールプログラムを展開します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ tar xvf <installation_program>.tar.gz
```

4. Red Hat OpenShift Cluster Manager サイトの「[Pull Secret](#)」ページから、インストールプルシークレットを `.txt` ファイルとして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。

1.1.7. バイナリーのダウンロードによる CLI のインストール

コマンドラインインターフェースを使用して 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 4.3 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。

1.1.7.1. Linux への 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 OpenShift Cluster Manager サイトの「[Infrastructure Provider](#)」ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Linux** を選択し、**Download command-line tools** をクリックします。

4. アーカイブを展開します。

```
$ tar xvzf <file>
```

5. **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
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.1.7.2. Windows での 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 OpenShift Cluster Manager サイトの「[Infrastructure Provider](#)」ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Windows** を選択し、**Download command-line tools** をクリックします。
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
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

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

1.1.7.3. macOS への 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 OpenShift Cluster Manager サイトの「[Infrastructure Provider](#)」ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **MacOS** を選択し、**Download command-line tools** をクリックします。
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
```

CLIのインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.1.8. インストール設定ファイルの手動作成

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 OpenShift Container Platform のインストールでは、インストール設定ファイルを手動で生成する必要があります。

前提条件

- OpenShift Container Platform インストーラープログラムおよびクラスターのアクセストークンを取得します。

手順

1. 必要なインストールアセットを保存するためのインストールディレクトリーを作成します。

```
$ mkdir <installation_directory>
```



重要

ディレクトリーを作成する必要があります。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の有効期限は短く設定されているため、インストールディレクトリーを再利用することができません。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. 以下の **install-config.yaml** ファイルテンプレートをカスタマイズし、これを **<installation_directory>** に保存します。



注記

この設定ファイル **install-config.yaml** に名前を付ける必要があります。

3. **install-config.yaml** ファイルをバックアップし、これを複数のクラスターをインストールするため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の次の手順で使用されます。この時点でこれをバックアップする必要があります。

1.1.8.1. ベアメタルのサンプル 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、または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:
- hyperthreading: Enabled ② ③
  name: worker
  replicas: 0 ④
controlPlane:
  hyperthreading: Enabled ⑤ ⑥
  name: master ⑦
  replicas: 3 ⑧
metadata:
  name: test ⑨
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14 ⑩
    hostPrefix: 23 ⑪
  networkType: OpenShiftSDN
  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、コンピュート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す。どちらのセクションも、現時点では単一のマシンプールを定義しますが、OpenShift Container Platform の今後のバージョンでは、インストール時の複数のコンピュートプールの定義をサポートする可能性があります。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 ⑦ 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。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。一部のクラスターマシンで同時マルチスレッドを無効にする場合は、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。



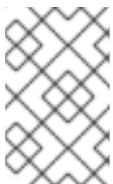
重要

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。

④ **replicas** パラメーターの値を **0** に設定する必要があります。このパラメーターはクラスターが作成し、管理するワーカーの数を制御します。これは、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する場合にクラスターが実行しない機能です。OpenShift Container

Platform のインストールが終了する前に、クラスターが使用するワーカーマシンを手動でデプロイする必要があります。

- 8 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。
- 9 DNS レコードに指定したクラスター名。
- 10 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。
- 11 それぞれの個別ノードに割り当てるサブネットプレフィックスの長さ。たとえば、**hostPrefix** が **23** に設定され、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 (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。
- 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。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。ベアメタル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。
- 15 Red Hat OpenShift Cluster Manager サイトの「[Pull Secret](#)」ページから取得したプルシークレット。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** ユーザーのデフォルト SSH キーの公開部分。



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がある実稼働用の 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。

1.1.8.2. インストール時のクラスター全体のプロキシの設定

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 プロキシを使用することができます。プロキシ設定を **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を使用するように設定できます。



注記

ベアメタルインストールでは、**install-config.yaml** ファイルの **networking.machineCIDR** フィールドで指定される範囲にある IP アドレスをノードに割り当てない場合、それらを **proxy.noProxy** フィールドに含める必要があります。

前提条件

- 既存の **install-config.yaml** ファイルが必要です。
- クラスタがアクセスする必要のあるサイトを確認し、プロキシをバイパスする必要があるかどうかを判別します。デフォルトで、すべてのクラスタ egress トラフィック (クラスタをホストするクラウドについてのクラウドプロバイダー API に対する呼び出しを含む) はプロキシされます。プロキシオブジェクトの **spec.noProxy** フィールドにサイトを追加し、必要に応じてプロキシをバイパスします。



注記

プロキシオブジェクトの **status.noProxy** フィールドは、デフォルトでインスタンスメタデータエンドポイント (**169.254.169.254**) およびインストール設定の **networking.machineCIDR**、**networking.clusterNetwork.cidr**、および **networking.serviceNetwork[]** フィールドの値で設定されます。

手順

1. **install-config.yaml** ファイルを編集し、プロキシ設定を追加します。以下は例になります。

```

apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> 1
  httpsProxy: http://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
  additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
  ...

```

- 1 クラスタ外の HTTP 接続を作成するために使用するプロキシ URL。URL スキームは **http** である必要があります。追加のプロキシ設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシネットワークを使用する場合には、**httpProxy** 値を指定することはできません。
- 2 クラスタ外で HTTPS 接続を作成するために使用するプロキシ URL。このフィールドが指定されていない場合、HTTP および HTTPS 接続の両方に **httpProxy** が使用されます。追加のプロキシ設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシネットワークを使用する場合には、**httpsProxy** 値を指定することはできません。
- 3 プロキシを除外するための宛先ドメイン名、ドメイン、IP アドレス、または他のネットワーク CIDR のカンマ区切りの一覧。ドメインのすべてのサブドメインを組み込むために、ドメインの前に . を入力します。* を使用し、すべての宛先のプロキシをバイパスします。

- 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 ConfigMap を



注記

インストールプログラムは、プロキシの **readinessEndpoints** フィールドをサポートしません。

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を作成します。プロキシ設定が指定されていない場合、**cluster** のプロキシ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 という名前のプロキシオブジェクトのみがサポートされ、追加のプロキシを作成することはできません。

1.1.9. Kubernetes マニフェストおよび Ignition 設定ファイルの作成

一部のクラスター定義ファイルを変更し、クラスターマシンを手動で起動する必要があるため、クラスターがマシンを作成するために必要な Kubernetes マニフェストと Ignition 設定ファイルを生成する必要があります。



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る証明書が含まれます。最初の証明書のローテーションが正常に実行されるようにするには、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。

前提条件

- OpenShift Container Platform インストールプログラムを取得します。
- **install-config.yaml** インストール設定ファイルを作成します。

手順

1. クラスターの Kubernetes マニフェストを生成します。

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

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

- 1 **<installation_directory>** については、作成した **install-config.yaml** ファイルが含まれるインストールディレクトリーを指定します。

インストールプロセスの後の部分で独自のコンピュータマシンを作成するため、この警告を無視しても問題がありません。

2. `<installation_directory>/manifests/cluster-scheduler-02-config.yml` Kubernetes マニフェストファイルを変更し、Pod がコントロールプレーンマシンにスケジュールされないようにします。
 - a. `<installation_directory>/manifests/cluster-scheduler-02-config.yml` ファイルを開きます。
 - b. `mastersSchedulable` パラメーターを見つけ、その値を **False** に設定します。
 - c. ファイルを保存し、終了します。



注記

現時点では、[Kubernetes の制限](#)により、コントロールプレーンマシンで実行されるルーター Pod に Ingress ロードバランサーがアクセスすることができません。この手順は、OpenShift Container Platform の今後のマイナーバージョンで不要になる可能性があります。

3. Ignition 設定ファイルを取得します。

```
$ ./openshift-install create ignition-configs --dir=<installation_directory> 1
```

- 1** `<installation_directory>` については、同じインストールディレクトリーを指定します。

以下のファイルはディレクトリーに生成されます。

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

1.1.10. Red Hat Enterprise Linux CoreOS (RHCOS) マシンの作成

ユーザーによってプロビジョニングされるベアメタル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前に、それが使用する RHCOS マシンを作成する必要があります。ISO イメージまたはネットワーク PXE ブートを使用する手順を実行してマシンを作成することができます。

1.1.10.1. ISO イメージを使用した Red Hat Enterprise Linux CoreOS (RHCOS) マシンの作成

ユーザーによってプロビジョニングされるベアメタル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前に、それが使用する RHCOS マシンを作成する必要があります。ISO イメージを使用してマシンを作成することができます。

前提条件

- クラスターの Ignition 設定ファイルを取得していること。

- お使いのコンピューターからアクセスでき、作成するマシンがアクセスできる HTTP サーバーへのアクセスがあること。

手順

1. インストールプログラムが作成したコントロールプレーン、コンピュート、およびブートストラップ Ignition 設定を HTTP サーバーにアップロードします。これらのファイルの URL をメモします。



重要

インストールの完了後にコンピュートマシンをさらにクラスターに追加する予定の場合には、これらのファイルを削除しないでください。

2. Red Hat カスタマーポータル[の「製品のダウンロード」](#)ページまたは「[RRHCOS イメージミラー](#)」ページからオペレーティングシステムのインスタンスをインストールするために優先される方法で必要な RHCOS イメージを取得します。



重要

RHCOS イメージは OpenShift Container Platform の各リリースごとに変更されない可能性があります。インストールする OpenShift Container Platform バージョンと等しいか、それ以下のバージョンの中で最も新しいバージョンのイメージをダウンロードする必要があります。利用可能な場合は、OpenShift Container Platform バージョンに一致するイメージのバージョンを使用します。

ISO ファイルおよび RAW ディスクファイルをダウンロードする必要があります。これらのファイルの名前は以下の例のようになります。

- ISO: `rhcos-<version>-installer.<architecture>.iso`
- 圧縮された metal RAW: `rhcos-<version>-metal.<architecture>.raw.gz`

3. RAW RHCOS イメージファイルのいずれかを HTTP サーバーにアップロードし、その URL をメモします。



重要

インストールの完了後にコンピュートマシンをさらにクラスターに追加する予定の場合には、これらのファイルを削除しないでください。

4. ISO を使用し、RHCOS インストールを開始します。以下のインストールオプションのいずれかを使用します。
 - ディスクに ISO イメージを書き込み、これを直接起動します。
 - LOM インターフェースで ISO リダイレクトを使用します。
5. インスタンスの起動後に、**TAB** または **E** キーを押してカーネルコマンドラインを編集します。
6. パラメーターをカーネルコマンドラインに追加します。

```
coreos.inst=yes
coreos.inst.install_dev=sda 1
```

```
coreos.inst.image_url=<image_URL> 2
coreos.inst.ignition_url=http://example.com/config.ign 3
ip=<dhcp or static IP address> 4 5
```

- 1 インストール先のシステムのブロックデバイスを指定します。
 - 2 サーバーにアップロードした RAW イメージの URL を指定します。
 - 3 このマシンタイプの Ignition 設定ファイルの URL を指定します。
 - 4 **ip=dhcp** を設定するか、各ノードに個別の静的 IP アドレス (**ip=**) および DNS サーバー (**nameserver=**) を設定します。たとえば、**ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp1s0:none nameserver=4.4.4.41** を設定すると、以下のように設定されます。
 - ノードの IP アドレス: **10.10.10.2**
 - ゲートウェイアドレス: **10.10.10.254**
 - ネットマスク: **255.255.255.0**
 - ホスト名: **core0.example.com**
 - DNS サーバーアドレス: **4.4.4.41**
 - 5 複数のネットワークインターフェースまたは DNS サーバーを使用する場合は、以下の方法で IP アドレス設定をさらにカスタマイズできます。
 - システムに複数のネットワークインターフェースがある場合には、複数の **ip=** エントリーを指定して、複数の IP アドレス設定を追加できます。
 - システムに複数のネットワークインターフェースがある場合には、DHCP と静的 IP の設定の組み合わせを使用できます。使用するインターフェースを指定する必要があります (例: **ip=enp1s0:dhcp ip=10.10.10.2::255.255.255.0:core0.example.com:enp2s0:none**) 。
 - 各サーバーに **nameserver=** エントリーを追加して、複数の DNS サーバーを指定できます (例: **nameserver=1.1.1.1 nameserver=8.8.8.8**)。
7. Enter を押してインストールを完了します。RHCOS のインストール後に、システムは再起動します。システムの再起動後、指定した Ignition 設定ファイルを適用します。
8. 継続してクラスターのマシンを作成します。



重要

この時点でブートストラップおよびコントロールプレーンマシンを作成する必要があります。一部の Pod はデフォルトでコンピュートマシンにデプロイされるため、クラスターのインストール前に、2つ以上のコンピュートマシンを作成します。

1.1.10.2. PXE または iPXE ブートによる Red Hat Enterprise Linux CoreOS (RHCOS) マシンの作成

ユーザーによってプロビジョニングされるベアメタル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前に、それが使用する RHCOS マシンを作成する必要があります。PXE または iPXE ブートを使用してマシンを作成することができます。

前提条件

- クラスターの Ignition 設定ファイルを取得していること。
- 適切な PXE または iPXE インフラストラクチャーを設定していること。
- 使用しているコンピューターからアクセス可能な HTTP サーバーへのアクセスがあること。

手順

1. インストールプログラムが作成したマスター、ワーカーおよびブートストラップの Ignition 設定を HTTP サーバーにアップロードします。これらのファイルの URL をメモします。



重要

インストールの完了後にコンピュートマシンをさらにクラスターに追加する予定の場合には、これらのファイルを削除しないでください。

2. Red Hat カスタマーポータル[の「製品のダウンロード」](#)ページまたは「[RHCOS イメージミラー](#)」ページから圧縮された metal RAW イメージ、**kernel** および **initramfs** ファイルを取得します。



重要

RHCOS イメージは OpenShift Container Platform の各リリースごとに変更されない可能性があります。インストールする OpenShift Container Platform バージョンと等しいか、それ以下のバージョンの中で最も新しいバージョンのイメージをダウンロードする必要があります。利用可能な場合は、OpenShift Container Platform バージョンに一致するイメージのバージョンを使用します。この手順には RAW イメージのみを使用します。RHCOS qcow2 イメージは、ベアメタルのインストールではサポートされません。

ファイル名には、OpenShift Container Platform のバージョン名が含まれます。以下の例のようになります。

- 圧縮されたメタル RAW イメージ: **rhcos-<version>metal.<architecture>.raw.gz**
 - **kernel**: **rhcos-<version>-installer-kernel-<architecture>**
 - **initramfs**: **rhcos-<version>-installer-initramfs.<architecture>.img**
3. 圧縮された metal RAW イメージおよび **kernel** および **initramfs** ファイルを HTTP サーバーにアップロードします。



重要

インストールの完了後にコンピュートマシンをさらにクラスターに追加する予定の場合には、これらのファイルを削除しないでください。

1. RHCOS のインストールの完了後に、マスター、ワーカー、ブートストラップの各コンポーネントをクラスターに追加します。

4. RHCOS のインストール後にマシンがローカルディスクから起動されるようにネットワークブートインフラストラクチャーを設定します。
5. RHCOS イメージに PXE または iPXE インストールを設定します。
ご使用の環境についての以下の例で示されるメニューエントリーのいずれかを変更し、イメージおよび Ignition ファイルが適切にアクセスできることを確認します。

- PXE の場合:

```

DEFAULT pxeboot
TIMEOUT 20
PROMPT 0
LABEL pxeboot
  KERNEL http://<HTTP_server>/rhcos-<version>-installer-kernel-<architecture> ❶
  APPEND ip=dhcp rd.neednet=1 initrd=http://<HTTP_server>/rhcos-<version>-installer-
initramfs.<architecture>.img console=tty0 console=ttyS0 coreos.inst=yes
coreos.inst.install_dev=sda coreos.inst.image_url=http://<HTTP_server>/rhcos-
<version>-metal.<architecture>.raw.gz
coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign ❷ ❸

```

- ❶ HTTP サーバーにアップロードした **kernel** ファイルの場所を指定します。
- ❷ 複数の NIC を使用する場合、**ip** オプションに単一インターフェースを指定します。たとえば、**eno1** という名前の NIC で DHCP を使用するには、**ip=eno1:dhcp** を設定します。
- ❸ HTTP サーバーにアップロードした RHCOS ファイルの場所を指定します。**initrd** パラメーター値は **initramfs** ファイルの場所であり、**coreos.inst.image_url** パラメーター値は圧縮された metal RAW イメージの場所、および **coreos.inst.ignition_url** パラメーター値はブートストラップ Ignition 設定ファイルの場所になります。

- iPXE の場合:

```

kernel http://<HTTP_server>/rhcos-<version>-installer-kernel-<architecture> ip=dhcp
rd.neednet=1 initrd=http://<HTTP_server>/rhcos-<version>-installer-initramfs.
<architecture>.img console=tty0 console=ttyS0 coreos.inst=yes
coreos.inst.install_dev=sda coreos.inst.image_url=http://<HTTP_server>/rhcos-
<version>-metal.<architecture>.raw.gz
coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign ❶ ❷
initrd http://<HTTP_server>/rhcos-<version>-installer-initramfs.<architecture>.img ❸
boot

```

- ❶ HTTP サーバーにアップロードした RHCOS ファイルの場所を指定します。**kernel** パラメーター値は **kernel** ファイルの場所であり、**initrd** パラメーター値は **initramfs** ファイルの場所、**coreos.inst.image_url** パラメーター値は圧縮された metal RAW イメージの場所、および **coreos.inst.ignition_url** パラメーター値はブートストラップ Ignition 設定ファイルの場所になります。
- ❷ 複数の NIC を使用する場合、**ip** オプションに単一インターフェースを指定します。たとえば、**eno1** という名前の NIC で DHCP を使用するには、**ip=eno1:dhcp** を設定します。
- ❸ HTTP サーバーにアップロードした **initramfs** ファイルの場所を指定します。

6. 続けてクラスタのマシンを作成します。

**重要**

この時点でブートストラップおよびコントロールプレーンマシンを作成する必要があります。一部の Pod はデフォルトでコンピュートマシンにデプロイされるため、クラスタのインストール前に、2つ以上のコンピュートマシンを作成します。

1.1.11. クラスタの作成

OpenShift Container Platform クラスタを作成するには、ブートストラッププロセスが、インストールプログラムで生成した Ignition 設定ファイルを使用してプロビジョニングしたマシンで完了するのを待機します。

前提条件

- クラスタに必要なインフラストラクチャーを作成します。
- インストールプログラムを取得し、クラスタの Ignition 設定ファイルを生成済みです。
- Ignition 設定ファイルを使用して、クラスタの RHCOS マシンを作成済みです。
- 使用するマシンでインターネットに直接アクセスできます。

手順

1. ブートストラッププロセスをモニターします。

```
$ ./openshift-install --dir=<installation_directory> wait-for bootstrap-complete \ 1
--log-level=info 2
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com...
INFO API v1.16.2 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
```

- 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。

Kubernetes API サーバーでこれがコントロールプレーンマシンにブートストラップされていることを示すシグナルが出されるとコマンドは成功します。

2. ブートストラッププロセスが完了したら、ブートストラップマシンをロードバランサーから削除します。

**重要**

この時点で、ブートストラップマシンをロードバランサーから削除する必要があります。さらに、マシン自体を削除し、再フォーマットすることができます。

1.1.12. 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ついての情報が含まれます。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ます。
- **oc** CLI をインストールします。

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
system:admin
```

1.1.13. マシンの CSR の承認

マシンをクラスターに追加する際に、追加したそれぞれのマシンについて 2 つの保留状態の証明書署名要求 (CSR) が生成されます。これらの CSR が承認されていることを確認するか、または必要な場合はそれらを承認してください。

前提条件

- マシンがクラスターに追加されています。

手順

1. クラスターがマシンを認識していることを確認します。

```
$ oc get nodes

NAME      STATUS   ROLES    AGE   VERSION
master-0  Ready    master   63m   v1.16.2
master-1  Ready    master   63m   v1.16.2
master-2  Ready    master   64m   v1.16.2
worker-0  NotReady worker   76s   v1.16.2
worker-1  NotReady worker   70s   v1.16.2
```

出力には作成したすべてのマシンが一覧表示されます。

2. 保留中の証明書署名要求 (CSR) を確認し、クラスターに追加したそれぞれのマシンのクライアントおよびサーバー要求に **Pending** または **Approved** ステータスが表示されていることを確認します。

```
$ oc get csr
```

```
NAME      AGE  REQUESTOR                                     CONDITION
csr-8b2br  15m  system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper Pending 1
csr-8vnps  15m  system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper Pending
csr-bfd72  5m26s system:node:ip-10-0-50-126.us-east-2.compute.internal
Pending 2
csr-c57lv  5m26s system:node:ip-10-0-95-157.us-east-2.compute.internal
Pending
...
```

1 クライアント要求の CSR。

2 サーバー要求の CSR。

この例では、2つのマシンがクラスターに参加しています。この一覧にはさらに多くの承認された CSR が表示される可能性があります。

3. 追加したマシンの保留中の CSR すべてが **Pending** ステータスになった後に CSR が承認されない場合には、クラスターマシンの CSR を承認します。



注記

CSR のローテーションは自動的に実行されるため、クラスターにマシンを追加後1時間以内に CSR を承認してください。1時間以内に承認しない場合には、証明書のローテーションが行われ、各ノードに3つ以上の証明書が存在するようになります。これらの証明書すべてを承認する必要があります。最初の CSR の承認後、後続のノードクライアント CSR はクラスターの **kube-controller-manger** によって自動的に承認されます。kubelet 提供証明書の要求を自動的に承認する方法を実装する必要があります。

- それらを個別に承認するには、それぞれの有効な CSR について以下のコマンドを実行します。

```
$ oc adm certificate approve <csr_name> 1
```

1 **<csr_name>** は、現行の CSR の一覧からの CSR の名前です。

- すべての保留中の CSR を承認するには、以下のコマンドを実行します。

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{"\n"}
{{end}}{{end}}' | xargs oc adm certificate approve
```

追加情報

- CSR の詳細は、「[Certificate Signing Requests](#)」を参照してください。

1.1.14. Operator の初期設定

コントロールプレーンの初期化後に、一部の Operator を利用可能にするためにそれらをすぐに設定する必要があります。

前提条件

- コントロールプレーンが初期化されています。

手順

1. クラスターコンポーネントがオンラインになることを確認します。

```
$ watch -n5 oc get clusteroperators
```

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
authentication	4.3.0	True	False	False	69s
cloud-credential	4.3.0	True	False	False	12m
cluster-autoscaler	4.3.0	True	False	False	11m
console	4.3.0	True	False	False	46s
dns	4.3.0	True	False	False	11m
image-registry	4.3.0	True	False	False	5m26s
ingress	4.3.0	True	False	False	5m36s
kube-apiserver	4.3.0	True	False	False	8m53s
kube-controller-manager	4.3.0	True	False	False	7m24s
kube-scheduler	4.3.0	True	False	False	12m
machine-api	4.3.0	True	False	False	12m
machine-config	4.3.0	True	False	False	7m36s
marketplace	4.3.0	True	False	False	7m54m
monitoring	4.3.0	True	False	False	7h54s
network	4.3.0	True	False	False	5m9s
node-tuning	4.3.0	True	False	False	11m
openshift-apiserver	4.3.0	True	False	False	11m
openshift-controller-manager	4.3.0	True	False	False	5m943s
openshift-samples	4.3.0	True	False	False	3m55s
operator-lifecycle-manager	4.3.0	True	False	False	11m
operator-lifecycle-manager-catalog	4.3.0	True	False	False	11m
service-ca	4.3.0	True	False	False	11m
service-catalog-apiserver	4.3.0	True	False	False	5m26s
service-catalog-controller-manager	4.3.0	True	False	False	5m25s
storage	4.3.0	True	False	False	5m30s

2. 利用不可の Operator を設定します。

1.1.14.1. インストール時に削除されたイメージレジストリー

共有可能なオブジェクトストレージを提供しないプラットフォームでは、OpenShift イメージレジストリー Operator 自体が **Removed** としてブートストラップされます。これにより、**openshift-installer** がそれらのプラットフォームタイプでのインストールを完了できます。

インストール後に、イメージレジストリー Operator 設定を編集して **managementState** を **Removed** から **Managed** に切り替える必要があります。



注記

Prometheus コンソールは、以下のような **ImageRegistryRemoved** アラートを提供しません。

"Image Registry has been removed.**ImageStreamTags**, **BuildConfigs** and **DeploymentConfigs** which reference **ImageStreamTags** may not work as expected.Please configure storage and update the config to **Managed** state by editing `configs.imageregistry.operator.openshift.io`."

1.1.14.2. イメージレジストリーストレージの設定

Image-registry Operator は、デフォルトストレージを提供しないプラットフォームでは最初は利用できません。インストール後に、レジストリー Operator を使用できるようにレジストリーをストレージを使用するように設定する必要があります。

実稼働クラスターに必要な PersistentVolume の設定方法と、実稼働用ではないクラスターにのみ使用できる空のディレクトリーをストレージの場所として設定する方法が表示されます。

1.1.14.2.1. ベアメタルの場合のレジストリーストレージの設定

クラスター管理者は、インストール後にレジストリーをストレージを使用できるように設定する必要があります。

前提条件

- クラスター管理者のパーミッション。
- ベアメタル上のクラスター。
- Red Hat OpenShift Container Storage などのクラスターの永続ストレージをプロビジョニングします。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。
- 容量は「100Gi」以上である。

手順

1. レジストリーをストレージを使用できるように設定するには、`configs.imageregistry/cluster` リソースの `spec.storage.pvc` を変更します。



注記

NFS などの共有ストレージを使用する場合は、**fsGroup** ID ではなく、セキュリティーコンテキストの許可される補助グループを定める **supplementalGroups** ストラテジーを使用することが強く推奨されます。詳細は、NFS の **グループ ID** についてのドキュメントを参照してください。

2. レジストリー Pod がないことを確認します。

```
$ oc get pod -n openshift-image-registry
```



注記

- ストレージタイプが **emptyDIR** の場合、レプリカ数が **1** を超えることはありません。
- ストレージタイプが **NFS** の場合、**no_wdelay** および **root_squash** マウントオプションを有効にする必要があります。以下は例になります。

```
# cat /etc/exports
/mnt/data *(rw,sync,no_wdelay,root_squash,insecure,fsid=0)
sh-4.3# exportfs -rv
exporting */mnt/data
```

3. レジストリー設定を確認します。

```
$ oc edit configs.imageregistry.operator.openshift.io

storage:
pvc:
claim:
```

claim フィールドを空のままにし、**image-registry-storage** PVC の自動作成を可能にします。

4. **clusteroperator** ステータスを確認します。

```
$ oc get clusteroperator image-registry
```

1.1.14.2.2. 実稼働以外のクラスターでのイメージレジストリーのストレージの設定

イメージレジストリー Operator のストレージを設定する必要があります。実稼働用以外のクラスターの場合、イメージレジストリーは空のディレクトリーに設定することができます。これを実行する場合、レジストリーを再起動するとすべてのイメージが失われます。

手順

- イメージレジストリーストレージを空のディレクトリーに設定するには、以下を実行します。

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec": {"storage":{"emptyDir":{}}}'
```



警告

実稼働用以外のクラスターにのみこのオプションを設定します。

イメージレジストリー Operator がそのコンポーネントを初期化する前にこのコマンドを実行する場合、**oc patch** コマンドは以下のエラーを出して失敗します。

```
Error from server (NotFound): configs.imageregistry.operator.openshift.io "cluster" not found
```

数分待機した後に、このコマンドを再び実行します。

1.1.15. 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の完了

Operator 設定の完了後に、提供するインフラストラクチャーでのクラスターのインストールを終了できます。

前提条件

- コントロールプレーンが初期化されています。
- Operator の初期設定を完了済みです。

手順

1. すべてのクラスターコンポーネントがオンラインであることを確認します。

```
$ watch -n5 oc get clusteroperators
```

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
authentication	4.3.0	True	False	False	10m
cloud-credential	4.3.0	True	False	False	22m
cluster-autoscaler	4.3.0	True	False	False	21m
console	4.3.0	True	False	False	10m
dns	4.3.0	True	False	False	21m
image-registry	4.3.0	True	False	False	16m
ingress	4.3.0	True	False	False	16m
kube-apiserver	4.3.0	True	False	False	19m
kube-controller-manager	4.3.0	True	False	False	18m
kube-scheduler	4.3.0	True	False	False	22m
machine-api	4.3.0	True	False	False	22m
machine-config	4.3.0	True	False	False	18m
marketplace	4.3.0	True	False	False	18m
monitoring	4.3.0	True	False	False	18m
network	4.3.0	True	False	False	16m
node-tuning	4.3.0	True	False	False	21m
openshift-apiserver	4.3.0	True	False	False	21m
openshift-controller-manager	4.3.0	True	False	False	17m
openshift-samples	4.3.0	True	False	False	14m
operator-lifecycle-manager	4.3.0	True	False	False	21m
operator-lifecycle-manager-catalog	4.3.0	True	False	False	21m
service-ca	4.3.0	True	False	False	21m
service-catalog-apiserver	4.3.0	True	False	False	16m
service-catalog-controller-manager	4.3.0	True	False	False	16m
storage	4.3.0	True	False	False	16m

すべてのクラスター Operator が **AVAILABLE** の場合、インストールを完了することができます。

2. クラスターの完了をモニターします。

```
$ ./openshift-install --dir=<installation_directory> wait-for install-complete 1
INFO Waiting up to 30m0s for the cluster to initialize...
```

-

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

Cluster Version Operator が Kubernetes API サーバーから OpenShift Container Platform クラスターのデプロイを終了するとコマンドは成功します。



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る証明書が含まれます。最初の証明書のローテーションが正常に実行されるようにするには、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。

3. Kubernetes API サーバーが Pod と通信していることを確認します。
 - a. すべての Pod の一覧を表示するには、以下のコマンドを使用します。

```
$ oc get pods --all-namespaces
```

NAMESPACE	NAME	READY	STATUS
RESTARTS	AGE		
openshift-apiserver-operator	openshift-apiserver-operator-85cb746d55-zqhs8	1/1	
Running	1 9m		
openshift-apiserver	apiserver-67b9g	1/1	Running 0
3m			
openshift-apiserver	apiserver-ljcmx	1/1	Running 0
1m			
openshift-apiserver	apiserver-z25h4	1/1	Running 0
2m			
openshift-authentication-operator	authentication-operator-69d5d8bf84-vh2n8	1/1	
Running	0 5m		
...			

- b. 以下のコマンドを使用して、直前のコマンドの出力に一覧表示される Pod のログを表示します。

```
$ oc logs <pod_name> -n <namespace> 1
```

- 1 直前のコマンドの出力にあるように、Pod 名および namespace を指定します。

Pod のログが表示される場合、Kubernetes API サーバーはクラスターマシンと通信できません。

1.1.16. 次のステップ

- [クラスターをカスタマイズ](#)します。
- 必要な場合は、[リモートの健全性レポートをオプトアウト](#)することができます。
- [レジストリーをセットアップ](#)し、[レジストリーストレージを設定](#)します。

1.2. ネットワークのカスタマイズによるベアメタルへのクラスタのインストール

OpenShift Container Platform バージョン 4.3 では、カスタマイズされたネットワーク設定オプションでプロビジョニングするベアメタルインフラストラクチャーにクラスタをインストールできます。ネットワーク設定をカスタマイズすることにより、クラスタは環境内の既存の IP アドレスの割り当てと共存でき、既存の MTU および VXLAN 設定と統合できます。

大半のネットワーク設定パラメーターはインストール時に設定する必要があり、実行中のクラスタで変更できるのは **kubeProxy** 設定パラメーターのみになります。

1.2.1. 前提条件

- [OpenShift Container Platform のインストールおよび更新](#) プロセスについての詳細を確認します。
- ファイアウォールを使用する場合、[Red Hat Insights にアクセスできるように設定](#) する必要があります。

1.2.2. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス

OpenShift Container Platform 4.3 では、クラスタ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。クラスタ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、クラスタは [Red Hat OpenShift Cluster Manager \(OCM\)](#) に登録されます。

Red Hat OpenShift Cluster Manager インベントリーが Telemetry によって自動的に維持されるか、または OCM を手動で使用しているかのいずれによって正常である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レベルで OpenShift Container Platform サブスクリプションを追跡します。

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。

- [Red Hat OpenShift Cluster Manager](#) ページにアクセスし、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を自動的に使用します。
- クラスタ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クラスタのインストールおよびインストールプログラムの生成に必要な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。

1.2.3. 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場合のクラスタのマシン要件

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含むクラスタの場合、必要なマシンすべてをデプロイする必要があります。

1.2.3.1. 必要なマシン

最小の OpenShift Container Platform クラスタでは以下のホストが必要です。

- 1つの一時的なブートストラップマシン
- 3つのコントロールプレーン、またはマスター、マシン
- 少なくとも2つのコンピュータマシン (ワーカーマシンとしても知られる)。



注記

クラスタでは、ブートストラップマシンが OpenShift Container Platform クラスタを3つのコントロールプレーンマシンにデプロイする必要があります。クラスタのインストール後にブートストラップマシンを削除できます。



重要

クラスタの高可用性を維持するには、これらのクラスタマシンについて別個の物理ホストを使用します。

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タマシンでは、Red Hat Enterprise Linux CoreOS (RHCOS) をオペレーティングシステムとして使用する必要があります。

RHCOS は Red Hat Enterprise Linux 8 をベースとしており、そのハードウェア認定および要件が継承されることに注意してください。「[Red Hat Enterprise Linux テクノロジーの機能と制限](#)」を参照してください。

1.2.3.2. ネットワーク接続の要件

すべての Red Hat Enterprise Linux CoreOS (RHCOS) マシンでは、起動時に **initramfs** のネットワークがマシン設定サーバーから 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、Ignition 設定ファイルをダウンロードできるようにネットワーク接続を確立するために、マシンには DHCP サーバーまたはその静的 IP アドレスが設定されている必要になります。

1.2.3.3. 最小リソース要件

それぞれのクラスタマシンは、以下の最小要件を満たしている必要があります。

マシン	Operating System	vCPU	仮想 RAM	ストレージ
ブートストラップ	RHCOS	4	16 GB	120 GB
コントロールプレーン	RHCOS	4	16 GB	120 GB

マシン	Operating System	vCPU	仮想 RAM	ストレージ
コンピュータ	RHCOS または RHEL 7.6	2	8 GB	120 GB

1.2.3.4. 証明書署名要求の管理

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を使用する場合、クラスターの自動マシン管理へのアクセスは制限されるため、インストール後にクラスターの証明書署名要求 (CSR) のメカニズムを提供する必要があります。**kube-controller-manager** は kubelet クライアント CSR のみを承認します。**machine-approver** は、kubelet 認証情報を使用して要求される提供証明書の有効性を保証できません。適切なマシンがこの要求を発行したかどうかを確認できないためです。kubelet 提供証明書の要求の有効性を検証し、それらを承認する方法を判別し、実装する必要があります。

1.2.4. 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作成

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 OpenShift Container Platform クラスターをデプロイする前に、基礎となるインフラストラクチャーを作成する必要があります。

前提条件

- クラスターでサポートするインフラストラクチャーを作成する前に、「[OpenShift Container Platform 4.x のテスト済みインテグレーション](#)」ページを参照してください。

手順

1. 各ノードに DHCP を設定するか、または静的 IP アドレスを設定します。
2. 必要なロードバランサーをプロビジョニングします。
3. マシンのポートを設定します。
4. DNS を設定します。
5. ネットワーク接続を確認します。

1.2.4.1. 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** のネットワークがマシン設定サーバーから Ignition 設定をフェッチする必要があります。

初回の起動時に、Ignition 設定ファイルをダウンロードできるようネットワーク接続を確立するために、マシンには DHCP サーバーまたはその静的 IP アドレスが設定されている必要があります。

クラスターのマシンを長期間管理するために DHCP サーバーを使用することが推奨されています。DHCP サーバーが永続 IP アドレス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。

クラスターの正常なインストール後に各マスターノードで実行される Kubernetes API サーバーは、クラスターマシンのノード名を解決する必要があります。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することができます。

マシン間のネットワーク接続を、クラスタのコンポーネントが通信できるように設定する必要があります。すべてのマシンではクラスタ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。

表1.6 すべてのマシンに対応するすべてのマシン

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	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 および Geneve
	6081	VXLAN および Geneve
	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。
TCP/UDP	30000-32767	Kubernetes NodePort

表1.7 コントロールプレーンへのすべてのマシン

プロトコル	ポート	説明
TCP	2379-2380	etcd サーバー、ピア、およびメトリクスポート
	6443	Kubernetes API

ネットワークポロジリー要件

クラスタ用にプロビジョニングするインフラストラクチャーは、ネットワークポロジリーの以下の要件を満たす必要があります。



重要

OpenShift Container Platform では、すべてのノードが、プラットフォームコンテナのイメージをプルし、Telemetry データを Red Hat に提供するためにインターネットへの直接のアクセスが必要です。

ロードバランサー

OpenShift Container Platform をインストールする前に、以下の要件を満たす2つのロードバランサーをプロビジョニングする必要があります。

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、SSL パススルー、または SSL ブリッジモードと呼ばれます。SSL ブリッジモードを使用する場合は、API ルートの Server Name Indication (SNI) を有効にする必要があります。
 - 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。



注記

API ロードバランサーが適切に機能するには、セッション永続性は必要ありません。

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1.8 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
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が、5 秒または 10 秒ごとにプローブし、2 つの正常な要求が正常な状態になり、3 つの要求が正常な状態になりません。これら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。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、SSL パススルー、または SSL ブリッジモ

- ▼ Layer 7 の負荷分散方法。シール、raw TCP、SSL バッファリング、または SSL プロキシモードと呼ばれます。SSL ブリッジモードを使用する場合は、Ingress ルートの Server Name Indication (SNI) を有効にする必要があります。
- 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。

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1.9 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 ルーター 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 ルーター 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

ヒント

クライアントの実際の IP アドレスが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。



注記

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。コントロールプレーンの初期化後に Ingress ルーターを設定する必要があります。

1.2.4.2. 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 DNS 要件

以下の DNS レコードは、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する OpenShift Container Platform クラスターに必要です。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クラスターの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1.10 必要な DNS レコード

コンポーネント	レコード	説明
---------	------	----

コンポーネント	レコード	説明
Kubernetes API	api.<cluster_name>.<base_domain>.	この DNS A/AAAA または CNAME レコードは、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必要があります。このレコードは、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>.	この DNS A/AAAA または CNAME レコードは、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必要があります。このレコードは、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。これがノード名を解決できない場合、プロキシされる 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。
ルート	*.apps.<cluster_name>.<base_domain>.	Ingress ルーター Pod を実行するマシンをターゲットにする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 レコードです。このレコードは、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。

コンポーネント	レコード	説明
etcd	etcd-<code><index></code>.<code><cluster_name></code>.<code><base_domain></code>.	<p>OpenShift Container Platform では、各 etcd インスタンスの DNS A/AAAA レコードがインスタンスをホストするコントロールプレーンマシンを参照する必要があります。etcd インスタンスは <code><index></code> 値によって差別化されます。この値は 0 で始まり、n-1 で終了します。ここで、n はクラスターのコントロールプレーンマシンの数です。DNS レコードはコントロールプレーンマシンのユニキャスト IPv4 アドレスに解決し、レコードはクラスター内のすべてのノードで解決可能である必要があります。</p>
	<code>_etcd-server-ssl._tcp.<cluster_name></code>. <code><base_domain></code>.	<p>それぞれのコントロールプレーンマシンについて、OpenShift Container Platform では、そのマシンに優先度 0、重み 10 およびポート 2380 の etcd サーバーの SRV DNS レコードも必要になります。3つのコントロールプレーンマシンを使用するクラスターには以下のレコードが必要です。</p> <pre># _service._proto.name. TTL class SRV priority weight port target. _etcd-server-ssl._tcp. <cluster_name>. <base_domain>. 86400 IN SRV 0 10 2380 etcd- 0.<cluster_name>. <base_domain> _etcd-server-ssl._tcp. <cluster_name>. <base_domain>. 86400 IN SRV 0 10 2380 etcd- 1.<cluster_name>. <base_domain> _etcd-server-ssl._tcp. <cluster_name>. <base_domain>. 86400 IN SRV 0 10 2380 etcd- 2.<cluster_name>. <base_domain></pre>

1.2.5. SSH プライベートキーの生成およびエージェントへの追加

クラスターで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がある場合、**ssh-agent** とインストールプログラムの両方に SSH キーを指定する必要があります。



注記

実稼働環境では、障害復旧およびデバッグが必要です。

このキーを使用して、ユーザー **core** としてマスターノードに対して SSH を実行できます。クラスターをデプロイする際に、キーは **core** ユーザーの `~/.ssh/authorized_keys` 一覧に追加されます。



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. パスワードなしの認証に設定されている SSH キーがコンピューター上にない場合は、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 rsa -b 4096 -N "" \
-f <path>/<file_name> ①
```

- ① `~/.ssh/id_rsa` などの、SSH キーのパスおよびファイル名を指定します。既存の SSH キーは上書きされるため、指定しないでください。

このコマンドを実行すると、指定した場所にパスワードを必要としない SSH キーが生成されます。

2. **ssh-agent** プロセスをバックグラウンドタスクとして開始します。

```
$ eval "$(ssh-agent -s)"
Agent pid 31874
```

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

```
$ ssh-add <path>/<file_name> ①
Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

- ① `~/.ssh/id_rsa` 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。

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。

1.2.6. 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、インストールファイルをローカルコンピューターにダウンロードします。

前提条件

- Linux または macOS を使用するコンピューターからクラスターをインストールする必要があります。
- インストールプログラムをダウンロードするには、500 MB のローカルディスク領域が必要です。

手順

1. Red Hat OpenShift Cluster Manager サイトの「[Infrastructure Provider](#)」ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使ってログインします。アカウントがない場合はこれを作成します。
2. 選択するインストールタイプのページに移動し、オペレーティングシステムのインストールプログラムをダウンロードし、ファイルをインストール設定ファイルを保存するディレクトリに配置します。



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。クラスターインストールの完了後は、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の両方を保持する必要があります。



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。特定のクラウドプロバイダー用に記載された OpenShift Container Platform のアンインストール手順を完了して、クラスターを完全に削除する必要があります。

3. 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 <installation_program>.tar.gz
```

4. Red Hat OpenShift Cluster Manager サイトの「[Pull Secret](#)」ページから、インストールプルシークレットを **.txt** ファイルとして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。

1.2.7. バイナリーのダウンロードによる CLI のインストール

コマンドラインインターフェースを使用して 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 4.3 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。

1.2.7.1. Linux への 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 OpenShift Cluster Manager サイトの「[Infrastructure Provider](#)」ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Linux** を選択し、**Download command-line tools** をクリックします。
4. アーカイブを展開します。

```
$ tar xvzf <file>
```

5. **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
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.2.7.2. Windows での 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 OpenShift Cluster Manager サイトの「[Infrastructure Provider](#)」ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Windows** を選択し、**Download command-line tools** をクリックします。
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
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

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

1.2.7.3. macOS への 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 OpenShift Cluster Manager サイトの「[Infrastructure Provider](#)」ページに移動します。
2. インフラストラクチャプロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **MacOS** を選択し、**Download command-line tools** をクリックします。
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
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.2.8. インストール設定ファイルの手動作成

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 OpenShift Container Platform のインストールでは、インストール設定ファイルを手動で生成する必要があります。

前提条件

- OpenShift Container Platform インストーラープログラムおよびクラスターのアクセストークンを取得します。

手順

1. 必要なインストールアセットを保存するためのインストールディレクトリーを作成します。

```
$ mkdir <installation_directory>
```



重要

ディレクトリを作成する必要があります。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の有効期限は短く設定されているため、インストールディレクトリを再利用することができません。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。

- 以下の **install-config.yaml** ファイルテンプレートをカスタマイズし、これを **<installation_directory>** に保存します。



注記

この設定ファイル **install-config.yaml** に名前を付ける必要があります。

- install-config.yaml** ファイルをバックアップし、これを複数のクラスターをインストールするため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の次の手順で使用されます。この時点でこれをバックアップする必要があります。

1.2.8.1. ベアメタルのサンプル 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、または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:
- hyperthreading: Enabled 2 3
  name: worker
  replicas: 0 4
controlPlane:
  hyperthreading: Enabled 5 6
  name: master 7
  replicas: 3 8
metadata:
  name: test 9
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14 10
    hostPrefix: 23 11
  networkType: OpenShiftSDN
  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ます。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、コンピュート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。どちらのセクションも、現時点では単一のマシンプールを定義しますが、OpenShift Container Platform の今後のバージョンでは、インストール時の複数のコンピュートプールの定義をサポートする可能性があります。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 7** 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。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。一部のクラスターマシンで同時マルチスレッドを無効にする場合は、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。



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。

- 4** **replicas** パラメーターの値を **0** に設定する必要があります。このパラメーターはクラスターが作成し、管理するワーカーの数を制御します。これは、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する場合にクラスターが実行しない機能です。OpenShift Container Platform のインストールが終了する前に、クラスターが使用するワーカーマシンを手動でデプロイする必要があります。
- 8** 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。
- 9** DNS レコードに指定したクラスター名。
- 10** 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。
- 11** それぞれの個別ノードに割り当てるサブネットプレフィックスの長さ。たとえば、**hostPrefix** が **23** に設定され、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 (510 (2³²⁻²³ - 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。
- 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。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。ベアメタル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。

- 15 Red Hat OpenShift Cluster Manager サイトの「[Pull Secret](#)」ページから取得したプルシークレット。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** ユーザーのデフォルト SSH キーの公開部分。



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がある実稼働用の 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。

1.2.8.2. ネットワーク設定パラメーター

クラスターのネットワーク設定パラメーターは **install-config.yaml** 設定ファイルで変更できます。以下の表では、これらのパラメーターについて説明しています。



注記

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

表1.11 必要なネットワークパラメーター

パラメーター	説明	値
networking.networkType	デプロイするデフォルトの Container Network Interface (CNI) ネットワークプロバイダープラグイン。 OpenShiftSDN プラグインのみが OpenShift Container Platform 4.3 でサポートされているプラグインです。	デフォルト値は OpenShiftSDN です。
networking.clusterNetwork.cidr	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。 OpenShiftSDN ネットワークプラグインは複数のクラスターネットワークをサポートします。複数のクラスターネットワークのアドレスブロックには重複が許可されません。予想されるワークロードに適したサイズのアドレスプールを選択してください。	CIDR 形式の IP アドレスの割り当て。デフォルト値は 10.128.0.0/14 です。
networking.clusterNetwork.hostPrefix	それぞれの個別ノードに割り当てるサブネットプレフィックスの長さ。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 (510 (2 ³²⁻²³) - 2) Pod IP アドレスが許可されます)。	サブネットプレフィックス。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 アドレスのブロック。 OpenShiftSDN は1つの serviceNetwork ブロックのみを許可します。このアドレスブロックは他のネットワークブロックと重複できません。	CIDR 形式の IP アドレスの割り当て。デフォルト値は 172.30.0.0/16 です。
networking.machineCIDR	クラスターのインストール中に OpenShift Container Platform インストールプログラムによって使用される IP アドレスのブロック。このアドレスブロックは他のネットワークブロックと重複できません。	CIDR 形式の IP アドレスの割り当て。デフォルト値は 10.0.0.0/16 です。

1.2.9. 高度なネットワーク設定パラメーターの変更

高度なネットワーク設定パラメーターは、クラスターのインストール前にのみ変更することができます。高度な設定のカスタマイズにより、クラスターを既存のネットワーク環境に統合させることができます。これを実行するには、MTU または VXLAN ポートを指定し、**kube-proxy** 設定のカスタマイズを許可し、**openshiftSDNConfig** パラメーターに異なる **mode** を指定します。



重要

OpenShift Container Platform マニフェストファイルの直接の変更はサポートされていません。

前提条件

- **install-config.yaml** ファイルを作成し、これに対する変更を完了します。
- クラスターの Ignition 設定ファイルを生成します。

手順

1. 以下のコマンドを使用してマニフェストを作成します。

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

- 1** **<installation_directory>** については、クラスターの **install-config.yaml** ファイルが含まれるディレクトリーの名前を指定します。

2. **<installation_directory>/manifests/cluster-scheduler-02-config.yml** Kubernetes マニフェストファイルを変更し、Pod がコントロールプレーンマシンにスケジュールされないようにします。
 - a. **manifests/cluster-scheduler-02-config.yml** ファイルを開きます。
 - b. **mastersSchedulable** パラメーターを見つけ、その値を **False** に設定します。
 - c. ファイルを保存し、終了します。



注記

現時点では、[Kubernetes の制限](#)により、コントロールプレーンマシンで実行されるルーター Pod に Ingress ロードバランサーがアクセスすることができません。

3. **cluster-network-03-config.yml** という名前のファイルを `<installation_directory>/manifests/` ディレクトリーに作成します。

```
$ touch <installation_directory>/manifests/cluster-network-03-config.yml ❶
```

- ❶ `<installation_directory>` については、クラスターの **manifests/** ディレクトリーが含まれるディレクトリー名を指定します。

ファイルの作成後は、以下のようにいくつかのネットワーク設定ファイルが **manifests/** ディレクトリーに置かれます。

```
$ ls <installation_directory>/manifests/cluster-network-*
```

出力例

```
cluster-network-01-crd.yml
cluster-network-02-config.yml
cluster-network-03-config.yml
```

4. エディターで **cluster-network-03-config.yml** ファイルを開き、必要な Operator 設定を記述する CR を入力します。

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec: ❶
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  serviceNetwork:
    - 172.30.0.0/16
  defaultNetwork:
    type: OpenShiftSDN
    openshiftSDNConfig:
      mode: NetworkPolicy
      mtu: 1450
      vxlanPort: 4789
```

- ❶ **spec** パラメーターのパラメーターは例です。CR に Cluster Network Operator の設定を指定します。

CNO は CR にパラメーターのデフォルト値を提供するため、変更が必要なパラメーターのみを指定する必要があります。

5. **cluster-network-03-config.yml** ファイルを保存し、テキストエディターを終了します。

- オプション: **manifests/cluster-network-03-config.yml** ファイルをバックアップします。インストールプログラムは、クラスタの作成時に **manifests/** ディレクトリを削除します。

1.2.10. Cluster Network Operator (CNO) の設定

クラスタネットワークの設定は、Cluster Network Operator (CNO) 設定の一部として指定され、**cluster** という名前の CR オブジェクトに保存されます。CR は **operator.openshift.io** API グループの **Network** API のパラメーターを指定します。

defaultNetwork パラメーターのパラメーター値を CNO CR に設定することにより、OpenShift Container Platform クラスタのクラスタネットワーク設定を指定できます。以下の CR は、CNO のデフォルト設定を表示し、設定可能なパラメーターと有効なパラメーターの値の両方について説明しています。

Cluster Network Operator CR

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  clusterNetwork: ①
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  serviceNetwork: ②
  - 172.30.0.0/16
  defaultNetwork: ③
  ...
  kubeProxyConfig: ④
  iptablesSyncPeriod: 30s ⑤
  proxyArguments:
    iptables-min-sync-period: ⑥
    - 0s
```

- ① ② **install-config.yaml** ファイルに指定されます。
- ③ クラスタネットワークのデフォルトの Container Network Interface (CNI) ネットワークプロバイダーを設定します。
- ④ このオブジェクトのパラメーターは、**kube-proxy** 設定を指定します。パラメーターの値を指定しない場合、クラスタネットワーク Operator は表示されるデフォルトのパラメーター値を適用します。OVN-Kubernetes デフォルト CNI ネットワークプロバイダーを使用している場合、**kube-proxy** 設定は機能しません。
- ⑤ **iptables** ルールの更新期間。デフォルト値は **30s** です。有効なサフィックスには、**s**、**m**、および **h** などが含まれ、これらについては、[Go Package time](#) ドキュメントで説明されています。



注記

OpenShift Container Platform 4.3 以降で強化されたパフォーマンスの向上により、**iptablesSyncPeriod** パラメーターを調整する必要はなくなりました。

- ⑥ **iptables** ルールを更新する前の最小期間。このパラメーターにより、更新の頻度が高くなり過ぎないようにできます。有効なサフィックスには、**s**、**m**、および **h** などが含まれ、これらについては、[Go Package time](#) ドキュメントで説明されています。

は、[Go Package time](#) で説明されています。

1.2.10.1. OpenShift SDN デフォルト CNI ネットワークプロバイダーの設定パラメーター

以下の YAML オブジェクトは、OpenShift SDN デフォルト Container Network Interface (CNI) ネットワークプロバイダーの設定パラメーターについて説明しています。

```
defaultNetwork:
  type: OpenShiftSDN ①
  openshiftSDNConfig: ②
    mode: NetworkPolicy ③
    mtu: 1450 ④
    vxlanPort: 4789 ⑤
```

- ① **install-config.yaml** ファイルに指定されます。
- ② OpenShift SDN 設定の一部を上書きする必要がある場合にのみ指定します。
- ③ OpenShift SDN のネットワーク分離モードを設定します。許可される値は **Multitenant**、**Subnet**、または **NetworkPolicy** です。デフォルト値は **NetworkPolicy** です。
- ④ VXLAN オーバーレイネットワークの maximum transmission unit (MTU)。この値は通常は自動的に設定されますが、クラスターにあるノードすべてが同じ MTU を使用しない場合、これを最小のノード MTU 値よりも 50 小さくする必要があります。
- ⑤ すべての VXLAN パケットに使用するポート。デフォルト値は **4789** です。別の VXLAN ネットワークの一部である既存ノードと共に仮想化環境で実行している場合は、これを変更する必要があります。たとえば、OpenShift SDN オーバーレイを VMware NSX-T 上で実行する場合は、両方の SDN が同じデフォルトの VXLAN ポート番号を使用するため、VXLAN の別のポートを選択する必要があります。

Amazon Web Services (AWS) では、VXLAN にポート **9000** とポート **9999** 間の代替ポートを選択できます。

1.2.10.2. Cluster Network Operator の設定例

以下の例のように、CNO の完全な CR オブジェクトが表示されます。

Cluster Network Operator のサンプル CR

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
  serviceNetwork:
    - 172.30.0.0/16
  defaultNetwork:
    type: OpenShiftSDN
    openshiftSDNConfig:
```

```

mode: NetworkPolicy
mtu: 1450
vxlanPort: 4789
kubeProxyConfig:
  iptablesSyncPeriod: 30s
  proxyArguments:
    iptables-min-sync-period:
      - 0s

```

1.2.11. Ignition 設定ファイルの作成

クラスターマシンは手動で起動する必要があるため、クラスターがマシンを作成するために必要な Ignition 設定ファイルを生成する必要があります。



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る証明書が含まれます。最初の証明書のローテーションが正常に実行されるようにするには、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。

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します。

手順

1. Ignition 設定ファイルを取得します。

```
$ ./openshift-install create ignition-configs --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。



重要

install-config.yaml ファイルを作成している場合、それが含まれるディレクトリーを指定します。または、空のディレクトリーを指定します。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の有効期限は短く設定されているため、インストールディレクトリーを再利用することができません。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。

以下のファイルはディレクトリーに生成されます。

```

.
├── auth
└── kubeadmin-password

```



1.2.12. Red Hat Enterprise Linux CoreOS (RHCOS) マシンの作成

ユーザーによってプロビジョニングされるベアメタル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前に、それが使用する RHCOS マシンを作成する必要があります。ISO イメージまたはネットワーク PXE ブートを使用する手順を実行してマシンを作成することができます。

1.2.12.1. ISO イメージを使用した Red Hat Enterprise Linux CoreOS (RHCOS) マシンの作成

ユーザーによってプロビジョニングされるベアメタル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前に、それが使用する RHCOS マシンを作成する必要があります。ISO イメージを使用してマシンを作成することができます。

前提条件

- クラスターの Ignition 設定ファイルを取得していること。
- お使いのコンピューターからアクセスでき、作成するマシンがアクセスできる HTTP サーバーへのアクセスがあること。

手順

1. インストールプログラムが作成したコントロールプレーン、コンピュート、およびブートストラップ Ignition 設定を HTTP サーバーにアップロードします。これらのファイルの URL をメモします。



重要

インストールの完了後にコンピュートマシンをさらにクラスターに追加する予定の場合には、これらのファイルを削除しないでください。

2. Red Hat カスタマーポータル[の「製品のダウンロード」](#)ページまたは「[RRHCOS イメージミラー](#)」ページからオペレーティングシステムのインスタンスをインストールするために優先される方法で必要な RHCOS イメージを取得します。



重要

RHCOS イメージは OpenShift Container Platform の各リリースごとに変更されない可能性があります。インストールする OpenShift Container Platform バージョンと等しいか、それ以下のバージョンの中で最も新しいバージョンのイメージをダウンロードする必要があります。利用可能な場合は、OpenShift Container Platform バージョンに一致するイメージのバージョンを使用します。

ISO ファイルおよび RAW ディスクファイルをダウンロードする必要があります。これらのファイルの名前は以下の例のようになります。

- ISO: `rhcos-<version>-installer.<architecture>.iso`

- 圧縮された metal RAW: `rhcos-<version>-metal.<architecture>.raw.gz`
3. RAW RHCOS イメージファイルのいずれかを HTTP サーバーにアップロードし、その URL をメモします。



重要

インストールの完了後にコンピュータマシンをさらにクラスターに追加する予定の場合には、これらのファイルを削除しないでください。

4. ISO を使用し、RHCOS インストールを開始します。以下のインストールオプションのいずれかを使用します。
 - ディスクに ISO イメージを書き込み、これを直接起動します。
 - LOM インターフェースで ISO リダイレクトを使用します。
5. インスタンスの起動後に、**TAB** または **E** キーを押してカーネルコマンドラインを編集します。
6. パラメーターをカーネルコマンドラインに追加します。

```
coreos.inst=yes
coreos.inst.install_dev=sda ①
coreos.inst.image_url=<image_URL> ②
coreos.inst.ignition_url=http://example.com/config.ign ③
ip=<dhcp or static IP address> ④ ⑤
```

- ① インストール先のシステムのブロックデバイスを指定します。
- ② サーバーにアップロードした RAW イメージの URL を指定します。
- ③ このマシンタイプの Ignition 設定ファイルの URL を指定します。
- ④ `ip=dhcp` を設定するか、各ノードに個別の静的 IP アドレス (`ip=`) および DNS サーバー (`nameserver=`) を設定します。たとえば、`ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp1s0:none nameserver=4.4.4.41` を設定すると、以下のように設定されます。
 - ノードの IP アドレス: **10.10.10.2**
 - ゲートウェイアドレス: **10.10.10.254**
 - ネットマスク: **255.255.255.0**
 - ホスト名: **core0.example.com**
 - DNS サーバーアドレス: **4.4.4.41**
- ⑤ 複数のネットワークインターフェースまたは DNS サーバーを使用する場合は、以下の方法で IP アドレス設定をさらにカスタマイズできます。
 - システムに複数のネットワークインターフェースがある場合には、複数の `ip=` エントリーを指定して、複数の IP アドレス設定を追加できます。
 - システムに複数のネットワークインターフェースがある場合には、DHCP と静的 IP の設定の組み合わせを使用できます。使用するインターフェースを指定する必要があります。

まず (例: `ip=enp1s0:dhcp`
`ip=10.10.10.2::255.255.255.0:core0.example.com:enp2s0:none`)。

- 各サーバーに `nameserver=` エントリーを追加して、複数の DNS サーバーを指定できます (例: `nameserver=1.1.1.1 nameserver=8.8.8.8`)。
7. Enter を押してインストールを完了します。RHCOS のインストール後に、システムは再起動します。システムの再起動後、指定した Ignition 設定ファイルを適用します。
 8. 継続してクラスターのマシンを作成します。



重要

この時点でブートストラップおよびコントロールプレーンマシンを作成する必要があります。一部の Pod はデフォルトでコンピュータマシンにデプロイされるため、クラスターのインストール前に、2 つ以上のコンピュータマシンを作成します。

1.2.12.2. PXE または iPXE ブートによる Red Hat Enterprise Linux CoreOS (RHCOS) マシンの作成

ユーザーによってプロビジョニングされるベアメタル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前に、それが使用する RHCOS マシンを作成する必要があります。PXE または iPXE ブートを使用してマシンを作成することができます。

前提条件

- クラスターの Ignition 設定ファイルを取得していること。
- 適切な PXE または iPXE インフラストラクチャーを設定していること。
- 使用しているコンピューターからアクセス可能な HTTP サーバーへのアクセスがあること。

手順

1. インストールプログラムが作成したマスター、ワーカーおよびブートストラップの Ignition 設定を HTTP サーバーにアップロードします。これらのファイルの URL をメモします。



重要

インストールの完了後にコンピュータマシンをさらにクラスターに追加する予定の場合には、これらのファイルを削除しないでください。

2. Red Hat カスタマーポータル[の「製品のダウンロード」](#)ページまたは[「RHCOS イメージミラー」](#)ページから圧縮された metal RAW イメージ、`kernel` および `initramfs` ファイルを取得します。



重要

RHCOS イメージは OpenShift Container Platform の各リリースごとに変更されない可能性があります。インストールする OpenShift Container Platform バージョンと等しいか、それ以下のバージョンの中で最も新しいバージョンのイメージをダウンロードする必要があります。利用可能な場合は、OpenShift Container Platform バージョンに一致するイメージのバージョンを使用します。この手順には RAW イメージのみを使用します。RHCOS qcow2 イメージは、ベアメタルのインストールではサポートされません。

ファイル名には、OpenShift Container Platform のバージョン名が含まれます。以下の例のようになります。

- 圧縮されたメタル RAW イメージ: `rhcos-<version>metal.<architecture>.raw.gz`
 - **kernel**: `rhcos-<version>-installer-kernel-<architecture>`
 - **initramfs**: `rhcos-<version>-installer-initramfs.<architecture>.img`
3. 圧縮された metal RAW イメージおよび **kernel** および **initramfs** ファイルを HTTP サーバーにアップロードします。



重要

インストールの完了後にコンピュータマシンをさらにクラスターに追加する予定の場合には、これらのファイルを削除しないでください。

4. RHCOS のインストール後にマシンがローカルディスクから起動されるようにネットワークブートインフラストラクチャーを設定します。
5. RHCOS イメージに PXE または iPXE インストールを設定します。
ご使用の環境についての以下の例で示されるメニューエントリーのいずれかを変更し、イメージおよび Ignition ファイルが適切にアクセスできることを確認します。
- PXE の場合:

```

DEFAULT pxeboot
TIMEOUT 20
PROMPT 0
LABEL pxeboot
  KERNEL http://<HTTP_server>/rhcos-<version>-installer-kernel-<architecture> 1
  APPEND ip=dhcp rd.neednet=1 initrd=http://<HTTP_server>/rhcos-<version>-installer-
initramfs.<architecture>.img console=tty0 console=ttyS0 coreos.inst=yes
coreos.inst.install_dev=sda coreos.inst.image_url=http://<HTTP_server>/rhcos-
<version>-metal.<architecture>.raw.gz
coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign 2 3

```

- 1** HTTP サーバーにアップロードした **kernel** ファイルの場所を指定します。
- 2** 複数の NIC を使用する場合、**ip** オプションに単一インターフェースを指定します。たとえば、**eno1** という名前の NIC で DHCP を使用するには、**ip=eno1:dhcp** を設定します。
- 3** HTTP サーバーにアップロードした RHCOS ファイルの場所を指定します。**initrd** パ

- iPXEの場合:

```
kernel http://<HTTP_server>/rhcos-<version>-installer-kernel-<architecture> ip=dhcp
rd.neednet=1 initrd=http://<HTTP_server>/rhcos-<version>-installer-initramfs.
<architecture>.img console=tty0 console=ttyS0 coreos.inst=yes
coreos.inst.install_dev=sda coreos.inst.image_url=http://<HTTP_server>/rhcos-
<version>-metal.<architecture>.raw.gz
coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign ❶ ❷
initrd http://<HTTP_server>/rhcos-<version>-installer-initramfs.<architecture>.img ❸
boot
```

- ❶ HTTP サーバーにアップロードした RHCOS ファイルの場所を指定します。 **kernel** パラメーター値は **kernel** ファイルの場所であり、 **initrd** パラメーター値は **initramfs** ファイルの場所、 **coreos.inst.image_url** パラメーター値は圧縮された metal RAW イメージの場所、および **coreos.inst.ignition_url** パラメーター値はブートストラップ Ignition 設定ファイルの場所になります。
- ❷ 複数の NIC を使用する場合、 **ip** オプションに単一インターフェースを指定します。たとえば、 **eno1** という名前の NIC で DHCP を使用するには、 **ip=eno1:dhcp** を設定します。
- ❸ HTTP サーバーにアップロードした **initramfs** ファイルの場所を指定します。

6. 継続してクラスターのマシンを作成します。



重要

この時点でブートストラップおよびコントロールプレーンマシンを作成する必要があります。一部の Pod はデフォルトでコンピュータマシンにデプロイされるため、クラスターのインストール前に、2 つ以上のコンピュータマシンを作成します。

1.2.13. クラスターの作成

OpenShift Container Platform クラスターを作成するには、ブートストラッププロセスが、インストールプログラムで生成した Ignition 設定ファイルを使用してプロビジョニングしたマシンで完了するのを待機します。

前提条件

- クラスターに必要なインフラストラクチャーを作成します。
- インストールプログラムを取得し、クラスターの Ignition 設定ファイルを生成済みです。
- Ignition 設定ファイルを使用して、クラスターの RHCOS マシンを作成済みです。
- 使用するマシンでインターネットに直接アクセスできます。

手順

1. ブートストラッププロセスをモニターします。

```
$ ./openshift-install --dir=<installation_directory> wait-for bootstrap-complete \ ❶
--log-level=info ❷
```

```
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com...
INFO API v1.16.2 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
```

- 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。

Kubernetes API サーバーでこれがコントロールプレーンマシンにブートストラップされていることを示すシグナルが出されるとコマンドは成功します。

2. ブートストラッププロセスが完了したら、ブートストラップマシンをロードバランサーから削除します。



重要

この時点で、ブートストラップマシンをロードバランサーから削除する必要があります。さらに、マシン自体を削除し、再フォーマットすることができます。

1.2.14. 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ついての情報が含まれます。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ます。
- **oc** CLI をインストールします。

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
system:admin
```

1.2.15. マシンの CSR の承認

マシンをクラスターに追加する際に、追加したそれぞれのマシンについて2つの保留状態の証明書署名要求 (CSR) が生成されます。これらの CSR が承認されていることを確認するか、または必要な場合はそれらを承認してください。

前提条件

- マシンがクラスターに追加されています。

手順

1. クラスターがマシンを認識していることを確認します。

```
$ oc get nodes

NAME      STATUS   ROLES    AGE   VERSION
master-0  Ready    master   63m   v1.16.2
master-1  Ready    master   63m   v1.16.2
master-2  Ready    master   64m   v1.16.2
worker-0  NotReady worker   76s   v1.16.2
worker-1  NotReady worker   70s   v1.16.2
```

出力には作成したすべてのマシンが一覧表示されます。

2. 保留中の証明書署名要求 (CSR) を確認し、クラスターに追加したそれぞれのマシンのクライアントおよびサーバー要求に **Pending** または **Approved** ステータスが表示されていることを確認します。

```
$ oc get csr

NAME      AGE   REQUESTOR                                     CONDITION
csr-8b2br 15m   system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper Pending 1
csr-8vnps 15m   system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper Pending
csr-bfd72 5m26s system:node:ip-10-0-50-126.us-east-2.compute.internal
Pending 2
csr-c57lv 5m26s system:node:ip-10-0-95-157.us-east-2.compute.internal
Pending
...
```

1 クライアント要求の CSR。

2 サーバー要求の CSR。

この例では、2つのマシンがクラスターに参加しています。この一覧にはさらに多くの承認された CSR が表示される可能性があります。

3. 追加したマシンの保留中の CSR すべてが **Pending** ステータスになった後に CSR が承認されない場合には、クラスターマシンの CSR を承認します。



注記

CSR のローテーションは自動的に実行されるため、クラスターにマシンを追加後1時間以内に CSR を承認してください。1時間以内に承認しない場合には、証明書のローテーションが行われ、各ノードに3つ以上の証明書が存在するようになります。これらの証明書すべてを承認する必要があります。最初の CSR の承認後、後続のノードクライアント CSR はクラスターの **kube-controller-manger** によって自動的に承認されます。kubelet 提供証明書の要求を自動的に承認する方法を実装する必要があります。

- それらを個別に承認するには、それぞれの有効な CSR について以下のコマンドを実行します。

```
$ oc adm certificate approve <csr_name> 1
```

- 1** **<csr_name>** は、現行の CSR の一覧からの CSR の名前です。

- すべての保留中の CSR を承認するには、以下のコマンドを実行します。

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{"\n"}{{end}}{{end}}' | xargs oc adm certificate approve
```

追加情報

- CSR の詳細は、「[Certificate Signing Requests](#)」を参照してください。

1.2.16. Operator の初期設定

コントロールプレーンの初期化後に、一部の Operator を利用可能にするためにそれらをすぐに設定する必要があります。

前提条件

- コントロールプレーンが初期化されています。

手順

- クラスターコンポーネントがオンラインになることを確認します。

```
$ watch -n5 oc get clusteroperators
```

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
authentication	4.3.0	True	False	False	69s
cloud-credential	4.3.0	True	False	False	12m
cluster-autoscaler	4.3.0	True	False	False	11m
console	4.3.0	True	False	False	46s
dns	4.3.0	True	False	False	11m
image-registry	4.3.0	True	False	False	5m26s
ingress	4.3.0	True	False	False	5m36s
kube-apiserver	4.3.0	True	False	False	8m53s
kube-controller-manager	4.3.0	True	False	False	7m24s
kube-scheduler	4.3.0	True	False	False	12m

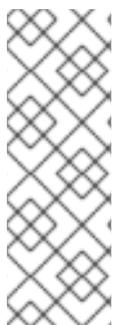
machine-api	4.3.0	True	False	False	12m
machine-config	4.3.0	True	False	False	7m36s
marketplace	4.3.0	True	False	False	7m54m
monitoring	4.3.0	True	False	False	7h54s
network	4.3.0	True	False	False	5m9s
node-tuning	4.3.0	True	False	False	11m
openshift-apiserver	4.3.0	True	False	False	11m
openshift-controller-manager	4.3.0	True	False	False	5m943s
openshift-samples	4.3.0	True	False	False	3m55s
operator-lifecycle-manager	4.3.0	True	False	False	11m
operator-lifecycle-manager-catalog	4.3.0	True	False	False	11m
service-ca	4.3.0	True	False	False	11m
service-catalog-apiserver	4.3.0	True	False	False	5m26s
service-catalog-controller-manager	4.3.0	True	False	False	5m25s
storage	4.3.0	True	False	False	5m30s

2. 利用不可の Operator を設定します。

1.2.16.1. インストール時に削除されたイメージレジストリー

共有可能なオブジェクトストレージを提供しないプラットフォームでは、OpenShift イメージレジストリー Operator 自体が **Removed** としてブートストラップされます。これにより、**openshift-installer** がそれらのプラットフォームタイプでのインストールを完了できます。

インストール後に、イメージレジストリー Operator 設定を編集して **managementState** を **Removed** から **Managed** に切り替える必要があります。



注記

Prometheus コンソールは、以下のような **ImageRegistryRemoved** アラートを提供しません。

"Image Registry has been removed.**ImageStreamTags**, **BuildConfigs** and **DeploymentConfigs** which reference **ImageStreamTags** may not work as expected.Please configure storage and update the config to **Managed** state by editing `configs.imageregistry.operator.openshift.io`."

1.2.16.2. イメージレジストリーストレージの設定

Image-registry Operator は、デフォルトストレージを提供しないプラットフォームでは最初は利用できません。インストール後に、レジストリー Operator を使用できるようにレジストリーをストレージを使用するように設定する必要があります。

実稼働クラスターに必要な **PersistentVolume** の設定方法と、実稼働用ではないクラスターにのみ使用できる空のディレクトリーをストレージの場所として設定する方法が表示されます。

1.2.17. 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の完了

Operator 設定の完了後に、提供するインフラストラクチャーでのクラスターのインストールを終了できます。

前提条件

- コントロールプレーンが初期化されています。
- Operator の初期設定を完了済みです。

手順

1. すべてのクラスターコンポーネントがオンラインであることを確認します。

```
$ watch -n5 oc get clusteroperators
```

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
authentication	4.3.0	True	False	False	10m
cloud-credential	4.3.0	True	False	False	22m
cluster-autoscaler	4.3.0	True	False	False	21m
console	4.3.0	True	False	False	10m
dns	4.3.0	True	False	False	21m
image-registry	4.3.0	True	False	False	16m
ingress	4.3.0	True	False	False	16m
kube-apiserver	4.3.0	True	False	False	19m
kube-controller-manager	4.3.0	True	False	False	18m
kube-scheduler	4.3.0	True	False	False	22m
machine-api	4.3.0	True	False	False	22m
machine-config	4.3.0	True	False	False	18m
marketplace	4.3.0	True	False	False	18m
monitoring	4.3.0	True	False	False	18m
network	4.3.0	True	False	False	16m
node-tuning	4.3.0	True	False	False	21m
openshift-apiserver	4.3.0	True	False	False	21m
openshift-controller-manager	4.3.0	True	False	False	17m
openshift-samples	4.3.0	True	False	False	14m
operator-lifecycle-manager	4.3.0	True	False	False	21m
operator-lifecycle-manager-catalog	4.3.0	True	False	False	21m
service-ca	4.3.0	True	False	False	21m
service-catalog-apiserver	4.3.0	True	False	False	16m
service-catalog-controller-manager	4.3.0	True	False	False	16m
storage	4.3.0	True	False	False	16m

すべてのクラスター Operator が **AVAILABLE** の場合、インストールを完了することができます。

2. クラスターの完了をモニターします。

```
$ ./openshift-install --dir=<installation_directory> wait-for install-complete 1
INFO Waiting up to 30m0s for the cluster to initialize...
```

- 1** **<installation_directory>** については、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。

Cluster Version Operator が Kubernetes API サーバーから OpenShift Container Platform クラスターのデプロイを終了するとコマンドは成功します。



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る証明書が含まれます。最初の証明書のローテーションが正常に実行されるようにするには、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。

3. Kubernetes API サーバーが Pod と通信していることを確認します。

a. すべての Pod の一覧を表示するには、以下のコマンドを使用します。

```
$ oc get pods --all-namespaces
```

NAMESPACE	NAME	READY	STATUS
RESTARTS	AGE		
openshift-apiserver-operator	openshift-apiserver-operator-85cb746d55-zqhs8	1/1	
Running	1 9m		
openshift-apiserver	apiserver-67b9g	1/1	Running 0
3m			
openshift-apiserver	apiserver-ljcmx	1/1	Running 0
1m			
openshift-apiserver	apiserver-z25h4	1/1	Running 0
2m			
openshift-authentication-operator	authentication-operator-69d5d8bf84-vh2n8	1/1	
Running	0 5m		
...			

b. 以下のコマンドを使用して、直前のコマンドの出力に一覧表示される Pod のログを表示します。

```
$ oc logs <pod_name> -n <namespace> ①
```

① 直前のコマンドの出力にあるように、Pod 名および namespace を指定します。

Pod のログが表示される場合、Kubernetes API サーバーはクラスターマシンと通信できません。

1.2.18. 次のステップ

- [クラスターをカスタマイズ](#)します。
- 必要な場合は、[リモートの健全性レポートをオプトアウト](#)することができます。
- [レジストリーをセットアップ](#)し、[レジストリーストレージを設定](#)します。

1.3. ネットワークが制限された環境でのクラスターのベアメタルへのインストール

OpenShift Container Platform バージョン 4.3 では、クラスターをネットワークが制限された環境でプロビジョニングするベアメタルインフラストラクチャーにインストールできます。



重要

以下の手順に従って仮想化環境またはクラウド環境にクラスターをデプロイすることができますが、ベアメタルプラットフォーム以外の場合は追加の考慮事項に注意してください。このような環境で OpenShift Container Platform クラスターのインストールを試行する前に、「[Deploying OpenShift 4.x on non-tested platforms using the bare metal install method](#)」にある情報を確認してください。

1.3.1. 前提条件

- ミラーホストでレジストリーを作成し、OpenShift Container Platform の使用しているバージョン用の **imageContentSources** データを取得します。



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します。

- クラスターの[永続ストレージ](#)をプロビジョニングします。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。
- [OpenShift Container Platform のインストールおよび更新](#)プロセスについての詳細を確認します。
- ファイアウォールを使用し、Telemetry を使用する予定がある場合は、クラスターがアクセスする必要のある[サイトを許可するようにファイアウォールを設定](#)する必要があります。



注記

プロキシを設定する場合は、このサイト一覧も確認してください。

1.3.2. 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 4.3 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。インストールプログラムでプロビジョニングされるインフラストラクチャーではなく、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上でのみネットワークが制限された環境でのインストールを実行します。そのため、プラットフォームの選択は制限されます。

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 IAM サービスなどの一部のクラウド機能はインターネットアクセスを必要とするため、インターネットアクセスが依然として必要になる場合があります。ネットワークによっては、ベアメタルハードウェアまたは VMware vSphere へのインストールには、インターネットアクセスが必要になる場合があります。

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 Container Platform 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。



重要

ネットワークが制限されたインストール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常に使用します。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の設定は複雑であるため、ネットワークが制限されたインストールを試行する前に、標準的な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実行することを検討してください。このテストが完了すると、ネットワークが制限されたインストール時に発生する可能性のある問題の切り分けやトラブルシューティングがより容易になります。

1.3.2.1. 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 ImageStreamTag にアクセスできないために使用できません。

1.3.3. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス

OpenShift Container Platform 4.3 では、クラスターをインストールするために必要なイメージを取得するために、インターネットアクセスが必要になります。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 [Red Hat OpenShift Cluster Manager \(OCM\)](#) に登録されます。

Red Hat OpenShift Cluster Manager インベントリーが Telemetry によって自動的に維持されるか、または OCM を手動で使用しているかのいずれによって正常である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。

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。

- [Red Hat OpenShift Cluster Manager](#) ページにアクセスし、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クラスターのインストールおよびインストールプログラムの生成に必要な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。

1.3.4. 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場合のクラスタのマシン要件

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含むクラスタの場合、必要なマシンすべてをデプロイする必要があります。

1.3.4.1. 必要なマシン

最小の OpenShift Container Platform クラスタでは以下のホストが必要です。

- 1つの一時的なブートストラップマシン
- 3つのコントロールプレーン、またはマスター、マシン
- 少なくとも2つのコンピュータマシン (ワーカーマシンとしても知られる)。



注記

クラスタでは、ブートストラップマシンが OpenShift Container Platform クラスタを3つのコントロールプレーンマシンにデプロイする必要があります。クラスタのインストール後にブートストラップマシンを削除できます。



重要

クラスタの高可用性を維持するには、これらのクラスタマシンについて別個の物理ホストを使用します。

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タマシンでは、Red Hat Enterprise Linux CoreOS (RHCOS) をオペレーティングシステムとして使用する必要があります。

RHCOS は Red Hat Enterprise Linux 8 をベースとしており、そのハードウェア認定および要件が継承されることに注意してください。「[Red Hat Enterprise Linux テクノロジーの機能と制限](#)」を参照してください。

1.3.4.2. ネットワーク接続の要件

すべての Red Hat Enterprise Linux CoreOS (RHCOS) マシンでは、起動時に `initramfs` のネットワークがマシン設定サーバーから 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、Ignition 設定ファイルをダウンロードできるようネットワーク接続を確立するために、マシンには DHCP サーバーまたはその静的 IP アドレスが設定されている必要になります。

1.3.4.3. 最小リソース要件

それぞれのクラスタマシンは、以下の最小要件を満たしている必要があります。

マシン	Operating System	vCPU	仮想 RAM	ストレージ
ブートストラップ	RHCOS	4	16 GB	120 GB
コントロールプレーン	RHCOS	4	16 GB	120 GB

マシン	Operating System	vCPU	仮想 RAM	ストレージ
コンピュータ	RHCOS または RHEL 7.6	2	8 GB	120 GB

1.3.4.4. 証明書署名要求の管理

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を使用する場合、クラスターの自動マシン管理へのアクセスは制限されるため、インストール後にクラスターの証明書署名要求 (CSR) のメカニズムを提供する必要があります。**kube-controller-manager** は kubelet クライアント CSR のみを承認します。**machine-approver** は、kubelet 認証情報を使用して要求される提供証明書の有効性を保証できません。適切なマシンがこの要求を発行したかどうかを確認できないためです。kubelet 提供証明書の要求の有効性を検証し、それらを承認する方法を判別し、実装する必要があります。

1.3.5. 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作成

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 OpenShift Container Platform クラスターをデプロイする前に、基礎となるインフラストラクチャーを作成する必要があります。

前提条件

- クラスターでサポートするインフラストラクチャーを作成する前に、「[OpenShift Container Platform 4.x のテスト済みインテグレーション](#)」ページを参照してください。

手順

1. 各ノードに DHCP を設定するか、または静的 IP アドレスを設定します。
2. 必要なロードバランサーをプロビジョニングします。
3. マシンのポートを設定します。
4. DNS を設定します。
5. ネットワーク接続を確認します。

1.3.5.1. 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** のネットワークがマシン設定サーバーから Ignition 設定をフェッチする必要があります。

初回の起動時に、Ignition 設定ファイルをダウンロードできるようネットワーク接続を確立するために、マシンには DHCP サーバーまたはその静的 IP アドレスが設定されている必要があります。

クラスターのマシンを長期間管理するために DHCP サーバーを使用することが推奨されています。DHCP サーバーが永続 IP アドレス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。

クラスターの正常なインストール後に各マスターノードで実行される Kubernetes API サーバーは、クラスターマシンのノード名を解決する必要があります。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することができます。

マシン間のネットワーク接続を、クラスターのコンポーネントが通信できるように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。

表1.12 すべてのマシンに対応するすべてのマシン

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	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 および Geneve
	6081	VXLAN および Geneve
	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。
TCP/UDP	30000-32767	Kubernetes NodePort

表1.13 コントロールプレーンへのすべてのマシン

プロトコル	ポート	説明
TCP	2379-2380	etcd サーバー、ピア、およびメトリクスポート
	6443	Kubernetes API

ネットワークポロジリー要件

クラスター用にプロビジョニングするインフラストラクチャーは、ネットワークポロジリーの以下の要件を満たす必要があります。



重要

OpenShift Container Platform では、すべてのノードが、プラットフォームコンテナのイメージをプルし、Telemetry データを Red Hat に提供するためにインターネットへの直接のアクセスが必要です。

ロードバランサー

OpenShift Container Platform をインストールする前に、以下の要件を満たす2つのロードバランサーをプロビジョニングする必要があります。

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、SSL パススルー、または SSL ブリッジモードと呼ばれます。SSL ブリッジモードを使用する場合は、API ルートの Server Name Indication (SNI) を有効にする必要があります。
 - 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。



注記

API ロードバランサーが適切に機能するには、セッション永続性は必要ありません。

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1.14 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
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が、5 秒または 10 秒ごとにプローブし、2 つの正常な要求が正常な状態になり、3 つの要求が正常な状態になりません。これら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。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、SSL パススルー、または SSL ブリッジモード

- Layer 7 の負荷分散方法。シール、raw TCP、SSL バッファ、または SSL プロキシモードと呼ばれます。SSL ブリッジモードを使用する場合は、Ingress ルートの Server Name Indication (SNI) を有効にする必要があります。
- 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。

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1.15 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 ルーター 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 ルーター 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

ヒント

クライアントの実際の IP アドレスが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。



注記

Ingress ルーターの作業用の設定が OpenShift Container Platform クラスタに必要です。コントロールプレーンの初期化後に Ingress ルーターを設定する必要があります。

1.3.5.2. 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 DNS 要件

以下の DNS レコードは、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する OpenShift Container Platform クラスタに必要です。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クラスターの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1.16 必要な DNS レコード

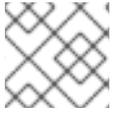
コンポーネント	レコード	説明
---------	------	----

コンポーネント	レコード	説明
Kubernetes API	api.<cluster_name>.<base_domain>.	この DNS A/AAAA または CNAME レコードは、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必要があります。このレコードは、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>.	この DNS A/AAAA または CNAME レコードは、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必要があります。このレコードは、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。これがノード名を解決できない場合、プロキシされる 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。
ルート	*.apps.<cluster_name>.<base_domain>.	Ingress ルーター Pod を実行するマシンをターゲットにする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 レコードです。このレコードは、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。

コンポーネント	レコード	説明
etcd	etcd-<code><index></code>.<code><cluster_name></code>.<code><base_domain></code>.	<p>OpenShift Container Platform では、各 etcd インスタンスの DNS A/AAAA レコードがインスタンスをホストするコントロールプレーンマシンを参照する必要があります。etcd インスタンスは <code><index></code> 値によって差別化されます。この値は 0 で始まり、n-1 で終了します。ここで、n はクラスターのコントロールプレーンマシンの数です。DNS レコードはコントロールプレーンマシンのユニキャスト IPv4 アドレスに解決し、レコードはクラスター内のすべてのノードで解決可能である必要があります。</p>
	<code>_etcd-server-ssl._tcp.<cluster_name></code>. <code><base_domain></code>.	<p>それぞれのコントロールプレーンマシンについて、OpenShift Container Platform では、そのマシンに優先度 0、重み 10 およびポート 2380 の etcd サーバーの SRV DNS レコードも必要になります。3つのコントロールプレーンマシンを使用するクラスターには以下のレコードが必要です。</p> <pre># _service._proto.name. TTL class SRV priority weight port target. _etcd-server-ssl._tcp. <cluster_name>. <base_domain>. 86400 IN SRV 0 10 2380 etcd- 0.<cluster_name>. <base_domain> _etcd-server-ssl._tcp. <cluster_name>. <base_domain>. 86400 IN SRV 0 10 2380 etcd- 1.<cluster_name>. <base_domain> _etcd-server-ssl._tcp. <cluster_name>. <base_domain>. 86400 IN SRV 0 10 2380 etcd- 2.<cluster_name>. <base_domain></pre>

1.3.6. SSH プライベートキーの生成およびエージェントへの追加

クラスターで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がある場合、**ssh-agent** とインストールプログラムの両方に SSH キーを指定する必要があります。



注記

実稼働環境では、障害復旧およびデバッグが必要です。

このキーを使用して、ユーザー **core** としてマスターノードに対して SSH を実行できます。クラスターをデプロイする際に、キーは **core** ユーザーの `~/.ssh/authorized_keys` 一覧に追加されます。



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. パスワードなしの認証に設定されている SSH キーがコンピューター上にない場合は、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 rsa -b 4096 -N "" \
-f <path>/<file_name> ①
```

- ① `~/.ssh/id_rsa` などの、SSH キーのパスおよびファイル名を指定します。既存の SSH キーは上書きされるため、指定しないでください。

このコマンドを実行すると、指定した場所にパスワードを必要としない SSH キーが生成されます。

2. **ssh-agent** プロセスをバックグラウンドタスクとして開始します。

```
$ eval "$(ssh-agent -s)"
Agent pid 31874
```

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

```
$ ssh-add <path>/<file_name> ①
Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

- ① `~/.ssh/id_rsa` 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。

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。クラスターを独自に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場合は、このキーをクラスターのマシンに指定する必要があります。

1.3.7. インストール設定ファイルの手動作成

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 OpenShift Container Platform のインストールでは、インストール設定ファイルを手動で生成する必要があります。

前提条件

- OpenShift Container Platform インストーラープログラムおよびクラスターのアクセストークンを取得します。
- リポジトリのミラーリングに使用するコマンドの出力で **imageContentSources** セクションを取得します。
- ミラーレジストリーの証明書の内容を取得します。

手順

1. 必要なインストールアセットを保存するためのインストールディレクトリーを作成します。

```
$ mkdir <installation_directory>
```



重要

ディレクトリーを作成する必要があります。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の有効期限は短く設定されているため、インストールディレクトリーを再利用することができません。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. 以下の **install-config.yaml** ファイルテンプレートをカスタマイズし、これを **<installation_directory>** に保存します。



注記

この設定ファイル **install-config.yaml** に名前を付ける必要があります。

- **docker.io** などの、RHCOS がデフォルトで信頼するレジストリーを使用しない限り、**additionalTrustBundle** セクションにミラーリポジトリの証明書の内容を指定する必要があります。ほとんどの場合、ミラーの証明書を指定する必要があります。
 - リポジトリのミラーリングに使用するコマンドの出力の **imageContentSources** セクションを組み込む必要があります。
3. **install-config.yaml** ファイルをバックアップし、これを複数のクラスターをインストールするため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の次の手順で使用されます。この時点でこれをバックアップする必要があります。

チスレッドを無効にする場合は、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。



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。

- 4 **replicas** パラメーターの値を **0** に設定する必要があります。このパラメーターはクラスターが作成し、管理するワーカーの数を制御します。これは、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する場合にクラスターが実行しない機能です。OpenShift Container Platform のインストールが終了する前に、クラスターが使用するワーカーマシンを手動でデプロイする必要があります。
- 8 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。
- 9 DNS レコードに指定したクラスター名。
- 10 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。
- 11 それぞれの個別ノードに割り当てるサブネットプレフィックスの長さ。たとえば、**hostPrefix** が **23** に設定され、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 (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。
- 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。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。ベアメタル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。
- 15 **<local_registry>** については、レジストリドメイン名と、ミラーレジストリーがコンテンツを提供するために使用するポートをオプションで指定します。例: **registry.example.com** または **registry.example.com:5000<credentials>** について、ミラーレジストリーの base64 でエンコードされたユーザー名およびパスワードを指定します。
- 16 Red Hat Enterprise Linux CoreOS (RHCOS) の **core** ユーザーのデフォルト SSH キーの公開部分。



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がある実稼働用の 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。

- 17 ミラーレジストリーに使用した証明書ファイルの内容を指定します。
- 18 リポジトリーのミラーリングに使用するコマンドの出力の **imageContentSources** セクションを指定します。

1.3.7.2. インストール時のクラスター全体のプロキシの設定

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 プロキシを使用することができます。プロキシ設定を **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を使用するように設定できます。

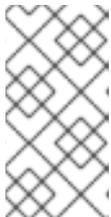


注記

ベアメタルインストールでは、**install-config.yaml** ファイルの **networking.machineCIDR** フィールドで指定される範囲にある IP アドレスをノードに割り当てない場合、それらを **proxy.noProxy** フィールドに含める必要があります。

前提条件

- 既存の **install-config.yaml** ファイルが必要です。
- クラスターがアクセスする必要があるサイトを確認し、プロキシをバイパスする必要があるかどうかを判別します。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ついてのクラウドプロバイダー API に対する呼び出しを含む) はプロキシされます。プロキシオブジェクトの **spec.noProxy** フィールドにサイトを追加し、必要に応じてプロキシをバイパスします。



注記

プロキシオブジェクトの **status.noProxy** フィールドは、デフォルトでインスタンスメタデータエンドポイント (**169.254.169.254**) およびインストール設定の **networking.machineCIDR**、**networking.clusterNetwork.cidr**、および **networking.serviceNetwork[]** フィールドの値で設定されます。

手順

1. **install-config.yaml** ファイルを編集し、プロキシ設定を追加します。以下は例になります。

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> 1
  httpsProxy: http://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
  additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
  ...
```

- 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 URL。URL スキームは **http** である必要があります。追加のプロキシ設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシネットワークを使用する場合には、**httpProxy** 値を指定

することはできません。

- 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 URL。このフィールドが指定されていない場合、HTTP および HTTPS 接続の両方に **httpProxy** が使用されます。追加のプロキシ設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシネットワークを使用する場合には、**httpsProxy** 値を指定することはできません。
- 3 プロキシを除外するための宛先ドメイン名、ドメイン、IP アドレス、または他のネットワーク CIDR のカンマ区切りの一覧。ドメインのすべてのサブドメインを組み込むために、ドメインの前に . を入力します。* を使用し、すべての宛先のプロキシをバイパスします。
- 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 ConfigMap 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** ConfigMap を作成し、この ConfigMap はプロキシオブジェクトの **trustedCA** フィールドで参照されます。**additionalTrustBundle** フィールドは、プロキシのアイデンティティ証明書が 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。追加のプロキシ設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシネットワークを使用する場合には、MITM CA 証明書を指定する必要があります。



注記

インストールプログラムは、プロキシの **readinessEndpoints** フィールドをサポートしません。

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を作成します。プロキシ設定が指定されていない場合、**cluster** のプロキシ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 という名前のプロキシオブジェクトのみがサポートされ、追加のプロキシを作成することはできません。

1.3.8. Kubernetes マニフェストおよび Ignition 設定ファイルの作成

一部のクラスター定義ファイルを変更し、クラスターマシンを手動で起動する必要があるため、クラスターがマシンを作成するために必要な Kubernetes マニフェストと Ignition 設定ファイルを生成する必要があります。



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る証明書が含まれます。最初の証明書のローテーションが正常に実行されるようにするには、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。

前提条件

- OpenShift Container Platform インストールプログラムを取得します。ネットワークが制限されたインストールでは、これらのファイルがミラーホスト上に置かれます。
- **install-config.yaml** インストール設定ファイルを作成します。

手順

1. クラスターの Kubernetes マニフェストを生成します。

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

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

- 1** **<installation_directory>** については、作成した **install-config.yaml** ファイルが含まれるインストールディレクトリーを指定します。

インストールプロセスの後の部分で独自のコンピュータマシンを作成するため、この警告を無視しても問題がありません。

2. **<installation_directory>/manifests/cluster-scheduler-02-config.yml** Kubernetes マニフェストファイルを変更し、Pod がコントロールプレーンマシンにスケジュールされないようにします。
 - a. **<installation_directory>/manifests/cluster-scheduler-02-config.yml** ファイルを開きま
す。
 - b. **mastersSchedulable** パラメーターを見つけ、その値を **False** に設定します。
 - c. ファイルを保存し、終了します。



注記

現時点では、[Kubernetes の制限](#) により、コントロールプレーンマシンで実行されるルーター Pod に Ingress ロードバランサーがアクセスすることができません。この手順は、OpenShift Container Platform の今後のマイナーバージョンで不要になる可能性があります。

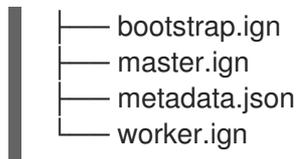
3. Ignition 設定ファイルを取得します。

```
$ ./openshift-install create ignition-configs --dir=<installation_directory> 1
```

- 1** **<installation_directory>** については、同じインストールディレクトリーを指定します。

以下のファイルはディレクトリーに生成されます。

```
.
├── auth
│   ├── kubeadmin-password
│   └── kubeconfig
```



1.3.9. Red Hat Enterprise Linux CoreOS (RHCOS) マシンの作成

ユーザーによってプロビジョニングされるベアメタル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前に、それが使用する RHCOS マシンを作成する必要があります。ISO イメージまたはネットワーク PXE ブートを使用する手順を実行してマシンを作成することができます。

1.3.9.1. ISO イメージを使用した Red Hat Enterprise Linux CoreOS (RHCOS) マシンの作成

ユーザーによってプロビジョニングされるベアメタル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前に、それが使用する RHCOS マシンを作成する必要があります。ISO イメージを使用してマシンを作成することができます。

前提条件

- クラスターの Ignition 設定ファイルを取得していること。
- お使いのコンピューターからアクセスでき、作成するマシンがアクセスできる HTTP サーバーへのアクセスがあること。

手順

1. インストールプログラムが作成したコントロールプレーン、コンピュート、およびブートストラップ Ignition 設定を HTTP サーバーにアップロードします。これらのファイルの URL をメモします。



重要

インストールの完了後にコンピュートマシンをさらにクラスターに追加する予定の場合には、これらのファイルを削除しないでください。

2. Red Hat カスタマーポータル「[製品のダウンロード](#)」ページまたは「[RRHCOS イメージミラー](#)」ページからオペレーティングシステムのインスタンスをインストールするために優先される方法で必要な RHCOS イメージを取得します。



重要

RHCOS イメージは OpenShift Container Platform の各リリースごとに変更されない可能性があります。インストールする OpenShift Container Platform バージョンと等しいか、それ以下のバージョンの中で最も新しいバージョンのイメージをダウンロードする必要があります。利用可能な場合は、OpenShift Container Platform バージョンに一致するイメージのバージョンを使用します。

ISO ファイルおよび RAW ディスクファイルをダウンロードする必要があります。これらのファイルの名前は以下の例のようになります。

- ISO: `rhcos-<version>-installer.<architecture>.iso`
- 圧縮された metal RAW: `rhcos-<version>-metal.<architecture>.raw.gz`

- RAW RHCOS イメージファイルのいずれかを HTTP サーバーにアップロードし、その URL をメモします。



重要

インストールの完了後にコンピュータマシンをさらにクラスターに追加する予定の場合には、これらのファイルを削除しないでください。

- ISO を使用し、RHCOS インストールを開始します。以下のインストールオプションのいずれかを使用します。
 - ディスクに ISO イメージを書き込み、これを直接起動します。
 - LOM インターフェースで ISO リダイレクトを使用します。
- インスタンスの起動後に、**TAB** または **E** キーを押してカーネルコマンドラインを編集します。
- パラメーターをカーネルコマンドラインに追加します。

```
coreos.inst=yes
coreos.inst.install_dev=sda ①
coreos.inst.image_url=<image_URL> ②
coreos.inst.ignition_url=http://example.com/config.ign ③
ip=<dhcp or static IP address> ④ ⑤
```

- ① インストール先のシステムのブロックデバイスを指定します。
- ② サーバーにアップロードした RAW イメージの URL を指定します。
- ③ このマシンタイプの Ignition 設定ファイルの URL を指定します。
- ④ **ip=dhcp** を設定するか、各ノードに個別の静的 IP アドレス (**ip=**) および DNS サーバー (**nameserver=**) を設定します。たとえば、**ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp1s0:none nameserver=4.4.4.41** を設定すると、以下のように設定されます。
 - ノードの IP アドレス: **10.10.10.2**
 - ゲートウェイアドレス: **10.10.10.254**
 - ネットマスク: **255.255.255.0**
 - ホスト名: **core0.example.com**
 - DNS サーバーアドレス: **4.4.4.41**
- ⑤ 複数のネットワークインターフェースまたは DNS サーバーを使用する場合は、以下の方法で IP アドレス設定をさらにカスタマイズできます。
 - システムに複数のネットワークインターフェースがある場合には、複数の **ip=** エントリーを指定して、複数の IP アドレス設定を追加できます。
 - システムに複数のネットワークインターフェースがある場合には、DHCP と静的 IP の設定の組み合わせを使用できます。使用するインターフェースを指定する必要があります (例: **ip=enp1s0:dhcp ip=10.10.10.2::255.255.255.0:core0.example.com:enp2s0:none**) 。

- 各サーバーに `nameserver=` エントリーを追加して、複数の DNS サーバーを指定できます (例: `nameserver=1.1.1.1 nameserver=8.8.8.8`)。
7. Enter を押してインストールを完了します。RHCOS のインストール後に、システムは再起動します。システムの再起動後、指定した Ignition 設定ファイルを適用します。
 8. 継続してクラスターのマシンを作成します。



重要

この時点でブートストラップおよびコントロールプレーンマシンを作成する必要があります。一部の Pod はデフォルトでコンピュートマシンにデプロイされるため、クラスターのインストール前に、2 つ以上のコンピュートマシンを作成します。

1.3.9.2. PXE または iPXE ブートによる Red Hat Enterprise Linux CoreOS (RHCOS) マシンの作成

ユーザーによってプロビジョニングされるベアメタル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前に、それが使用する RHCOS マシンを作成する必要があります。PXE または iPXE ブートを使用してマシンを作成することができます。

前提条件

- クラスターの Ignition 設定ファイルを取得していること。
- 適切な PXE または iPXE インフラストラクチャーを設定していること。
- 使用しているコンピューターからアクセス可能な HTTP サーバーへのアクセスがあること。

手順

1. インストールプログラムが作成したマスター、ワーカーおよびブートストラップの Ignition 設定を HTTP サーバーにアップロードします。これらのファイルの URL をメモします。



重要

インストールの完了後にコンピュートマシンをさらにクラスターに追加する予定の場合には、これらのファイルを削除しないでください。

2. Red Hat カスタマーポータルで「[製品のダウンロード](#)」ページまたは「[RHCOS イメージミラー](#)」ページから圧縮された metal RAW イメージ、`kernel` および `initramfs` ファイルを取得します。



重要

RHCOS イメージは OpenShift Container Platform の各リリースごとに変更されない可能性があります。インストールする OpenShift Container Platform バージョンと等しいか、それ以下のバージョンの中で最も新しいバージョンのイメージをダウンロードする必要があります。利用可能な場合は、OpenShift Container Platform バージョンに一致するイメージのバージョンを使用します。この手順には RAW イメージのみを使用します。RHCOS qcow2 イメージは、ベアメタルのインストールではサポートされません。

ファイル名には、OpenShift Container Platform のバージョン名が含まれます。以下の例のようになります。

- 圧縮されたメタル RAW イメージ: **rhcos-<version>metal.<architecture>.raw.gz**
 - **kernel: rhcos-<version>-installer-kernel-<architecture>**
 - **initramfs: rhcos-<version>-installer-initramfs.<architecture>.img**
3. 圧縮された metal RAW イメージおよび **kernel** および **initramfs** ファイルを HTTP サーバーにアップロードします。



重要

インストールの完了後にコンピュータマシンをさらにクラスターに追加する予定の場合には、これらのファイルを削除しないでください。

4. RHCOS のインストール後にマシンがローカルディスクから起動されるようにネットワークブートインフラストラクチャーを設定します。
5. RHCOS イメージに PXE または iPXE インストールを設定します。
ご使用の環境についての以下の例で示されるメニューエントリーのいずれかを変更し、イメージおよび Ignition ファイルが適切にアクセスできることを確認します。

- PXE の場合:

```

DEFAULT pxeboot
TIMEOUT 20
PROMPT 0
LABEL pxeboot
  KERNEL http://<HTTP_server>/rhcos-<version>-installer-kernel-<architecture> ❶
  APPEND ip=dhcp rd.neednet=1 initrd=http://<HTTP_server>/rhcos-<version>-installer-
initramfs.<architecture>.img console=tty0 console=ttyS0 coreos.inst=yes
coreos.inst.install_dev=sda coreos.inst.image_url=http://<HTTP_server>/rhcos-
<version>-metal.<architecture>.raw.gz
coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign ❷ ❸

```

- ❶ HTTP サーバーにアップロードした **kernel** ファイルの場所を指定します。
- ❷ 複数の NIC を使用する場合、**ip** オプションに単一インターフェースを指定します。たとえば、**eno1** という名前の NIC で DHCP を使用するには、**ip=eno1:dhcp** を設定します。
- ❸ HTTP サーバーにアップロードした RHCOS ファイルの場所を指定します。**initrd** パラメーター値は **initramfs** ファイルの場所であり、**coreos.inst.image_url** パラメーター値は圧縮された metal RAW イメージの場所、および **coreos.inst.ignition_url** パラメーター値はブートストラップ Ignition 設定ファイルの場所になります。

- iPXE の場合:

```

kernel http://<HTTP_server>/rhcos-<version>-installer-kernel-<architecture> ip=dhcp
rd.neednet=1 initrd=http://<HTTP_server>/rhcos-<version>-installer-initramfs.
<architecture>.img console=tty0 console=ttyS0 coreos.inst=yes
coreos.inst.install_dev=sda coreos.inst.image_url=http://<HTTP_server>/rhcos-

```

```
<version>-metal.<architecture>.raw.gz
coreos.inst.ignition_url=http://<HTTP_server>/bootstrap.ign ❶ ❷
initrd http://<HTTP_server>/rhcos-<version>-installer-initramfs.<architecture>.img ❸
boot
```

- ❶ HTTP サーバーにアップロードした RHCOS ファイルの場所を指定します。 **kernel** パラメーター値は **kernel** ファイルの場所であり、 **initrd** パラメーター値は **initramfs** ファイルの場所、 **coreos.inst.image_url** パラメーター値は圧縮された metal RAW イメージの場所、 および **coreos.inst.ignition_url** パラメーター値はブートストラップ Ignition 設定ファイルの場所になります。
- ❷ 複数の NIC を使用する場合、 **ip** オプションに単一インターフェースを指定します。たとえば、 **eno1** という名前の NIC で DHCP を使用するには、 **ip=eno1:dhcp** を設定します。
- ❸ HTTP サーバーにアップロードした **initramfs** ファイルの場所を指定します。

6. 継続してクラスターのマシンを作成します。



重要

この時点でブートストラップおよびコントロールプレーンマシンを作成する必要があります。一部の Pod はデフォルトでコンピュータマシンにデプロイされるため、クラスターのインストール前に、2つ以上のコンピュータマシンを作成します。

1.3.10. クラスターの作成

OpenShift Container Platform クラスターを作成するには、ブートストラッププロセスが、インストールプログラムで生成した Ignition 設定ファイルを使用してプロビジョニングしたマシンで完了するのを待機します。

前提条件

- クラスターに必要なインフラストラクチャーを作成します。
- インストールプログラムを取得し、クラスターの Ignition 設定ファイルを生成済みです。
- Ignition 設定ファイルを使用して、クラスターの RHCOS マシンを作成済みです。

手順

1. ブートストラッププロセスをモニターします。

```
$ ./openshift-install --dir=<installation_directory> wait-for bootstrap-complete \ ❶
--log-level=info ❷
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com...
INFO API v1.16.2 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
```

- ❶ **<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。

Kubernetes API サーバーでこれがコントロールプレーンマシンにブートストラップされていることを示すシグナルが出されるとコマンドは成功します。

- ブートストラッププロセスが完了したら、ブートストラップマシンをロードバランサーから削除します。



重要

この時点で、ブートストラップマシンをロードバランサーから削除する必要があります。さらに、マシン自体を削除し、再フォーマットすることができます。

1.3.11. 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ついての情報が含まれます。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ます。
- oc** CLI をインストールします。

手順

- 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。

- 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
system:admin
```

1.3.12. マシンの CSR の承認

マシンをクラスターに追加する際に、追加したそれぞれのマシンについて 2 つの保留状態の証明書署名要求 (CSR) が生成されます。これらの CSR が承認されていることを確認するか、または必要な場合はそれらを承認してください。

前提条件

- マシンがクラスターに追加されています。

手順

1. クラスタがマシンを認識していることを確認します。

```
$ oc get nodes

NAME      STATUS    ROLES    AGE   VERSION
master-0  Ready     master   63m   v1.16.2
master-1  Ready     master   63m   v1.16.2
master-2  Ready     master   64m   v1.16.2
worker-0  NotReady  worker   76s   v1.16.2
worker-1  NotReady  worker   70s   v1.16.2
```

出力には作成したすべてのマシンが一覧表示されます。

2. 保留中の証明書署名要求 (CSR) を確認し、クラスタに追加したそれぞれのマシンのクライアントおよびサーバー要求に **Pending** または **Approved** ステータスが表示されていることを確認します。

```
$ oc get csr

NAME      AGE   REQUESTOR                                     CONDITION
csr-8b2br 15m   system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper Pending 1
csr-8vnps 15m   system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper Pending
csr-bfd72 5m26s system:node:ip-10-0-50-126.us-east-2.compute.internal
Pending 2
csr-c57lv 5m26s system:node:ip-10-0-95-157.us-east-2.compute.internal
Pending
...
```

1 クライアント要求の CSR。

2 サーバー要求の CSR。

この例では、2つのマシンがクラスタに参加しています。この一覧にはさらに多くの承認された CSR が表示される可能性があります。

3. 追加したマシンの保留中の CSR すべてが **Pending** ステータスになった後に CSR が承認されない場合には、クラスタマシンの CSR を承認します。



注記

CSR のローテーションは自動的に実行されるため、クラスタにマシンを追加後1時間以内に CSR を承認してください。1時間以内に承認しない場合には、証明書のローテーションが行われ、各ノードに3つ以上の証明書が存在するようになります。これらの証明書すべてを承認する必要があります。最初の CSR の承認後、後続のノードクライアント CSR はクラスタの **kube-controller-manger** によって自動的に承認されます。kubelet 提供証明書の要求を自動的に承認する方法を実装する必要があります。

- それらを個別に承認するには、それぞれの有効な CSR について以下のコマンドを実行します。

```
$ oc adm certificate approve <csr_name> 1
```

-

① `<csr_name>` は、現行の CSR の一覧からの CSR の名前です。

- すべての保留中の CSR を承認するには、以下のコマンドを実行します。

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{"\n"}
{{end}}{{end}}' | xargs oc adm certificate approve
```

追加情報

- CSR の詳細は、「[Certificate Signing Requests](#)」を参照してください。

1.3.13. Operator の初期設定

コントロールプレーンの初期化後に、一部の Operator を利用可能にするためにそれらをすぐに設定する必要があります。

前提条件

- コントロールプレーンが初期化されています。

手順

1. クラスターコンポーネントがオンラインになることを確認します。

```
$ watch -n5 oc get clusteroperators
```

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
authentication	4.3.0	True	False	False	69s
cloud-credential	4.3.0	True	False	False	12m
cluster-autoscaler	4.3.0	True	False	False	11m
console	4.3.0	True	False	False	46s
dns	4.3.0	True	False	False	11m
image-registry	4.3.0	True	False	False	5m26s
ingress	4.3.0	True	False	False	5m36s
kube-apiserver	4.3.0	True	False	False	8m53s
kube-controller-manager	4.3.0	True	False	False	7m24s
kube-scheduler	4.3.0	True	False	False	12m
machine-api	4.3.0	True	False	False	12m
machine-config	4.3.0	True	False	False	7m36s
marketplace	4.3.0	True	False	False	7m54m
monitoring	4.3.0	True	False	False	7h54s
network	4.3.0	True	False	False	5m9s
node-tuning	4.3.0	True	False	False	11m
openshift-apiserver	4.3.0	True	False	False	11m
openshift-controller-manager	4.3.0	True	False	False	5m943s
openshift-samples	4.3.0	True	False	False	3m55s
operator-lifecycle-manager	4.3.0	True	False	False	11m
operator-lifecycle-manager-catalog	4.3.0	True	False	False	11m
service-ca	4.3.0	True	False	False	11m
service-catalog-apiserver	4.3.0	True	False	False	5m26s
service-catalog-controller-manager	4.3.0	True	False	False	5m25s
storage	4.3.0	True	False	False	5m30s

-
- 2. 利用不可の Operator を設定します。

1.3.13.1. イメージレジストリーストレージの設定

Image-registry Operator は、デフォルトストレージを提供しないプラットフォームでは最初は利用できません。インストール後に、レジストリー Operator を使用できるようにレジストリーをストレージを使用するように設定する必要があります。

実稼働クラスターに必要な PersistentVolume の設定方法と、実稼働用ではないクラスターにのみ使用できる空のディレクトリーをストレージの場所として設定する方法が表示されます。

1.3.13.1.1. ベアメタルの場合のレジストリーストレージの設定

クラスター管理者は、インストール後にレジストリーをストレージを使用できるように設定する必要があります。

前提条件

- クラスター管理者のパーミッション。
- ベアメタル上のクラスター。
- Red Hat OpenShift Container Storage などのクラスターの永続ストレージをプロビジョニングします。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。
- 容量は「100Gi」以上である。

手順

1. レジストリーをストレージを使用できるように設定するには、**configs.imageregistry/cluster** リソースの **spec.storage.pvc** を変更します。



注記

NFS などの共有ストレージを使用する場合は、**fsGroup** ID ではなく、セキュリティーコンテキストの許可される補助グループを定める **supplementalGroups** ストラテジーを使用することが強く推奨されます。詳細は、NFS の **グループ ID** についてのドキュメントを参照してください。

2. レジストリー Pod がないことを確認します。

```
$ oc get pod -n openshift-image-registry
```



注記

- ストレージタイプが **emptyDIR** の場合、レプリカ数が **1** を超えることはありません。
- ストレージタイプが **NFS** の場合、**no_wdelay** および **root_squash** マウントオプションを有効にする必要があります。以下は例になります。

```
# cat /etc/exports
/mnt/data *(rw,sync,no_wdelay,root_squash,insecure,fsid=0)
sh-4.3# exportfs -rv
exporting */mnt/data
```

3. レジストリー設定を確認します。

```
$ oc edit configs.imageregistry.operator.openshift.io

storage:
pvc:
claim:
```

claim フィールドを空のままにし、**image-registry-storage** PVC の自動作成を可能にします。

4. **clusteroperator** ステータスを確認します。

```
$ oc get clusteroperator image-registry
```

1.3.13.1.2. 実稼働以外のクラスターでのイメージレジストリーのストレージの設定

イメージレジストリー Operator のストレージを設定する必要があります。実稼働用以外のクラスターの場合、イメージレジストリーは空のディレクトリーに設定することができます。これを実行する場合、レジストリーを再起動するとすべてのイメージが失われます。

手順

- イメージレジストリーストレージを空のディレクトリーに設定するには、以下を実行します。

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec": {"storage":{"emptyDir":{}}}'
```



警告

実稼働用以外のクラスターにのみこのオプションを設定します。

イメージレジストリー Operator がそのコンポーネントを初期化する前にこのコマンドを実行する場合、**oc patch** コマンドは以下のエラーを出して失敗します。

```
Error from server (NotFound): configs.imageregistry.operator.openshift.io "cluster" not found
```

数分待機した後に、このコマンドを再び実行します。

1.3.14. 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の完了

Operator 設定の完了後に、提供するインフラストラクチャーでのクラスターのインストールを終了できます。

前提条件

- コントロールプレーンが初期化されています。
- Operator の初期設定を完了済みです。

手順

1. すべてのクラスターコンポーネントがオンラインであることを確認します。

```
$ watch -n5 oc get clusteroperators
```

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
authentication	4.3.0	True	False	False	10m
cloud-credential	4.3.0	True	False	False	22m
cluster-autoscaler	4.3.0	True	False	False	21m
console	4.3.0	True	False	False	10m
dns	4.3.0	True	False	False	21m
image-registry	4.3.0	True	False	False	16m
ingress	4.3.0	True	False	False	16m
kube-apiserver	4.3.0	True	False	False	19m
kube-controller-manager	4.3.0	True	False	False	18m
kube-scheduler	4.3.0	True	False	False	22m
machine-api	4.3.0	True	False	False	22m
machine-config	4.3.0	True	False	False	18m
marketplace	4.3.0	True	False	False	18m
monitoring	4.3.0	True	False	False	18m
network	4.3.0	True	False	False	16m
node-tuning	4.3.0	True	False	False	21m
openshift-apiserver	4.3.0	True	False	False	21m
openshift-controller-manager	4.3.0	True	False	False	17m
openshift-samples	4.3.0	True	False	False	14m
operator-lifecycle-manager	4.3.0	True	False	False	21m
operator-lifecycle-manager-catalog	4.3.0	True	False	False	21m
service-ca	4.3.0	True	False	False	21m
service-catalog-apiserver	4.3.0	True	False	False	16m
service-catalog-controller-manager	4.3.0	True	False	False	16m
storage	4.3.0	True	False	False	16m

すべてのクラスター Operator が **AVAILABLE** の場合、インストールを完了することができます。

2. クラスターの完了をモニターします。

```
$ ./openshift-install --dir=<installation_directory> wait-for install-complete 1
INFO Waiting up to 30m0s for the cluster to initialize...
```

-

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

Cluster Version Operator が Kubernetes API サーバーから OpenShift Container Platform クラスターのデプロイを終了するとコマンドは成功します。



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る証明書が含まれます。最初の証明書のローテーションが正常に実行されるようにするには、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。

3. Kubernetes API サーバーが Pod と通信していることを確認します。
 - a. すべての Pod の一覧を表示するには、以下のコマンドを使用します。

```
$ oc get pods --all-namespaces
```

NAMESPACE	NAME	READY	STATUS
RESTARTS	AGE		
openshift-apiserver-operator	openshift-apiserver-operator-85cb746d55-zqhs8	1/1	
Running	1 9m		
openshift-apiserver	apiserver-67b9g	1/1	Running 0
3m			
openshift-apiserver	apiserver-ljcmx	1/1	Running 0
1m			
openshift-apiserver	apiserver-z25h4	1/1	Running 0
2m			
openshift-authentication-operator	authentication-operator-69d5d8bf84-vh2n8	1/1	
Running	0 5m		
...			

- b. 以下のコマンドを使用して、直前のコマンドの出力に一覧表示される Pod のログを表示します。

```
$ oc logs <pod_name> -n <namespace> 1
```

- 1 直前のコマンドの出力にあるように、Pod 名および namespace を指定します。

Pod のログが表示される場合、Kubernetes API サーバーはクラスターマシンと通信できません。

4. 「[Cluster registration](#)」 ページでクラスターを登録します。

1.3.15. 次のステップ

- [クラスターをカスタマイズ](#) します。
- 必要な場合は、[リモートの健全性レポートをオプトアウト](#) することができます。

