OpenShift Container Platform 4.2

GCP へのインストール

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

Last Updated: 2020-06-17

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

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

法律上の通知

概要

本書では、Google Cloud Platform に OpenShift Container Platform 4.2 クラスターをインストールし、アンインストールする方法について説明します。

目次

第1章 GCP へのインストール

1.1. GCP プロジェクトの設定

OpenShift Container Platform をインストールする前に、これをホストするように Google Cloud Platform (GCP) プロジェクトを設定する必要があります。

1.1.1. GCP プロジェクトの作成

OpenShift Container Platform をインストールするには、クラスターをホストするために Google Cloud Platform (GCP) アカウントでプロジェクトを作成する必要があります。

手順

- OpenShift Container Platform クラスターをホストするプロジェクトを作成します。GCP ドキュメントの「Creating and Managing Projects」を参照してください。

1.1.2. GCP での API サービスの有効化

Google Cloud Platform (GCP) プロジェクトでは、OpenShift Container Platform インストールを完了するために複数の API サービスへのアクセスが必要です。

前提条件

- クラスターをホストするプロジェクトを作成している。

手順

- クラスターをホストするプロジェクトで以下の必要な API サービスを有効にします。GCP ドキュメントの「サービスの有効化」を参照してください。

表1.1 必要な API サービス

API サービス	コンソールサービス名
Compute Engine API	**compute.googleapis.com**
Google Cloud API	**cloudapis.googleapis.com**
Cloud Resource Manager API	**cloudresourcemanager.googleapis.com**
Google DNS API	**dns.googleapis.com**
IAM Service Account Credentials API	**iamcredentials.googleapis.com**
Identity and Access Management (IAM) API	**iam.googleapis.com**
Service Management API	**servicemanagement.googleapis.com**

API サービス	コンソールサービス名
Service Usage API	**serviceusage.googleapis.com**
Google Cloud Storage JSON API	**storage-api.googleapis.com**
Cloud Storage	**storage-component.googleapis.com**

1.1.3. GCP の DNS の設定

OpenShift Container Platform をインストールするには、使用する Google Cloud Platform (GCP) アカウントに、OpenShift Container Platform クラスターをホストする同じプロジェクトに専用のパブリックホストゾーンがなければなりません。このゾーンはドメインに対する権威を持っている必要があります。DNS サービスは、クラスターへの外部接続のためのクラスターの DNS 解決および名前検索を提供します。

手順

1. ドメイン、またはサブドメイン、およびレジストラーを特定します。既存のドメインおよびレジストラーを移行するか、GCP または他のソースから新規のものを取得できます。

注記

新規ドメインを購入する場合、関連する DNS の変更が伝播するのに時間がかかる場合があります。Google 経由でドメインを購入する方法についての詳細は、「Google ドメイン」を参照してください。

2. GCP プロジェクトにドメインまたはサブドメインのパブリックホストゾーンを作成します。GCP ドキュメントの「Creating public zones」を参照してください。
openshiftcorp.com などのルートドメインや、 **clusters.openshiftcorp.com** などのサブドメインを使用します。

3. ホストゾーンレコードから新規の権威ネームサーバーを抽出します。GCP ドキュメントの「Look up your Cloud DNS name servers」を参照してください。
通常は、4 つのネームサーバーがあります。

4. ドメインが使用するネームサーバーのレジストラーレコードを更新します。たとえば、ドメインを Google ドメインに登録している場合は、Google Domains Help で「How to switch to custom name servers」のトピックを参照してください。

5. ルートドメインを Google Cloud DNS に移行している場合は、DNS レコードを移行します。GCP ドキュメントの「Migrating to Cloud DNS」を参照してください。

6. サブドメインを使用する場合は、所属する会社の手順に従ってその委任レコードを親ドメインに追加します。このプロセスには、所属企業の IT 部門や、会社のルートドメインと DNS サービスを制御する部門へのリクエストが含まれる場合があります。

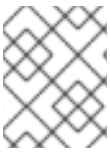
1.1.4. GCP アカウントの制限

OpenShift Container Platform クラスターは多くの Google Cloud Platform (GCP) コンポーネントを使用しますが、デフォルトの 割り当て (Quota) はデフォルトの OpenShift Container Platform クラスターをインストールする機能に影響を与えません。

3 つのコンピュートマシンおよび 3 つのコントロールプレーンマシンが含まれるデフォルトクラスターは以下のリソースを使用します。一部のリソースはブートストラッププロセス時にのみ必要となり、クラスターのデプロイ後に削除されることに注意してください。

表1.2 デフォルトのクラスターで使用される GCP リソース

サービス	Component	場所	必要なリソースの合計	ブートストラップ後に削除されるリソース
サービスアカウント	IAM	グローバル	5	0
ファイアウォールのルール	コンピュート	グローバル	11	1
転送ルール	コンピュート	グローバル	2	0
使用中のグローバル IP アドレス	コンピュート	グローバル	4	1
ヘルスチェック	コンピュート	グローバル	3	0
イメージ	コンピュート	グローバル	1	0
ネットワーク	コンピュート	グローバル	2	0
静的 IP アドレス	コンピュート	リージョン	4	1
ルーター	コンピュート	グローバル	1	0
ルート	コンピュート	グローバル	2	0
サブネットワーク	コンピュート	グローバル	2	0
ターゲットプール	コンピュート	グローバル	3	0
CPU	コンピュート	リージョン	28	4
永続ディスク SSD (GB)	コンピュート	リージョン	896	128



注記

インストール時にクォータが十分ではない場合、インストールプログラムは超過したクォータとリージョンの両方を示すエラーを表示します。

実際のクラスターサイズ、計画されるクラスターの拡張、およびアカウントに関連付けられた他のクラスターからの使用法を考慮してください。CPU、静的 IP アドレス、および永続ディスク SSD（ストレージ）のクォータは、ほとんどの場合に不十分になる可能性のあるものです。

以下のリージョンのいずれかにクラスターをデプロイする予定の場合、ストレージクォータの最大値を超え、CPU クォータ制限を超える可能性が高くなります。

- asia-east2
- asia-northeast2
- asia-south1
- australia-southeast1
- europe-north1
- europe-west2
- europe-west3
- europe-west6
- northamerica-northeast1
- southamerica-east1
- us-west2

GCP コンソールからリソースクォータを増やすことは可能ですが、サポートチケットを作成する必要がある場合があります。OpenShift Container Platform クラスターをインストールする前にサポートチケットを解決できるように、クラスターのサイズを早期に計画してください。

1.1.5. GCP でのサービスアカウントの作成

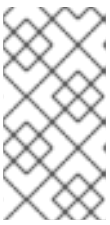
OpenShift Container Platform には、Google API でデータにアクセスするための認証および承認を提供する Google Cloud Platform (GCP) サービスアカウントが必要です。プロジェクトに必要なロールが含まれる既存の IAM サービスアカウントがない場合は、これを作成する必要があります。

前提条件

- クラスターをホストするプロジェクトを作成している。

手順

1. OpenShift Container Platform クラスターをホストするために使用するプロジェクトでサービスアカウントを作成します。GCP ドキュメントで「Creating a service account 」を参照してください。
2. サービスアカウントに適切なパーミッションを付与します。付随する個別のパーミッションを付与したり、**オーナー**ロールをこれに割り当てることができます。「特定のリソースのサービス アカウントへの役割の付与」を参照してください。



注記

サービスアカウントをプロジェクトの所有者にすることが必要なパーミッションを取得する最も簡単な方法になります。 つまりこれは、サービスアカウントはプロジェクトを完全に制御できることを意味します。この権限を提供することに伴うリスクが受け入れ可能であるかどうかを判別する必要があります。

3. JSON 形式でサービスアカウントキーを作成します。GCP ドキュメントの「サービスアカウント キーの作成」を参照してください。
クラスターを作成するには、サービスアカウントキーが必要になります。

1.1.5.1. 必要な GCP パーミッション

作成するサービスアカウントに**オーナー**ロールを割り当てると、OpenShift Container Platform のインストールに必要なパーミッションも含め、そのサービスアカウントにすべてのパーミッションが付与されます。 OpenShift Container Platform クラスターをデプロイするには、サービスアカウントに以下のパーミッションが必要です。

インストールプログラムに必要なロール

- Compute 管理者
- DNS 管理者
- セキュリティー管理者
- サービスアカウント管理者
- サービスアカウントユーザー
- ストレージ管理者

インストール時のネットワークリソースの作成に必要なロール

- DNS 管理者

オプションのロール

クラスターで Operator の制限された認証情報を新たに作成できるようにするには、以下のロールを追加します。

- サービスアカウントキー管理者

ロールは、コントロールプレーンおよびコンピュートマシンが使用するサービスアカウントに適用されます。

表1.3 GCP サービスアカウントのパーミッション

アカウント	ロール
コントロールプレーン	**roles/compute.instanceAdmin**
	roles/compute.networkAdmin
	roles/compute.securityAdmin
	roles/storage.admin
	roles/iam.serviceAccountUser

アカウント	ロール
コンピュート	**roles/compute.viewer**
	roles/storage.admin

1.1.6. サポートされている GCP リージョン

OpenShift Container Platform クラスターを以下の Google Cloud Platform (GCP) リージョンにデプロイできます。

- asia-east1 (Changhua County, Taiwan)
- asia-east2 (Hong Kong)
- asia-northeast1 (Tokyo, Japan)
- asia-northeast2 (Osaka, Japan)
- asia-south1 (Mumbai, India)
- asia-southeast1 (Jurong West, Singapore)
- australia-southeast1 (Sydney, Australia)
- europe-north1 (Hamina, Finland)
- europe-west1 (St. Ghislain, Belgium)
- europe-west2 (London, England, UK)
- europe-west3 (Frankfurt, Germany)
- europe-west4 (Eemshaven, Netherlands)
- europe-west6 (Zürich, Switzerland)
- northamerica-northeast1 (Montréal, Québec, Canada)
- southamerica-east1 (São Paulo, Brazil)
- us-central1 (Council Bluffs, Iowa, USA)
- us-east1 (Moncks Corner, South Carolina, USA)
- us-east4 (Ashburn, Northern Virginia, USA)
- us-west1 (The Dalles, Oregon, USA)
- us-west2 (Los Angeles, California, USA)

次のステップ

- GCP に OpenShift Container Platform クラスターをインストールします。カスタマイズされたクラスターのインストール、またはデフォルトのオプションでクラスターのクイックインストールを実行できます。

1.2. GCP へのクラスターのクイックインストール

OpenShift Container Platform バージョン 4.2 では、デフォルトの設定オプションを使用してクラスターを Google Cloud Platform (GCP) にインストールできます。

前提条件

- OpenShift Container Platform のインストールおよび更新 プロセスについての詳細を確認します。
- GCP アカウントを設定してクラスターをホストします。
- 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。

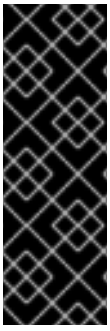
1.2.1. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス

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. 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。

1.2.3. 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

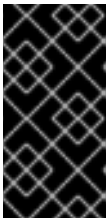
ローカルコンピューターにダウンロードします。

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。

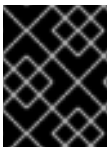
3. インストールプログラムを展開します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ 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。

1.2.4. クラスターのデプロイ

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



重要

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

前提条件

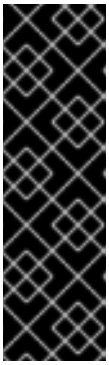
- クラスターをホストするクラウドプラットフォームでアカウントを設定します。
- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

手順

1. クラスターに設定した GCP アカウントのサービスアカウントキーを使用しない既存の GCP 認証情報で、以下の場所に保存されているものを削除します。
   - **GOOGLE_CREDENTIALS**、 **GOOGLE_CLOUD_KEYFILE_JSON**、または **GCLOUD_KEYFILE_JSON** 環境変数
   - **~/.gcp/osServiceAccount.json** ファイル
   - **gcloud cli** デフォルト認証情報
2. インストールプログラムを実行します。

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

1. **<installation_directory>** には、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。
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。



重要

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。

プロンプト時に値を指定します。

a. オプション: クラスターマシンにアクセスするために使用する SSH キーを選択します。

注記

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

b. ターゲットに設定するプラットフォームとして **gcp** を選択します。

c. コンピューター上で GCP アカウント用のサービスアカウントキーを設定していない場合、GCP からこれを取得してファイルの内容を貼り付けるか、またはファイルへの絶対パスを入力する必要があります。

d. クラスターのプロビジョニングに使用するプロジェクト ID を選択します。デフォルト値は、設定したサービスアカウントによって指定されます。

e. クラスターをデプロイするリージョンを選択します。

f. クラスターをデプロイするベースドメインを選択します。ベースドメインは、クラスターに作成したパブリック DNS ゾーンに対応します。

g. クラスターの記述名を入力します。7 文字以上の名前を指定すると、クラスター名から生成されるインフラストラクチャー ID で最初の 6 文字のみが使用されます。

h. 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を貼り付けます。

注記

ホストに設定した AWS アカウントにクラスターをデプロイするための十分なパーミッションがない場合、インストールプログラムは停止し、不足しているパーミッションが表示されます。

クラスターのデプロイメントが完了すると、Web コンソールへのリンクや **kubeadmin** ユーザーの認証情報を含む、クラスターにアクセスするための指示がターミナルに表示されます。

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。

重要

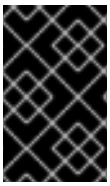
インストールプログラム、またはインストールプログラムが作成するファイルを削除することはできません。これらはいずれもクラスターを削除するために必要になります。

3. オプション: クラスターをインストールするために使用したサービスアカウントのパーミッションの数を減らすことができます。

   - **Owner** ロールをサービスアカウントに割り当てている場合、 そのロールを削除し、これを **Viewer** ロールに置き換えることができます。
   - **Service Account Key Admin** ロールが含まれている場合は、これを削除することができます。

1.2.5. 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>
```

1.2.6. 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
```

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. カスタマイズを使用した GCP 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、インストールプログラムが Google Cloud Platform (GCP) にプロビジョニングするインフラストラクチャーにカスタマイズされたクラスターをインストールできます。インストールをカスタマイズするには、クラスターをインストールする前に、**install-config.yaml** ファイルでパラメーターを変更します。

前提条件

- OpenShift Container Platform のインストールおよび更新 プロセスについての詳細を確認します。
- GCP アカウントを設定してクラスターをホストします。
- 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。

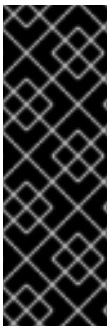
1.3.1. OpenShift Container Platform のインターネットアクセスおよび Telemetry アクセス

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。

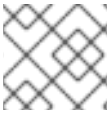


重要

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。

1.3.2. 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。

1.3.3. 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

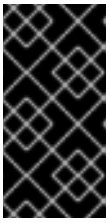
ローカルコンピューターにダウンロードします。

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、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。

1.3.4. インストール設定ファイルの作成

Google Cloud Platform (GCP) での OpenShift Container Platform のインストールをカスタマイズできます。

前提条件

- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

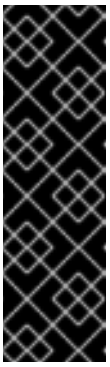
手順

1. **install-config.yaml** ファイルを作成します。
   a. 次のコマンドを実行します。

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

```
-
```

1. **<installation_directory>** には、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

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。

b. プロンプト時に、クラウドの設定の詳細情報を指定します。

i. オプション: クラスターマシンにアクセスするために使用する SSH キーを選択します。

注記

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

ii. ターゲットに設定するプラットフォームとして **gcp** を選択します。

iii. コンピューター上で GCP アカウント用のサービスアカウントキーを設定していない場合、GCP からこれを取得してファイルの内容を貼り付けるか、またはファイルへの絶対パスを入力する必要があります。

iv. クラスターのプロビジョニングに使用するプロジェクト ID を選択します。デフォルト値は、設定したサービスアカウントによって指定されます。

v. クラスターをデプロイするリージョンを選択します。

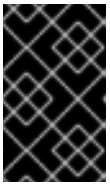
vi. クラスターをデプロイするベースドメインを選択します。ベースドメインは、クラスターに作成したパブリック DNS ゾーンに対応します。

vii. クラスターの記述名を入力します。7 文字以上の名前を指定すると、クラスター名から生成されるインフラストラクチャー ID で最初の 6 文字のみが使用されます。

viii. 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を貼り付けます。

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

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時に使用されます。このファイルを再利用する必要がある場合は、この段階でこれをバックアップしてください。

1.3.4.1. インストール設定パラメーター

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

注記

インストール後は、**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1.4 必須パラメーター

パラメーター	説明	値
baseDomain	クラウドプロバイダーのベースドメイン。この値は、OpenShift Container Platform クラスターコンポーネントへのルートを作成するために使用されます。クラスターの完全な DNS 名は、**baseDomain** と **<metadata.name>.<baseDomain>** 形式を使用する **metadata.name** パラメーターの値の組み合わせです。	**example.com** などの完全修飾ドメインまたはサブドメイン名。
controlPlane.platform	コントロールプレーンマシンをホストするためのクラウドプロバイダー。このパラメーターの値は **compute.platform** パラメーターの値に一致する必要があります。	**aws**、**azure**、**gcp**、**openstack**、または **{}**
compute.platform	ワーカーマシンをホストするためのクラウドプロバイダー。このパラメーターの値は **controlPlane.platform** パラメーターの値に一致する必要があります。	**aws**、**azure**、**gcp**、**openstack**、または **{}**
metadata.name	クラスターの名前。	**dev** などの大文字または小文字を含む文字列。

パラメーター	説明	値
platform.<platform>.region	クラスターをデプロイするリージョン。	AWS の **us-east-1**、Azure の **centralus**、または Red Hat OpenStack Platform (RHOSP) の **region1** などのクラウドの有効なリージョン。

pullSecret	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。	<pre>{
		  "auths":{
		  "cloud.openshift.com":{
		  "auth":"b3Blb=",
		  "email":"you@example.com"
		  },
		  "quay.io":{
		  "auth":"b3Blb=",
		  "email":"you@example.com"
		  }
		  }
		}</pre>

表1.5 オプションのパラメーター

パラメーター	説明	値
sshKey	クラスターマシンにアクセスするために使用する SSH キー。 **注記** インストールのデバッグまたは障害復旧を実行する必要のある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。	**ssh-agent** プロセスに追加した、有効なローカルのパブリック SSH キー。

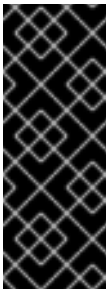
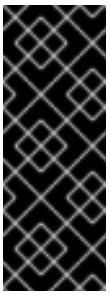
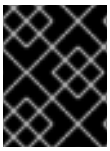
パラメーター	説明	値
compute.hyperthreading	コンピュートマシンで同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。  **重要** 同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。	**Enabled** または **Disabled**
compute.replicas	プロビジョニングするコンピュートマシン（ワーカーマシンとしても知られる）の数。	**2** 以上の正の整数。デフォルト値は **3** です。
controlPlane.hyperthreading	コントロールプレーンマシンで同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。  **重要** 同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。	**Enabled** または **Disabled**
controlPlane.replicas	プロビジョニングするコントロールプレーンマシンの数。	**3** 以上の正の整数。デフォルト値は **3** です。

表1.6 追加の Google Cloud Platform (GCP) パラメーター

パラメーター	説明	値
platform.gcp.type	GCP マシンタイプ。	GCP マシンタイプ。
platform.gcp.zones	インストールプログラムが指定される MachinePool のマシンを作成するアベイラビリティーゾーン。	YAML シーケンスの **us-central1-a** などの有効な GCP アベイラビリティーゾーンの一覧。

1.3.4.2. GCP のカスタマイズされた install-config.yaml ファイルのサンプル

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。



重要

このサンプルの YAML ファイルは参照用にのみ提供されます。インストールプログラムを使用して **install-config.yaml** ファイルを取得し、これを変更する必要があります。

```
apiVersion: v1
baseDomain: example.com (1)
controlPlane: (2)
  hyperthreading: Enabled (3) (4)
  name: master
  platform:
    gcp:
      type: n2-standard-4
      zones:
      - us-central1-a
      - us-central1-c
  replicas: 3
compute: (5)
- hyperthreading: Enabled (6)
  name: worker
  platform:
    gcp:
      type: n2-standard-4
      zones:
      - us-central1-a
      - us-central1-c
  replicas: 3
metadata:
  name: test-cluster (7)
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineCIDR: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  gcp:
```

```
    ProjectID: openshift-production (8)
    region: us-central-1 (9)
pullSecret: '{"auths": ...}' (10)
sshKey: ssh-ed25519 AAAA... (11)
```

(1) (7) (8) (9) (10) 必須。インストールプログラムはこの値の入力を求めるプロンプトを出します。

(2) (5) これらのパラメーターおよび値を指定しない場合、インストールプログラムはデフォルトの値を指定します。

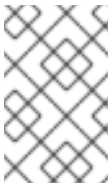
(3) (6) **controlPlane** セクションは単一マッピングですが、コンピュートセクションはマッピングのシーケンスになります。複数の異なるデータ構造の要件を満たすには、 **compute** セクションの最初の行はハイフン **-** で始め、**controlPlane** セクションの最初の行はハイフンで始めることができません。どちらのセクションも、現時点では単一のマシンプールを定義しますが、OpenShift Container Platform の今後のバージョンでは、インストール時の複数のコンピュートプールの定義をサポートする可能性があります。1 つのコントロールプ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

(4) 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、マシンに対して **n1-standard-8**などの大規模なマシンタイプを使用します。

(11) クラスター内のマシンにアクセスするために使用する **sshKey** 値をオプションで指定できます。



注記

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

1.3.5. クラスターのデプロイ

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

重要

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

前提条件

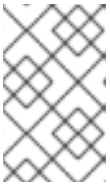
- クラスターをホストするクラウドプラットフォームでアカウントを設定します。
- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

手順

1. クラスターに設定した GCP アカウントのサービスアカウントキーを使用しない既存の GCP 認証情報で、以下の場所に保存されているものを削除します。
   - **GOOGLE_CREDENTIALS**、 **GOOGLE_CLOUD_KEYFILE_JSON**、または **GCLOUD_KEYFILE_JSON** 環境変数
   - **~/.gcp/osServiceAccount.json** ファイル
   - **gcloud cli** デフォルト認証情報
2. インストールプログラムを実行します。

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

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



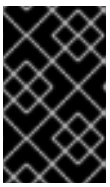
注記

ホストに設定した AWS アカウントにクラスターをデプロイするための十分なパーミッションがない場合、インストールプログラムは停止し、不足しているパーミッションが表示されます。

クラスターのデプロイメントが完了すると、Web コンソールへのリンクや **kubeadmin** ユーザーの認証情報を含む、クラスターにアクセスするための指示がターミナルに表示されます。

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。



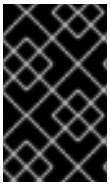
重要

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

3. オプション: クラスターをインストールするために使用したサービスアカウントのパーミッションの数を減らすことができます。
   - **Owner** ロールをサービスアカウントに割り当てている場合、 そのロールを削除し、これを **Viewer** ロールに置き換えることができます。
   - **Service Account Key Admin** ロールが含まれている場合は、これを削除することができます。

1.3.6. 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>
```

1.3.7. 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
```

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。

1.4. ネットワークのカスタマイズによる GCP 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、インストールプログラムが Google Cloud Platform (GCP) にプロビジョニングするインフラストラクチャーにカスタマイズされたネットワーク設定で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。インストールをカスタマイズするには、クラスターをインストールする前に、**install-config.yaml** ファイルでパラメーターを変更します。

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

- 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。
- GCP アカウントを設定してクラスターをホストします。
- 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。

1.4.1. 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。

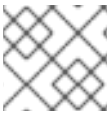
- Red Hat OpenShift Cluster Manager ページにアクセスし、インストールプログラムをダウンロードし、サブスクリプション管理を実行します。クラスターにインターネットアクセスがあり、Telemetry を無効にしない場合、そのサービスは有効なサブスクリプションでクラスターを自動的に使用します。
- クラスターのインストールに必要なパッケージを取得するために Quay.io にアクセスします。
- クラスターの更新を実行するために必要なパッケージを取得します。

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。

1.4.2. 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。

1.4.3. 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、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。

1.4.4. インストール設定ファイルの作成

Google Cloud Platform (GCP) での OpenShift Container Platform のインストールをカスタマイズできます。

前提条件

- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

手順

1. **install-config.yaml** ファイルを作成します。

   a. 次のコマンドを実行します。

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

   1 **<installation_directory>** には、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

   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。

   b. プロンプト時に、クラウドの設定の詳細情報を指定します。

      i. オプション: クラスターマシンにアクセスするために使用する SSH キーを選択します。

      注記

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

      ii. ターゲットに設定するプラットフォームとして **gcp** を選択します。

      iii. コンピューター上で GCP アカウント用のサービスアカウントキーを設定していない場合、GCP からこれを取得してファイルの内容を貼り付けるか、またはファイルへの絶対パスを入力する必要があります。

iv. クラスターのプロビジョニングに使用するプロジェクト ID を選択します。デフォルト値は、設定したサービスアカウントによって指定されます。

v. クラスターをデプロイするリージョンを選択します。

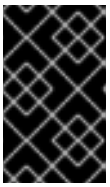
vi. クラスターをデプロイするベースドメインを選択します。ベースドメインは、クラスターに作成したパブリック DNS ゾーンに対応します。

vii. クラスターの記述名を入力します。7 文字以上の名前を指定すると、クラスター名から生成されるインフラストラクチャー ID で最初の 6 文字のみが使用されます。

viii. 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を貼り付けます。

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

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。

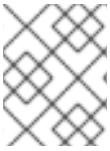


重要

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

1.4.4.1. インストール設定パラメーター

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



注記

インストール後は、**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 必須パラメーター

パラメーター	説明	値

パラメーター	説明	値
baseDomain	クラウドプロバイダーのベースドメイン。この値は、OpenShift Container Platform クラスターコンポーネントへのルートを作成するために使用されます。クラスターの完全な DNS 名は、**baseDomain** と **<metadata.name>.<baseDomain>** 形式を使用する **metadata.name** パラメーターの値の組み合わせです。	**example.com** などの完全修飾ドメインまたはサブドメイン名。
controlPlane.platform	コントロールプレーンマシンをホストするためのクラウドプロバイダー。このパラメーターの値は **compute.platform** パラメーターの値に一致する必要があります。	**aws**、**azure**、**gcp**、**openstack**、または **{}**
compute.platform	ワーカーマシンをホストするためのクラウドプロバイダー。このパラメーターの値は **controlPlane.platform** パラメーターの値に一致する必要があります。	**aws**、**azure**、**gcp**、**openstack**、または **{}**
metadata.name	クラスターの名前。	**dev** などの大文字または小文字を含む文字列。
platform.<platform>.region	クラスターをデプロイするリージョン。	AWS の **us-east-1**、Azure の **centralus**、または Red Hat OpenStack Platform (RHOSP) の **region1** などのクラウドの有効なリージョン。

pullSecret	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。	{
		"auths":{
		"cloud.openshift.com":{
		"auth":"b3Blb=",
		"email":"you@example.com"
		},
		"quay.io":{
		"auth":"b3Blb=",
		"email":"you@example.com"
		}
		}
		}

表1.8 オプションのパラメーター

パラメーター	説明	値
sshKey	クラスターマシンにアクセスするために使用する SSH キー。  **注記** インストールのデバッグまたは障害復旧を実行する必要のある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。	**ssh-agent** プロセスに追加した、有効なローカルのパブリック SSH キー。
compute.hyperthreading	コンピュートマシンで同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。  **重要** 同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。	**Enabled** または **Disabled**
compute.replicas	プロビジョニングするコンピュートマシン（ワーカーマシンとしても知られる）の数。	**2** 以上の正の整数。デフォルト値は **3** です。

パラメーター	説明	値
controlPlane.hyperthreading	コントロールプレーンマシンで同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。  **重要** 同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。	**Enabled** または **Disabled**
controlPlane.replicas	プロビジョニングするコントロールプレーンマシンの数。	**3** 以上の正の整数。デフォルト値は **3** です。

表1.9 追加の Google Cloud Platform (GCP) パラメーター

パラメーター	説明	値
platform.gcp.type	GCP マシンタイプ。	GCP マシンタイプ。
platform.gcp.zones	インストールプログラムが指定される MachinePool のマシンを作成するアベイラビリティーゾーン。	YAML シーケンスの **us-central1-a** などの有効な GCP アベイラビリティーゾーンの一覧。

1.4.4.2. 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1.10 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。

1.4.4.3. GCP のカスタマイズされた install-config.yaml ファイルのサンプル

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。

重要

このサンプルの YAML ファイルは参照用にのみ提供されます。インストールプログラムを使用して **install-config.yaml** ファイルを取得し、これを変更する必要があります。

```
apiVersion: v1
baseDomain: example.com (1)
controlPlane: (2)
  hyperthreading: Enabled (3) (4)
  name: master
  platform:
    gcp:
      type: n2-standard-4
      zones:
      - us-central1-a
```

```
      - us-central1-c
  replicas: 3
compute: 5
- hyperthreading: Enabled 6
  name: worker
  platform:
    gcp:
      type: n2-standard-4
      zones:
      - us-central1-a
      - us-central1-c
  replicas: 3
metadata:
  name: test-cluster 7
networking: 8
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineCIDR: 10.0.0.0/16
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  gcp:
    ProjectID: openshift-production 9
    region: us-central-1 10
pullSecret: '{"auths": ...}' 11
sshKey: ssh-ed25519 AAAA... 12
```

1 7 9 10 11 必須。インストールプログラムはこの値の入力を求めるプロンプトを出します。

2 5 8 これらのパラメーターおよび値を指定しない場合、インストールプログラムはデフォルトの値を指定します。

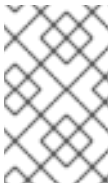
3 6 **controlPlane** セクションは単一マッピングですが、コンピュートセクションはマッピングのシーケンスになります。複数の異なるデータ構造の要件を満たすには、 **compute** セクションの最初の行はハイフン **-** で始め、**controlPlane** セクションの最初の行はハイフンで始めることができません。どちらのセクションも、現時点では単一のマシンプールを定義しますが、OpenShift Container Platform の今後のバージョンでは、インストール時の複数のコンピュートプールの定義をサポートする可能性があります。1 つのコントロールプレーンプールのみが使用されます。

4 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、マシンに対して **n1-standard-8**などの大規模なマシンタイプを使用します。

12 クラスター内のマシンにアクセスするために使用する **sshKey** 値をオプションで指定できます。



注記

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

1.4.5. 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。

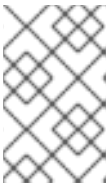
手順

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。

1.4.6. 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。

1.4.6.1. 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

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。

1.4.6.2. 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
```

1.4.7. クラスターのデプロイ

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

重要

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

前提条件

- クラスターをホストするクラウドプラットフォームでアカウントを設定します。
- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

手順

1. インストールプログラムを実行します。

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

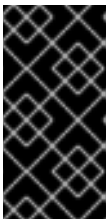
1 **<installation_directory>** については、カスタマイズされたファイルの場所を指定します。

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

注記

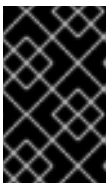
ホストに設定した AWS アカウントにクラスターをデプロイするための十分なパーミッションがない場合、インストールプログラムは停止し、不足しているパーミッションが表示されます。

クラスターのデプロイメントが完了すると、Web コンソールへのリンクや **kubeadmin** ユーザーの認証情報を含む、クラスターにアクセスするための指示がターミナルに表示されます。



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。

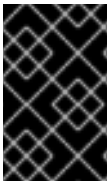


重要

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

1.4.8. 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>
```

1.4.9. 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
```

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。

1.5. DEPLOYMENT MANAGER テンプレートを使用した GCP 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、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クラスターを Google Cloud Platform (GCP) にインストールできます。

以下に、ユーザーによってプロビジョニングされるインフラストラクチャーのインストールを実行する手順を要約します。これらの手順を実行するか、独自の手順を作成するのに役立つ複数の Deployment Manager テンプレートが提供されます。他の方法を使用して必要なリソースを作成することもできます。これらのテンプレートはサンプルとしてのみ提供されます。

前提条件

- OpenShift Container Platform のインストールおよび更新 プロセスについての詳細を確認します。
- ファイアウォールを使用し、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。

1.5.1. 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。

1.5.2. GCP プロジェクトの設定

OpenShift Container Platform をインストールする前に、これをホストするように Google Cloud Platform (GCP) プロジェクトを設定する必要があります。

1.5.2.1. GCP プロジェクトの作成

OpenShift Container Platform をインストールするには、クラスターをホストするために Google Cloud Platform (GCP) アカウントでプロジェクトを作成する必要があります。

手順

- OpenShift Container Platform クラスターをホストするプロジェクトを作成します。GCP ドキュメントの「Creating and Managing Projects」を参照してください。

1.5.2.2. GCP での API サービスの有効化

Google Cloud Platform (GCP) プロジェクトでは、OpenShift Container Platform インストールを完了するために複数の API サービスへのアクセスが必要です。

前提条件

- クラスターをホストするプロジェクトを作成している。

手順

- クラスターをホストするプロジェクトで以下の必要な API サービスを有効にします。GCP ドキュメントの「サービスの有効化」を参照してください。

表1.11 必要な API サービス

API サービス	コンソールサービス名
Cloud Deployment Manager V2 API	**deploymentmanager.googleapis.com**
Compute Engine API	**compute.googleapis.com**
Google Cloud API	**cloudapis.googleapis.com**
Cloud Resource Manager API	**cloudresourcemanager.googleapis.com**
Google DNS API	**dns.googleapis.com**
IAM Service Account Credentials API	**iamcredentials.googleapis.com**
Identity and Access Management (IAM) API	**iam.googleapis.com**
Service Management API	**servicemanagement.googleapis.com**
Service Usage API	**serviceusage.googleapis.com**
Google Cloud Storage JSON API	**storage-api.googleapis.com**
Cloud Storage	**storage-component.googleapis.com**

1.5.2.3. GCP の DNS の設定

OpenShift Container Platform をインストールするには、使用する Google Cloud Platform (GCP) アカウントに、OpenShift Container Platform クラスターをホストする同じプロジェクトに専用のパブリックホストゾーンがなければなりません。このゾーンはドメインに対する権威を持っている必要があります。DNS サービスは、クラスターへの外部接続のためのクラスターの DNS 解決および名前検索を提供します。

手順

1. ドメイン、またはサブドメイン、およびレジストラーを特定します。既存のドメインおよびレジストラーを移行するか、GCP または他のソースから新規のものを取得できます。

注記

新規ドメインを購入する場合、関連する DNS の変更が伝播するのに時間がかかる場合があります。Google 経由でドメインを購入する方法についての詳細は、「Google ドメイン」を参照してください。

2. GCP プロジェクトにドメインまたはサブドメインのパブリックホストゾーンを作成します。GCP ドキュメントの「Creating public zones」を参照してください。
openshiftcorp.com などのルートドメインや、 **clusters.openshiftcorp.com** などのサブドメインを使用します。

3. ホストゾーンレコードから新規の権威ネームサーバーを抽出します。GCP ドキュメントの「Look up your Cloud DNS name servers」を参照してください。
通常は、4 つのネームサーバーがあります。

4. ドメインが使用するネームサーバーのレジストラーレコードを更新します。たとえば、ドメインを Google ドメインに登録している場合は、Google Domains Help で「How to switch to custom name servers」のトピックを参照してください。

5. ルートドメインを Google Cloud DNS に移行している場合は、DNS レコードを移行します。GCP ドキュメントの「Migrating to Cloud DNS」を参照してください。

6. サブドメインを使用する場合は、所属する会社の手順に従ってその委任レコードを親ドメインに追加します。このプロセスには、所属企業の IT 部門や、会社のルートドメインと DNS サービスを制御する部門へのリクエストが含まれる場合があります。

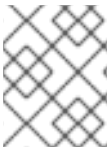
1.5.2.4. GCP アカウントの制限

OpenShift Container Platform クラスターは多くの Google Cloud Platform (GCP) コンポーネントを使用しますが、デフォルトの 割り当て (Quota) はデフォルトの OpenShift Container Platform クラスターをインストールする機能に影響を与えません。

3 つのコンピュートマシンおよび 3 つのコントロールプレーンマシンが含まれるデフォルトクラスターは以下のリソースを使用します。一部のリソースはブートストラッププロセス時にのみ必要となり、クラスターのデプロイ後に削除されることに注意してください。

表1.12 デフォルトのクラスターで使用される GCP リソース

サービス	Component	場所	必要なリソースの合計	ブートストラップ後に削除されるリソース
サービスアカウント	IAM	グローバル	5	0
ファイアウォールのルール	ネットワーク	グローバル	11	1
転送ルール	コンピュート	グローバル	2	0
ヘルスチェック	コンピュート	グローバル	2	0
イメージ	コンピュート	グローバル	1	0
ネットワーク	ネットワーク	グローバル	1	0
ルーター	ネットワーク	グローバル	1	0
ルート	ネットワーク	グローバル	2	0
サブネットワーク	コンピュート	グローバル	2	0
ターゲットプール	ネットワーク	グローバル	2	0



注記

インストール時にクォータが十分ではない場合、インストールプログラムは超過したクォータとリージョンの両方を示すエラーを表示します。

実際のクラスターサイズ、計画されるクラスターの拡張、およびアカウントに関連付けられた他のクラスターからの使用法を考慮してください。CPU、静的 IP アドレス、および永続ディスク SSD（ストレージ）のクォータは、ほとんどの場合に不十分になる可能性のあるものです。

以下のリージョンのいずれかにクラスターをデプロイする予定の場合、ストレージクォータの最大値を超え、CPU クォータ制限を超える可能性が高くなります。

- asia-east2
- asia-northeast2
- asia-south1
- australia-southeast1
- europe-north1
- europe-west2
- europe-west3
- europe-west6
- northamerica-northeast1
- southamerica-east1
- us-west2

GCP コンソールからリソースクォータを増やすことは可能ですが、サポートチケットを作成する必要がある場合があります。OpenShift Container Platform クラスターをインストールする前にサポートチケットを解決できるように、クラスターのサイズを早期に計画してください。

1.5.2.5. GCP でのサービスアカウントの作成

OpenShift Container Platform には、Google API でデータにアクセスするための認証および承認を提供する Google Cloud Platform (GCP) サービスアカウントが必要です。プロジェクトに必要なロールが含まれる既存の IAM サービスアカウントがない場合は、これを作成する必要があります。

前提条件

- クラスターをホストするプロジェクトを作成している。

手順

1. OpenShift Container Platform クラスターをホストするために使用するプロジェクトでサービスアカウントを作成します。GCP ドキュメントで「Creating a service account 」を参照してください。

2. サービスアカウントに適切なパーミッションを付与します。付随する個別のパーミッションを付与したり、**オーナー**ロールをこれに割り当てることができます。「特定のリソースのサービス アカウントへの役割の付与」を参照してください。

注記

サービスアカウントをプロジェクトの所有者にすることが必要なパーミッションを取得する最も簡単な方法になります。 つまりこれは、サービスアカウントはプロジェクトを完全に制御できることを意味します。この権限を提供することに伴うリスクが受け入れ可能であるかどうかを判別する必要があります。

3. JSON 形式でサービスアカウントキーを作成します。GCP ドキュメントの「サービスアカウント キーの作成」を参照してください。
クラスターを作成するには、サービスアカウントキーが必要になります。

1.5.2.5.1. 必要な GCP パーミッション

作成するサービスアカウントに**オーナー**ロールを割り当てると、OpenShift Container Platform のインストールに必要なパーミッションも含め、そのサービスアカウントにすべてのパーミッションが付与されます。 OpenShift Container Platform クラスターをデプロイするには、サービスアカウントに以下のパーミッションが必要です。

インストールプログラムに必要なロール

- Compute 管理者
- DNS 管理者
- セキュリティー管理者
- サービスアカウント管理者
- サービスアカウントユーザー
- ストレージ管理者

インストール時のネットワークリソースの作成に必要なロール

- DNS 管理者

ユーザーによってプロビジョニングされる GCP インフラストラクチャーに必要なロール

- Deployment Manager Editor
- サービスアカウントキー管理者

オプションのロール

クラスターで Operator の制限された認証情報を新たに作成できるようにするには、以下のロールを追加します。

- サービスアカウントキー管理者

ロールは、コントロールプレーンおよびコンピュートマシンが使用するサービスアカウントに適用されます。

表1.13 GCP サービスアカウントのパーミッション

アカウント	ロール
コントロールプレーン	**roles/compute.instanceAdmin**
	roles/compute.networkAdmin
	roles/compute.securityAdmin
	roles/storage.admin
	roles/iam.serviceAccountUser
コンピュート	**roles/compute.viewer**
	roles/storage.admin

1.5.2.6. サポートされている GCP リージョン

OpenShift Container Platform クラスターを以下の Google Cloud Platform (GCP) リージョンにデプロイできます。

- asia-east1 (Changhua County, Taiwan)
- asia-east2 (Hong Kong)
- asia-northeast1 (Tokyo, Japan)
- asia-northeast2 (Osaka, Japan)
- asia-south1 (Mumbai, India)
- asia-southeast1 (Jurong West, Singapore)
- australia-southeast1 (Sydney, Australia)
- europe-north1 (Hamina, Finland)
- europe-west1 (St. Ghislain, Belgium)
- europe-west2 (London, England, UK)
- europe-west3 (Frankfurt, Germany)
- europe-west4 (Eemshaven, Netherlands)
- europe-west6 (Zürich, Switzerland)
- northamerica-northeast1 (Montréal, Québec, Canada)
- southamerica-east1 (São Paulo, Brazil)

- us-central1 (Council Bluffs, Iowa, USA)
- us-east1 (Moncks Corner, South Carolina, USA)
- us-east4 (Ashburn, Northern Virginia, USA)
- us-west1 (The Dalles, Oregon, USA)
- us-west2 (Los Angeles, California, USA)

1.5.2.7. GCP の CLI ツールのインストールおよび設定

ユーザーによってプロビジョニングされるインフラストラクチャーを使用して Google Cloud Platform (GCP) に OpenShift Container Platform をインストールするには、GCP の CLI ツールをインストールし、設定する必要があります。

前提条件

- クラスターをホストするプロジェクトを作成している。
- サービスアカウントを作成し、これに必要なパーミッションを付与している。

手順

1. **$PATH** で以下のバイナリーをインストールします。
   - **gcloud**
   - **gsutil**

   GCP ドキュメントの「Install the latest Cloud SDK version」を参照してください。
2. 設定したサービスアカウントで、**gcloud** ツールを使用して認証します。

1.5.3. GCP のインストール設定ファイルの作成

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して OpenShift Container Platform を Google Cloud Platform (GCP) にインストールするには、インストールプログラムがクラスターをデプロイするために必要なファイルを生成し、クラスターが使用するマシンのみを作成するようにそれらのファイルを変更する必要があります。**install-config.yaml** ファイル、Kubernetes マニフェスト、および Ignition 設定ファイルを生成し、カスタマイズします。

1.5.3.1. インストール設定ファイルの作成

Google Cloud Platform (GCP) での OpenShift Container Platform のインストールをカスタマイズできます。

前提条件

- OpenShift Container Platform インストールプログラム、およびクラスターのプルシークレットを取得します。

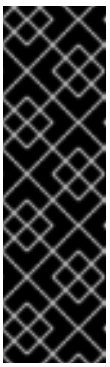
手順

1. **install-config.yaml** ファイルを作成します。

a. 次のコマンドを実行します。

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

1 **<installation_directory>** には、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

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。

b. プロンプト時に、クラウドの設定の詳細情報を指定します。

i. オプション: クラスターマシンにアクセスするために使用する SSH キーを選択します。

注記

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

ii. ターゲットに設定するプラットフォームとして **gcp** を選択します。

iii. コンピューター上で GCP アカウント用のサービスアカウントキーを設定していない場合、GCP からこれを取得してファイルの内容を貼り付けるか、またはファイルへの絶対パスを入力する必要があります。

iv. クラスターのプロビジョニングに使用するプロジェクト ID を選択します。デフォルト値は、設定したサービスアカウントによって指定されます。

v. クラスターをデプロイするリージョンを選択します。

vi. クラスターをデプロイするベースドメインを選択します。ベースドメインは、クラスターに作成したパブリック DNS ゾーンに対応します。

vii. クラスターの記述名を入力します。7 文字以上の名前を指定すると、クラスター名から生成されるインフラストラクチャー ID で最初の 6 文字のみが使用されます。

viii. 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を貼り付けます。

c. オプション: クラスターでコンピュートマシンをプロビジョニングするよう設定する必要がない場合は、 **install-config.yaml** ファイルで **compute** プールの **replicas** を **0** に設定してコンピュートプールを空にします。

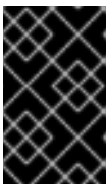
```
compute:
- hyperthreading: Enabled
```

```
name: worker
platform: {}
replicas: 0 (1)
```

(1) **0** に設定します。

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

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時に使用されます。このファイルを再利用する必要がある場合は、この段階でこれをバックアップしてください。

1.5.3.2. 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。

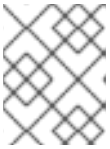
手順

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。

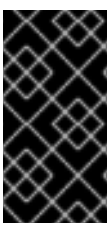
インストールプログラムは、指定の **install-config.yaml** ファイルのプロキシー設定を使用する **cluster** という名前のクラスター全体のプロキシーを作成します。プロキシー設定が指定されていない場合、**cluster** のプロキシーオブジェクトが依然として作成されますが、これには **spec** がありません。

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。

1.5.3.3. 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. コントロールプレーンマシンを定義する Kubernetes マニフェストファイルを削除します。

```
$ rm -f openshift/99_openshift-cluster-api_master-machines-*.yaml
```

これらのファイルを削除することで、クラスターがコントロールプレーンマシンを自動的に生成するのを防ぐことができます。

3. オプション: クラスターでコンピュートマシンをプロビジョニングする必要がない場合は、ワーカーマシンを定義する Kubernetes マニフェストファイルを削除します。

```
$ rm -f openshift/99_openshift-cluster-api_worker-machineset-*.yaml
```

ワーカーマシンは独自に作成し、管理するため、これらのマシンを初期化する必要はありません。

4. **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。

5. オプション: [illegible] を DNS ゾーンを作成するよう設定する必要がない場合

5. オプション: Ingress Operator を DNS レコードを作成するよう設定する必要がない場合は、**manifests/cluster-dns-02-config.yml** DNS 設定ファイルから **privateZone** および **publicZone** セクションを削除します。

```
apiVersion: config.openshift.io/v1
kind: DNS
metadata:
  creationTimestamp: null
  name: cluster
spec:
  baseDomain: example.openshift.com
  privateZone: 1
    id: mycluster-100419-private-zone
  publicZone: 2
    id: example.openshift.com
status: {}
```

1 2 これらのセクションを完全に削除します。

これを実行する場合、後のステップで Ingress DNS レコードを手動で追加する必要があります。

6. 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
```

追加リソース

- オプション: Ingress DNS レコードの追加

1.5.4. 一般的な変数のエクスポート

1.5.4.1. インフラストラクチャー名の抽出

Ignition 設定には、Google Cloud Platform (GCP) でクラスターを一意に識別するために使用できる一意のクラスター ID が含まれます。提供される Deployment Manager テンプレートにはこのインフラストラクチャー名への参照が含まれるため、これを抽出する必要があります。

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。
- クラスターの Ignition 設定ファイルを生成します。
- **jq** パッケージのインストール。

手順

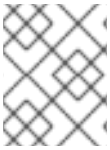
- Ignition 設定ファイルメタデータからインフラストラクチャー名を抽出し、表示するには、以下のコマンドを実行します。

```
$ jq -r .infraID /<installation_directory>/metadata.json 1
openshift-vw9j6 2
```

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. このコマンドの出力はクラスター名とランダムな文字列です。

1.5.4.2. Deployment Manager テンプレートの一般的な変数のエクスポート

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 Google Cloud Platform (GCP) で実行するのに役立つ指定の Deployment Manager テンプレートで使用される一般的な変数のセットをエクスポートする必要があります。



注記

特定の Deployment Manager テンプレートには、追加のエクスポートされる変数が必要になる場合があります。これについては、関連する手順で詳しく説明されています。

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。
- クラスターの Ignition 設定ファイルを生成します。
- **jq** パッケージのインストール。

手順

- 提供される Deployment Manager テンプレートで使用される以下の一般的な変数をエクスポートします。

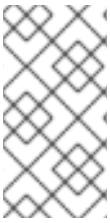
```
$ export BASE_DOMAIN='<base_domain>'
$ export BASE_DOMAIN_ZONE_NAME='<base_domain_zone_name>'
$ export NETWORK_CIDR='10.0.0.0/16'
$ export MASTER_SUBNET_CIDR='10.0.0.0/19'
$ export WORKER_SUBNET_CIDR='10.0.32.0/19'

$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
$ export CLUSTER_NAME=`jq -r .clusterName <installation_directory>/metadata.json`
```

```
$ export INFRA_ID=`jq -r .infraID <installation_directory>/metadata.json`
$ export PROJECT_NAME=`jq -r .gcp.projectID <installation_directory>/metadata.json`
$ export REGION=`jq -r .gcp.region <installation_directory>/metadata.json`
```

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。

1.5.5. GCP での VPC の作成

OpenShift Container Platform クラスターで使用する VPC を Google Cloud Platform (GCP) で作成する必要があります。各種の要件を満たすよう VPC をカスタマイズできます。VPC を作成する 1 つの方法として、提供されている Deployment Manager テンプレートを変更することができます。



注記

提供される Deployment Manager テンプレートを使用して GCP インフラストラクチャーを使用しない場合、提供される情報を確認し、インフラストラクチャーを手動で作成する必要があります。クラスターが適切に初期化されない場合、インストールログを用意して Red Hat サポートに問い合わせする必要がある可能性があります。

前提条件

- GCP アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。

手順

1. 本トピックの「**VPC の Deployment Manager テンプレート**」セクションを確認し、これを **01_vpc.py** としてコンピューターに保存します。 このテンプレートは、クラスターに必要な VPC について記述しています。
2. **01_xvdb.yaml** リソース定義ファイルを作成します。

```
$ cat <<EOF >01_vpc.yaml
imports:
- path: 01_vpc.py

resources:
- name: cluster-vpc
 type: 01_vpc.py
 properties:
   infra_id: '${INFRA_ID}' 1
   region: '${REGION}' 2

   master_subnet_cidr: '${MASTER_SUBNET_CIDR}' 3
   worker_subnet_cidr: '${WORKER_SUBNET_CIDR}' 4
EOF
```

1 **infra_id** は抽出手順で得られる **INFRA_ID** インフラストラクチャー名です。

2 **region** はクラスターをデプロイするリージョンです (例: **us-east1**)。

3. **master_subnet_cidr** はマスターサブネットの CIDR です (例: **10.0.0.0/19**)。
4. **worker_subnet_cidr** はワーカーサブネットの CIDR です (例: **10.0.32.0/19**)。

3. **gcloud** CLI を使用してデプロイメントを作成します。

```
$ gcloud deployment-manager deployments create ${INFRA_ID}-vpc --config 01_vpc.yaml
```

1.5.5.1. VPC の Deployment Manager テンプレート

以下の Deployment Manager テンプレートを使用して、OpenShift Container Platform クラスターに必要な VPC をデプロイすることができます。

01_vpc.py Deployment Manager テンプレート

```
def GenerateConfig(context):

    resources = [{
        'name': context.properties['infra_id'] + '-network',
        'type': 'compute.v1.network',
        'properties': {
            'region': context.properties['region'],
            'autoCreateSubnetworks': False
        }
    }, {
        'name': context.properties['infra_id'] + '-master-subnet',
        'type': 'compute.v1.subnetwork',
        'properties': {
            'region': context.properties['region'],
            'network': '$(ref.' + context.properties['infra_id'] + '-network.selfLink)',
            'ipCidrRange': context.properties['master_subnet_cidr']
        }
    }, {
        'name': context.properties['infra_id'] + '-worker-subnet',
        'type': 'compute.v1.subnetwork',
        'properties': {
            'region': context.properties['region'],
            'network': '$(ref.' + context.properties['infra_id'] + '-network.selfLink)',
            'ipCidrRange': context.properties['worker_subnet_cidr']
        }
    }, {
        'name': context.properties['infra_id'] + '-master-nat-ip',
        'type': 'compute.v1.address',
        'properties': {
            'region': context.properties['region']
        }
    }, {
        'name': context.properties['infra_id'] + '-worker-nat-ip',
        'type': 'compute.v1.address',
        'properties': {
            'region': context.properties['region']
        }
    }, {
        'name': context.properties['infra_id'] + '-router',
```

```
        'type': 'compute.v1.router',
        'properties': {
            'region': context.properties['region'],
            'network': '$(ref.' + context.properties['infra_id'] + '-network.selfLink)',
            'nats': [{
                'name': context.properties['infra_id'] + '-nat-master',
                'natIpAllocateOption': 'MANUAL_ONLY',
                'natIps': ['$(ref.' + context.properties['infra_id'] + '-master-nat-ip.selfLink)'],
                'minPortsPerVm': 7168,
                'sourceSubnetworkIpRangesToNat': 'LIST_OF_SUBNETWORKS',
                'subnetworks': [{
                    'name': '$(ref.' + context.properties['infra_id'] + '-master-subnet.selfLink)',
                    'sourceIpRangesToNat': ['ALL_IP_RANGES']
                }]
            }, {
                'name': context.properties['infra_id'] + '-nat-worker',
                'natIpAllocateOption': 'MANUAL_ONLY',
                'natIps': ['$(ref.' + context.properties['infra_id'] + '-worker-nat-ip.selfLink)'],
                'minPortsPerVm': 128,
                'sourceSubnetworkIpRangesToNat': 'LIST_OF_SUBNETWORKS',
                'subnetworks': [{
                    'name': '$(ref.' + context.properties['infra_id'] + '-worker-subnet.selfLink)',
                    'sourceIpRangesToNat': ['ALL_IP_RANGES']
                }]
            }]
        }
    }]

    return {'resources': resources}
```

1.5.6. GCP でのネットワークおよび負荷分散コンポーネントの作成

OpenShift Container Platform クラスターで使用するネットワークおよびロードバランシングを Google Cloud Platform (GCP) で設定する必要があります。これらのコンポーネントを作成する方法として、提供される Deployment Manager テンプレートを変更することができます。

注記

提供される Deployment Manager テンプレートを使用して GCP インフラストラクチャーを使用しない場合、提供される情報を確認し、インフラストラクチャーを手動で作成する必要があります。クラスターが適切に初期化されない場合、インストールログを用意して Red Hat サポートに問い合わせする必要がある可能性があります。

前提条件

- GCP アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。
- GCP で VPC および関連するサブネットを作成し、設定します。

手順

1. 本トピックの「**ネットワークおよびロードバランサーの Deployment Manager テンプレート**」セクションからテンプレートをコピーし、これを **02_infra.py** としてコンピューターに保存し

ます。 このテンプレートは、クラスターに必要なネットワークおよび負荷分散オブジェクトについて記述しています。

2. リソース定義で必要な以下の変数をエクスポートします。

```
$ export CLUSTER_NETWORK=`gcloud compute networks describe ${INFRA_ID}-network --format json | jq -r .selfLink`
```

3. **02_infra.yaml** リソース定義ファイルを作成します。

```
$ cat <<EOF >02_infra.yaml
imports:
- path: 02_infra.py

resources:
- name: cluster-infra
  type: 02_infra.py
  properties:
    infra_id: '${INFRA_ID}' (1)
    region: '${REGION}' (2)

    cluster_domain: '${CLUSTER_NAME}.${BASE_DOMAIN}' (3)
    cluster_network: '${CLUSTER_NETWORK}' (4)
EOF
```

(1) **infra_id** は抽出手順で得られる **INFRA_ID** インフラストラクチャー名です。

(2) **region** はクラスターをデプロイするリージョンです (例: **us-east1**)。

(3) **cluster_domain** はクラスターのドメインです (例: **openshift.example.com**)。

(4) **cluster_network** はクラスターネットワークの **selfLink** URL です。

4. **gcloud** CLI を使用してデプロイメントを作成します。

```
$ gcloud deployment-manager deployments create ${INFRA_ID}-infra --config 02_infra.yaml
```

5. このテンプレートは Deployment Manager の制限により DNS エントリーを作成しないので、手動で作成する必要があります。

   a. 以下の変数をエクスポートします。

```
$ export CLUSTER_IP=`gcloud compute addresses describe ${INFRA_ID}-cluster-public-ip --region=${REGION} --format json | jq -r .address`
```

   b. 外部 DNS エントリーを追加します。

```
$ if [ -f transaction.yaml ]; then rm transaction.yaml; fi
$ gcloud dns record-sets transaction start --zone ${BASE_DOMAIN_ZONE_NAME}
$ gcloud dns record-sets transaction add ${CLUSTER_IP} --name api.${CLUSTER_NAME}.${BASE_DOMAIN}. --ttl 60 --type A --zone ${BASE_DOMAIN_ZONE_NAME}
$ gcloud dns record-sets transaction execute --zone ${BASE_DOMAIN_ZONE_NAME}
```

c. 内部 DNS エントリーを追加します。

```
$ if [ -f transaction.yaml ]; then rm transaction.yaml; fi
$ gcloud dns record-sets transaction start --zone ${INFRA_ID}-private-zone
$ gcloud dns record-sets transaction add ${CLUSTER_IP} --name
api.${CLUSTER_NAME}.${BASE_DOMAIN}. --ttl 60 --type A --zone ${INFRA_ID}-
private-zone
$ gcloud dns record-sets transaction add ${CLUSTER_IP} --name api-
int.${CLUSTER_NAME}.${BASE_DOMAIN}. --ttl 60 --type A --zone ${INFRA_ID}-
private-zone
$ gcloud dns record-sets transaction execute --zone ${INFRA_ID}-private-zone
```

1.5.6.1. ネットワークおよびロードバランサーの Deployment Manager テンプレート

以下の Deployment Manager テンプレートを使用して、OpenShift Container Platform クラスターに必要なネットワークオブジェクトおよびロードバランサーをデプロイすることができます。

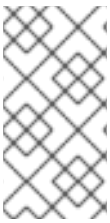
02_infra.py Deployment Manager テンプレート

```
def GenerateConfig(context):

    resources = [{
        'name': context.properties['infra_id'] + '-cluster-public-ip',
        'type': 'compute.v1.address',
        'properties': {
            'region': context.properties['region']
        }
    }, {
        'name': context.properties['infra_id'] + '-api-http-health-check',
        'type': 'compute.v1.httpHealthCheck',
        'properties': {
            'port': 6080,
            'requestPath': '/readyz'
        }
    }, {
        'name': context.properties['infra_id'] + '-api-target-pool',
        'type': 'compute.v1.targetPool',
        'properties': {
            'region': context.properties['region'],
            'healthChecks': ['$(ref.' + context.properties['infra_id'] + '-api-http-health-check.selfLink)'],
            'instances': []
        }
    }, {
        'name': context.properties['infra_id'] + '-api-forwarding-rule',
        'type': 'compute.v1.forwardingRule',
        'properties': {
            'region': context.properties['region'],
            'IPAddress': '$(ref.' + context.properties['infra_id'] + '-cluster-public-ip.selfLink)',
            'target': '$(ref.' + context.properties['infra_id'] + '-api-target-pool.selfLink)',
            'portRange': '6443'
        }
    }, {
        'name': context.properties['infra_id'] + '-ign-http-health-check',
        'type': 'compute.v1.httpHealthCheck',
        'properties': {
```

```
            'port': 22624,
            'requestPath': '/healthz'
        }
    }, {
        'name': context.properties['infra_id'] + '-ign-target-pool',
        'type': 'compute.v1.targetPool',
        'properties': {
            'region': context.properties['region'],
            'healthChecks': ['$(ref.' + context.properties['infra_id'] + '-ign-http-health-check.selfLink)'],
            'instances': []
        }
    }, {
        'name': context.properties['infra_id'] + '-ign-forwarding-rule',
        'type': 'compute.v1.forwardingRule',
        'properties': {
            'region': context.properties['region'],
            'IPAddress': '$(ref.' + context.properties['infra_id'] + '-cluster-public-ip.selfLink)',
            'target': '$(ref.' + context.properties['infra_id'] + '-ign-target-pool.selfLink)',
            'portRange': '22623'
        }
    }, {
        'name': context.properties['infra_id'] + '-private-zone',
        'type': 'dns.v1.managedZone',
        'properties': {
            'description': '',
            'dnsName': context.properties['cluster_domain'] + '.',
            'visibility': 'private',
            'privateVisibilityConfig': {
                'networks': [{
                    'networkUrl': context.properties['cluster_network']
                }]
            }
        }
    }]

    return {'resources': resources}
```

1.5.7. GCP でのファイアウォールルールおよび IAM ロールの作成

OpenShift Container Platform クラスターで使用するセキュリティーグループおよびロールを Google Cloud Platform (GCP) で作成する必要があります。これらのコンポーネントを作成する方法として、提供される Deployment Manager テンプレートを変更することができます。



注記

提供される Deployment Manager テンプレートを使用して GCP インフラストラクチャーを使用しない場合、提供される情報を確認し、インフラストラクチャーを手動で作成する必要があります。クラスターが適切に初期化されない場合、インストールログを用意して Red Hat サポートに問い合わせする必要がある可能性があります。

前提条件

- GCP アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。

- GCP で VPC および関連するサブネットを作成し、設定します。

手順

1. 本トピックの「**ファイアウォールおよび IAM ロールの Deployment Manager テンプレート**」セクションからテンプレートをコピーし、これを **03_security.py** としてコンピューターに保存します。このテンプレートは、クラスターに必要なセキュリティーグループおよびロールについて記述しています。

2. リソース定義で必要な以下の変数をエクスポートします。

```
$ export MASTER_NAT_IP=`gcloud compute addresses describe ${INFRA_ID}-master-nat-ip --region ${REGION} --format json | jq -r .address`
$ export WORKER_NAT_IP=`gcloud compute addresses describe ${INFRA_ID}-worker-nat-ip --region ${REGION} --format json | jq -r .address`
```

3. **03_security.yaml** リソース定義ファイルを作成します。

```
$ cat <<EOF >03_security.yaml
imports:
- path: 03_security.py

resources:
- name: cluster-security
  type: 03_security.py
  properties:
    infra_id: '${INFRA_ID}' (1)
    region: '${REGION}' (2)

    cluster_network: '${CLUSTER_NETWORK}' (3)
    network_cidr: '${NETWORK_CIDR}' (4)
    master_nat_ip: '${MASTER_NAT_IP}' (5)
    worker_nat_ip: '${WORKER_NAT_IP}' (6)
EOF
```

(1) **infra_id** は抽出手順で得られる **INFRA_ID** インフラストラクチャー名です。

(2) **region** はクラスターをデプロイするリージョンです (例: **us-east1**)。

(3) **cluster_network** はクラスターネットワークの **selfLink** URL です。

(4) **network_cidr** は VPC ネットワークの CIDR です (例: **10.0.0.0/16**)。

(5) **master_nat_ip** はマスター NAT の IP アドレスです (例: **34.94.100.1**)。

(6) **worker_nat_ip** はワーカー NAT の IP アドレスです (例: **34.94.200.1**)。

4. **gcloud** CLI を使用してデプロイメントを作成します。

```
$ gcloud deployment-manager deployments create ${INFRA_ID}-security --config 03_security.yaml
```

5. このテンプレートは Deployment Manager の制限によりポリシーバインディングを作成しないため、これらを手動で作成する必要があります。

```
$ export MASTER_SA=${INFRA_ID}-m@${PROJECT_NAME}.iam.gserviceaccount.com
$ gcloud projects add-iam-policy-binding ${PROJECT_NAME} --member
"serviceAccount:${MASTER_SA}" --role "roles/compute.instanceAdmin"
$ gcloud projects add-iam-policy-binding ${PROJECT_NAME} --member
"serviceAccount:${MASTER_SA}" --role "roles/compute.networkAdmin"
$ gcloud projects add-iam-policy-binding ${PROJECT_NAME} --member
"serviceAccount:${MASTER_SA}" --role "roles/compute.securityAdmin"
$ gcloud projects add-iam-policy-binding ${PROJECT_NAME} --member
"serviceAccount:${MASTER_SA}" --role "roles/iam.serviceAccountUser"
$ gcloud projects add-iam-policy-binding ${PROJECT_NAME} --member
"serviceAccount:${MASTER_SA}" --role "roles/storage.admin"

$ export WORKER_SA=${INFRA_ID}-w@${PROJECT_NAME}.iam.gserviceaccount.com
$ gcloud projects add-iam-policy-binding ${PROJECT_NAME} --member
"serviceAccount:${WORKER_SA}" --role "roles/compute.viewer"
$ gcloud projects add-iam-policy-binding ${PROJECT_NAME} --member
"serviceAccount:${WORKER_SA}" --role "roles/storage.admin"
```

6. サービスアカウントキーを作成し、後で使用できるようにこれをローカルに保存します。

```
$ gcloud iam service-accounts keys create service-account-key.json --iam-
account=${MASTER_SA}
```

1.5.7.1. ファイアウォールルールおよび IAM ロール用の Deployment Manager テンプレート

以下の Deployment Manager テンプレートを使用して、OpenShift Container Platform クラスターに必要なセキュリティーオブジェクトをデプロイすることができます。

03_security.py Deployment Manager テンプレート

```
def GenerateConfig(context):

  resources = [{
    'name': context.properties['infra_id'] + '-api',
    'type': 'compute.v1.firewall',
    'properties': {
        'network': context.properties['cluster_network'],
        'allowed': [{
            'IPProtocol': 'tcp',
            'ports': ['6443']
        }],
        'sourceRanges':  ['0.0.0.0/0'],
        'targetTags': [context.properties['infra_id'] + '-master']
    }
  }, {
    'name': context.properties['infra_id'] + '-mcs',
    'type': 'compute.v1.firewall',
    'properties': {
        'network': context.properties['cluster_network'],
        'allowed': [{
            'IPProtocol': 'tcp',
            'ports': ['22623']
```

```
        }],
        'sourceRanges':  [
            context.properties['network_cidr'],
            context.properties['master_nat_ip'],
            context.properties['worker_nat_ip']
        ],
        'targetTags': [context.properties['infra_id'] + '-master']
    }
}, {
    'name': context.properties['infra_id'] + '-health-checks',
    'type': 'compute.v1.firewall',
    'properties': {
        'network': context.properties['cluster_network'],
        'allowed': [{
            'IPProtocol': 'tcp',
            'ports': ['6080', '22624']
        }],
        'sourceRanges':  ['35.191.0.0/16', '209.85.152.0/22', '209.85.204.0/22'],
        'targetTags': [context.properties['infra_id'] + '-master']
    }
}, {
    'name': context.properties['infra_id'] + '-etcd',
    'type': 'compute.v1.firewall',
    'properties': {
        'network': context.properties['cluster_network'],
        'allowed': [{
            'IPProtocol': 'tcp',
            'ports': ['2379-2380']
        }],
        'sourceTags': [context.properties['infra_id'] + '-master'],
        'targetTags': [context.properties['infra_id'] + '-master']
    }
}, {
    'name': context.properties['infra_id'] + '-control-plane',
    'type': 'compute.v1.firewall',
    'properties': {
        'network': context.properties['cluster_network'],
        'allowed': [{
            'IPProtocol': 'tcp',
            'ports': ['10257']
        },{
            'IPProtocol': 'tcp',
            'ports': ['10259']
        }],
        'sourceTags': [
            context.properties['infra_id'] + '-master',
            context.properties['infra_id'] + '-worker'
        ],
        'targetTags': [context.properties['infra_id'] + '-master']
    }
}, {
    'name': context.properties['infra_id'] + '-internal-network',
    'type': 'compute.v1.firewall',
    'properties': {
        'network': context.properties['cluster_network'],
        'allowed': [{
```

```
            'IPProtocol': 'icmp'
        },{
            'IPProtocol': 'tcp',
            'ports': ['22']
        }],
        'sourceRanges':  [context.properties['network_cidr']],
        'targetTags': [
            context.properties['infra_id'] + '-master',
            context.properties['infra_id'] + '-worker'
        ]
    }
}, {
    'name': context.properties['infra_id'] + '-internal-cluster',
    'type': 'compute.v1.firewall',
    'properties': {
        'network': context.properties['cluster_network'],
        'allowed': [{
            'IPProtocol': 'udp',
            'ports': ['4789', '6081']
        },{
            'IPProtocol': 'tcp',
            'ports': ['9000-9999']
        },{
            'IPProtocol': 'udp',
            'ports': ['9000-9999']
        },{
            'IPProtocol': 'tcp',
            'ports': ['10250']
        },{
            'IPProtocol': 'tcp',
            'ports': ['30000-32767']
        },{
            'IPProtocol': 'udp',
            'ports': ['30000-32767']
        }],
        'sourceTags': [
            context.properties['infra_id'] + '-master',
            context.properties['infra_id'] + '-worker'
        ],
        'targetTags': [
            context.properties['infra_id'] + '-master',
            context.properties['infra_id'] + '-worker'
        ]
    }
}, {
    'name': context.properties['infra_id'] + '-master-node-sa',
    'type': 'iam.v1.serviceAccount',
    'properties': {
        'accountId': context.properties['infra_id'] + '-m',
        'displayName': context.properties['infra_id'] + '-master-node'
    }
}, {
    'name': context.properties['infra_id'] + '-worker-node-sa',
    'type': 'iam.v1.serviceAccount',
    'properties': {
        'accountId': context.properties['infra_id'] + '-w',
```

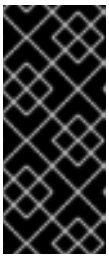
```
            'displayName': context.properties['infra_id'] + '-worker-node'
        }
    }]

    return {'resources': resources}
```

1.5.8. GCP インフラストラクチャー用の RHCOS クラスターイメージの作成

OpenShift Container Platform ノードに Google Cloud Platform (GCP) 用の有効な Red Hat Enterprise Linux CoreOS (RHCOS) イメージを使用する必要があります。

手順

1. 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 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>-gcp.tar** 形式の OpenShift Container Platform バージョン番号が含まれます。

2. 以下の変数をエクスポートします。

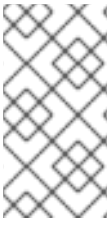
```
$ export IMAGE_SOURCE=<downloaded_image_file_path>
```

3. クラスターイメージを作成します。

```
$ gcloud compute images create "${INFRA_ID}-rhcos-image" \
    --source-uri="${IMAGE_SOURCE}"
```

1.5.9. GCP でのブートストラップマシンの作成

OpenShift Container Platform クラスターの初期化を実行する際に使用するブートストラップマシンを Google Cloud Platform (GCP) で作成する必要があります。このマシンを作成する方法として、提供される Deployment Manager テンプレートを変更することができます。



注記

提供されている Deployment Manager テンプレートを使用してブートストラップマシンを作成しない場合、指定される情報を確認し、インフラストラクチャーを手動で作成する必要があります。クラスターが適切に初期化されない場合、インストールログを用意して Red Hat サポートに問い合わせする必要がある可能性があります。

前提条件

- GCP アカウントを設定します。

- クラスターの Ignition 設定ファイルを生成します。
- GCP で VPC および関連するサブネットを作成し、設定します。
- GCP でネットワークおよびロードバランサーを作成し、設定します。
- コントロールプレーンおよびコンピュートロールを作成します。

手順

1. 本トピックの「**ブートストラップマシンの Deployment Manager テンプレート**」セクションからテンプレートをコピーし、これを **04_bootstrap.py** としてコンピューターに保存します。 このテンプレートは、クラスターに必要なブートストラップマシンについて記述しています。
2. リソース定義で必要な以下の変数をエクスポートします。

```
$ export CONTROL_SUBNET=`gcloud compute networks subnets describe ${INFRA_ID}-
master-subnet --region=${REGION} --format json | jq -r .selfLink`
$ export CLUSTER_IMAGE=`gcloud compute images describe ${INFRA_ID}-rhcos-image --
format json | jq -r .selfLink`
$ export ZONE_0=`gcloud compute regions describe ${REGION} --format=json | jq -r
.zones[0] | cut -d "/" -f9`
$ export ZONE_1=`gcloud compute regions describe ${REGION} --format=json | jq -r
.zones[1] | cut -d "/" -f9`
$ export ZONE_2=`gcloud compute regions describe ${REGION} --format=json | jq -r
.zones[2] | cut -d "/" -f9`
```

3. バケットを作成し、**bootstrap.ign** ファイルをアップロードします。

```
$ gsutil mb gs://${INFRA_ID}-bootstrap-ignition
$ gsutil cp bootstrap.ign gs://${INFRA_ID}-bootstrap-ignition/
```

4. Ignition 設定にアクセスするために使用するブートストラップインスタンスの署名付き URL を作成します。出力から URL を変数としてエクスポートします。

```
$ export BOOTSTRAP_IGN=`gsutil signurl -d 1h service-account-key.json \
   gs://${INFRA_ID}-bootstrap-ignition/bootstrap.ign | grep "^gs:" | awk '{print $5}'`
```

5. **04_bootstrap.yaml** リソース定義ファイルを作成します。

```
$ cat <<EOF >04_bootstrap.yaml
imports:
- path: 04_bootstrap.py

resources:
- name: cluster-bootstrap
  type: 04_bootstrap.py
  properties:
    infra_id: '${INFRA_ID}' (1)
    region: '${REGION}' (2)
    zone: '${ZONE_0}' (3)

    cluster_network: '${CLUSTER_NETWORK}' (4)
    control_subnet: '${CONTROL_SUBNET}' (5)
```

```
    image: '${CLUSTER_IMAGE}' (6)
    machine_type: 'n1-standard-4' (7)
    root_volume_size: '128' (8)

    bootstrap_ign: '${BOOTSTRAP_IGN}' (9)
EOF
```

(1) **infra_id** は抽出手順で得られる **INFRA_ID** インフラストラクチャー名です。

(2) **region** はクラスターをデプロイするリージョンです (例: **us-east1**)。

(3) **zone** はブートストラップインスタンスをデプロイするゾーンです (例: **us-east1-b**)。

(4) **cluster_network** はクラスターネットワークの **selfLink** URL です。

(5) **control_subnet** は、コントロールサブセットの **selfLink** URL です。

(6) **image** は RHCOS イメージの **selfLink** URL です。

(7) **machine_type** はインスタンスのマシンタイプです (例: **n1-standard-4**)。

(8) **bootstrap_ign** は上記の署名付き URL の作成時の URL 出力です。

6. **gcloud** CLI を使用してデプロイメントを作成します。

```
$ gcloud deployment-manager deployments create ${INFRA_ID}-bootstrap --config
04_bootstrap.yaml
```

7. Deployment Manager の制限によりテンプレートではロードバランサーのメンバーシップを管理しないため、ブートストラップマシンは手動で追加する必要があります。

```
$ gcloud compute target-pools add-instances \
    ${INFRA_ID}-api-target-pool --instances-zone="${ZONE_0}" --instances=${INFRA_ID}-
bootstrap
$ gcloud compute target-pools add-instances \
    ${INFRA_ID}-ign-target-pool --instances-zone="${ZONE_0}" --instances=${INFRA_ID}-
bootstrap
```

1.5.9.1. ブートストラップマシンの Deployment Manager テンプレート

以下の Deployment Mananger テンプレートを使用し、OpenShift Container Platform クラスターに必要なブートストラップマシンをデプロイすることができます。

04_bootstrap.py Deployment Manager テンプレート

```
def GenerateConfig(context):

    resources = [{
        'name': context.properties['infra_id'] + '-bootstrap-public-ip',
        'type': 'compute.v1.address',
        'properties': {
            'region': context.properties['region']
        }
    }, {
```

```
        'name': context.properties['infra_id'] + '-bootstrap-in-ssh',
        'type': 'compute.v1.firewall',
        'properties': {
            'network': context.properties['cluster_network'],
            'allowed': [{
                'IPProtocol': 'tcp',
                'ports': ['22']
            }],
            'sourceRanges':  ['0.0.0.0/0'],
            'targetTags': [context.properties['infra_id'] + '-bootstrap']
        }
    }, {
        'name': context.properties['infra_id'] + '-bootstrap',
        'type': 'compute.v1.instance',
        'properties': {
            'disks': [{
                'autoDelete': True,
                'boot': True,
                'initializeParams': {
                    'diskSizeGb': context.properties['root_volume_size'],
                    'sourceImage': context.properties['image']
                }
            }],
            'machineType': 'zones/' + context.properties['zone'] + '/machineTypes/' +
context.properties['machine_type'],
            'metadata': {
                'items': [{
                    'key': 'user-data',
                    'value': '{"ignition":{"config":{"replace":{"source":"' + context.properties['bootstrap_ign'] +
'","verification":{}}},"timeouts":{},"version":"2.1.0"},"networkd":{},"passwd":{},"storage":{},"systemd":{}}',
                }]
            },
            'networkInterfaces': [{
                'subnetwork': context.properties['control_subnet'],
                'accessConfigs': [{
                    'natIP': '$(ref.' + context.properties['infra_id'] + '-bootstrap-public-ip.address)'
                }]
            }],
            'tags': {
                'items': [
                    context.properties['infra_id'] + '-master',
                    context.properties['infra_id'] + '-bootstrap'
                ]
            },
            'zone': context.properties['zone']
        }
    }]

    return {'resources': resources}
```

1.5.10. GCP でのコントロールプレーンマシンの作成

クラスターで使用するコントロールプレーンマシンを Google Cloud Platform (GCP) で作成する必要があります。これらのマシンを作成する方法として、提供される Deployment Manager テンプレートを変更することができます。

注記

提供される Deployment Manager テンプレートを使用してコントロールプレーンマシンを使用しない場合、指定される情報を確認し、インフラストラクチャーを手動で作成する必要があります。クラスターが適切に初期化されない場合、インストールログを用意して Red Hat サポートに問い合わせする必要がある可能性があります。

前提条件

- GCP アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。
- GCP で VPC および関連するサブネットを作成し、設定します。
- GCP でネットワークおよびロードバランサーを作成し、設定します。
- コントロールプレーンおよびコンピュートロールを作成します。
- ブートストラップマシンを作成します。

手順

1. 本トピックの「**コントロールプレーンマシンの Deployment Manager テンプレート**」セクションからテンプレートをコピーし、これを **05_control_plane.py** としてコンピューターに保存します。このテンプレートは、クラスターに必要なコントロールプレーンのマシンについて記述しています。
2. リソース定義で必要な以下の変数をエクスポートします。

```
$ export MASTER_SERVICE_ACCOUNT_EMAIL=`gcloud iam service-accounts list | grep
"^${INFRA_ID}-master-node " | awk '{print $2}'`
$ export MASTER_IGNITION=`cat master.ign`
```

3. **05_control_plane.yaml** リソース定義ファイルを作成します。

```
$ cat <<EOF >05_control_plane.yaml
imports:
- path: 05_control_plane.py

resources:
- name: cluster-control-plane
  type: 05_control_plane.py
  properties:
    infra_id: '${INFRA_ID}' 1
    region: '${REGION}' 2
    zones: 3
    - '${ZONE_0}'
    - '${ZONE_1}'
    - '${ZONE_2}'

    control_subnet: '${CONTROL_SUBNET}' 4
    image: '${CLUSTER_IMAGE}' 5
    machine_type: 'n1-standard-4' 6
    root_volume_size: '128'
```

```
    service_account_email: '${MASTER_SERVICE_ACCOUNT_EMAIL}' 7

    ignition: '${MASTER_IGNITION}' 8
EOF
```

1. **infra_id** は抽出手順で得られる **INFRA_ID** インフラストラクチャー名です。
2. **region** はクラスターをデプロイするリージョンです (例: **us-east1**)。
3. **zones** は、ブートストラップインスタンスをデプロイするゾーンです (例: **us-east1-b**、**us-east1-c**、および **us-east1-d**)。
4. **control_subnet** は、コントロールサブセットの **selfLink** URL です。
5. **image** は RHCOS イメージの **selfLink** URL です。
6. **machine_type** はインスタンスのマシンタイプです (例: **n1-standard-4**)。
7. **service_account_email** は上記の手順で作成したマスターサービスアカウントのメールアドレスです。
8. **ignition** は **master.ign** ファイルの内容です。

4. **gcloud** CLI を使用してデプロイメントを作成します。

```
$ gcloud deployment-manager deployments create ${INFRA_ID}-control-plane --config
05_control_plane.yaml
```

5. Deployment Manager の制限によりテンプレートでは DNS エントリーを管理しないため、etcd エントリーを手動で追加する必要があります。

```
$ export MASTER0_IP=`gcloud compute instances describe ${INFRA_ID}-m-0 --zone
${ZONE_0} --format json | jq -r .networkInterfaces[0].networkIP`
$ export MASTER1_IP=`gcloud compute instances describe ${INFRA_ID}-m-1 --zone
${ZONE_1} --format json | jq -r .networkInterfaces[0].networkIP`
$ export MASTER2_IP=`gcloud compute instances describe ${INFRA_ID}-m-2 --zone
${ZONE_2} --format json | jq -r .networkInterfaces[0].networkIP`
$ if [ -f transaction.yaml ]; then rm transaction.yaml; fi
$ gcloud dns record-sets transaction start --zone ${INFRA_ID}-private-zone
$ gcloud dns record-sets transaction add ${MASTER0_IP} --name etcd-
0.${CLUSTER_NAME}.${BASE_DOMAIN}. --ttl 60 --type A --zone ${INFRA_ID}-private-
zone
$ gcloud dns record-sets transaction add ${MASTER1_IP} --name etcd-
1.${CLUSTER_NAME}.${BASE_DOMAIN}. --ttl 60 --type A --zone ${INFRA_ID}-private-
zone
$ gcloud dns record-sets transaction add ${MASTER2_IP} --name etcd-
2.${CLUSTER_NAME}.${BASE_DOMAIN}. --ttl 60 --type A --zone ${INFRA_ID}-private-
zone
$ gcloud dns record-sets transaction add \
  "0 10 2380 etcd-0.${CLUSTER_NAME}.${BASE_DOMAIN}." \
  "0 10 2380 etcd-1.${CLUSTER_NAME}.${BASE_DOMAIN}." \
  "0 10 2380 etcd-2.${CLUSTER_NAME}.${BASE_DOMAIN}." \
  --name _etcd-server-ssl._tcp.${CLUSTER_NAME}.${BASE_DOMAIN}. --ttl 60 --type SRV -
-zone ${INFRA_ID}-private-zone
$ gcloud dns record-sets transaction execute --zone ${INFRA_ID}-private-zone
```

6. Deployment Manager の制限により、テンプレートではロードバランサーのメンバーシップを管理しないため、コントロールプレーンマシンを手動で追加する必要があります。

```
$ gcloud compute target-pools add-instances ${INFRA_ID}-api-target-pool --instances-zone="${ZONE_0}" --instances=${INFRA_ID}-m-0
$ gcloud compute target-pools add-instances ${INFRA_ID}-api-target-pool --instances-zone="${ZONE_1}" --instances=${INFRA_ID}-m-1
$ gcloud compute target-pools add-instances ${INFRA_ID}-api-target-pool --instances-zone="${ZONE_2}" --instances=${INFRA_ID}-m-2
$ gcloud compute target-pools add-instances ${INFRA_ID}-ign-target-pool --instances-zone="${ZONE_0}" --instances=${INFRA_ID}-m-0
$ gcloud compute target-pools add-instances ${INFRA_ID}-ign-target-pool --instances-zone="${ZONE_1}" --instances=${INFRA_ID}-m-1
$ gcloud compute target-pools add-instances ${INFRA_ID}-ign-target-pool --instances-zone="${ZONE_2}" --instances=${INFRA_ID}-m-2
```

1.5.10.1. コントロールプレーンマシンの Deployment Manager テンプレート

以下の Deployment Mananger テンプレートを使用して、OpenShift Container Platform クラスターに必要なコントロールプレーンマシンをデプロイすることができます。

05_control_plane.py Deployment Manager テンプレート

```
def GenerateConfig(context):

    resources = [{
        'name': context.properties['infra_id'] + '-m-0',
        'type': 'compute.v1.instance',
        'properties': {
            'disks': [{
                'autoDelete': True,
                'boot': True,
                'initializeParams': {
                    'diskSizeGb': context.properties['root_volume_size'],
                    'diskType': 'zones/' + context.properties['zones'][0] + '/diskTypes/pd-ssd',
                    'sourceImage': context.properties['image']
                }
            }],
            'machineType': 'zones/' + context.properties['zones'][0] + '/machineTypes/' +
context.properties['machine_type'],
            'metadata': {
                'items': [{
                    'key': 'user-data',
                    'value': context.properties['ignition']
                }]
            },
            'networkInterfaces': [{
                'subnetwork': context.properties['control_subnet']
            }],
            'serviceAccounts': [{
                'email': context.properties['service_account_email'],
                'scopes': ['https://www.googleapis.com/auth/cloud-platform']
            }],
            'tags': {
                'items': [
```

```
                context.properties['infra_id'] + '-master',
            ]
        },
        'zone': context.properties['zones'][0]
    }
}, {
    'name': context.properties['infra_id'] + '-m-1',
    'type': 'compute.v1.instance',
    'properties': {
        'disks': [{
            'autoDelete': True,
            'boot': True,
            'initializeParams': {
                'diskSizeGb': context.properties['root_volume_size'],
                'diskType': 'zones/' + context.properties['zones'][1] + '/diskTypes/pd-ssd',
                'sourceImage': context.properties['image']
            }
        }],
        'machineType': 'zones/' + context.properties['zones'][1] + '/machineTypes/' +
context.properties['machine_type'],
        'metadata': {
            'items': [{
                'key': 'user-data',
                'value': context.properties['ignition']
            }]
        },
        'networkInterfaces': [{
            'subnetwork': context.properties['control_subnet']
        }],
        'serviceAccounts': [{
            'email': context.properties['service_account_email'],
            'scopes': ['https://www.googleapis.com/auth/cloud-platform']
        }],
        'tags': {
            'items': [
                context.properties['infra_id'] + '-master',
            ]
        },
        'zone': context.properties['zones'][1]
    }
}, {
    'name': context.properties['infra_id'] + '-m-2',
    'type': 'compute.v1.instance',
    'properties': {
        'disks': [{
            'autoDelete': True,
            'boot': True,
            'initializeParams': {
                'diskSizeGb': context.properties['root_volume_size'],
                'diskType': 'zones/' + context.properties['zones'][2] + '/diskTypes/pd-ssd',
                'sourceImage': context.properties['image']
            }
        }],
        'machineType': 'zones/' + context.properties['zones'][2] + '/machineTypes/' +
context.properties['machine_type'],
        'metadata': {
```

```
            'items': [{
                'key': 'user-data',
                'value': context.properties['ignition']
            }]
        },
        'networkInterfaces': [{
            'subnetwork': context.properties['control_subnet']
        }],
        'serviceAccounts': [{
            'email': context.properties['service_account_email'],
            'scopes': ['https://www.googleapis.com/auth/cloud-platform']
        }],
        'tags': {
            'items': [
                context.properties['infra_id'] + '-master',
            ]
        },
        'zone': context.properties['zones'][2]
    }
}]

return {'resources': resources}
```

1.5.11. ブートストラップの完了を待機し、GCP のブートストラップリソースを削除します。

Google Cloud Platform (GCP) ですべての必要なインフラストラクチャーを作成した後に、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

- GCP アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。
- GCP で VPC および関連するサブネットを作成し、設定します。
- GCP でネットワークおよびロードバランサーを作成し、設定します。
- コントロールプレーンおよびコンピュートロールを作成します。
- ブートストラップマシンを作成します。
- コントロールプレーンマシンを作成します。

手順

1. インストールプログラムが含まれるディレクトリーに切り替え、以下のコマンドを実行します。

```
$ ./openshift-install wait-for bootstrap-complete --dir=<installation_directory> \ 1
    --log-level info 2
```

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。

コマンドが **FATAL** 警告を出さずに終了する場合、実稼働用のコントロールプレーンは初期化されています。

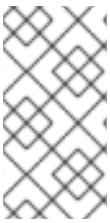
2. ブートストラップリソースを削除します。

```
$ gcloud compute target-pools remove-instances ${INFRA_ID}-api-target-pool --instances-zone="${ZONE_0}" --instances=${INFRA_ID}-bootstrap
$ gcloud compute target-pools remove-instances ${INFRA_ID}-ign-target-pool --instances-zone="${ZONE_0}" --instances=${INFRA_ID}-bootstrap
$ gsutil rm gs://${INFRA_ID}-bootstrap-ignition/bootstrap.ign
$ gsutil rb gs://${INFRA_ID}-bootstrap-ignition
$ gcloud deployment-manager deployments delete ${INFRA_ID}-bootstrap
```

1.5.12. GCP での追加のワーカーマシンの作成

Google Cloud Platform (GCP) でクラスターが使用するワーカーマシンを作成するには、それぞれのインスタンスを個別に起動するか、または自動スケーリンググループなどのクラスター外にある自動プロセスを実行します。OpenShift Container Platform の組み込まれたクラスタースケーリングメカニズムやマシン API を利用できます。

この例では、Deployment Manager テンプレートを使用して 1 つのインスタンスを手動で起動します。追加のインスタンスは、ファイル内に **06_worker.py** というタイプのリソースを追加して起動することができます。



注記

ワーカーマシンを使用するために提供される Deployment Manager テンプレートを使用しない場合は、提供される情報を確認し、インフラストラクチャーを手動で作成する必要があります。クラスターが適切に初期化されない場合、インストールログを用意して Red Hat サポートに問い合わせする必要がある可能性があります。

前提条件

- GCP アカウントを設定します。
- クラスターの Ignition 設定ファイルを生成します。
- GCP で VPC および関連するサブネットを作成し、設定します。
- GCP でネットワークおよびロードバランサーを作成し、設定します。
- コントロールプレーンおよびコンピュートロールを作成します。
- ブートストラップマシンを作成します。
- コントロールプレーンマシンを作成します。

手順

1. 本トピックの「**ワーカーマシンの Deployment Manager テンプレート**」からテンプレートをコピーし、これを **06_worker.py** としてコンピューターに保存します。このテンプレートは、クラスターに必要なワーカーマシンについて記述しています。

2. リソース定義で必要な以下の変数をエクスポートします。

```
$ export COMPUTE_SUBNET=`gcloud compute networks subnets describe ${INFRA_ID}-worker-subnet --region=${REGION} --format json | jq -r .selfLink`
$ export WORKER_SERVICE_ACCOUNT_EMAIL=`gcloud iam service-accounts list | grep "^${INFRA_ID}-worker-node " | awk '{print $2}'`
$ export WORKER_IGNITION=`cat worker.ign`
```

3. **06_worker.yaml** リソース定義ファイルを作成します。

```
$ cat <<EOF >06_worker.yaml
imports:
- path: 06_worker.py

resources:
- name: 'w-a-0' (1)
  type: 06_worker.py
  properties:
    infra_id: '${INFRA_ID}' (2)
    region: '${REGION}' (3)
    zone: '${ZONE_0}' (4)

    compute_subnet: '${COMPUTE_SUBNET}' (5)
    image: '${CLUSTER_IMAGE}' (6)
    machine_type: 'n1-standard-4' (7)
    root_volume_size: '128'
    service_account_email: '${WORKER_SERVICE_ACCOUNT_EMAIL}' (8)

    ignition: '${WORKER_IGNITION}' (9)
EOF
```

(1) **name** はワーカーマシンの名前です (例 : **w-a-0**)。

(2) **infra_id** は抽出手順で得られる **INFRA_ID** インフラストラクチャー名です。

(3) **region** はクラスターをデプロイするリージョンです (例: **us-east1**)。

(4) **zone** はワーカーマシンをデプロイするゾーンです (例: **us-east1-b**)。

(5) **compute_subnet** はコンピュートサブネットの **selfLink** URL です。

(6) **image** は RHCOS イメージの **selfLink** URL です。

(7) **machine_type** はインスタンスのマシンタイプです (例: **n1-standard-4**)。

(8) **service_account_email** は上記の手順で作成したワーカーサービスアカウントのメールアドレスです。

(9) **Ignition** は **worker.ign** ファイルの内容です。

4. オプション: 追加のインスタンスを起動する必要がある場合には、**06_worker.py** タイプの追加のリソースを **06_worker.yaml** リソース定義ファイルに組み込みます。

5. **gcloud** CLI を使用してデプロイメントを作成します。

```
$ gcloud deployment-manager deployments create ${INFRA_ID}-worker --config
06_worker.yaml
```

1.5.12.1. ワーカーマシンの Deployment Manager テンプレート

以下の Deloyment Manager テンプレートを使用し、OpenShift Container Platform クラスターに必要なワーカーマシンをデプロイすることができます。

06_worker.py Deployment Manager テンプレート

```
def GenerateConfig(context):

    resources = [{
        'name': context.properties['infra_id'] + '-' + context.env['name'],
        'type': 'compute.v1.instance',
        'properties': {
            'disks': [{
                'autoDelete': True,
                'boot': True,
                'initializeParams': {
                    'diskSizeGb': context.properties['root_volume_size'],
                    'sourceImage': context.properties['image']
                }
            }],
            'machineType': 'zones/' + context.properties['zone'] + '/machineTypes/' +
context.properties['machine_type'],
            'metadata': {
                'items': [{
                    'key': 'user-data',
                    'value': context.properties['ignition']
                }]
            },
            'networkInterfaces': [{
                'subnetwork': context.properties['compute_subnet']
            }],
            'serviceAccounts': [{
                'email': context.properties['service_account_email'],
                'scopes': ['https://www.googleapis.com/auth/cloud-platform']
            }],
            'tags': {
                'items': [
                    context.properties['infra_id'] + '-worker',
                ]
            },
            'zone': context.properties['zone']
        }
    }]

    return {'resources': resources}
```

1.5.13. CLI のインストール

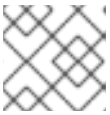
コマンドラインインターフェースを使用して OpenShift Container Platform と対話するために CLI をインストールすることができます。

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>
```

1.5.14. 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
```

1.5.15. 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
```

1.5.16. オプション: Ingress DNS レコードの追加

Kubernetes マニフェストの作成および Ignition 設定の生成時に DNS ゾーン設定を削除した場合、Ingress ロードバランサーをポイントする DNS レコードを手動で作成する必要があります。ワイルドカード ***.apps.{baseDomain}.** または特定のレコードのいずれかを作成できます。要件に基づいて A、CNAME その他のレコードを使用できます。

前提条件

- GCP アカウントを設定します。
- Kubernetes マニフェストの作成および Ignition 設定の生成時に DNS ゾーン設定を削除します。
- GCP で VPC および関連するサブネットを作成し、設定します。
- GCP でネットワークおよびロードバランサーを作成し、設定します。
- コントロールプレーンおよびコンピュートロールを作成します。
- ブートストラップマシンを作成します。
- コントロールプレーンマシンを作成します。
- ワーカーマシンを作成します。

手順

1. Ingress ルーターがロードバランサーを作成し、 **EXTERNAL-IP** フィールドにデータを設定するのを待機します。

```
$ oc -n openshift-ingress get service router-default
NAME             TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)                      AGE
router-default   LoadBalancer   172.30.18.154   35.233.157.184
80:32288/TCP,443:31215/TCP   98
```

2. パブリックおよびプライベートゾーンに A レコードを追加します。

```
$ export ROUTER_IP=`oc -n openshift-ingress get service router-default --no-headers | awk
'{print $4}'`

$ if [ -f transaction.yaml ]; then rm transaction.yaml; fi
$ gcloud dns record-sets transaction start --zone ${BASE_DOMAIN_ZONE_NAME}
$ gcloud dns record-sets transaction add ${ROUTER_IP} --name
\*.apps.${CLUSTER_NAME}.${BASE_DOMAIN}. --ttl 300 --type A --zone
${BASE_DOMAIN_ZONE_NAME}
$ gcloud dns record-sets transaction execute --zone ${BASE_DOMAIN_ZONE_NAME}

$ if [ -f transaction.yaml ]; then rm transaction.yaml; fi
$ gcloud dns record-sets transaction start --zone ${INFRA_ID}-private-zone
$ gcloud dns record-sets transaction add ${ROUTER_IP} --name
\*.apps.${CLUSTER_NAME}.${BASE_DOMAIN}. --ttl 300 --type A --zone ${INFRA_ID}-
private-zone
$ gcloud dns record-sets transaction execute --zone ${INFRA_ID}-private-zone
```

ワイルドカードを使用する代わりに明示的なドメインを追加する場合は、クラスターのそれぞれの現行ルートのエントリーを作成できます。

```
$ oc get --all-namespaces -o jsonpath='{range .items[*]}{range .status.ingress[*]}{.host}{"\n"}
{end}{end}' routes
oauth-openshift.apps.your.cluster.domain.example.com
console-openshift-console.apps.your.cluster.domain.example.com
downloads-openshift-console.apps.your.cluster.domain.example.com
alertmanager-main-openshift-monitoring.apps.your.cluster.domain.example.com
grafana-openshift-monitoring.apps.your.cluster.domain.example.com
prometheus-k8s-openshift-monitoring.apps.your.cluster.domain.example.com
```

1.5.17. 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 GCP インストールの完了

Google Cloud Platform (GCP) の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 OpenShift Container Platform のインストールを開始した後は、クラスターが準備状態になるまでクラスターのイベントをモニターできます。

前提条件

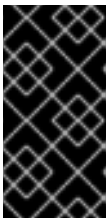
- OpenShift Container Platform クラスターのブートストラップマシンを、ユーザーによってプロビジョニングされる GCP インフラストラクチャーにデプロイします。
- **oc** CLI のインストールおよびログイン。

手順

1. クラスターのインストールを完了します。

```
$ ./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。



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。

2. クラスターの稼働状態を確認します。

   a. 以下のコマンドを実行し、現在のクラスターバージョンとステータスを表示します。

```
$ oc get clusterversion
NAME      VERSION   AVAILABLE   PROGRESSING   SINCE   STATUS
version             False       True          24m     Working towards 4.2.0-0: 99% complete
```

   b. 以下のコマンドを実行し、 Cluster Version Operator (CVO) を使用してコントロールプレーンで管理される Operator を表示します。

```
$ oc get clusteroperators
NAME                                 VERSION   AVAILABLE   PROGRESSING   DEGRADED
SINCE
authentication                       4.2.0-0   True        False         False      6m18s
cloud-credential                     4.2.0-0   True        False         False      17m
cluster-autoscaler                   4.2.0-0   True        False         False      80s
console                              4.2.0-0   True        False         False      3m57s
dns                                  4.2.0-0   True        False         False      22m
image-registry                       4.2.0-0   True        False         False      5m4s
ingress                              4.2.0-0   True        False         False      4m38s
insights                             4.2.0-0   True        False         False      21m
kube-apiserver                       4.2.0-0   True        False         False      12m
kube-controller-manager              4.2.0-0   True        False         False      12m
kube-scheduler                       4.2.0-0   True        False         False      11m
machine-api                          4.2.0-0   True        False         False      18m
machine-config                       4.2.0-0   True        False         False      22m
marketplace                          4.2.0-0   True        False         False      5m38s
monitoring                           4.2.0-0   True        False         False      86s
network                              4.2.0-0   True        False         False      14m
node-tuning                          4.2.0-0   True        False         False      6m8s
openshift-apiserver                  4.2.0-0   True        False         False      6m48s
openshift-controller-manager         4.2.0-0   True        False         False      12m
openshift-samples                    4.2.0-0   True        False         False      67s
operator-lifecycle-manager           4.2.0-0   True        False         False      15m
operator-lifecycle-manager-catalog   4.2.0-0   True        False         False      15m
```

```
operator-lifecycle-manager-packageserver   4.2.0-0   True        False         False
6m48s
service-ca                                 4.2.0-0   True        False         False      17m
service-catalog-apiserver                  4.2.0-0   True        False         False      6m18s
service-catalog-controller-manager         4.2.0-0   True        False         False      6m19s
storage                                    4.2.0-0   True        False         False      6m20s
```

c. 以下のコマンドを実行して、クラスター Pod を表示します。

```
$ oc get pods --all-namespaces
NAMESPACE                                               NAME
READY     STATUS      RESTARTS   AGE
kube-system                                             etcd-member-ip-10-0-3-111.us-east-
2.compute.internal              1/1       Running     0          35m
kube-system                                             etcd-member-ip-10-0-3-239.us-east-
2.compute.internal              1/1       Running     0          37m
kube-system                                             etcd-member-ip-10-0-3-24.us-east-
2.compute.internal              1/1       Running     0          35m
openshift-apiserver-operator                            openshift-apiserver-operator-6d6674f4f4-
h7t2t                           1/1       Running     1          37m
openshift-apiserver                                     apiserver-fm48r
1/1       Running     0          30m
openshift-apiserver                                     apiserver-fxkvv
1/1       Running     0          29m
openshift-apiserver                                     apiserver-q85nm
1/1       Running     0          29m
...
openshift-service-ca-operator                           openshift-service-ca-operator-66ff6dc6cd-
9r257                           1/1       Running     0          37m
openshift-service-ca                                    apiservice-cabundle-injector-695b6bcbc-cl5hm
1/1       Running     0          35m
openshift-service-ca                                    configmap-cabundle-injector-8498544d7-
25qn6                           1/1       Running     0          35m
openshift-service-ca                                    service-serving-cert-signer-6445fc9c6-wqdqn
1/1       Running     0          35m
openshift-service-catalog-apiserver-operator            openshift-service-catalog-apiserver-
operator-549f44668b-b5q2w       1/1       Running     0          32m
openshift-service-catalog-controller-manager-operator   openshift-service-catalog-
controller-manager-operator-b78cr2lnm     1/1       Running     0          31m
```

現在のクラスターバージョンが **AVAILABLE**の場合、インストールが完了します。

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。

1.6. GCP でのクラスターのアンインストール

Google Cloud Platform (GCP) にデプロイしたクラスターを削除できます。

1.6.1. インストーラーでプロビジョニングされるインフラストラクチャーを使用するクラスターの削除

インストーラーでプロビジョニングされるインフラストラクチャーを使用するクラスターは、クラウドから削除できます。

前提条件

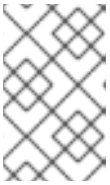
- クラスターをデプロイするために使用したインストールプログラムのコピーがあること。
- クラスター作成時にインストールプログラムが生成したファイルがあること。

手順

1. クラスターをインストールするために使用したコンピューターから、以下のコマンドを実行します。

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

1. **<installation_directory>** には、インストールファイルを保存したディレクトリーへのパスを指定します。
2. 異なる詳細情報を表示するには、 **info** ではなく、**warn**、**debug**、または **error** を指定します。



注記

クラスターのクラスター定義ファイルが含まれるディレクトリーを指定する必要があります。クラスターを削除するには、インストールプログラムでこのディレクトリーにある **metadata.json** ファイルが必要になります。

2. オプション: **<installation_directory>** ディレクトリーおよび OpenShift Container Platform インストールプログラムを削除します。