OpenShift Container Platform 4.2

vSphere へのインストール

OpenShift Container Platform 4.2 vSphere クラスターのインストール

Last Updated: 2020-06-17

OpenShift Container Platform 4.2 vSphere へのインストール

OpenShift Container Platform 4.2 vSphere クラスターのインストール

法律上の通知

概要

本書では、VMware vSphere に OpenShift Container Platform 4.2 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 VSPHERE へのインストール

1.1. クラスターの VSPHERE へのインストール

OpenShift Container Platform バージョン 4.2 では、プロビジョニングする VMware vSphere インフラストラクチャーにクラスターをインストールできます。

前提条件

- クラスターの永続ストレージ 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

プロキシーを設定する場合は、このサイト一覧も確認してください。

1.1.1. 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 4.2 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。

1.1.2. VMware vSphere インフラストラクチャーの要件

OpenShift Container Platform クラスターを、VMware vSphere のバージョン 6.5 または 6.7U2 以降のインスタンスにインストールする必要があります。

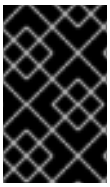
VMware では、vSphere バージョン 6.7 U2 以降を OpenShift Container Platform クラスターで使用することを推奨しています。 vSphere 6.7U2 には以下が含まれます。

- VMware NSX-T のサポート
- インツリー (in-tree) VCP を使用する vSAN、VMFS および NFS のサポート

vSphere 6.5 とハードウェアのバージョン 13 の使用がサポートされていますが、OpenShift Container Platform クラスターは以下の制限を受けます。

- NSX-T SDN はサポートされません。
- OpenShift Container Platform がサポートする別の SDN またはストレージプロバイダーを使用する必要があります。

vSphere バージョン 6.5 インスタンスを使用している場合は、OpenShift Container Platform をインストールする前に 6.7U2 にアップグレードすることを検討してください。



重要

OpenShift Container Platform をインストールする前に、ESXi ホストの時間が同期されていることを確認する必要があります。VMware ドキュメントの「Edit Time Configuration for a Host」を参照してください。

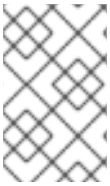
1.1.3. ユーザーによってプロビジョニングされるインフラストラクチャーを使用する場合のクラスターのマシン要件

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 つのコンピュートマシン (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トマシンでは、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 サーバーが必要になります。

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
コンピュート	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 Tested Integrations」ページを参照してください。

手順

1. DHCP 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 サーバーが必要になります。

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

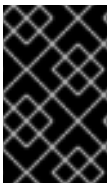
プロトコル	ポート	説明
ICMP	該当なし	ネットワーク到達性のテスト
TCP	**9000**-**9999**	ホストレベルのサービス。 ポート **9100**-**9101** のノードエクスポーター、ポート **9099** の Cluster Version Operator が含まれます。
	10250-**10259**	Kubernetes が予約するデフォルトポート

プロトコル	ポート	説明
	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

ネットワークトポロジー要件
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、ネットワークトポロジ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ー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、2 つの Layer 4 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。API には 1 つのロードバランサーが必要で、デフォルトの Ingress コントローラーには、Ingress をアプリケーションに提供する 2 番目のロードバランサーが必要です。

ポート	マシン	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。	x	x	Kubernetes API サーバー

ポート	マシン	内部	外部	説明
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
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

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。

Ethernet アダプターのハードウェアアドレス要件
クラスターの仮想マシンをプロビジョニングする場合、各仮想マシンに設定されたイーサネットインターフェースは VMware Organizationally Unique Identifier (OUI) 割り当て範囲から MAC アドレスを使用する必要があります。

- **00:05:69:00:00:00** - **00:05:69:FF:FF:FF**
- **00:0c:29:00:00:00** - **00:0c:29:FF:FF:FF**
- **00:1c:14:00:00:00** - **00:1c:14:FF:FF:FF**
- **00:50:56:00:00:00** - **00:50:56:FF:FF:FF**

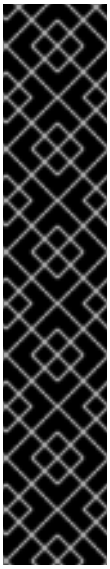
VMware OUI 外の MAC アドレスが使用される場合、クラスターのインストールは成功しません。

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1.3 必要な DNS レコード

Component	レコード	説明

Component	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、プロキシーされる API 呼び出しが失敗し、Pod からログを取得できなくなる可能性があります。
Routes	***.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。

Component	レコード	説明
etcd	**etcd-<index>.<cluster_name>.<base_domain>.**	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 **<index>** 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。
	_etcd-server-ssl._tcp.<cluster_name>.<base_domain>.	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。 `# _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>`

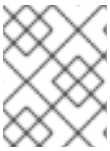
1.1.5. SSH プライベートキーの生成およびエージェントへの追加

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。

   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。

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ーイメージを提供する、Quay.io などの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. 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

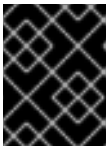
ディレクトリーを作成する必要があります。ブートストラップ X.509 証明書などの一部のインストールアセットの有効期限は短く設定されているため、インストールディレクトリーを再利用することができません。別のクラスターインストールの個別のファイルを再利用する必要がある場合は、それらをディレクトリーにコピーすることができます。ただし、インストールアセットのファイル名はリリース間で変更される可能性があります。インストールファイルを以前のバージョンの OpenShift Container Platform からコピーする場合は注意してコピーを行ってください。

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。

1.1.7.1. VMware vSphere 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
  replicas: 3 (7)
metadata:
  name: test (8)
platform:
  vsphere:
    vcenter: your.vcenter.server (9)
    username: username (10)
    password: password (11)
    datacenter: datacenter (12)
```

```
    defaultDatastore: datastore 13
pullSecret: '{"auths": ...}' 14
sshKey: 'ssh-ed25519 AAAA...' 15
```

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 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。同時マルチスレッドを無効にする場合は、マシンで最低でも 8 CPU および 32 GB の RAM を使用する必要があります。

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。

7 クラスターに追加するコントロールプレーンマシンの数。クラスターをこの値をクラスターの etcd エンドポイント数として使用するため、値はデプロイするコントロールプレーンマシンの数に一致する必要があります。

8 DNS レコードに指定したクラスター名。

9 vCenter サーバーの完全修飾ホスト名または IP アドレス。

10 サーバーにアクセスするユーザーの名前。このユーザーには、少なくとも vSphere の静的または動的な永続ボリュームのプロビジョニングに必要なロールおよび権限がなければなりません。

11 vSphere ユーザーに関連付けられたパスワード。

12 vSphere データセンター。

13 使用するデフォルトの vSphere データストア。

14 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ーイメージを提供する、Quay.io などの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。

15 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

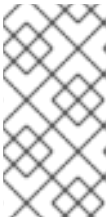
インストールのデバッグまたは障害復旧を実行する必要のある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

1.1.7.2. インストール時のクラスター全体のプロキシーの設定

実稼働環境では、インターネットへの直接アクセスを拒否し、代わりに HTTP または HTTPS プロキシーを使用することができます。プロキシー設定を **install-config.yaml** ファイルで行うことにより、新規の OpenShift Container Platform クラスターをプロキシーを使用するように設定できます。

前提条件

- 既存の **install-config.yaml** ファイル。
- クラスターがアクセスする必要のあるサイトを確認し、プロキシーをバイパスする必要があるかどうかを判別する。デフォルトで、すべてのクラスター egress トラフィック (クラスターをホストするクラウドに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) はプロキシーされます。プロキシーオブジェクトの **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ーをバイパスします。



注記

プロキシーオブジェクトの **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、プロキシー設定を追加します。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ー URL。URL スキームは **http**である必要があります。
2. クラスター外で HTTPS 接続を作成するために使用するプロキシー URL。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。URL スキームは **http** である必要があります。 **https** は現在サポートされていません。

3 プロキシー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

4 指定されている場合、インストールプログラムは HTTPS 接続のプロキシーに必要な 1 つ以上の追加の CA 証明書が含まれる **user-ca-bundle** という名前の ConfigMap を **openshift-config** namespace に生成します。次に、Cluster Network Operator は 3 つのコンテンツを 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 はプロキシーオブジェクトの **trustedCA** フィールドで参照されます。**additionalTrustBundle** フィールドは、プロキシーのアイデンティティー証明書が RHCOS 信頼バンドルからの認証局によって署名されない限り必要になります。

注記

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

2. ファイルを保存し、OpenShift Container Platform のインストール時にこれを参照します。

インストールプログラムは、指定の **install-config.yaml** ファイルのプロキシー設定を使用する **cluster** という名前のクラスター全体のプロキシーを作成します。プロキシー設定が指定されていない場合、**cluster** のプロキシー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ーオブジェクトのみがサポートされ、追加のプロキシーを作成することはできません。

1.1.8. 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。
- **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
```

```
WARNING There are no compute nodes specified. The cluster will not fully initialize without
compute nodes.
INFO Consuming "Install Config" from target directory
```

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ト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. **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。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
```

1.1.9. vSphere での 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インフラストラクチャーが含まれるクラスターを VMware vSphere にインストールする前に、それが使用する RHCOS マシンを vSphere ホストに作成する必要があります。

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。
- vSphere クラスターを作成します。

手順

1. **<installation_directory>/bootstrap.ign** という名前のインストールプログラムが作成したブートストラップ Ignition 設定ファイルを HTTP サーバーにアップロードします。このファイルの URL をメモします。
   ブートストラップ Ignition 設定ファイルはサイズが大きすぎて vApp プロパティーに適さないため、これをホストする必要があります。

2. ブートストラップノードの以下の二次的な Ignition 設定ファイルを、**<installation_directory>/append-bootstrap.ign** としてコンピューターに保存します。

```
{
  "ignition": {
    "config": {
      "append": [
        {
          "source": "<bootstrap_ignition_config_url>", 1
          "verification": {}
        }
      ]
    },
    "timeouts": {},
    "version": "2.1.0"
  },
  "networkd": {},
  "passwd": {},
  "storage": {},
  "systemd": {}
}
```

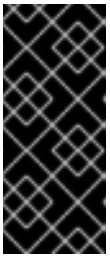
   1 ホストしているブートストラップの Ignition 設定ファイルの URL を指定します。

   ブートストラップマシンの仮想マシン (VM) を作成する場合に、この Ignition 設定ファイルを使用します。

3. マスター、ワーカー、および二次的なブートストラップ Ignition 設定ファイルを Base64 エンコーディングに変換します。
   たとえば、Linux オペレーティングシステムを使用する場合、 **base64** コマンドを使用してファイルをエンコードできます。

```
$ base64 -w0 <installation_directory>/master.ign > <installation_directory>/master.64
$ base64 -w0 <installation_directory>/worker.ign > <installation_directory>/worker.64
$ base64 -w0 <installation_directory>/append-bootstrap.ign >
<installation_directory>/append-bootstrap.64
```

4. 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 RHCOS OVA イメージを取得します。

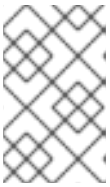


重要

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内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。

ファイル名には、**rhcos-<version>-<architecture>-vmware.ova** 形式の OpenShift Container Platform のバージョン番号が含まれます。

5. vSphere クライアントで、仮想マシンを保管するフォルダーをデータセンターに作成します。
   a. **VMs and Templates** ビューをクリックします。
   b. データセンターの名前を右クリックします。
   c. **New Folder → New VM and Template Folder**をクリックします。
   d. 表示されるウィンドウで、フォルダー名を入力します。フォルダー名は、**install-config.yaml** ファイルで指定したクラスター名と一致している必要があります。
6. vSphere クライアントで、OVA イメージのテンプレートを作成します。



注記

以下の手順では、すべてのクラスターマシンに同じテンプレートを使用し、仮想マシンのプロビジョニングの際に該当するマシンタイプの Ignition 設定ファイルの場所を指定します。

   a. **Hosts and Clusters** タブで、クラスターの名前を右クリックし、**Deploy OVF Template** をクリックします。
   b. **Select an OVF** タブで、ダウンロードした RHCOS OVA ファイルの名前を指定します。
   c. **Select a name and folder**タブで、RHCOS などの **Virtual machine name**を設定し、vSphere クラスターの名前をクリックし、直前のステップで作成したフォルダーを選択します。
   d. **Select a compute resource** タブで、vSphere クラスターの名前をクリックします。
   e. **Select storage** タブで、仮想マシンのストレージオプションを設定します。
      - **Thin Provision** を選択します。
      - **install-config.yaml** ファイルで指定したデータストアを選択します。
   f. **Select network** タブで、クラスターに設定したネットワークを指定します (ある場合)。
   g. すべてのクラスターマシンタイプに同じテンプレートを使用する予定の場合、**Customize template** タブに値を指定しないでください。
7. テンプレートがデプロイされた後に、マシンの仮想マシンをクラスターにデプロイします。
   a. テンプレートの名前を右クリックし、**Clone → Clone to Virtual Machine**をクリックします。

b. **Select a name and folder**タブで、仮想マシンの名前を指定します。**control-plane-0** または **compute-1** などのように、マシンタイプを名前に含めることができるかもしれません。

c. **Select a name and folder**タブで、クラスターに作成したフォルダーの名前を選択します。

d. **Select a compute resource** タブで、データセンター内のホストの名前を選択します。

e. オプション: **Select storage** タブで、ストレージオプションをカスタマイズします。

f. **Select clone options** で、**Customize this virtual machine's hardware**を選択します。

g. **Customize hardware** タブで、**VM Options→ Advanced** をクリックします。

- オプション: クラスターのパフォーマンスに問題が生じる場合は、**Latency Sensitivity** 一覧から **High** を選択します。
- **Edit Configuration** をクリックし、**Configuration Parameters** ウィンドウで **Add Configuration Params** をクリックします。以下のパラメーター名および値を定義します。
  - **guestinfo.ignition.config.data**: このマシンファイルの base64 でエンコードした Ignition 設定ファイルの内容を貼り付けます。
  - **guestinfo.ignition.config.data.encoding**: **base64** を指定します。
  - **disk.EnableUUID**: **TRUE** を指定します。
- または、仮想マシンの電源を入れる前に vApp プロパティーを使用して追加します。
  - vCenter Server インベントリーから仮想マシンに移動します。
  - **Configure** タブで **Settings** を展開し、**vAPP options** を選択します。
  - スクロールダウンし、**Properties** の下で上記の設定を適用します。

h. **Customize hardware** タブの **Virtual Hardware** パネルで、必要に応じて指定した値を変更します。RAM、CPU、およびディスクストレージの量がマシンタイプの最小要件を満たすことを確認してください。

i. 設定を完了し、仮想マシンの電源をオンにします。

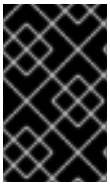
8. 各マシンごとに先の手順に従って、クラスターの残りのマシンを作成します。

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. 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 をインストールすることができます。



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 4.2 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。

手順

1. Red Hat OpenShift Cluster Manager サイトの Infrastructure Provider ページから、選択するインストールタイプのページに移動し、**Download Command-line Tools**をクリックします。
2. オペレーティングシステムおよびアーキテクチャーのフォルダーをクリックしてから、圧縮されたファイルをクリックします。

注記

oc は Linux、Windows、または macOS にインストールできます。

3. ファイルをファイルシステムに保存します。
4. 圧縮ファイルを展開します。
5. これを **PATH** にあるディレクトリーに配置します。

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>
```

1.1.11. 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すること。
- 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していること。
- クラスターの RHCOS マシンを作成するために Ignition 設定ファイルを使用していること。
- 使用するマシンでインターネットに直接アクセスできること。

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:6443...
INFO API v1.14.6+c4799753c up
INFO Waiting up to 30m0s for the bootstrap-complete event...
```

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。

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 クラスターのデプロイ。
- **oc** CLI のインストール。

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
```

1.1.13. 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

- マシンをクラスターに追加していること。

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  v1.14.6+c4799753c
master-1  Ready     master  63m  v1.14.6+c4799753c
master-2  Ready     master  64m  v1.14.6+c4799753c
worker-0  NotReady  worker  76s  v1.14.6+c4799753c
worker-1  NotReady  worker  70s  v1.14.6+c4799753c
```

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
```

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ること。

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                       4.2.0     True        False         False      69s
cloud-credential                     4.2.0     True        False         False      12m
cluster-autoscaler                   4.2.0     True        False         False      11m
console                              4.2.0     True        False         False      46s
dns                                  4.2.0     True        False         False      11m
image-registry                       4.2.0     False       True          False      5m26s
ingress                              4.2.0     True        False         False      5m36s
kube-apiserver                       4.2.0     True        False         False      8m53s
kube-controller-manager              4.2.0     True        False         False      7m24s
kube-scheduler                       4.2.0     True        False         False      12m
machine-api                          4.2.0     True        False         False      12m
machine-config                       4.2.0     True        False         False      7m36s
marketplace                          4.2.0     True        False         False      7m54m
monitoring                           4.2.0     True        False         False      7h54s
network                              4.2.0     True        False         False      5m9s
node-tuning                          4.2.0     True        False         False      11m
openshift-apiserver                  4.2.0     True        False         False      11m
openshift-controller-manager         4.2.0     True        False         False      5m943s
openshift-samples                    4.2.0     True        False         False      3m55s
operator-lifecycle-manager           4.2.0     True        False         False      11m
operator-lifecycle-manager-catalog   4.2.0     True        False         False      11m
service-ca                           4.2.0     True        False         False      11m
```

```
service-catalog-apiserver           4.2.0     True        False         False      5m26s
service-catalog-controller-manager  4.2.0     True        False         False      5m25s
storage                             4.2.0     True        False         False      5m30s
```

2. 利用不可の Operator を設定します。

1.1.14.1. イメージレジストリーストレージの設定

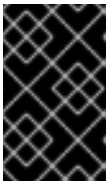
image-registry Operator が利用できない場合、そのストレージを設定する必要があります。実稼働クラスターに必要な PersistentVolume の設定方法と、実稼働用ではないクラスターにのみ使用できる空のディレクトリーをストレージの場所として設定する方法が表示されます。

1.1.14.1.1. VMware vSphere 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。
- VMware vSphere 上のクラスター。
- **ReadWriteMany** アクセスモードのプロビジョニングされた永続ボリューム (PV)(例: **NFS**)。



重要

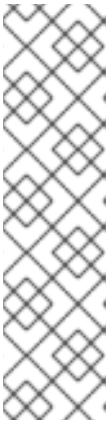
vSphere ボリュームは **ReadWriteMany** アクセスモードをサポートしません。レジストリーストレージを設定するには、**NFS**などの異なるストレージバックエンドを使用する必要があります。

- 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。
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** で、レジストリー Pod を **replica>1** を設定してスケールアップする必要がある場合、**no_wdelay** 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,no_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. オプション: 新しいストレージクラスを PV に追加します。

   a. PV を作成します。

```
$ oc create -f -
```

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: image-registry-pv
spec:
  accessModes:
    ReadWriteMany
    capacity:
      storage: 100Gi
  nfs:
    path: /registry
    server: 172.16.231.181
  persistentVolumeReclaimPolicy: Retain
  storageClassName: nfs01
```

```
$ oc get pv
```

   b. PVC を作成します。

```
$ oc create -n openshift-image-registry -f -
```

```
apiVersion: "v1"
kind: "PersistentVolumeClaim"
metadata:
  name: "image-registry-pvc"
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: nfs01
  volumeMode: Filesystem
```

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

   最後に、PVC の名前を追加します。

```
$ oc edit configs.imageregistry.operator.openshift.io -o yaml
```

```
storage:
  pvc:
    claim: image-registry-pvc (1)
```

(1) カスタム PVC を作成すると、**image-registry-storage** PVC のデフォルトの自動作成の **claim** フィールドを空のままにすることができます。

5. **clusteroperator** ステータスを確認します。

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

1.1.14.1.2. 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。

1.1.15. 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ること。

- Operator の初期設定を完了していること。

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                       4.2.0     True        False         False      10m
cloud-credential                     4.2.0     True        False         False      22m
cluster-autoscaler                   4.2.0     True        False         False      21m
console                              4.2.0     True        False         False      10m
dns                                  4.2.0     True        False         False      21m
image-registry                       4.2.0     True        False         False      16m
ingress                              4.2.0     True        False         False      16m
kube-apiserver                       4.2.0     True        False         False      19m
kube-controller-manager              4.2.0     True        False         False      18m
kube-scheduler                       4.2.0     True        False         False      22m
machine-api                          4.2.0     True        False         False      22m
machine-config                       4.2.0     True        False         False      18m
marketplace                          4.2.0     True        False         False      18m
monitoring                           4.2.0     True        False         False      18m
network                              4.2.0     True        False         False      16m
node-tuning                          4.2.0     True        False         False      21m
openshift-apiserver                  4.2.0     True        False         False      21m
openshift-controller-manager         4.2.0     True        False         False      17m
openshift-samples                    4.2.0     True        False         False      14m
operator-lifecycle-manager           4.2.0     True        False         False      21m
operator-lifecycle-manager-catalog   4.2.0     True        False         False      21m
service-ca                           4.2.0     True        False         False      21m
service-catalog-apiserver            4.2.0     True        False         False      16m
service-catalog-controller-manager   4.2.0     True        False         False      16m
storage                              4.2.0     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す。

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。

1.2. ネットワークのカスタマイズによる VSPHERE へのクラスターのインストール

OpenShift Container Platform バージョン 4.2 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 VMware vSphere インフラストラクチャーにクラスター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、クラスターは環境内の既存の 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、実行中のクラスターで変更できるのは **kubeProxy** 設定パラメーターのみになります。

前提条件

- クラスターの永続ストレージ 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。

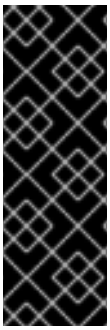
1.2.1. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス

OpenShift Container Platform 4.2 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。

1.2.2. VMware vSphere インフラストラクチャーの要件

OpenShift Container Platform クラスターを、VMware vSphere のバージョン 6.5 または 6.7U2 以降のインスタンスにインストールする必要があります。

VMware では、vSphere バージョン 6.7 U2 以降を OpenShift Container Platform クラスターで使用することを推奨しています。 vSphere 6.7U2 には以下が含まれます。

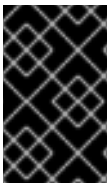
- VMware NSX-T のサポート

- インツリー (in-tree) VCP を使用する vSAN、VMFS および NFS のサポート

vSphere 6.5 とハードウェアのバージョン 13 の使用がサポートされていますが、OpenShift Container Platform クラスターは以下の制限を受けます。

- NSX-T SDN はサポートされません。
- OpenShift Container Platform がサポートする別の SDN またはストレージプロバイダーを使用する必要があります。

vSphere バージョン 6.5 インスタンスを使用している場合は、OpenShift Container Platform をインストールする前に 6.7U2 にアップグレードすることを検討してください。



重要

OpenShift Container Platform をインストールする前に、ESXi ホストの時間が同期されていることを確認する必要があります。VMware ドキュメントの「Edit Time Configuration for a Host」を参照してください。

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。

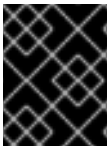
1.2.3.1. 必要なマシン

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 つのコンピュートマシン (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トマシンでは、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 サーバーが必要になります。

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
コンピュート	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 Tested Integrations」ページを参照してください。

手順

1. DHCP 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 サーバーが必要になります。

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1.4 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1.5 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

ネットワークトポロジー要件

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、ネットワークトポロジーの以下の要件を満たす必要があります。

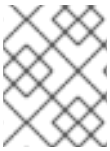
重要

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ー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、2 つの Layer 4 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。API には 1 つのロードバランサーが必要で、デフォルトの Ingress コントローラーには、Ingress をアプリケーションに提供する 2 番目のロードバランサーが必要です。

ポート	マシン	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。	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
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



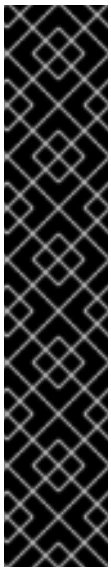
注記

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1.6 必要な DNS レコード

Component	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、プロキシーされる API 呼び出しが失敗し、Pod からログを取得できなくなる可能性があります。
Routes	***.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。

Component	レコード	説明
etcd	**etcd-<index>.<cluster_name>.<base_domain>.**	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 **<index>** 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。
	_etcd-server-ssl._tcp.<cluster_name>.<base_domain>.	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。 `# _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>`

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。

   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

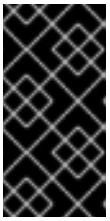
ローカルコンピューターにダウンロードします。

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。

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ーイメージを提供する、Quay.io などの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. 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

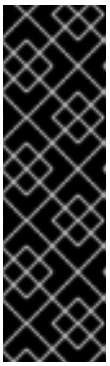
- OpenShift Container Platform インストーラープログラムおよびクラスターのアクセストークンを取得します。

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

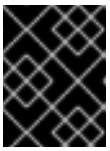
ディレクトリーを作成する必要があります。ブートストラップ X.509 証明書などの一部のインストールアセットの有効期限は短く設定されているため、インストールディレクトリーを再利用することができません。別のクラスターインストールの個別のファイルを再利用する必要がある場合は、それらをディレクトリーにコピーすることができます。ただし、インストールアセットのファイル名はリリース間で変更される可能性があります。インストールファイルを以前のバージョンの OpenShift Container Platform からコピーする場合は注意してコピーを行ってください。

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。

1.2.7.1. VMware vSphere 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
  replicas: 3 (7)
metadata:
  name: test (8)
platform:
  vsphere:
    vcenter: your.vcenter.server (9)
    username: username (10)
    password: password (11)
    datacenter: datacenter (12)
```

```
    defaultDatastore: datastore 13
pullSecret: '{"auths": ...}' 14
sshKey: 'ssh-ed25519 AAAA...' 15
```

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 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。同時マルチスレッドを無効にする場合は、マシンで最低でも 8 CPU および 32 GB の RAM を使用する必要があります。

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。

7 クラスターに追加するコントロールプレーンマシンの数。クラスターをこの値をクラスターの etcd エンドポイント数として使用するため、値はデプロイするコントロールプレーンマシンの数に一致する必要があります。

8 DNS レコードに指定したクラスター名。

9 vCenter サーバーの完全修飾ホスト名または IP アドレス。

10 サーバーにアクセスするユーザーの名前。このユーザーには、少なくとも vSphere の静的または動的な永続ボリュームのプロビジョニングに必要なロールおよび権限がなければなりません。

11 vSphere ユーザーに関連付けられたパスワード。

12 vSphere データセンター。

13 使用するデフォルトの vSphere データストア。

14 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ーイメージを提供する、Quay.io などの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。

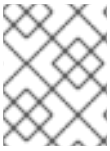
15 Red Hat Enterprise Linux CoreOS (RHCOS) の **core** ユーザーのデフォルト SSH キーの公開部分。

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のある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

1.2.7.2. 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1.7 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	デプロイするネットワークプラグイン。**OpenShiftSDN** プラグインのみが OpenShift Container Platform 4.2 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。
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。

1.2.8. 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 1
```

1 **<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-*
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: 1
  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
```

1 **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。

6. 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、クラスターの作成時に **manifests/** ディレクトリーを削除します。

1.2.9. クラスターネットワーク Operator のカスタムリソース (CR、Custom Resource)

Network.operator.openshift.io カスタムリソース (CR) のクラスターネットワーク設定は、Cluster Network Operator (CNO) の設定内容を保存します。Operator はクラスターネットワークを管理します。

defaultNetwork パラメーターのパラメーターを CNO CR に設定することにより、OpenShift Container Platform クラスターのクラスター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
      - 30s
```

1 2 **install-config.yaml** ファイルに指定されます。

3 クラスターネットワークの SDN (software-defined networking) 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 、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。

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 time package ドキュメントで説明されています。

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** が含まれ、これらについては、Go time package で説明されています。

1.2.9.1. OpenShift SDN の設定パラメーター

以下の YAML オブジェクトは OpenShift SDN 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 **OpenShiftSDN** 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 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る可能性があります。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。

1.2.9.2. Cluster Network Operator のサンプル CR

以下の例のように、CNO の完全な CR 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:
      - 30s
```

1.2.10. 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>** については、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
│   └── kubeconfig
├── bootstrap.ign
├── master.ign
├── metadata.json
└── worker.ign
```

1.2.11. vSphere での 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インフラストラクチャーが含まれるクラスターを VMware vSphere にインストールする前に、それが使用する RHCOS マシンを vSphere ホストに作成する必要があります。

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。
- vSphere クラスターを作成します。

手順

1. **<installation_directory>/bootstrap.ign** という名前のインストールプログラムが作成したブートストラップ Ignition 設定ファイルを HTTP サーバーにアップロードします。このファイルの URL をメモします。
   ブートストラップ Ignition 設定ファイルはサイズが大きすぎて vApp プロパティーに適さないため、これをホストする必要があります。
2. ブートストラップノードの以下の二次的な Ignition 設定ファイルを、**<installation_directory>/append-bootstrap.ign** としてコンピューターに保存します。

```
{
```

```
{
  "ignition": {
    "config": {
      "append": [
        {
          "source": "<bootstrap_ignition_config_url>", 1
          "verification": {}
        }
      ]
    },
    "timeouts": {},
    "version": "2.1.0"
  },
  "networkd": {},
  "passwd": {},
  "storage": {},
  "systemd": {}
}
```

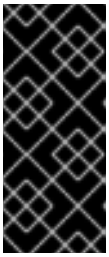
1 ホストしているブートストラップの Ignition 設定ファイルの URL を指定します。

ブートストラップマシンの仮想マシン (VM) を作成する場合に、この Ignition 設定ファイルを使用します。

3. マスター、ワーカー、および二次的なブートストラップ Ignition 設定ファイルを Base64 エンコーディングに変換します。
たとえば、Linux オペレーティングシステムを使用する場合、 **base64** コマンドを使用してファイルをエンコードできます。

```
$ base64 -w0 <installation_directory>/master.ign > <installation_directory>/master.64
$ base64 -w0 <installation_directory>/worker.ign > <installation_directory>/worker.64
$ base64 -w0 <installation_directory>/append-bootstrap.ign >
<installation_directory>/append-bootstrap.64
```

4. 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 RHCOS OVA 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内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。

ファイル名には、**rhcos-<version>-<architecture>-vmware.ova** 形式の OpenShift Container Platform のバージョン番号が含まれます。

5. vSphere クライアントで、仮想マシンを保管するフォルダーをデータセンターに作成します。

   a. **VMs and Templates** ビューをクリックします。

   b. データセンターの名前を右クリックします。

   c. **New Folder → New VM and Template Folder**をクリックします。

d. 表示されるウィンドウで、フォルダー名を入力します。フォルダー名は、**install-config.yaml** ファイルで指定したクラスター名と一致している必要があります。

6. vSphere クライアントで、OVA イメージのテンプレートを作成します。

注記

以下の手順では、すべてのクラスターマシンに同じテンプレートを使用し、仮想マシンのプロビジョニングの際に該当するマシンタイプの Ignition 設定ファイルの場所を指定します。

a. **Hosts and Clusters** タブで、クラスターの名前を右クリックし、**Deploy OVF Template** をクリックします。

b. **Select an OVF** タブで、ダウンロードした RHCOS OVA ファイルの名前を指定します。

c. **Select a name and folder**タブで、RHCOS などの **Virtual machine name**を設定し、vSphere クラスターの名前をクリックし、直前のステップで作成したフォルダーを選択します。

d. **Select a compute resource** タブで、vSphere クラスターの名前をクリックします。

e. **Select storage** タブで、仮想マシンのストレージオプションを設定します。

- **Thin Provision** を選択します。
- **install-config.yaml** ファイルで指定したデータストアを選択します。

f. **Select network** タブで、クラスターに設定したネットワークを指定します (ある場合)。

g. すべてのクラスターマシンタイプに同じテンプレートを使用する予定の場合、**Customize template** タブに値を指定しないでください。

7. テンプレートがデプロイされた後に、マシンの仮想マシンをクラスターにデプロイします。

a. テンプレートの名前を右クリックし、**Clone → Clone to Virtual Machine**をクリックします。

b. **Select a name and folder**タブで、仮想マシンの名前を指定します。**control-plane-0** または **compute-1** などのように、マシンタイプを名前に含めることができるかもしれません。

c. **Select a name and folder**タブで、クラスターに作成したフォルダーの名前を選択します。

d. **Select a compute resource** タブで、データセンター内のホストの名前を選択します。

e. オプション: **Select storage** タブで、ストレージオプションをカスタマイズします。

f. **Select clone options** で、**Customize this virtual machine's hardware**を選択します。

g. **Customize hardware** タブで、**VM Options→ Advanced** をクリックします。

- オプション: クラスターのパフォーマンスに問題が生じる場合は、**Latency Sensitivity** 一覧から **High** を選択します。

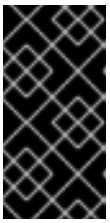
- **Edit Configuration** をクリックし、**Configuration Parameters** ウィンドウで **Add**

- Edit Configuration をクリックし、Configuration Parameters ウィンドウで Add Configuration Params をクリックします。以下のパラメーター名および値を定義します。
  - **guestinfo.ignition.config.data**: このマシンファイルの base64 でエンコードした Ignition 設定ファイルの内容を貼り付けます。
  - **guestinfo.ignition.config.data.encoding**: **base64** を指定します。
  - **disk.EnableUUID**: **TRUE** を指定します。
- または、仮想マシンの電源を入れる前に vApp プロパティーを使用して追加します。
  - vCenter Server インベントリーから仮想マシンに移動します。
  - **Configure** タブで **Settings** を展開し、**vAPP options** を選択します。
  - スクロールダウンし、**Properties** の下で上記の設定を適用します。

h. **Customize hardware** タブの **Virtual Hardware** パネルで、必要に応じて指定した値を変更します。RAM、CPU、およびディスクストレージの量がマシンタイプの最小要件を満たすことを確認してください。

i. 設定を完了し、仮想マシンの電源をオンにします。

8. 各マシンごとに先の手順に従って、クラスターの残り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。

1.2.12. 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 をインストールすることができます。

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 4.2 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。

手順

1. Red Hat OpenShift Cluster Manager サイトの Infrastructure Provider ページから、選択するインストールタイプのページに移動し、**Download Command-line Tools**をクリックします。
2. オペレーティングシステムおよびアーキテクチャーのフォルダーをクリックしてから、圧縮されたファイルをクリックします。

注記

oc は Linux、Windows、または macOS にインストールできます。

3. ファイルをファイルシステムに保存します。
4. 圧縮ファイルを展開します。
5. これを **PATH** にあるディレクトリーに配置します。

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>
```

1.2.13. 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すること。
- 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していること。
- クラスターの RHCOS マシンを作成するために Ignition 設定ファイルを使用していること。
- 使用するマシンでインターネットに直接アクセスできること。

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:6443...
INFO API v1.14.6+c4799753c up
INFO Waiting up to 30m0s for the bootstrap-complete event...
```

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。

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 クラスターのデプロイ。
- **oc** CLI のインストール。

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

- マシンをクラスターに追加していること。

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  v1.14.6+c4799753c
master-1  Ready     master  63m  v1.14.6+c4799753c
master-2  Ready     master  64m  v1.14.6+c4799753c
worker-0  NotReady  worker  76s  v1.14.6+c4799753c
worker-1  NotReady  worker  70s  v1.14.6+c4799753c
```

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
```

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ること。

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                             4.2.0     True        False         False      69s
cloud-credential                           4.2.0     True        False         False      12m
cluster-autoscaler                         4.2.0     True        False         False      11m
console                                    4.2.0     True        False         False      46s
dns                                        4.2.0     True        False         False      11m
image-registry                             4.2.0     False       True          False      5m26s
ingress                                    4.2.0     True        False         False      5m36s
kube-apiserver                             4.2.0     True        False         False      8m53s
kube-controller-manager                    4.2.0     True        False         False      7m24s
kube-scheduler                             4.2.0     True        False         False      12m
machine-api                                4.2.0     True        False         False      12m
machine-config                             4.2.0     True        False         False      7m36s
marketplace                                4.2.0     True        False         False      7m54m
monitoring                                 4.2.0     True        False         False      7h54s
network                                    4.2.0     True        False         False      5m9s
node-tuning                                4.2.0     True        False         False      11m
openshift-apiserver                        4.2.0     True        False         False      11m
openshift-controller-manager               4.2.0     True        False         False      5m943s
openshift-samples                          4.2.0     True        False         False      3m55s
operator-lifecycle-manager                 4.2.0     True        False         False      11m
operator-lifecycle-manager-catalog         4.2.0     True        False         False      11m
service-ca                                 4.2.0     True        False         False      11m
service-catalog-apiserver                  4.2.0     True        False         False      5m26s
service-catalog-controller-manager         4.2.0     True        False         False      5m25s
storage                                    4.2.0     True        False         False      5m30s
```

2. 利用不可の Operator を設定します。

1.2.16.1. イメージレジストリーストレージの設定

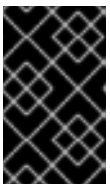
image-registry Operator が利用できない場合、そのストレージを設定する必要があります。実稼働クラスターに必要な PersistentVolume の設定方法と、実稼働用ではないクラスターにのみ使用できる空のディレクトリーをストレージの場所として設定する方法が表示されます。

1.2.16.1.1. VMware vSphere 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。
- VMware vSphere 上のクラスター。
- **ReadWriteMany** アクセスモードのプロビジョニングされた永続ボリューム (PV)(例: **NFS**)。



重要

vSphere ボリュームは **ReadWriteMany** アクセスモードをサポートしません。レジストリーストレージを設定するには、**NFS**などの異なるストレージバックエンドを使用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。
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** で、レジストリー Pod を **replica>1** を設定してスケールアップする必要がある場合、**no_wdelay** 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,no_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. オプション: 新しいストレージクラスを PV に追加します。
   a. PV を作成します。

```
$ oc create -f -
```

```
apiVersion: v1
kind: PersistentVolume
```

```
metadata:
  name: image-registry-pv
spec:
  accessModes:
    ReadWriteMany
    capacity:
      storage: 100Gi
  nfs:
    path: /registry
    server: 172.16.231.181
  persistentVolumeReclaimPolicy: Retain
  storageClassName: nfs01
```

```
$ oc get pv
```

b. PVC を作成します。

```
$ oc create -n openshift-image-registry -f -
```

```
apiVersion: "v1"
kind: "PersistentVolumeClaim"
metadata:
  name: "image-registry-pvc"
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: nfs01
  volumeMode: Filesystem
```

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

最後に、PVC の名前を追加します。

```
$ oc edit configs.imageregistry.operator.openshift.io -o yaml
```

```
storage:
  pvc:
    claim: image-registry-pvc 1
```

1 カスタム PVC を作成すると、**image-registry-storage** PVC のデフォルトの自動作成の **claim** フィールドを空のままにすることができます。

5. **clusteroperator** ステータスを確認します。

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

1.2.16.1.2. 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。

1.2.17. 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ること。
- Operator の初期設定を完了していること。

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                       4.2.0     True        False         False      10m
cloud-credential                     4.2.0     True        False         False      22m
cluster-autoscaler                   4.2.0     True        False         False      21m
console                              4.2.0     True        False         False      10m
dns                                  4.2.0     True        False         False      21m
image-registry                       4.2.0     True        False         False      16m
ingress                              4.2.0     True        False         False      16m
```

```
kube-apiserver                 4.2.0     True        False         False      19m
kube-controller-manager        4.2.0     True        False         False      18m
kube-scheduler                 4.2.0     True        False         False      22m
machine-api                    4.2.0     True        False         False      22m
machine-config                 4.2.0     True        False         False      18m
marketplace                    4.2.0     True        False         False      18m
monitoring                     4.2.0     True        False         False      18m
network                        4.2.0     True        False         False      16m
node-tuning                    4.2.0     True        False         False      21m
openshift-apiserver            4.2.0     True        False         False      21m
openshift-controller-manager   4.2.0     True        False         False      17m
openshift-samples              4.2.0     True        False         False      14m
operator-lifecycle-manager     4.2.0     True        False         False      21m
operator-lifecycle-manager-catalog   4.2.0     True        False         False      21m
service-ca                     4.2.0     True        False         False      21m
service-catalog-apiserver      4.2.0     True        False         False      16m
service-catalog-controller-manager   4.2.0     True        False         False      16m
storage                        4.2.0     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す。

次のステップ

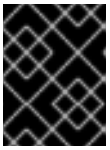
- クラスターをカスタマイズします。
- 必要な場合は、リモートの健全性レポートをオプトアウトすることができます。

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

OpenShift Container Platform バージョン 4.2 では、クラスターを制限されたネットワークでプロビジョニングする VMware vSphere インフラストラクチャーにインストールできます。

前提条件

- bastion ホストでミラー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 bastion 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 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

プロキシーを設定する場合は、このサイト一覧も確認してください。

1.3.1. 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 4.2 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 bastion 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。

1.3.1.1. 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。

1.3.2. 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 4.2 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。

1.3.3. VMware vSphere インフラストラクチャーの要件

OpenShift Container Platform クラスターを、VMware vSphere のバージョン 6.5 または 6.7U2 以降のインスタンスにインストールする必要があります。

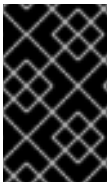
VMware では、vSphere バージョン 6.7 U2 以降を OpenShift Container Platform クラスターで使用することを推奨しています。 vSphere 6.7U2 には以下が含まれます。

- VMware NSX-T のサポート
- インツリー (in-tree) VCP を使用する vSAN、VMFS および NFS のサポート

vSphere 6.5 とハードウェアのバージョン 13 の使用がサポートされていますが、OpenShift Container Platform クラスターは以下の制限を受けます。

- NSX-T SDN はサポートされません。
- OpenShift Container Platform がサポートする別の SDN またはストレージプロバイダーを使用する必要があります。

vSphere バージョン 6.5 インスタンスを使用している場合は、OpenShift Container Platform をインストールする前に 6.7U2 にアップグレードすることを検討してください。



重要

OpenShift Container Platform をインストールする前に、ESXi ホストの時間が同期されていることを確認する必要があります。VMware ドキュメントの「Edit Time Configuration for a Host」を参照してください。

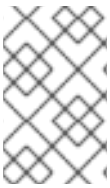
1.3.4. ユーザーによってプロビジョニングされるインフラストラクチャーを使用する場合のクラスターのマシン要件

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。

1.3.4.1. 必要なマシン

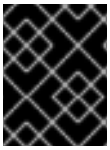
最小の OpenShift Container Platform クラスターでは以下のホストが必要です。

- 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 つのコンピュートマシン (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トマシンでは、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 サーバーが必要になります。

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
コンピュート	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 Tested Integrations」ページを参照してください。

手順

1. DHCP 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 サーバーが必要になります。

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1.8 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

プロトコル	ポート	説明
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1.9 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

ネットワークトポロジー要件
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、ネットワークトポロジ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ー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、2 つの Layer 4 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。API には 1 つのロードバランサーが必要で、デフォルトの Ingress コントローラーには、Ingress をアプリケーションに提供する 2 番目のロードバランサーが必要です。

ポート	マシン	内部	外部	説明

ポート	マシン	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。	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
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

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。

1.3.5.2. 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

Component	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。

Component	レコード	説明
	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、プロキシーされる API 呼び出しが失敗し、Pod からログを取得できなくなる可能性があります。
Routes	***.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。

Component	レコード	説明
etcd	**etcd-<index>.<cluster_name>.<base_domain>.**	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 **<index>** 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。
	_etcd-server-ssl._tcp.<cluster_name>.<base_domain>.	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。 `# _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>`

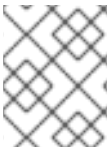
1.3.6. SSH プライベートキーの生成およびエージェントへの追加

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。

   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。

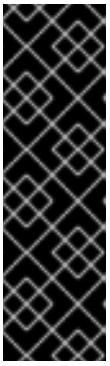
前提条件

- 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。
- リポジトリー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。

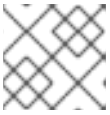
```
$ mkdir <installation_directory>
```



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。

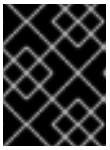


注記

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ー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。

1.3.7.1. VMware vSphere 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
  replicas: 3 (7)
metadata:
  name: test (8)
platform:
  vsphere:
    vcenter: your.vcenter.server (9)
    username: username (10)
    password: password (11)
    datacenter: datacenter (12)
    defaultDatastore: datastore (13)
pullSecret: '{"auths":{"<bastion_host_name>:5000": {"auth": "<credentials>","email":
"you@example.com"}}}' (14)
sshKey: 'ssh-ed25519 AAAA...' (15)
additionalTrustBundle: | (16)
  -----BEGIN CERTIFICATE-----
  ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
  -----END CERTIFICATE-----
imageContentSources: (17)
- mirrors:
  - <bastion_host_name>:5000/<repo_name>/release
  source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - <bastion_host_name>:5000/<repo_name>/release
  source: registry.svc.ci.openshift.org/ocp/release
```

(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) 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。同時マルチスレッドを無効にする場合は、マシンで最低でも 8 CPU および 32 GB の RAM を使用する必要があります。

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。

7 クラスターに追加するコントロールプレーンマシンの数。クラスターをこの値をクラスターの etcd エンドポイント数として使用するため、値はデプロイするコントロールプレーンマシンの数に一致する必要があります。

8 DNS レコードに指定したクラスター名。

9 vCenter サーバーの完全修飾ホスト名または IP アドレス。

10 サーバーにアクセスするユーザーの名前。このユーザーには、少なくとも vSphere の静的または動的な永続ボリュームのプロビジョニングに必要なロールおよび権限がなければなりません。

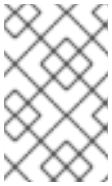
11 vSphere ユーザーに関連付けられたパスワード。

12 vSphere データセンター。

13 使用するデフォルトの vSphere データストア。

14 **bastion_host_name**の場合、ミラーレジストリーの証明書で指定したレジストリードメイン名を指定し、**<credentials>** の場合は、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。

15 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のある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

16 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。

17 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ーの設定

実稼働環境では、インターネットへの直接アクセスを拒否し、代わりに HTTP または HTTPS プロキシーを使用することができます。プロキシー設定を **install-config.yaml** ファイルで行うことにより、新規の OpenShift Container Platform クラスターをプロキシーを使用するように設定できます。

前提条件

- 既存の **install-config.yaml** ファイル。
- クラスターがアクセスする必要のあるサイトを確認し、プロキシーをバイパスする必要があるかどうかを判別する。デフォルトで、すべてのクラスター egress トラフィック (クラスターをホストするクラウドに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) はプロキシーされます。プロキシーオブジェクトの **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ーをバイパスします。

注記

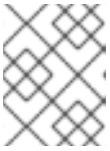
プロキシーオブジェクトの **status.noProxy** フィールドは、デフォルトでインスタンスメタデータエンドポイント (**169.254.169.254**) およびインストール設定の **networking.machineCIDR**、 **networking.clusterNetwork.cidr**、および **networking.serviceNetwork** フィールドの値で設定されます。

手順

1. **install-config.yaml** ファイルを編集し、プロキシー設定を追加します。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ー URL。URL スキームは **http**である必要があります。
2. クラスター外で HTTPS 接続を作成するために使用するプロキシー URL。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。URL スキームは **http** である必要があります。 **https** は現在サポートされていません。
3. プロキシー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ーをバイパスします。
4. 指定されている場合、インストールプログラムは HTTPS 接続のプロキシーに必要な 1 つ以上の追加の CA 証明書が含まれる **user-ca-bundle** という名前の ConfigMap を **openshift-config** namespace に生成します。次に、Cluster Network Operator は 3 つのコンテンツを 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 はプロキシーオブジェクトの **trustedCA** フィールドで参照されます。**additionalTrustBundle** フィールドは、プロキシーのアイデンティティー証明書が RHCOS 信頼バンドルからの認証局によって署名されない限り必要になります。

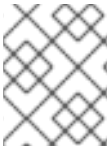


注記

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

2. ファイルを保存し、OpenShift Container Platform のインストール時にこれを参照します。

インストールプログラムは、指定の **install-config.yaml** ファイルのプロキシー設定を使用する **cluster** という名前のクラスター全体のプロキシーを作成します。プロキシー設定が指定されていない場合、**cluster** のプロキシー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ーオブジェクトのみがサポートされ、追加のプロキシーを作成することはできません。

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 bastion 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

WARNING There are no compute nodes specified. The cluster will not fully initialize without
compute nodes.
INFO Consuming "Install Config" from target directory
```

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。

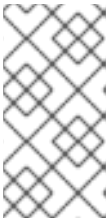
インストールプロセスの後の部分で独自のコンピュートマシンを作成するため、この警告を無視しても問題がありません。

2. **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。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
```

1.3.9. vSphere での 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インフラストラクチャーが含まれるクラスターを VMware vSphere にインストールする前に、それが使用する RHCOS マシンを vSphere ホストに作成する必要があります。

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。
- vSphere クラスターを作成します。

手順

1. **<installation_directory>/bootstrap.ign** という名前のインストールプログラムが作成したブートストラップ Ignition 設定ファイルを HTTP サーバーにアップロードします。このファイルの URL をメモします。

ブートストラップ Ignition 設定ファイルはサイズが大きすぎて vApp プロパティーに適さないため、これをホストする必要があります。

2. ブートストラップノードの以下の二次的な Ignition 設定ファイルを、**<installation_directory>/append-bootstrap.ign** としてコンピューターに保存します。

```
{
  "ignition": {
    "config": {
      "append": [
        {
          "source": "<bootstrap_ignition_config_url>", 1
          "verification": {}
        }
      ]
    },
    "timeouts": {},
    "version": "2.1.0"
  },
  "networkd": {},
  "passwd": {},
  "storage": {},
  "systemd": {}
}
```

1 ホストしているブートストラップの Ignition 設定ファイルの URL を指定します。

ブートストラップマシンの仮想マシン (VM) を作成する場合に、この Ignition 設定ファイルを使用します。

3. マスター、ワーカー、および二次的なブートストラップ Ignition 設定ファイルを Base64 エンコーディングに変換します。
たとえば、Linux オペレーティングシステムを使用する場合、 **base64** コマンドを使用してファイルをエンコードできます。

```
$ base64 -w0 <installation_directory>/master.ign > <installation_directory>/master.64
$ base64 -w0 <installation_directory>/worker.ign > <installation_directory>/worker.64
$ base64 -w0 <installation_directory>/append-bootstrap.ign >
<installation_directory>/append-bootstrap.64
```

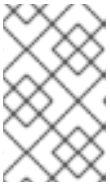
4. Red Hat カスタマーポータルの「製品のダウンロード」ページまたは「 RHCOS イメージミラー」ページから RHCOS OVA イメージを取得します。

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内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。

ファイル名には、**rhcos-<version>-<architecture>-vmware.ova** 形式の OpenShift Container Platform のバージョン番号が含まれます。

5. vSphere クライアントで、仮想マシンを保管するフォルダーをデータセンターに作成します。
   a. **VMs and Templates** ビューをクリックします。
   b. データセンターの名前を右クリックします。
   c. **New Folder → New VM and Template Folder**をクリックします。
   d. 表示されるウィンドウで、フォルダー名を入力します。フォルダー名は、**install-config.yaml** ファイルで指定したクラスター名と一致している必要があります。
6. vSphere クライアントで、OVA イメージのテンプレートを作成します。



注記

以下の手順では、すべてのクラスターマシンに同じテンプレートを使用し、仮想マシンのプロビジョニングの際に該当するマシンタイプの Ignition 設定ファイルの場所を指定します。

   a. **Hosts and Clusters** タブで、クラスターの名前を右クリックし、**Deploy OVF Template** をクリックします。
   b. **Select an OVF** タブで、ダウンロードした RHCOS OVA ファイルの名前を指定します。
   c. **Select a name and folder**タブで、RHCOS などの **Virtual machine name**を設定し、vSphere クラスターの名前をクリックし、直前のステップで作成したフォルダーを選択します。
   d. **Select a compute resource** タブで、vSphere クラスターの名前をクリックします。
   e. **Select storage** タブで、仮想マシンのストレージオプションを設定します。
      - **Thin Provision** を選択します。
      - **install-config.yaml** ファイルで指定したデータストアを選択します。
   f. **Select network** タブで、クラスターに設定したネットワークを指定します (ある場合)。
   g. すべてのクラスターマシンタイプに同じテンプレートを使用する予定の場合、**Customize template** タブに値を指定しないでください。
7. テンプレートがデプロイされた後に、マシンの仮想マシンをクラスターにデプロイします。
   a. テンプレートの名前を右クリックし、**Clone → Clone to Virtual Machine**をクリックします。
   b. **Select a name and folder**タブで、仮想マシンの名前を指定します。**control-plane-0** または **compute-1** などのように、マシンタイプを名前に含めることができるかもしれません。
   c. **Select a name and folder**タブで、クラスターに作成したフォルダーの名前を選択します。
   d. **Select a compute resource** タブで、データセンター内のホストの名前を選択します。
   e. オプション: **Select storage** タブで、ストレージオプションをカスタマイズします。
   f. **Select clone options** で、**Customize this virtual machine's hardware**を選択します。

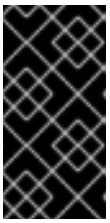
g. **Customize hardware** タブで、**VM Options→ Advanced** をクリックします。

- オプション: クラスターのパフォーマンスに問題が生じる場合は、**Latency Sensitivity** 一覧から **High** を選択します。
- **Edit Configuration** をクリックし、**Configuration Parameters** ウィンドウで **Add Configuration Params** をクリックします。以下のパラメーター名および値を定義します。
  - **guestinfo.ignition.config.data**: このマシンファイルの base64 でエンコードした Ignition 設定ファイルの内容を貼り付けます。
  - **guestinfo.ignition.config.data.encoding**: **base64** を指定します。
  - **disk.EnableUUID**: **TRUE** を指定します。
- または、仮想マシンの電源を入れる前に vApp プロパティーを使用して追加します。
  - vCenter Server インベントリーから仮想マシンに移動します。
  - **Configure** タブで **Settings** を展開し、**vAPP options** を選択します。
  - スクロールダウンし、**Properties** の下で上記の設定を適用します。

h. **Customize hardware** タブの **Virtual Hardware** パネルで、必要に応じて指定した値を変更します。RAM、CPU、およびディスクストレージの量がマシンタイプの最小要件を満たすことを確認してください。

i. 設定を完了し、仮想マシンの電源をオンにします。

8. 各マシンごとに先の手順に従って、クラスターの残り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。

1.3.10. 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すること。
- 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していること。
- クラスターの RHCOS マシンを作成するために Ignition 設定ファイルを使用していること。

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:6443...
INFO API v1.14.6+c4799753c up
INFO Waiting up to 30m0s for the bootstrap-complete event...
```

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。

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。

1.3.11. 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 クラスターのデプロイ。
- **oc** CLI のインストール。

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
```

1.3.12. 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

- マシンをクラスターに追加していること。

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  v1.14.6+c4799753c
master-1  Ready     master  63m  v1.14.6+c4799753c
master-2  Ready     master  64m  v1.14.6+c4799753c
worker-0  NotReady  worker  76s  v1.14.6+c4799753c
worker-1  NotReady  worker  70s  v1.14.6+c4799753c
```

   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。

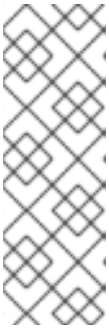
```
$ oc get csr

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
```

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ること。

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             4.2.0     True        False         False      69s
cloud-credential           4.2.0     True        False         False      12m
cluster-autoscaler         4.2.0     True        False         False      11m
console                    4.2.0     True        False         False      46s
dns                        4.2.0     True        False         False      11m
image-registry             4.2.0     False       True          False      5m26s
ingress                    4.2.0     True        False         False      5m36s
kube-apiserver             4.2.0     True        False         False      8m53s
kube-controller-manager    4.2.0     True        False         False      7m24s
kube-scheduler             4.2.0     True        False         False      12m
machine-api                4.2.0     True        False         False      12m
machine-config             4.2.0     True        False         False      7m36s
marketplace                4.2.0     True        False         False      7m54m
monitoring                 4.2.0     True        False         False      7h54s
```

```
network                                    4.2.0     True        False         False      5m9s
node-tuning                                4.2.0     True        False         False      11m
openshift-apiserver                        4.2.0     True        False         False      11m
openshift-controller-manager               4.2.0     True        False         False      5m943s
openshift-samples                          4.2.0     True        False         False      3m55s
operator-lifecycle-manager                 4.2.0     True        False         False      11m
operator-lifecycle-manager-catalog         4.2.0     True        False         False      11m
service-ca                                 4.2.0     True        False         False      11m
service-catalog-apiserver                  4.2.0     True        False         False      5m26s
service-catalog-controller-manager         4.2.0     True        False         False      5m25s
storage                                    4.2.0     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 が利用できない場合、そのストレージを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. VMware vSphere 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。
- VMware vSphere 上のクラスター。
- **ReadWriteMany** アクセスモードのプロビジョニングされた永続ボリューム (PV)(例: **NFS**)。

重要

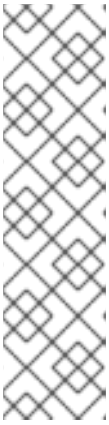
vSphere ボリュームは **ReadWriteMany** アクセスモードをサポートしません。レジストリーストレージを設定するには、**NFS**などの異なるストレージバックエンドを使用する必要があります。

- 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。
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** で、レジストリー Pod を **replica>1** を設定してスケールアップする必要がある場合、**no_wdelay** 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,no_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. オプション: 新しいストレージクラスを PV に追加します。

   a. PV を作成します。

```
$ oc create -f -
```

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: image-registry-pv
spec:
  accessModes:
    ReadWriteMany
    capacity:
      storage: 100Gi
  nfs:
    path: /registry
    server: 172.16.231.181
  persistentVolumeReclaimPolicy: Retain
  storageClassName: nfs01
```

```
$ oc get pv
```

   b. PVC を作成します。

```
$ oc create -n openshift-image-registry -f -
```

```
apiVersion: "v1"
kind: "PersistentVolumeClaim"
metadata:
  name: "image-registry-pvc"
```

```
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: nfs01
  volumeMode: Filesystem
```

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

最後に、PVC の名前を追加します。

```
$ oc edit configs.imageregistry.operator.openshift.io -o yaml
```

```
storage:
  pvc:
    claim: image-registry-pvc 1
```

1 カスタム PVC を作成すると、**image-registry-storage** PVC のデフォルトの自動作成の **claim** フィールドを空のままにすることができます。

5. **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ること。
- Operator の初期設定を完了していること。

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                       4.2.0     True        False         False      10m
cloud-credential                     4.2.0     True        False         False      22m
cluster-autoscaler                   4.2.0     True        False         False      21m
console                              4.2.0     True        False         False      10m
dns                                  4.2.0     True        False         False      21m
image-registry                       4.2.0     True        False         False      16m
ingress                              4.2.0     True        False         False      16m
kube-apiserver                       4.2.0     True        False         False      19m
kube-controller-manager              4.2.0     True        False         False      18m
kube-scheduler                       4.2.0     True        False         False      22m
machine-api                          4.2.0     True        False         False      22m
machine-config                       4.2.0     True        False         False      18m
marketplace                          4.2.0     True        False         False      18m
monitoring                           4.2.0     True        False         False      18m
network                              4.2.0     True        False         False      16m
node-tuning                          4.2.0     True        False         False      21m
openshift-apiserver                  4.2.0     True        False         False      21m
openshift-controller-manager         4.2.0     True        False         False      17m
openshift-samples                    4.2.0     True        False         False      14m
operator-lifecycle-manager           4.2.0     True        False         False      21m
operator-lifecycle-manager-catalog   4.2.0     True        False         False      21m
service-ca                           4.2.0     True        False         False      21m
service-catalog-apiserver            4.2.0     True        False         False      16m
service-catalog-controller-manager   4.2.0     True        False         False      16m
storage                              4.2.0     True        False         False      16m
```

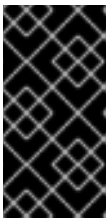
すべてのクラスター Operator が **AVAILABLE** の場合、インストールを完了することができます。

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す。

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

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。