OpenShift Container Platform 4.3

IBM Z および LinuxONE へのインストール

OpenShift Container Platform 4.3 IBM Z および LinuxONE クラスターのインストール

Last Updated: 2020-09-18

OpenShift Container Platform 4.3 IBM Z および LinuxONE へのインストール

OpenShift Container Platform 4.3 IBM Z および LinuxONE クラスターのインストール

法律上の通知

概要

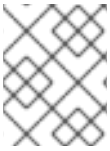
本書では、IBM Z および LinuxONE に OpenShift Container Platform 4.3 クラスターをインストールする方法について説明します。

目次

第1章 IBM Z および LINUXONE へのインストール

1.1. クラスターの IBM Z および LINUXONE へのインストール

OpenShift Container Platform version 4.3 では、プロビジョニングする IBM Z または LinuxONE 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。



注記

本書は IBM Z のみを参照しますが、これに含まれるすべての情報は LinuxONE にも適用されます。

重要

ベアメタルプラットフォーム以外の場合には、追加の考慮点を検討する必要があります。OpenShift Container Platform クラスターをインストールする前に、「guidelines for deploying OpenShift Container Platform on non-tested platforms」にある情報を確認してください。

1.1.1. 前提条件

- クラスターの NFS を使用する永続ストレージをプロビジョニングします。プライベートイメージレジストリーをデプロイするには、ストレージで ReadWriteMany アクセスモードを指定する必要があります。
- OpenShift Container Platform のインストールおよび更新 プロセスについての詳細を確認します。
- ファイアウォールを使用する場合、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。

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

OpenShift Container Platform 4.3 では、クラスターをインストールするためにインターネットアクセスが必要になります。クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは Red Hat OpenShift Cluster Manager (OCM) に登録されます。

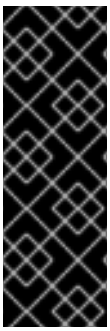
Red Hat OpenShift Cluster Manager インベントリーが Telemetry によって自動的に維持されるか、または OCM を手動で使用しているかのいずれによって正常であることを確認した後に、subscription watch を使用して、アカウントまたはマルチクラスターレベルで OpenShift Container Platform サブスクリプションを追跡します。

インターネットへのアクセスは以下を実行するために必要です。

- Red Hat OpenShift Cluster Manager ページにアクセスし、インストールプログラムをダウンロードし、サブスクリプション管理を実行します。クラスターにインターネットアクセスがあ

り、Telemetry を無効にしない場合、そのサービスは有効なサブスクリプションでクラスターを自動的に使用します。

- クラスターのインストールに必要なパッケージを取得するために Quay.io にアクセスします。
- クラスターの更新を実行するために必要なパッケージを取得します。



重要

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

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

ユーザーによってプロビジョニングされるインフラストラクチャーを含むクラスターの場合、必要なマシンすべてをデプロイする必要があります。

1.1.3.1. 必要なマシン

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

- 1 つの一時的なブートストラップマシン
- 3 つのコントロールプレーン、またはマスター、マシン
- 少なくとも 2 つのコンピュートマシン (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改善するには、2 つ以上の物理マシンの複数の異なる z/VM インスタンスにコントロールプレーンマシンを分散します。

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。マシンは静的 IP アドレスで設定されます。DHCP サーバーは必要ありません。

1.1.3.3. IBM Z ネットワーク接続の要件

IBM Z の z/VM でインストールするには、 レイヤー 2 モードの単一 z/VM 仮想 NIC が必要になります。以下も必要になります。

- 直接接続された OSA または RoCE ネットワークアダプター
- z/VM VSWITCH のセットアップ。推奨されるセットアップでは、OSA リンクアグリゲーションを使用します。

1.1.3.4. 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	RHCOS または RHEL 7.6	2	8 GB

1.1.3.5. 最小の IBM Z システム要件

OpenShift Container Platform バージョン 4.3 は、以下の IBM ハードウェアにインストールできます。

- IBM Z: z13、z13s、すべての z14 モデル、すべての z15 モデル
- LinuxONE: すべてのモデル

ハードウェア要件

- SMT2 をサポートする 3 IFL 搭載の 1 LPAR
- 1 OSA または RoCE ネットワークアダプター

オペレーティングシステム要件

- z/VM 7.1 の 1 インスタンス

z/VM インスタンスで以下をセットアップします。

- OpenShift Container Platform コントロールプレーンマシンの 3 ゲスト仮想マシン
- OpenShift Container Platform コンピュートマシンの 2 ゲスト仮想マシン
- 一時 OpenShift Container Platform ブートストラップマシンの 1 ゲスト仮想マシン

z/VM ゲスト仮想マシンのディスクストレージ

- FICON 接続のディスクストレージ (DASD)これらには z/VM ミニディスク、フルパックミニディスク、または専用の DASD を使用できます。Red Hat Enterprise Linux CoreOS (RHCOS) インストールに必要な最低限の DASD サイズに達するには、拡張アドレスボリューム (EAV) が必要です。利用可能な場合は、HyperPAV を使用して最適なパフォーマンスを確保します。
- FCP 接続のディスクストレージ

ストレージ/メインメモリー

- OpenShift Container Platform コントロールプレーンマシン用に 16 GB
- OpenShift Container Platform コンピュートマシン用に 8 GB
- 一時 OpenShift Container Platform ブートストラップマシン用に 16 GB

1.1.3.6. 推奨される IBM Z システム要件

ハードウェア要件

- SMT2 をサポートする 6 IFL 搭載の 3 LPAR
- 1 または 2 OSA または RoCE ネットワークアダプター、またはその両方
- HiperSockets。ノードに直接割り当てられるか、または z/VM ゲストに対して透過性を持たせるために z/VM VSWITCH でブリッジしてノードに割り当てられます。Hipersockets をノードに直接接続するには、RHEL 8 ゲスト経由で外部ネットワークにゲートウェイを設定し、Hipersockets ネットワークにブリッジする必要があります。

オペレーティングシステム要件

- 高可用性を確保する場合は z/VM 7.1 の 2 または 3 インスタンス

z/VM インスタンスで以下を設定します。

- OpenShift Container Platform コントロールプレーンマシン用に 3 ゲスト仮想マシン (z/VM インスタンスごとに 1 つ)
- OpenShift Container Platform コンピュートマシン用に 6 以上のゲスト仮想マシン (z/VM インスタンス全体に分散)
- 一時 OpenShift Container Platform ブートストラップマシンの 1 ゲスト仮想マシン

z/VM ゲスト仮想マシンのディスクストレージ

- FICON 接続のディスクストレージ (DASD)これらには z/VM ミニディスク、フルパックミニディスク、または専用の DASD を使用できます。Red Hat Enterprise Linux CoreOS (RHCOS) インストールに必要な最低限の DASD サイズに達するには、拡張アドレスボリューム (EAV) が必要です。利用可能な場合は、HyperPAV および High Performance FICON (zHPF) を使用して最適なパフォーマンスを確保します。
- FCP 接続のディスクストレージ

ストレージ/メインメモリー

- OpenShift Container Platform コントロールプレーンマシン用に 16 GB

- OpenShift Container Platform コンピュートマシン用に 8 GB
- 一時 OpenShift Container Platform ブートストラップマシン用に 16 GB

1.1.3.7. 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。

追加リソース

- IBM Knowledge Center の「Bridging a HiperSockets LAN with a z/VM Virtual Switch 」を参照してください。
- パフォーマンスの最適化については、「Scaling HyperPAV alias devices on Linux guests on z/VM」を参照してください。

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. 静的 IP アドレスをセットアップします。
2. FTP サーバーをセットアップします。
3. 必要なロードバランサーをプロビジョニングします。
4. マシンのポートを設定します。
5. DNS を設定します。
6. 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 FTP サーバーが必要になります。

マシンに永続 IP アドレスおよびホスト名があることを確認します。

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

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

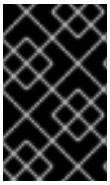
表1.1 すべてのマシンに対応するすべてのマシン

プロトコル	ポート	説明
ICMP	該当なし	ネットワーク到達性のテスト
TCP	**9000**-**9999**	ホストレベルのサービス。 ポート **9100**-**9101** のノードエクスポーター、ポート **9099** の Cluster Version Operator が含まれます。
	10250-**10259**	Kubernetes が予約するデフォルトポート
	10256	openshift-sdn
UDP	**4789**	VXLAN および GENEVE
	6081	VXLAN および GENEVE
	9000-**9999**	ポート **9100**-**9101** のノードエクスポーターを含む、ホストレベルのサービス。
TCP/UDP	**30000**-**32767**	Kubernetes NodePort

表1.2 コントロールプレーンへのすべてのマシン

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

ネットワークトポロジー要件
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。



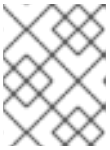
重要

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、以下の要件を満たす 2 つのロードバランサーをプロビジョニングする必要があります。

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

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

   注記

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

   ロードバランサーのフロントとバックの両方で以下のポートを設定します。

   表1.3 API ロードバランサー

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

   注記

   ロードバランサーは、API サーバーが **/readyz** エンドポイントをオフにしてからプールから API サーバーインスタンスを削除するまで最大 30 秒かかるように設定する必要があります。**/readyz** の後の時間枠内でエラーが返されたり、正常になったりする場合、エンドポイントは削除または追加されている必要があります。5 秒または 10 秒ごとにプローブし、2 つの正常な要求が正常な状態になり、3 つの要求が正常な状態になりません。これらは十分にテストされた値です。

2. **Application Ingress ロードバランサー**: クラスター外から送られるアプリケーショントラフィックの Ingress ポイントを提供します。以下の条件を設定します。

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

ロードバランサーのフロントとバックの両方で以下のポートを設定します。

表1.4 アプリケーション Ingress ロードバランサー

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

ヒント

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

注記

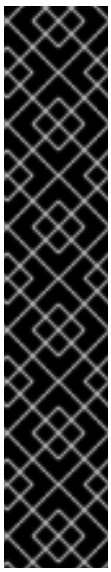
Ingress ルーターの作業用の設定が OpenShift Container Platform クラスターに必要です。コントロールプレーンの初期化後に Ingress ルーターを設定する必要があります。

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

以下の DNS レコードは、ユーザーによってプロビジョニングされるインフラストラクチャーを使用する OpenShift Container Platform クラスターに必要です。各レコードで、 **<cluster_name>** はクラスター名で、**<base_domain>** は、**install-config.yaml** ファイルに指定するクラスターのベースドメインです。完全な DNS レコードは **<component>.<cluster_name>.<base_domain>.** の形式を取ります。

表1.5 必要な DNS レコード

コンポーネント	レコード	説明
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。

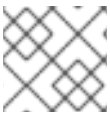
コンポーネント	レコード	説明
	api-int.<cluster_name>.<base_domain>.	この DNS A/AAAA または CNAME レコードは、コントロールプレーンマシンのロードバランサーを参照する必要があります。このレコードは、クラスター内のすべてのノードで解決できる必要があります。  **重要** API サーバーは、Kubernetes に記録されるホスト名でワーカーノードを解決できる必要があります。これがノード名を解決できない場合、プロキシーされる API 呼び出しが失敗し、Pod からログを取得できなくなる可能性があります。
Routes	***.apps.<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。

コンポーネント	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>`

コンポーネント	レコード	説明

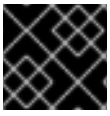
1.1.5. SSH プライベートキーの生成およびエージェントへの追加

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。



重要

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

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

注記

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

手順

1. パスワードなしの認証に設定されている SSH キーがコンピューター上にない場合は、これを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ ssh-keygen -t rsa -b 4096 -N '' \
    -f <path>/<file_name> 1
```

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

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

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

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

Agent pid 31874
```

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

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

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

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

次のステップ

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

1.1.6. 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プロビジョニングマシンにダウンロードします。

前提条件

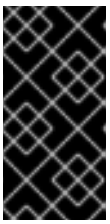
- Linux を実行するマシンからクラスターをインストールする必要があります (例: Red Hat Enterprise Linux 8)。

- インストールプログラムをダウンロードするには、500 MB のローカルディスク領域が必要です。

手順

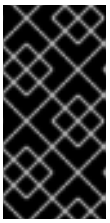
1. Red Hat OpenShift Cluster Manager サイトの「Infrastructure Provider」ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使ってログインします。アカウントがない場合はこれを作成します。

2. 選択するインストールタイプのページに移動し、オペレーティングシステムのインストールプログラムをダウンロードし、ファイルをインストール設定ファイルを保存するディレクトリーに配置します。



重要

インストールプログラムは、クラスターのインストールに使用するコンピューターにいくつかのファイルを作成します。クラスターインストールの完了後は、インストールプログラムおよびインストールプログラムが作成するファイルの両方を保持する必要があります。



重要

インストールプログラムで作成されたファイルを削除しても、クラスターがインストール時に失敗した場合でもクラスターは削除されません。特定のクラウドプロバイダー用に記載された OpenShift Container Platform のアンインストール手順を完了して、クラスターを完全に削除する必要があります。

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

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

4. Red Hat OpenShift Cluster Manager サイトの「Pull Secret」ページから、インストールプルシークレットを **.txt** ファイルとしてダウンロードします。このプルシークレットを使用し、OpenShift Container Platform コンポーネントのコンテナ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. バイナリーのダウンロードによる CLI のインストール

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

重要

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.3 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールします。

1.1.7.1. Linux への CLI のインストール

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

手順

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

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

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

```
$ echo $PATH
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.1.7.2. Windows での CLI のインストール

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

手順

1. Red Hat OpenShift Cluster Manager サイトの Infrastructure Provider ページに移動します。
2. インフラストラクチャープロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **Windows** を選択し、**Download command-line tools** をクリックします。
4. ZIP プログラムでアーカイブを解凍します。
5. **oc** バイナリーを、**PATH** にあるディレクトリーに移動します。
PATH を確認するには、コマンドプロンプトを開いて以下のコマンドを実行します。

```
C:\> path
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

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

1.1.7.3. macOS への CLI のインストール

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

手順

1. Red Hat OpenShift Cluster Manager サイトの Infrastructure Provider ページに移動します。
2. インフラストラクチャープロバイダーを選択し、(該当する場合は) インストールタイプを選択します。
3. **Command-line interface** セクションで、ドロップダウンメニューの **MacOS** を選択し、**Download command-line tools** をクリックします。
4. アーカイブを展開し、解凍します。
5. **oc** バイナリーをパスにあるディレクトリーに移動します。
PATH を確認するには、ターミナルを開き、以下のコマンドを実行します。

```
$ echo $PATH
```

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

1.1.8. インストール設定ファイルの手動作成

ユーザーによってプロビジョニングされるインフラストラクチャーを使用する OpenShift Container Platform のインストールでは、インストール設定ファイルを手動で生成する必要があります。

前提条件

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

手順

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

```
$ mkdir <installation_directory>
```

重要

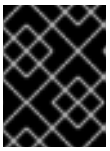
ディレクトリーを作成する必要があります。ブートストラップ X.509 証明書などの一部のインストールアセットの有効期限は短く設定されているため、インストールディレクトリーを再利用することができません。別のクラスターインストールの個別のファイルを再利用する必要がある場合は、それらをディレクトリーにコピーすることができます。ただし、インストールアセットのファイル名はリリース間で変更される可能性があります。インストールファイルを以前のバージョンの OpenShift Container Platform からコピーする場合は注意してコピーを行ってください。

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

注記

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

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



重要

install-config.yaml ファイルは、インストールプロセスの次の手順で使用されます。この時点でこれをバックアップする必要があります。

1.1.8.1. ベアメタルのサンプル install-config.yaml ファイル

1.1.8.2. IBM Z のサンプル install-config.yaml ファイル

install-config.yaml ファイルをカスタマイズして、OpenShift Container Platform クラスターのプラットフォームについての詳細を指定するか、または必要なパラメーターの値を変更することができます。

```
apiVersion: v1
baseDomain: example.com (1)
compute:
- hyperthreading: Enabled (2) (3)
  name: worker
  replicas: 0 (4)
controlPlane:
  hyperthreading: Enabled (5) (6)
  name: master (7)
  replicas: 3 (8)
metadata:
  name: test (9)
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14 (10)
    hostPrefix: 23 (11)
  networkType: OpenShiftSDN
  serviceNetwork: (12)
  - 172.30.0.0/16
platform:
  none: {} (13)
fips: false (14)
pullSecret: '{"auths": ...}' (15)
sshKey: 'ssh-ed25519 AAAA...' (16)
```

(1) クラスターのベースドメイン。すべての DNS レコードはこのベースのサブドメインである必要があり、クラスター名が含まれる必要があります。

(2) (5) **controlPlane** セクションは単一マッピングですが、コンピュートセクションはマッピングのシーケンスになります。複数の異なるデータ構造の要件を満たすには、 **compute** セクションの最初の行はハイフン **-** で始め、**controlPlane** セクションの最初の行はハイフンで始めることができません。どちらのセクションも、現時点では単一のマシンプールを定義しますが、OpenShift Container Platform の今後のバージョンでは、インストール時の複数のコンピュートプールの定義をサポートする可能性があります。1 つのコントロールプレーンプールのみが使用されます。

(3) (6) (7) 同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。パラメーター値を **Disabled** に設定するとこれを無効にすることができます。一部のクラスターマシンで同時マル

チスレッドを無効にする場合は、これをすべてのクラスターマシンで無効にする必要があります。

重要

同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。

4 **replicas** パラメーターの値を **0** に設定する必要があります。このパラメーターはクラスターが作成し、管理するワーカーの数を制御します。これは、ユーザーによってプロビジョニングされるインフラストラクチャーを使用する場合にクラスターが実行しない機能です。OpenShift Container Platform のインストールが終了する前に、クラスターが使用するワーカーマシンを手動でデプロイする必要があります。

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

9 DNS レコードに指定したクラスター名。

10 Pod IP アドレスの割り当てに使用する IP アドレスのブロック。このブロックは既存の物理ネットワークと重複できません。これらの IP アドレスは Pod ネットワークに使用されます。外部ネットワークから Pod にアクセスする必要がある場合、ロードバランサーおよびルーターを、トラフィックを管理するように設定する必要があります。

11 それぞれの個別ノードに割り当てるサブネットプレフィックスの長さ。たとえば、**hostPrefix** が **23** に設定され、各ノードに指定の **cidr** から **/23** サブネットが割り当てられます (510 (2^(32 - 23) - 2) Pod IP アドレスが許可されます)。外部ネットワークからのノードへのアクセスを提供する必要がある場合には、ロードバランサーおよびルーターを、トラフィックを管理するように設定します。

12 サービス IP アドレスに使用する IP アドレスプール。1 つの IP アドレスプールのみを入力できます。外部ネットワークからサービスにアクセスする必要がある場合、ロードバランサーおよびルーターを、トラフィックを管理するように設定します。

13 プラットフォームを **none**に設定する必要があります。ベアメタル IBM Z インフラストラクチャー用に追加のプラットフォーム設定変数を指定することはできません。

14 FIPS モードを有効または無効にするかどうか。デフォルトでは、FIPS モードは有効にされません。FIPS モードが有効にされている場合、OpenShift Container Platform が実行される Red Hat Enterprise Linux CoreOS (RHCOS) マシンがデフォルトの Kubernetes 暗号スイートをバイパスし、代わりに RHCOS で提供される暗号モジュールを使用します。

15 pullSecret の値には、レジストリーの認証情報が含まれます。<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。このプルシークレットを使用し、OpenShift Container Platform コンポーネントのコンテナーイメージを提供する Quay.io など、組み込まれた各種の認証局によって提供されるサービスで認証できます。

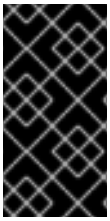
16 Red Hat Enterprise Linux CoreOS (RHCOS) の **core** ユーザーのデフォルト SSH キーの公開部分。

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。

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

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



重要

インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになる証明書が含まれます。最初の証明書のローテーションが正常に実行されるようにするには、クラスターのインストールを完了し、クラスターを動作が低下していない状態で 24 時間実行し続ける必要があります。

前提条件

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

手順

1. クラスターの Kubernetes マニフェストを生成します。

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

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

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

インストールプロセスの後の部分で独自のコンピュー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. **<installation_directory>/manifests/cluster-scheduler-02-config.yml** Kubernetes マニフェストファイルを変更し、Pod がコントロールプレーンマシンにスケジュールされないようにします。

   a. **<installation_directory>/manifests/cluster-scheduler-02-config.yml** ファイルを開きます。

   b. **mastersSchedulable** パラメーターを見つけ、その値を **False** に設定します。

   c. ファイルを保存し、終了します。

注記

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

3. Ignition 設定ファイルを取得します。

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

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

以下のファイルはディレクトリーに生成されます。

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

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

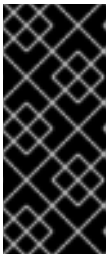
プロビジョニングする IBM Z インフラストラクチャーにクラスターをインストールする前に、クラスターが使用する RHCOS を z/VM ゲスト仮想マシンにインストールする必要があります。マシンを作成するには、以下の手順を実行します。

前提条件

- 作成するマシンがアクセスできるプロビジョニングマシンで稼働している FTP サーバー。

手順

1. プロビジョニングマシンで Linux にログインします。
2. RHCOS イメージミラーから Red Hat Enterprise Linux CoreOS インストールファイルをダウンロードします。



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。

以下のファイルをダウンロードします。

- initramfs: **rhcos-<version>-installer-initramfs.img**
- kernel: **rhcos-<version>-installer-kernel**
- RHCOS をインストールするディスクのオペレーティングシステムイメージ。このタイプは仮想マシンによって異なる場合があります。
  rhcos-<version>-s390x-dasd.s390x.raw.gz (DASD 用)

  rhcos-<version>-s390x-metal.s390x.raw.gz (FCP 用)

3. パラメーターファイルを作成します。以下のパラメーターは特定の仮想マシンに固有のものです。

   - **coreos.inst.install_dev=** の場合、DASD インストールに **dasda** を指定するか、または FCP に **sda** を指定します。FCP には **zfcp.allow_lun_scan=0** が必要なことに注意してください。
   - **rd.dasd=** の場合、RHCOS がインストールされる DASD を指定します。
   - **rd.zfcp=<adapter>,<wwpn>,<lun>** は、RHCOS をインストールする FCP ディスクを指定します。
   - **ip=**には、以下の 7 つのエントリーを指定します。
     i. マシンの IP アドレス。
     ii. 空の文字列。
     iii. ゲートウェイ。
     iv. ネットマスク。
     v. **hostname.domainname** 形式のマシンホストおよびドメイン名。RHCOS に決定し、設定させる場合は、この値を省略します。
     vi. ネットワークインターフェース名。RHCOS に決定し、設定させる場合は、この値を省略します。
     vii. 静的 IP アドレスを使用する場合、空の文字列になります。
   - **coreos.inst.ignition_url=** の場合、マシンロールの Ignition ファイルを指定します。**bootstrap.ign**、**master.ign**、または **worker.ign** を使用します。
   - その他のパラメーターはそのまま利用できます。
     ブートストラップマシンのパラメーターファイルのサンプル **bootstrap-0.parm**:

     ```
     rd.neednet=1 coreos.inst=yes coreos.inst.install_dev=dasda coreos.inst.image_url=ftp://
     cl1.provide.example.com:8080/assets/rhcos-43.80.20200430.0-s390x-dasd.390x.raw.gz
     coreos.inst.ignition_url=ftp://cl1.provide.example.com:8080/ignition-bootstrap-0
     ip=172.18.78.2::172.18.78.1:255.255.255.0:::none nameserver=172.18.78.1
     rd.znet=qeth,0.0.bdf0,0.0.bdf1,0.0.bdf2,layer2=1,portno=0 zfcp.allow_lun_scan=0
     cio_ignore=all,
     !condev rd.dasd=0.0.3490
     ```

4. FTP などを使用し、initramfs、kernel、パラメーターファイル、および RHCOS イメージを z/VM に転送します。FTP でファイルを転送し、仮想リーダーから起動する方法については、「Z/VM 環境へのインストール」を参照してください。

5. ブートストラップノードになる z/VM ゲスト仮想マシンの仮想リーダーに対してファイルの punch を実行します。
   IBM Knowledge Center で PUNCH を参照してください。

   ヒント

   CP PUNCH コマンドを使用するか、Linux を使用している場合は、 **vmur** コマンドを使用して 2 つの z/VM ゲスト仮想マシン間でファイルを転送できます。

6. ブートストラップマシンで CMS にログインします。
7. リーダーからブートストラップマシンに対して IPL を実行します。

```
$ ipl c
```

IBM Knowledge Center で IPL を参照してください。

8. クラスター内の他のマシンについてこの手順を繰り返します。

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。

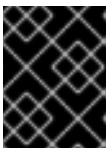
```
$ ./openshift-install --dir=<installation_directory> wait-for bootstrap-complete \ 1
    --log-level=info 2
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com...
INFO API v1.16.2 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove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

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

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

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

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



重要

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

1.1.12. クラスターへのログイン

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

前提条件

- OpenShift Container Platform クラスターのデプロイ。
- **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-01.example.com   Ready    master   40d   v1.16.2
master-02.example.com   Ready    master   40d   v1.16.2
master-03.example.com   Ready    master   40d   v1.16.2
worker-01.example.com   Ready    worker   40d   v1.16.2
worker-02.example.com   Ready    worker   40d   v1.16.2
```

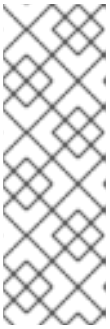
出力には作成したすべてのマシンが一覧表示されます。

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

```
$ oc get csr

NAME        AGE   REQUESTOR                                 CONDITION
csr-mddf5   20m   system:node:master-01.example.com   Approved,Issued
csr-z5rln   16m   system:node:worker-21.example.com   Approved,Issued
```

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             4.3.0         True        False       False     40d
cloud-credential           4.3.0         True        False       False     40d
```

```
cluster-autoscaler                         4.3.0       True        False         False      40d
console                                    4.3.0       True        False         False      6d2h
dns                                        4.3.0       True        False         False      22d
image-registry                             4.3.0       True        False         False      6d2h
ingress                                    4.3.0       True        False         False      6d2h
insights                                   4.3.0       True        False         False      40d
kube-apiserver                             4.3.0       True        False         False      40d
kube-controller-manager                    4.3.0       True        False         False      40d
kube-scheduler                             4.3.0       True        False         False      40d
machine-api                                4.3.0       True        False         False      40d
machine-config                             4.3.0       True        False         False      22d
marketplace                                4.3.0       True        False         False      40d
monitoring                                 4.3.0       True        False         False      6d16h
network                                    4.3.0       True        False         False      40d
node-tuning                                4.3.0       True        False         False      22d
openshift-apiserver                        4.3.0       True        False         False      22d
openshift-controller-manager               4.3.0       True        False         False      6d2h
openshift-samples                          4.3.0       True        False         False      40d
operator-lifecycle-manager                 4.3.0       True        False         False      40d
operator-lifecycle-manager-catalog         4.3.0       True        False         False      40d
operator-lifecycle-manager-packageserver   4.3.0       True        False         False      6d2h
service-ca                                 4.3.0       True        False         False      40d
service-catalog-apiserver                  4.3.0       True        False         False      40d
service-catalog-controller-manager         4.3.0       True        False         False      40d
storage                                    4.3.0       True        False         False      40d
```

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

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

1.1.14.1.1. ベアメタルの場合のレジストリーストレージの設定

1.1.14.1.2. IBM Z の場合のレジストリーストレージの設定

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

前提条件

- クラスター管理者のパーミッション。
- ベアメタル上のクラスター。IBM Z
- NFS などのクラスターの永続ストレージをプロビジョニングします。プライベートイメージレジストリーをデプロイするには、ストレージで ReadWriteMany アクセスモードを指定する必要があります。
- 容量は「100Gi」以上である。

手順

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

注記

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

2. レジストリー Pod がないことを確認します。

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

注記

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

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

3. レジストリー設定を確認します。

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

storage:
  pvc:
    claim:
```

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

4. **clusteroperator** ステータスを確認します。

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

1.1.14.1.3. 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             4.3.0     True        False         False      10m
cloud-credential           4.3.0     True        False         False      22m
cluster-autoscaler         4.3.0     True        False         False      21m
console                    4.3.0     True        False         False      10m
dns                        4.3.0     True        False         False      21m
image-registry             4.3.0     True        False         False      16m
ingress                    4.3.0     True        False         False      16m
kube-apiserver             4.3.0     True        False         False      19m
kube-controller-manager    4.3.0     True        False         False      18m
kube-scheduler             4.3.0     True        False         False      22m
machine-api                4.3.0     True        False         False      22m
machine-config             4.3.0     True        False         False      18m
marketplace                4.3.0     True        False         False      18m
monitoring                 4.3.0     True        False         False      18m
network                    4.3.0     True        False         False      16m
```

```
node-tuning                 4.3.0     True        False         False      21m
openshift-apiserver         4.3.0     True        False         False      21m
openshift-controller-manager 4.3.0    True        False         False      17m
openshift-samples           4.3.0     True        False         False      14m
operator-lifecycle-manager  4.3.0     True        False         False      21m
operator-lifecycle-manager-catalog 4.3.0 True     False         False      21m
service-ca                  4.3.0     True        False         False      21m
service-catalog-apiserver   4.3.0     True        False         False      16m
service-catalog-controller-manager 4.3.0 True     False         False      16m
storage                     4.3.0     True        False         False      16m
```

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

2. クラスターの完了をモニターします。

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

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

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

重要

インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになる証明書が含まれます。最初の証明書のローテーションが正常に実行されるようにするには、クラスターを動作が低下していない状態で 24 時間実行し続ける必要があります。

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

   a. すべての Pod の一覧を表示するには、以下のコマンドを使用します。

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

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

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

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

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

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

1.1.16. デバッグ情報の収集

IBM Z での OpenShift Container Platform インストールに関する特定の問題のトラブルシューティングおよびデバッグに役立つ可能性のあるデバッグ情報を収集できます。

前提条件

- **oc** CLI ツールをインストールしていること。

手順

1. クラスターにログインします。

```
$ oc login
```

2. ハードウェア情報を収集するノードで、デバッグコンテナーを起動します。

```
$ oc debug node/<nodename>
```

3. **/host** ファイルシステムに切り替え、**toolbox** を起動します。

```
$ chroot /host
$ toolbox
```

4. **dbginfo** データを収集します。

```
$ dbginfo.sh
```

5. その後に、**scp** を使用するなどしてデータを取得できます。

追加リソース

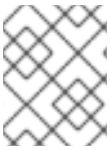
- 「How to generate SOSREPORT within OpenShift4 nodes without SSH 」も参照してください。

1.1.17. 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。

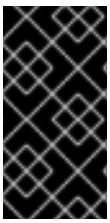
1.2. ネットワークが制限された環境でのクラスターの IBM Z および LINUXONE へのインストール

OpenShift Container Platform バージョン 4.3 では、クラスターを制限されたネットワークでプロビジョニングする IBM Z および LinuxONE 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。



注記

本書は IBM Z のみを参照しますが、これに含まれるすべての情報は LinuxONE にも適用されます。

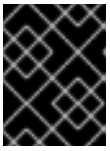


重要

ベアメタルプラットフォーム以外の場合には、追加の考慮点を検討する必要があります。OpenShift Container Platform クラスターをインストールする前に、「guidelines for deploying OpenShift Container Platform on non-tested platforms」にある情報を確認してください。

1.2.1. 前提条件

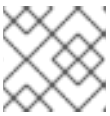
- bastion ホストでミラーレジストリーを作成し、OpenShift Container Platform の使用しているバージョン用の **imageContentSources** データを取得します。
- 非接続インストールプロセスを開始する前に、既存の **/usr/local/bin/openshift-install** bastion ファイルを移動するか、または削除する必要があります。これにより、更新された **/usr/local/bin/openshift-install** ファイルが非接続インストールのプロセスで作成されるようになります。

  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 NFS を使用して 永続ストレージをプロビジョニングします。プライベートイメージレジストリーをデプロイするには、ストレージで ReadWriteMany アクセスモードを指定する必要があります。
- OpenShift Container Platform のインストールおよび更新 プロセスについての詳細を確認します。
- ファイアウォールを使用し、Telemetry を使用する予定がある場合は、クラスターがアクセスする必要のあるサイトを許可するようにファイアウォールを設定する必要があります。

  注記

  プロキシ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。

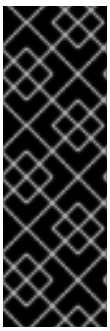
1.2.2. ネットワークが制限された環境でのインストールについて

OpenShift Container Platform 4.3 では、ソフトウェアコンポーネントを取得するためにインターネットへのアクティブな接続を必要としないインストールを実行できます。インストールプログラムでプロビジョニングされるインフラストラクチャーではなく、ユーザーによってプロビジョニングされるインフラストラクチャー上でのみネットワークが制限された環境でのインストールを実行します。そのため、プラットフォームの選択は制限されます。

クラウドプラットフォーム上でネットワークが制限されたインストールの実行を選択した場合でも、そ

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

ネットワークが制限されたインストールを完了するには、OpenShift Container Platform レジストリーのコンテンツをミラーリングし、インストールメディアを含むレジストリーを作成する必要があります。このミラーは、インターネットと制限されたネットワークの両方にアクセスできるミラーホストで、または制限に対応する他の方法を使用して作成できます。



重要

ネットワークが制限されたインストールはユーザーによってプロビジョニングされるインフラストラクチャーを常に使用します。ユーザーによってプロビジョニングされるインストールの設定は複雑であるため、ネットワークが制限されたインストールを試行する前に、標準的なユーザーによってプロビジョニングされるインフラストラクチャーを実行することを検討してください。このテストが完了すると、ネットワークが制限されたインストール時に発生する可能性のある問題の切り分けやトラブルシューティングがより容易になります。

1.2.2.1. その他の制限

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

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

1.2.3. 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。

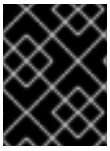
1.2.3.1. 必要なマシン

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改善するには、2 つ以上の物理マシンの複数の異なる z/VM インスタンスにコントロールプレーンマシンを分散します。

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。マシンは静的 IP アドレスで設定されます。DHCP サーバーは必要ありません。

1.2.3.3. IBM Z ネットワーク接続の要件

IBM Z の z/VM でインストールするには、 レイヤー 2 モードの単一 z/VM 仮想 NIC が必要になります。以下も必要になります。

- 直接接続された OSA または RoCE ネットワークアダプター
- z/VM VSWITCH のセットアップ。推奨されるセットアップでは、OSA リンクアグリゲーションを使用します。

1.2.3.4. 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	RHCOS または RHEL 7.6	2	8 GB

1.2.3.5. 最小の IBM Z システム要件

OpenShift Container Platform バージョン 4.3 は、以下の IBM ハードウェアにインストールできます。

- IBM Z: z13、z13s、すべての z14 モデル、すべての z15 モデル
- LinuxONE: すべてのモデル

ハードウェア要件

- SMT2 をサポートする 3 IFL 搭載の 1 LPAR

- 1 OSA または RoCE ネットワークアダプター

オペレーティングシステム要件

- z/VM 7.1 の 1 インスタンス

z/VM インスタンスで以下をセットアップします。

- OpenShift Container Platform コントロールプレーンマシンの 3 ゲスト仮想マシン
- OpenShift Container Platform コンピュートマシンの 2 ゲスト仮想マシン
- 一時 OpenShift Container Platform ブートストラップマシンの 1 ゲスト仮想マシン

z/VM ゲスト仮想マシンのディスクストレージ

- FICON 接続のディスクストレージ (DASD)これらには z/VM ミニディスク、フルパックミニディスク、または専用の DASD を使用できます。Red Hat Enterprise Linux CoreOS (RHCOS) インストールに必要な最低限の DASD サイズに達するには、拡張アドレスボリューム (EAV) が必要です。利用可能な場合は、HyperPAV を使用して最適なパフォーマンスを確保します。
- FCP 接続のディスクストレージ

ストレージ/メインメモリー

- OpenShift Container Platform コントロールプレーンマシン用に 16 GB
- OpenShift Container Platform コンピュートマシン用に 8 GB
- 一時 OpenShift Container Platform ブートストラップマシン用に 16 GB

1.2.3.6. 推奨される IBM Z システム要件

ハードウェア要件

- SMT2 をサポートする 6 IFL 搭載の 3 LPAR
- 1 または 2 OSA または RoCE ネットワークアダプター、またはその両方
- HiperSockets。ノードに直接割り当てられるか、または z/VM ゲストに対して透過性を持たせるために z/VM VSWITCH でブリッジしてノードに割り当てられます。Hipersockets をノードに直接接続するには、RHEL 8 ゲスト経由で外部ネットワークにゲートウェイを設定し、Hipersockets ネットワークにブリッジする必要があります。

オペレーティングシステム要件

- 高可用性を確保する場合は z/VM 7.1 の 2 または 3 インスタンス

z/VM インスタンスで以下を設定します。

- OpenShift Container Platform コントロールプレーンマシン用に 3 ゲスト仮想マシン (z/VM インスタンスごとに 1 つ)
- OpenShift Container Platform コンピュートマシン用に 6 以上のゲスト仮想マシン (z/VM インスタンス全体に分散)
- 一時 OpenShift Container Platform ブートストラップマシンの 1 ゲスト仮想マシン

z/VM ゲスト仮想マシンのディスクストレージ

- FICON 接続のディスクストレージ (DASD)これらには z/VM ミニディスク、フルパックミニディスク、または専用の DASD を使用できます。Red Hat Enterprise Linux CoreOS (RHCOS) インストールに必要な最低限の DASD サイズに達するには、拡張アドレスボリューム (EAV) が必要です。利用可能な場合は、HyperPAV および High Performance FICON (zHPF) を使用して最適なパフォーマンスを確保します。
- FCP 接続のディスクストレージ

ストレージ/メインメモリー

- OpenShift Container Platform コントロールプレーンマシン用に 16 GB
- OpenShift Container Platform コンピュートマシン用に 8 GB
- 一時 OpenShift Container Platform ブートストラップマシン用に 16 GB

1.2.3.7. 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。

追加リソース

- IBM Knowledge Center の「Bridging a HiperSockets LAN with a z/VM Virtual Switch 」を参照してください。
- パフォーマンスの最適化については、「Scaling HyperPAV alias devices on Linux guests on z/VM」を参照してください。

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

5. ネットワーク接続を確認します。

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

すべての Red Hat Enterprise Linux CoreOS (RHCOS) マシンでは、起動時に **initramfs** のネットワークがマシン設定サーバーから Ignition 設定をフェッチする必要があります。

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

クラスターのマシンを長期間管理するために DHCP サーバーを使用することが推奨されています。DHCP サーバーが永続 IP アドレスおよびホスト名をクラスターマシンに提供するように設定されていることを確認します。

クラスターの正常なインストール後に各マスターノードで実行される Kubernetes API サーバーは、クラスターマシンのノード名を解決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1.6 すべてのマシンに対応するすべてのマシン

プロトコル	ポート	説明
ICMP	該当なし	ネットワーク到達性のテスト
TCP	**9000**-**9999**	ホストレベルのサービス。 ポート **9100**-**9101** のノードエクスポーター、ポート **9099** の Cluster Version Operator が含まれます。
	10250-**10259**	Kubernetes が予約するデフォルトポート
	10256	openshift-sdn
UDP	**4789**	VXLAN および GENEVE
	6081	VXLAN および GENEVE
	9000-**9999**	ポート **9100**-**9101** のノードエクスポーターを含む、ホストレベルのサービス。
TCP/UDP	**30000**-**32767**	Kubernetes NodePort

表1.7 コントロールプレーンへのすべてのマシン

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

プロトコル	ポート	説明
	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。

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

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

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

注記

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

ロードバランサーのフロントとバックの両方で以下のポートを設定します。

表1.8 API ロードバランサー

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

注記

ロードバランサーは、API サーバーが **/readyz** エンドポイントをオフにしてからプールから API サーバーインスタンスを削除するまで最大 30 秒かかるように設定する必要があります。**/readyz** の後の時間枠内でエラーが返されたり、正常になったりする場合、エンドポイントは削除または追加されている必要があります。5 秒または 10 秒ごとにプローブし、2 つの正常な要求が正常な状態になり、3 つの要求が正常な状態になりません。これらは十分にテストされた値です。

2. **Application Ingress ロードバランサー**: クラスター外から送られるアプリケーショントラフィックの Ingress ポイントを提供します。以下の条件を設定します。

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

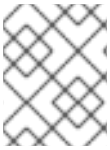
   ロードバランサーのフロントとバックの両方で以下のポートを設定します。

表1.9 アプリケーション Ingress ロードバランサー

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

ヒント

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



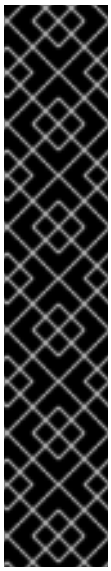
注記

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

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

以下の DNS レコードは、ユーザーによってプロビジョニングされるインフラストラクチャーを使用する OpenShift Container Platform クラスターに必要です。各レコードで、 **<cluster_name>** はクラスター名で、**<base_domain>** は、**install-config.yaml** ファイルに指定するクラスターのベースドメインです。完全な DNS レコードは **<component>.<cluster_name>.<base_domain>.** の形式を取ります。

表1.10 必要な DNS レコード

コンポーネント	レコード	説明
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。

コンポーネント	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。

重要

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

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

注記

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

手順

1. パスワードなしの認証に設定されている SSH キーがコンピューター上にない場合は、これを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ ssh-keygen -t rsa -b 4096 -N '' \
    -f <path>/<file_name> 1
```

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

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

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

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

Agent pid 31874
```

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

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

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

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

次のステップ

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

1.2.6. インストール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。

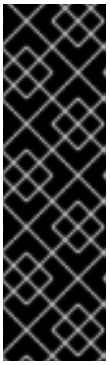
前提条件

- 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

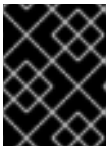
ディレクトリーを作成する必要があります。ブートストラップ X.509 証明書などの一部のインストールアセットの有効期限は短く設定されているため、インストールディレクトリーを再利用することができません。別のクラスターインストールの個別のファイルを再利用する必要がある場合は、それらをディレクトリーにコピーすることができます。ただし、インストールアセットのファイル名はリリース間で変更される可能性があります。インストールファイルを以前のバージョンの OpenShift Container Platform からコピーする場合は注意してコピーを行ってください。

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。

1.2.6.1. ベアメタルのサンプル install-config.yaml ファイル

1.2.6.2. IBM Z のサンプル install-config.yaml ファイル

install-config.yaml ファイルをカスタマイズして、OpenShift Container Platform クラスターのプラットフォームについての詳細を指定するか、または必要なパラメーターの値を変更することができます。

```
apiVersion: v1
baseDomain: example.com (1)
compute:
- hyperthreading: Enabled (2) (3)
  name: worker
  replicas: 0 (4)
controlPlane:
  hyperthreading: Enabled (5) (6)
  name: master (7)
  replicas: 3 (8)
metadata:
  name: test (9)
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14 (10)
    hostPrefix: 23 (11)
  networkType: OpenShiftSDN
  serviceNetwork: (12)
  - 172.30.0.0/16
platform:
  none: {} (13)
fips: false (14)
pullSecret: '{"auths":{"<mirror_registry>": {"auth": "<credentials>","email": "you@example.com"}}}' (15)
sshKey: 'ssh-ed25519 AAAA...' (16)
additionalTrustBundle: | (17)
  -----BEGIN CERTIFICATE-----
  ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
  -----END CERTIFICATE-----
imageContentSources: (18)
- mirrors:
  - <local_repository>/ocp4/openshift4
  source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - <local_repository>/ocp4/openshift4
  source: quay.io/openshift-release-dev/ocp-v4.0-art-devsshKey: 'ssh-ed25519 AAAA...'
```

(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) (7) 同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。パラメーター値を **Disabled** に設定するとこれを無効にすることができます。一部のクラスターマシンで同時マルチスレッドを無効にする場合は、これをすべてのクラスターマシンで無効にする必要があります。

重要

同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。

4 **replicas** パラメーターの値を **0** に設定する必要があります。このパラメーターはクラスターが作成し、管理するワーカーの数を制御します。これは、ユーザーによってプロビジョニングされるインフラストラクチャーを使用する場合にクラスターが実行しない機能です。OpenShift Container Platform のインストールが終了する前に、クラスターが使用するワーカーマシンを手動でデプロイする必要があります。

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

9 DNS レコードに指定したクラスター名。

10 Pod IP アドレスの割り当てに使用する IP アドレスのブロック。このブロックは既存の物理ネットワークと重複できません。これらの IP アドレスは Pod ネットワークに使用されます。外部ネットワークから Pod にアクセスする必要がある場合、ロードバランサーおよびルーターを、トラフィックを管理するように設定する必要があります。

11 それぞれの個別ノードに割り当てるサブネットプレフィックスの長さ。たとえば、**hostPrefix** が **23** に設定され、各ノードに指定の **cidr** から **/23** サブネットが割り当てられます (510 (2^(32 - 23) - 2) Pod IP アドレスが許可されます)。外部ネットワークからのノードへのアクセスを提供する必要がある場合には、ロードバランサーおよびルーターを、トラフィックを管理するように設定します。

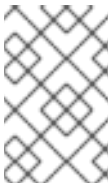
12 サービス IP アドレスに使用する IP アドレスプール。1 つの IP アドレスプールのみを入力できます。外部ネットワークからサービスにアクセスする必要がある場合、ロードバランサーおよびルーターを、トラフィックを管理するように設定します。

13 プラットフォームを **none**に設定する必要があります。ベアメタル IBM Z インフラストラクチャー用に追加のプラットフォーム設定変数を指定することはできません。

14 FIPS モードを有効または無効にするかどうか。デフォルトでは、FIPS モードは有効にされません。FIPS モードが有効にされている場合、OpenShift Container Platform が実行される Red Hat Enterprise Linux CoreOS (RHCOS) マシンがデフォルトの Kubernetes 暗号スイートをバイパスし、代わりに RHCOS で提供される暗号モジュールを使用します。

15 **<mirror_registry>** については、レジストリードメイン名と、ミラーレジストリーがコンテンツを提供するために使用するポートをオプションで指定します。例: **registry.example.com** または **registry.example.com:5000<credentials>** について、ミラーレジストリーの base64 でエンコードされたユーザー名およびパスワードを指定します。

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



注記

インストールのデバッグまたは障害復旧を実行する必要のある実稼働用の 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。

17 additionalTrustBundle パラメーターおよび値を追加します。この値は、ミラーレジストリーに使用した証明書ファイルの内容である必要があります。これはミラーレジストリー用に生成した既存の、信頼される認証局または自己署名証明書である可能性があります。

18 リポジトリーのミラーリングに使用するコマンドの出力の **imageContentSources** セクションを指定します。

1.2.6.3. 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。追加のプロキシー設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシーネットワークを使用する場合には、**httpProxy** 値を指定することはできません。

2 クラスター外で HTTPS 接続を作成するために使用するプロキシー 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。追加のプロキシー設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシーネットワークを使用する場合には、**httpsProxy** 値を指定することはできません。

3 プロキシ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、これらのコンテンツを Red Hat Enterprise Linux CoreOS (RHCOS) 信頼バンドルにマージする **trusted-ca-bundle** ConfigMap を作成し、この ConfigMap はプロキシーオブジェクトの **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。追加のプロキシー設定が必要ではなく、追加の CA を必要とする MITM の透過的なプロキシーネットワークを使用する場合には、MITM CA 証明書を指定する必要があります。

注記

インストールプログラムは、プロキシーの **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。

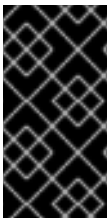
インストールプログラムは、指定の **install-config.yaml** ファイルのプロキシー設定を使用する **cluster** という名前のクラスター全体のプロキシーを作成します。プロキシー設定が指定されていない場合、**cluster** のプロキシーオブジェクトが依然として作成されますが、これには **spec** がありません。

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。

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

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



重要

インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになる証明書が含まれます。最初の証明書のローテーションが正常に実行されるようにするには、クラスターのインストールを完了し、クラスターを動作が低下していない状態で 24 時間実行し続ける必要があります。

前提条件

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

手順

1. クラスターの Kubernetes マニフェストを生成します。

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

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

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

インストールプロセスの後の部分で独自のコンピュー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. **<installation_directory>/manifests/cluster-scheduler-02-config.yml** Kubernetes マニフェストファイルを変更し、Pod がコントロールプレーンマシンにスケジュールされないようにします。

   a. **<installation_directory>/manifests/cluster-scheduler-02-config.yml** ファイルを開きます。

   b. **mastersSchedulable** パラメーターを見つけ、その値を **False** に設定します。

   c. ファイルを保存し、終了します。

注記

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

3. Ignition 設定ファイルを取得します。

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

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

以下のファイルはディレクトリーに生成されます。

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

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

プロビジョニングする IBM Z インフラストラクチャーにクラスターをインストールする前に、クラスターが使用する RHCOS を z/VM ゲスト仮想マシンにインストールする必要があります。マシンを作成するには、以下の手順を実行します。

前提条件

- 作成するマシンがアクセスできるプロビジョニングマシンで稼働している FTP サーバー。

手順

1. プロビジョニングマシンで Linux にログインします。
2. RHCOS イメージミラーから Red Hat Enterprise Linux CoreOS インストールファイルをダウンロードします。

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。

以下のファイルをダウンロードします。

- initramfs: **rhcos-<version>-installer-initramfs.img**
- kernel: **rhcos-<version>-installer-kernel**
- RHCOS をインストールするディスクのオペレーティングシステムイメージ。このタイプは仮想マシンによって異なる場合があります。
  rhcos-<version>-s390x-dasd.s390x.raw.gz (DASD 用)

  rhcos-<version>-s390x-metal.s390x.raw.gz (FCP 用)

3. パラメーターファイルを作成します。以下のパラメーターは特定の仮想マシンに固有のものです。
   - **coreos.inst.install_dev=** の場合、DASD インストールに **dasda** を指定するか、または FCP に **sda** を指定します。FCP には **zfcp.allow_lun_scan=0** が必要なことに注意してください。
   - **rd.dasd=** の場合、RHCOS がインストールされる DASD を指定します。
   - **rd.zfcp=<adapter>,<wwpn>,<lun>** は、RHCOS をインストールする FCP ディスクを指定します。
   - **ip=**には、以下の 7 つのエントリーを指定します。
     i. マシンの IP アドレス。
     ii. 空の文字列。
     iii. ゲートウェイ。
     iv. ネットマスク。
     v. **hostname.domainname** 形式のマシンホストおよびドメイン名。RHCOS に決定し、設定させる場合は、この値を省略します。

vi. ネットワークインターフェース名。RHCOS に決定し、設定させる場合は、この値を省略します。

vii. 静的 IP アドレスを使用する場合、空の文字列になります。

- **coreos.inst.ignition_url=** の場合、マシンロールの Ignition ファイルを指定します。**bootstrap.ign**、**master.ign**、または **worker.ign** を使用します。
- その他のパラメーターはそのまま利用できます。
ブートストラップマシンのパラメーターファイルのサンプル **bootstrap-0.parm**:

```
rd.neednet=1 coreos.inst=yes coreos.inst.install_dev=dasda coreos.inst.image_url=ftp://
cl1.provide.example.com:8080/assets/rhcos-43.80.20200430.0-s390x-dasd.390x.raw.gz
coreos.inst.ignition_url=ftp://cl1.provide.example.com:8080/ignition-bootstrap-0
ip=172.18.78.2::172.18.78.1:255.255.255.0:::none nameserver=172.18.78.1
rd.znet=qeth,0.0.bdf0,0.0.bdf1,0.0.bdf2,layer2=1,portno=0 zfcp.allow_lun_scan=0
cio_ignore=all,
!condev rd.dasd=0.0.3490
```

4. FTP などを使用し、initramfs、kernel、パラメーターファイル、および RHCOS イメージを z/VM に転送します。FTP でファイルを転送し、仮想リーダーから起動する方法については、「Z/VM 環境へのインストール」を参照してください。

5. ブートストラップノードになる z/VM ゲスト仮想マシンの仮想リーダーに対してファイルの punch を実行します。
IBM Knowledge Center で PUNCH を参照してください。

ヒント

CP PUNCH コマンドを使用するか、Linux を使用している場合は、 **vmur** コマンドを使用して 2 つの z/VM ゲスト仮想マシン間でファイルを転送できます。

6. ブートストラップマシンで CMS にログインします。

7. リーダーからブートストラップマシンに対して IPL を実行します。

```
$ ipl c
```

IBM Knowledge Center で IPL を参照してください。

8. クラスター内の他のマシンについてこの手順を繰り返します。

1.2.9. 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。

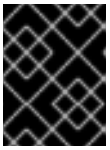
```
$ ./openshift-install --dir=<installation_directory> wait-for bootstrap-complete \ 1
    --log-level=info 2
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com...
INFO API v1.16.2 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove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

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

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

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

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



重要

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

1.2.10. 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
```

1.2.11. 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  v1.16.2
master-1  Ready     master  63m  v1.16.2
master-2  Ready     master  64m  v1.16.2
worker-0  NotReady  worker  76s  v1.16.2
worker-1  NotReady  worker  70s  v1.16.2
```

   出力には作成したすべてのマシンが一覧表示されます。

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

```
$ oc get csr

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

   (1) クライアント要求の CSR。

   (2) サーバー要求の CSR。

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

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

注記

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

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

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

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

- すべての保留中の CSR を承認するには、以下のコマンドを実行します。

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

1.2.12. 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                       4.3.0       True        False         False      40d
cloud-credential                     4.3.0       True        False         False      40d
cluster-autoscaler                   4.3.0       True        False         False      40d
console                              4.3.0       True        False         False      6d2h
dns                                  4.3.0       True        False         False      22d
image-registry                       4.3.0       True        False         False      6d2h
ingress                              4.3.0       True        False         False      6d2h
insights                             4.3.0       True        False         False      40d
kube-apiserver                       4.3.0       True        False         False      40d
kube-controller-manager              4.3.0       True        False         False      40d
kube-scheduler                       4.3.0       True        False         False      40d
machine-api                          4.3.0       True        False         False      40d
```

```
machine-config                             4.3.0       True        False         False      22d
marketplace                                4.3.0       True        False         False      40d
monitoring                                 4.3.0       True        False         False      6d16h
network                                    4.3.0       True        False         False      40d
node-tuning                                4.3.0       True        False         False      22d
openshift-apiserver                        4.3.0       True        False         False      22d
openshift-controller-manager               4.3.0       True        False         False      6d2h
openshift-samples                          4.3.0       True        False         False      40d
operator-lifecycle-manager                 4.3.0       True        False         False      40d
operator-lifecycle-manager-catalog         4.3.0       True        False         False      40d
operator-lifecycle-manager-packageserver   4.3.0       True        False         False      6d2h
service-ca                                 4.3.0       True        False         False      40d
service-catalog-apiserver                  4.3.0       True        False         False      40d
service-catalog-controller-manager         4.3.0       True        False         False      40d
storage                                    4.3.0       True        False         False      40d
```

2. 利用不可の Operator を設定します。

1.2.12.1. イメージレジストリーストレージの設定

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

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

1.2.12.1.1. ベアメタルの場合のレジストリーストレージの設定

1.2.12.1.2. IBM Z の場合のレジストリーストレージの設定

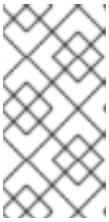
クラスター管理者は、インストール後にレジストリーをストレージを使用できるように設定する必要があります。

前提条件

- クラスター管理者のパーミッション。
- ベアメタル上のクラスター。IBM Z
- NFS などのクラスターの永続ストレージをプロビジョニングします。プライベートイメージレジストリーをデプロイするには、ストレージで ReadWriteMany アクセスモードを指定する必要があります。
- 容量は「100Gi」以上である。

手順

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



注記

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

2. レジストリー Pod がないことを確認します。

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

注記

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

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

3. レジストリー設定を確認します。

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

storage:
  pvc:
    claim:
```

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

4. **clusteroperator** ステータスを確認します。

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

1.2.12.1.3. 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。

1.2.13. 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                             4.3.0     True        False         False      10m
cloud-credential                           4.3.0     True        False         False      22m
cluster-autoscaler                         4.3.0     True        False         False      21m
console                                    4.3.0     True        False         False      10m
dns                                        4.3.0     True        False         False      21m
image-registry                             4.3.0     True        False         False      16m
ingress                                    4.3.0     True        False         False      16m
kube-apiserver                             4.3.0     True        False         False      19m
kube-controller-manager                    4.3.0     True        False         False      18m
kube-scheduler                             4.3.0     True        False         False      22m
machine-api                                4.3.0     True        False         False      22m
machine-config                             4.3.0     True        False         False      18m
marketplace                                4.3.0     True        False         False      18m
monitoring                                 4.3.0     True        False         False      18m
network                                    4.3.0     True        False         False      16m
node-tuning                                4.3.0     True        False         False      21m
openshift-apiserver                        4.3.0     True        False         False      21m
openshift-controller-manager               4.3.0     True        False         False      17m
openshift-samples                          4.3.0     True        False         False      14m
```

```
operator-lifecycle-manager          4.3.0     True        False         False      21m
operator-lifecycle-manager-catalog  4.3.0     True        False         False      21m
service-ca                          4.3.0     True        False         False      21m
service-catalog-apiserver           4.3.0     True        False         False      16m
service-catalog-controller-manager  4.3.0     True        False         False      16m
storage                             4.3.0     True        False         False      16m
```

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

2. クラスターの完了をモニターします。

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

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

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

重要

インストールプログラムが生成する Ignition 設定ファイルには、24 時間が経過すると期限切れになる証明書が含まれます。最初の証明書のローテーションが正常に実行されるようにするには、クラスターを動作が低下していない状態で 24 時間実行し続ける必要があります。

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

   a. すべての Pod の一覧を表示するには、以下のコマンドを使用します。

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

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

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

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

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

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

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

1.2.14. デバッグ情報の収集

IBM Z での OpenShift Container Platform インストールに関する特定の問題のトラブルシューティングおよびデバッグに役立つ可能性のあるデバッグ情報を収集できます。

前提条件

- **oc** CLI ツールをインストールしていること。

手順

1. クラスターにログインします。

```
$ oc login
```

2. ハードウェア情報を収集するノードで、デバッグコンテナーを起動します。

```
$ oc debug node/<nodename>
```

3. **/host** ファイルシステムに切り替え、**toolbox** を起動します。

```
$ chroot /host
$ toolbox
```

4. **dbginfo** データを収集します。

```
$ dbginfo.sh
```

5. その後に、**scp** を使用するなどしてデータを取得できます。

追加リソース

- 「How to generate SOSREPORT within OpenShift Container Platform version 4 nodes without SSH」も参照してください。

1.2.15. 次のステップ

- クラスターをカスタマイズします。