OpenShift Container Platform 4.13

Alibaba へのインストール

Alibaba Cloud に OpenShift Container Platform をインストールする

Last Updated: 2025-11-10

OpenShift Container Platform 4.13 Alibaba へのインストール

Alibaba Cloud に OpenShift Container Platform をインストールする

Legal Notice

Abstract

このドキュメントでは、Alibaba Cloud に OpenShift Container Platform をインストールする方法を説明します。

Table of Contents

第1章 ALIBABA CLOUD へのインストールの準備

重要

OpenShift Container Platform 上の Alibaba Cloud は、テクノロジープレビュー機能としてのみ利用できます。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行い、フィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

1.1. 前提条件

- OpenShift Container Platform のインストールおよび更新 プロセスの詳細を確認している。
- クラスターインストール方法の選択およびそのユーザー向けの準備 を確認している。

1.2. OPENSHIFT CONTAINER PLATFORM を ALIBABA CLOUD にインストールするための要件

OpenShift Container Platform を Alibaba Cloud にインストールする前に、ドメインを設定および登録し、インストール用の Resource Access Management (RAM) ユーザーを作成し、インストール用にサポートされている Alibaba Cloud データセンターのリージョンとゾーンを確認する必要があります。

1.3. ALIBABA CLOUD ドメインの登録と設定

OpenShift Container Platform をインストールするには、使用する Alibaba Cloud アカウントに専用のパブリックホストゾーンが必要です。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。このサービスは、クラスターへの外部接続のためのクラスター 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。既存のドメインとレジストラーを移行するか、Alibaba Cloud または別のソースから新しいドメインを取得することができます。

注記

Alibaba Cloud を介して新しいドメインを購入した場合、関連する DNS の変更が反映されるまでに時間がかかります。Alibaba Cloud を介したドメインの購入の詳細については、Alibaba Cloud ドメイン を参照してください。

2. 既存のドメインとレジストラーを使用している場合は、その DNS を Alibaba Cloud に移行します。Alibaba Cloud のドキュメントの Domain name transfer を参照してください。
3. ドメインの DNS を設定します。これには以下が含まれます。
   - ジェネリックドメイン名を登録します。

- ドメイン名の実名検証を完了します。
- インターネットコンテンツプロバイダー (ICP) のファイリングを申請します。
- ドメイン名解決を有効にします。
  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。

4. サブドメインを使用している場合は、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。

1.4. サポートされている ALIBABA リージョン

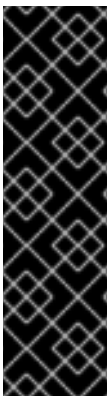
OpenShift Container Platform クラスターを Alibaba の**リージョンとゾーン**のドキュメント にリストされているリージョンにデプロイできます。

1.5. 次のステップ

- 必要な Alibaba Cloud リソースを作成します。

第2章 必要な ALIBABA CLOUD リソースの作成

OpenShift Container Platform をインストールする前に、Alibaba Cloud コンソールを使用して、OpenShift Container Platform を Alibaba Cloud にインストールするための十分な権限を持つ Resource Access Management (RAM) ユーザーを作成する必要があります。このユーザーには、新しい RAM ユーザーを作成するための権限も必要です。**ccoctl** ツールを設定および使用して、OpenShift Container Platform コンポーネントに必要な権限を持つ新しいクレデンシャルを作成することもできます。



重要

OpenShift Container Platform 上の Alibaba Cloud は、テクノロジープレビュー機能としてのみ利用できます。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行い、フィードバックを提供していただくことを目的としています。

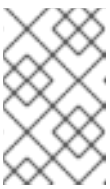
Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

2.1. 必要な RAM ユーザーの作成

インストールには、十分な権限を持つ Alibaba Cloud Resource Access Management (RAM) ユーザーが必要です。Alibaba Cloud Resource Access Management コンソールを使用して、新しいユーザーを作成したり、既存のユーザーを変更したりできます。後で、このユーザーの権限に基づいて、OpenShift Container Platform でクレデンシャルを作成します。

RAM ユーザーを設定するときは、次の要件を必ず考慮してください。

- ユーザーは、Alibaba Cloud AccessKey ID と AccessKey シークレットのペアを持っている必要があります。
  - 新規ユーザーの場合、ユーザーの作成時に Access Mode に **Open API Access** を選択できます。このモードでは、必要な AccessKey ペアが生成されます。
  - 既存のユーザーの場合は、AccessKey ペアを追加するか、そのユーザーの AccessKey のペアを取得 できます。



注記

作成されると、AccessKey シークレットは 1 回だけ表示されます。API 呼び出しには AccessKey ペアが必要であるため、AccessKey ペアをすぐに保存する必要があります。

- AccessKey ID とシークレットをローカルコンピューターの **~/.alibabacloud/credentials** ファイル に追加します。コンソールにログインすると、Alibaba Cloud によってこのファイルが自動的に作成されます。Cloud Credential Operator (CCO) ユーティリティー (ccoutil) は、**Credential Request** オブジェクトを処理するときにこれらの認証情報を使用します。以下に例を示します。

```
[default]                    # Default client
type = access_key            # Certification type: access_key
access_key_id = LTAI5t8cefXKmt           # Key (1)
```

```
access_key_secret = wYx56mszAN4Uunfh          # Secret
```

1 ここに AccessKeyID と AccessKeySecret を追加します。

- RAM ユーザーは、アカウントが OpenShift Container Platform クラスターを作成するための十分なパーミッションを持っていることを確認するために **AdministratorAccess** ポリシーを持っている必要があります。このポリシーは、すべての Alibaba Cloud リソースを管理するための権限を付与します。
AdministratorAccess ポリシーを RAM ユーザーにアタッチすると、そのユーザーにすべての Alibaba Cloud サービスとリソースへのフルアクセスが許可されます。フルアクセス権を持つユーザーを作成したくない場合は、インストールのために RAM ユーザーに追加できる次のアクションを使用してカスタムポリシーを作成します。これらのアクションは、OpenShift Container Platform をインストールするのに十分です。

ヒント

次の JSON コードをコピーして Alibaba Cloud コンソールに貼り付け、カスタムポリシーを作成できます。カスタムポリシー作成の詳細については、Alibaba Cloud のドキュメントの Create a custom policy を参照してください。

例2.1 カスタムポリシー JSON ファイルの例

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "tag:ListTagResources",
        "tag:UntagResources"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "vpc:DescribeVpcs",
        "vpc:DeleteVpc",
        "vpc:DescribeVSwitches",
        "vpc:DeleteVSwitch",
        "vpc:DescribeEipAddresses",
        "vpc:DescribeNatGateways",
        "vpc:ReleaseEipAddress",
        "vpc:DeleteNatGateway",
        "vpc:DescribeSnatTableEntries",
        "vpc:CreateSnatEntry",
        "vpc:AssociateEipAddress",
        "vpc:ListTagResources",
        "vpc:TagResources",
        "vpc:DescribeVSwitchAttributes",
        "vpc:CreateVSwitch",
        "vpc:CreateNatGateway",
        "vpc:DescribeRouteTableList",
        "vpc:CreateVpc",
        "vpc:AllocateEipAddress",
```

```
        "vpc:ListEnhanhcedNatGatewayAvailableZones"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "ecs:ModifyInstanceAttribute",
        "ecs:DescribeSecurityGroups",
        "ecs:DeleteSecurityGroup",
        "ecs:DescribeSecurityGroupReferences",
        "ecs:DescribeSecurityGroupAttribute",
        "ecs:RevokeSecurityGroup",
        "ecs:DescribeInstances",
        "ecs:DeleteInstances",
        "ecs:DescribeNetworkInterfaces",
        "ecs:DescribeInstanceRamRole",
        "ecs:DescribeUserData",
        "ecs:DescribeDisks",
        "ecs:ListTagResources",
        "ecs:AuthorizeSecurityGroup",
        "ecs:RunInstances",
        "ecs:TagResources",
        "ecs:ModifySecurityGroupPolicy",
        "ecs:CreateSecurityGroup",
        "ecs:DescribeAvailableResource",
        "ecs:DescribeRegions",
        "ecs:AttachInstanceRamRole"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "pvtz:DescribeRegions",
        "pvtz:DescribeZones",
        "pvtz:DeleteZone",
        "pvtz:DeleteZoneRecord",
        "pvtz:BindZoneVpc",
        "pvtz:DescribeZoneRecords",
        "pvtz:AddZoneRecord",
        "pvtz:SetZoneRecordStatus",
        "pvtz:DescribeZoneInfo",
        "pvtz:DescribeSyncEcsHostTask",
        "pvtz:AddZone"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "slb:DescribeLoadBalancers",
        "slb:SetLoadBalancerDeleteProtection",
        "slb:DeleteLoadBalancer",
        "slb:SetLoadBalancerModificationProtection",
        "slb:DescribeLoadBalancerAttribute",
```

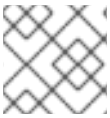
```
      "slb:AddBackendServers",
      "slb:DescribeLoadBalancerTCPListenerAttribute",
      "slb:SetLoadBalancerTCPListenerAttribute",
      "slb:StartLoadBalancerListener",
      "slb:CreateLoadBalancerTCPListener",
      "slb:ListTagResources",
      "slb:TagResources",
      "slb:CreateLoadBalancer"
    ],
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "ram:ListResourceGroups",
      "ram:DeleteResourceGroup",
      "ram:ListPolicyAttachments",
      "ram:DetachPolicy",
      "ram:GetResourceGroup",
      "ram:CreateResourceGroup",
      "ram:DeleteRole",
      "ram:GetPolicy",
      "ram:DeletePolicy",
      "ram:ListPoliciesForRole",
      "ram:CreateRole",
      "ram:AttachPolicyToRole",
      "ram:GetRole",
      "ram:CreatePolicy",
      "ram:CreateUser",
      "ram:DetachPolicyFromRole",
      "ram:CreatePolicyVersion",
      "ram:DetachPolicyFromUser",
      "ram:ListPoliciesForUser",
      "ram:AttachPolicyToUser",
      "ram:CreateUser",
      "ram:GetUser",
      "ram:DeleteUser",
      "ram:CreateAccessKey",
      "ram:ListAccessKeys",
      "ram:DeleteAccessKey",
      "ram:ListUsers",
      "ram:ListPolicyVersions"
    ],
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "oss:DeleteBucket",
      "oss:DeleteBucketTagging",
      "oss:GetBucketTagging",
      "oss:GetBucketCors",
      "oss:GetBucketPolicy",
      "oss:GetBucketLifecycle",
      "oss:GetBucketReferer",
      "oss:GetBucketTransferAcceleration",
```

```
            "oss:GetBucketLog",
            "oss:GetBucketWebSite",
            "oss:GetBucketInfo",
            "oss:PutBucketTagging",
            "oss:PutBucket",
            "oss:OpenOssService",
            "oss:ListBuckets",
            "oss:GetService",
            "oss:PutBucketACL",
            "oss:GetBucketLogging",
            "oss:ListObjects",
            "oss:GetObject",
            "oss:PutObject",
            "oss:DeleteObject"
          ],
          "Resource": "*",
          "Effect": "Allow"
        },
        {
          "Action": [
            "alidns:DescribeDomainRecords",
            "alidns:DeleteDomainRecord",
            "alidns:DescribeDomains",
            "alidns:DescribeDomainRecordInfo",
            "alidns:AddDomainRecord",
            "alidns:SetDomainRecordStatus"
          ],
          "Resource": "*",
          "Effect": "Allow"
        },
        {
          "Action": "bssapi:CreateInstance",
          "Resource": "*",
          "Effect": "Allow"
        },
        {
          "Action": "ram:PassRole",
          "Resource": "*",
          "Effect": "Allow",
          "Condition": {
            "StringEquals": {
              "acs:Service": "ecs.aliyuncs.com"
            }
          }
        }
      ]
    }
```

RAM ユーザーの作成と権限の付与の詳細については、Alibaba Cloud のドキュメントの Create a RAM user および Grant permissions to a RAM user を参照してください。

2.2. CLOUD CREDENTIAL OPERATOR ユーティリティーの設定

クラスター内コンポーネントごとに長寿命の RAM AccessKeys (AKs) を提供する RAM ユーザーとポリシーを割り当てるには、Cloud Credential Operator (CCO) ユーティリティー (**ccoctl**) バイナリーを抽出して準備します。



注記

ccoctl ユーティリティーは、Linux 環境で実行する必要がある Linux バイナリーです。

前提条件

- クラスター管理者のアクセスを持つ OpenShift Container Platform アカウントを使用できる。
- OpenShift CLI (**oc**) がインストールされている。

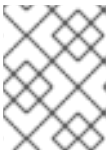
手順

1. 以下のコマンドを実行して、OpenShift Container Platform リリースイメージを取得します。

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 以下のコマンドを実行して、OpenShift Container Platform リリースイメージから CCO コンテナーイメージを取得します。

```
$ CCO_IMAGE=$(oc adm release info --image-for='cloud-credential-operator'
$RELEASE_IMAGE -a ~/.pull-secret)
```



注記

$RELEASE_IMAGE のアーキテクチャーが、**ccoctl** ツールを使用する環境のアーキテクチャーと一致していることを確認してください。

3. 以下のコマンドを実行して、OpenShift Container Platform リリースイメージ内の CCO コンテナーイメージから **ccoctl** バイナリーを抽出します。

```
$ oc image extract $CCO_IMAGE --file="/usr/bin/ccoctl" -a ~/.pull-secret
```

4. 次のコマンドを実行して、権限を変更して **ccoctl** を実行可能にします。

```
$ chmod 775 ccoctl
```

検証

- **ccoctl** が使用できることを確認するには、help ファイルを表示します。コマンドを実行するときは、相対ファイル名を使用します。以下に例を示します。

```
$ ./ccoctl.rhel9
```

出力例

```
OpenShift credentials provisioning tool

Usage:
```

```
  ccoctl [command]

Available Commands:
  alibabacloud Manage credentials objects for alibaba cloud
  aws          Manage credentials objects for AWS cloud
  gcp          Manage credentials objects for Google cloud
  help         Help about any command
  ibmcloud     Manage credentials objects for IBM Cloud
  nutanix      Manage credentials objects for Nutanix

Flags:
  -h, --help   help for ccoctl

Use "ccoctl [command] --help" for more information about a command.
```

関連情報

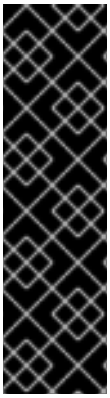
- 手動で維持された認証情報でクラスターを更新する準備

2.3. 次のステップ

- 次のいずれかの方法を使用して、OpenShift Container Platform インストールプログラムによってプロビジョニングされた Alibaba Cloud インフラストラクチャーにクラスターをインストールします。
  - **Alibaba Cloud へのクラスターの迅速なインストール**: デフォルトの設定オプションを使用して、クラスターを迅速にインストールできます。
  - **カスタマイズされたクラスターを Alibaba Cloud にインストール**: インストールプログラムを使用すると、インストールの段階で一部のカスタマイズを適用することができます。その他の数多くのカスタマイズオプションは、インストール後 に利用できます。

第3章 クラスターを ALIBABA CLOUD にすばやくインストールする

OpenShift Container Platform バージョン 4.13 では、デフォルトの設定オプションを使用するクラスターを Alibaba Cloud にインストールできます。



重要

OpenShift Container Platform 上の Alibaba Cloud は、テクノロジープレビュー機能としてのみ利用できます。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行い、フィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

3.1. 前提条件

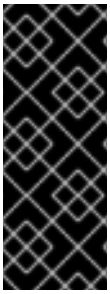
- OpenShift Container Platform のインストールおよび更新 プロセスの詳細を確認している。
- クラスターインストール方法の選択およびそのユーザー向けの準備 を確認している。
- ドメインを登録 している。
- ファイアウォールを使用する場合は、クラスターがアクセスを必要とするサイトを許可するようにファイアウォールを設定する必要がある。
- 必要な Alibaba Cloud リソースを作成している 。
- ご使用の環境でクラウド Resource Access Management (RAM) API にアクセスできない場合、または管理者レベルのクレデンシャルシークレットを kube-system namespace に保存したくない場合は、Resource Access Management (RAM) 認証情報を手動で作成および維持 することができます。

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

OpenShift Container Platform 4.13 では、クラスターをインストールするためにインターネットアクセスが必要になります。

インターネットへのアクセスは以下を実行するために必要です。

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



重要

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

3.3. クラスターノードの SSH アクセス用のキーペアの生成

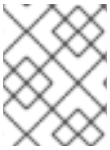
OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定できます。キーは、Ignition 設定ファイルを介して Red Hat Enterprise Linux CoreOS (RHCOS) ノードに渡され、ノードへの SSH アクセスを認証するために使用されます。このキーは各ノードの **core** ユーザーの **~/.ssh/authorized_keys** リストに追加され、パスワードなしの認証が可能になります。

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

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

重要

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



注記

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

手順

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

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

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

2. 公開 SSH キーを表示します。

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

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

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

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

注記

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

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

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

出力例

```
Agent pid 31874
```

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

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

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

出力例

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

次のステップ

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

3.4. インストールプログラムの取得

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

前提条件

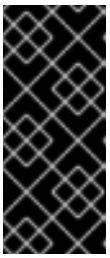
- 500 MB のローカルディスク領域がある Linux または macOS を実行するコンピューターが必要です。

手順

1. OpenShift Cluster Manager サイトの インフラストラクチャープロバイダー ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使用してログインします。アカウントがない場合はこれを作成します。

2. インフラストラクチャープロバイダーを選択します。

3. インストールタイプのページに移動し、ホストオペレーティングシステムとアーキテクチャーに対応するインストールプログラムをダウンロードして、インストール設定ファイルを保存するディレクトリーにファイルを配置します。



重要

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

重要

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

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

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

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

3.5. インストール設定ファイルの作成

Alibaba Cloud にインストールする OpenShift Container Platform クラスターをカスタマイズできます。

前提条件

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

手順

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

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

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

ディレクトリーを指定する場合:

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

注記

古い設定の再利用を回避するために、~/**.powervs** ディレクトリーは必ず削除してください。以下のコマンドを実行します。

```
$ rm -rf ~/.powervs
```

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. ターゲットとするプラットフォームとして **alibabacloud** を選択します。

iii. クラスターをデプロイするリージョンを選択します。

iv. 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。

v. クラスターの記述名を指定します。

vi. Red Hat OpenShift Cluster Manager からプルシークレット を貼り付けます。

2. クラスターを Alibaba Cloud にインストールするには、Cloud Credential Operator (CCO) が手動モードで動作する必要があります。**install-config.yaml** ファイルを変更して、**credentialsMode** パラメーターを **Manual** に設定します。

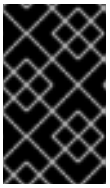
credentialsMode が Manual に設定された install-config.yaml 設定ファイルの例

```
apiVersion: v1
baseDomain: cluster1.example.com
credentialsMode: Manual 1
compute:
- architecture: amd64
  hyperthreading: Enabled
 ...
```

```
-
```

1 この行を追加して、 **credentialsMode** を **Manual** に設定します。

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



重要

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

3.6. 必要なインストールマニフェストの生成

クラスターがマシンを設定するために必要な 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。

手順

1. インストールプログラムが含まれているディレクトリーから次のコマンドを実行して、マニフェストを生成します。

```
$ openshift-install create manifests --dir <installation_directory>
```

ここで、

<installation_directory>
インストールプログラムがファイルを作成するディレクトリーを指定します。

3.7. CCOCTL ツールを使用した OPENSHIFT CONTAINER PLATFORM コンポーネントのクレデンシャルの作成

OpenShift Container Platform Cloud Credential Operator (CCO) ユーティリティーを使用して、Alibaba Cloud RAM ユーザーとクラスター内コンポーネントごとのポリシーの作成を自動化できます。

注記

デフォルトで、**ccoctl** はコマンドが実行されるディレクトリーにオブジェクトを作成します。オブジェクトを別のディレクトリーに作成するには、**--output-dir** フラグを使用します。この手順では、**<path_to_ccoctl_output_dir>** を使用してこの場所を参照します。

前提条件

以下が必要になります。

- **ccoctl** バイナリーを抽出して準備している。
- OpenShift Container Platform クラスターを作成するための十分な権限を持つ RAM ユーザーを作成している。

- その RAM ユーザーの AccessKeyID (**access_key_id**) と AccessKeySecret (**access_key_secret**) をローカルコンピューターの **~/.alibabacloud/credentials ファイル** に追加しました。

手順

1. 以下のコマンドを実行して、**$RELEASE_IMAGE** 変数を設定します。

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 以下のコマンドを実行して、OpenShift Container Platform リリースイメージから **CredentialsRequest** オブジェクトのリストを抽出します。

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --cloud=alibabacloud \
  --to=<path_to_directory_with_list_of_credentials_requests>/credrequests (1)
```

(1) **credrequests** は、**CredentialsRequest** オブジェクトのリストが格納されるディレクトリーです。ディレクトリーが存在しない場合、このコマンドはディレクトリーを作成します。

注記

このコマンドの実行には少し時間がかかる場合があります。

3. クラスターでクラスター機能を使用して 1 つ以上のオプションコンポーネントを無効にする場合は、無効なコンポーネントの **CredentialsRequest** カスタムリソースを削除します。

Alibaba Cloud 上の OpenShift Container Platform 4.13 の credrequests ディレクトリーの内容の例

```
0000_30_machine-api-operator_00_credentials-request.yaml (1)
0000_50_cluster-image-registry-operator_01-registry-credentials-request-alibaba.yaml (2)
0000_50_cluster-ingress-operator_00-ingress-credentials-request.yaml (3)
0000_50_cluster-storage-operator_03_credentials_request_alibaba.yaml (4)
```

(1) Machine API Operator CR が必要です。

(2) Image Registry Operator CR が必要です。

(3) Ingress Operator CR が必要です。

(4) Storage Operator CR はオプションのコンポーネントであり、クラスターで無効になっている場合があります。

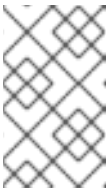
4. **ccoctl** ツールを使用して、**credrequests** ディレクトリーですべての **CredentialsRequest** オブジェクトを処理します。

   a. ツールを使用するには、次のコマンドを実行します。

```
$ ccoctl alibabacloud create-ram-users \
  --name <name> \
  --region=<alibaba_region> \
  --credentials-requests-dir=
<path_to_directory_with_list_of_credentials_requests>/credrequests \
  --output-dir=<path_to_ccoctl_output_dir>
```

ここで、

- **<name>** は、追跡用に作成されたクラウドリソースにタグを付けるために使用される名前です。
- **<alibaba_region>** は、クラウドリソースが作成される Alibaba Cloud リージョンです。
- **<path_to_directory_with_list_of_credentials_requests>/credrequests** は、コンポーネント **CredentialsRequest** オブジェクトのファイルを含むディレクトリーです。
- **<path_to_ccoctl_output_dir>** は、生成されたコンポーネントクレデンシャルシークレットが配置されるディレクトリーです。



注記

クラスターで **TechPreviewNoUpgrade** 機能セットによって有効化されたテクノロジープレビュー機能を使用している場合は、**--enable-tech-preview** パラメーターを含める必要があります。

出力例

```
2022/02/11 16:18:26 Created RAM User: user1-alicloud-openshift-machine-api-
alibabacloud-credentials
2022/02/11 16:18:27 Ready for creating new ram policy user1-alicloud-openshift-
machine-api-alibabacloud-credentials-policy-policy
2022/02/11 16:18:27 RAM policy user1-alicloud-openshift-machine-api-alibabacloud-
credentials-policy-policy has created
2022/02/11 16:18:28 Policy user1-alicloud-openshift-machine-api-alibabacloud-
credentials-policy-policy has attached on user user1-alicloud-openshift-machine-api-
alibabacloud-credentials
2022/02/11 16:18:29 Created access keys for RAM User: user1-alicloud-openshift-
machine-api-alibabacloud-credentials
2022/02/11 16:18:29 Saved credentials configuration to: user1-
alicloud/manifests/openshift-machine-api-alibabacloud-credentials-credentials.yaml
...
```

注記

RAM ユーザーは、同時に最大 2 つの AccessKey を持つことができます。**ccoctl alibabacloud create-ram-users** を 3 回以上実行すると、以前に生成されたマニフェストシークレットが古くなり、新しく生成されたシークレットを再適用する必要があります。

b. OpenShift Container Platform シークレットが作成されていることを確認します。

```
$ ls <path_to_ccoctl_output_dir>/manifests
```

出力例:

```
openshift-cluster-csi-drivers-alibaba-disk-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-alibabacloud-credentials-credentials.yaml
```

RAM ユーザーとポリシーが Alibaba Cloud にクエリーを実行して作成されていることを確認できます。詳細については、RAM ユーザーとポリシーのリスト表示に関する Alibaba Cloud のドキュメントを参照してください。

5. 生成されたクレデンシャルファイルをターゲットマニフェストディレクトリーにコピーします。

```
$ cp ./<path_to_ccoctl_output_dir>/manifests/*credentials.yaml
./<path_to_installation>dir>/manifests/
```

ここで、

<path_to_ccoctl_output_dir>

ccoctl alibabacloud create-ram-users コマンドによって作成されるディレクトリーを指定します。

<path_to_installation_dir>

インストールプログラムがファイルを作成するディレクトリーを指定します。

3.8. 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する。
- ホスト上のクラウドプロバイダーアカウントに、クラスターをデプロイするための適切な権限があることを確認してください。アカウントの権限が正しくないと、インストールプロセスが失敗し、不足している権限を示すエラーメッセージが表示されます。

手順

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

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

1. **<installation_directory>** に、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。

検証

クラスターのデプロイが正常に完了すると、次のようになります。

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

重要

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

出力例

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

重要

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

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

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

重要

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.13 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールします。

Linux への OpenShift CLI のインストール

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

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **Product Variant** ドロップダウンリストからアーキテクチャーを選択します。
3. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
4. **OpenShift v4.13 Linux Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。
5. アーカイブを展開します。

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

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

```
$ echo $PATH
```

検証

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

```
$ oc <command>
```

Windows への OpenShift CLI のインストール

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

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
3. **OpenShift v4.13 Windows Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。

4. ZIP プログラムでアーカイブを解凍します。
5. **oc** バイナリーを、**PATH** にあるディレクトリーに移動します。
   PATH を確認するには、コマンドプロンプトを開いて以下のコマンドを実行します。

```
C:\> path
```

検証

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

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

macOS への OpenShift CLI のインストール

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

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
3. **OpenShift v4.13 macOS Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。

注記

macOS arm64 の場合は、**OpenShift v4.13 macOS arm64 Client** エントリーを選択します。

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

```
$ echo $PATH
```

検証

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

```
$ oc <command>
```

3.10. CLI の使用によるクラスターへのログイン

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

前提条件

前提条件

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

手順

1. **kubeadmin** 認証情報をエクスポートします。

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

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

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

```
$ oc whoami
```

出力例

```
system:admin
```

3.11. WEB コンソールを使用したクラスターへのログイン

kubeadmin ユーザーは、OpenShift Container Platform のインストール後はデフォルトで存在します。OpenShift Container Platform Web コンソールを使用し、**kubeadmin** ユーザーとしてクラスターにログインできます。

前提条件

- インストールホストにアクセスできる。
- クラスターのインストールを完了しており、すべてのクラスター Operator が利用可能である。

手順

1. インストールホストで **kubeadmin-password** ファイルから **kubeadmin** ユーザーのパスワードを取得します。

```
$ cat <installation_directory>/auth/kubeadmin-password
```

注記

または、インストールホストで **<installation_directory>/.openshift_install.log** ログファイルから **kubeadmin** パスワードを取得できます。

2. OpenShift Container Platform Web コンソールルートをリスト表示します。

```
$ oc get routes -n openshift-console | grep 'console-openshift'
```

注記

または、インストールホストで **<installation_directory>/.openshift_install.log** ログファイルからで OpenShift Container Platform ルートを取得できます。

出力例

```
console     console-openshift-console.apps.<cluster_name>.<base_domain>           console
https   reencrypt/Redirect   None
```

3. Web ブラウザーで前述のコマンドの出力で詳細に説明されたルートに移動し、**kubeadmin** ユーザーとしてログインします。

3.12. OPENSHIFT CONTAINER PLATFORM の TELEMETRY アクセス

OpenShift Container Platform 4.13 では、クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは OpenShift Cluster Manager Hybrid Cloud Console に登録されます。

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

関連情報

- OpenShift Container Platform Web コンソールへ のアクセスと理解の詳細については、Web コンソールへのアクセス を参照してください。
- Telemetry サービスの詳細は、リモートヘルスモニタリング を参照して ください。

3.13. 次のステップ

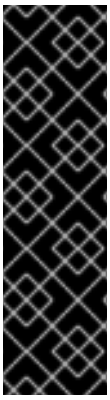
- インストールの検証
- クラスターをカスタマイズします。
- リモートヘルスレポート

第4章 カスタマイズによる ALIBABA CLOUD へのクラスターのインストール

OpenShift Container Platform バージョン 4.13 では、インストールプログラムが Alibaba Cloud で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。

注記

OpenShift Container Platform インストール設定のスコープは意図的に狭められています。単純さを確保し、確実にインストールを実行できるように設計されているためです。インストールが完了した後にさらに多くの OpenShift Container Platform 設定タスクを実行することができます。



重要

OpenShift Container Platform 上の Alibaba Cloud は、テクノロジープレビュー機能としてのみ利用できます。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行い、フィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

4.1. 前提条件

- OpenShift Container Platform のインストールおよび更新 プロセスの詳細を確認している。
- クラスターインストール方法の選択およびそのユーザー向けの準備 を確認している。
- ドメインを登録 している。
- ファイアウォールを使用する場合は、クラスターがアクセスを必要とするサイトを許可するようにファイアウォールを設定する必要がある。
- ご使用の環境でクラウド Resource Access Management (RAM) API にアクセスできない場合、または管理者レベルのクレデンシャルシークレットを **kube-system** namespace に保存したくない場合は、Resource Access Management (RAM) 認証情報を手動で作成および維持 することができます。

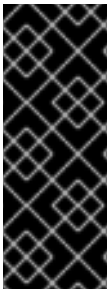
4.2. OPENSHIFT CONTAINER PLATFORM のインターネットアクセス

OpenShift Container Platform 4.13 では、クラスターをインストールするためにインターネットアクセスが必要になります。

インターネットへのアクセスは以下を実行するために必要です。

- OpenShift Cluster Manager Hybrid Cloud Console にアクセスし、インストールプログラムをダウンロードし、サブスクリプション管理を実行します。クラスターにインターネットアクセスがあり、Telemetry を無効にしない場合、そのサービスは有効なサブスクリプションでクラスターを自動的に使用します。

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



重要

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

4.3. クラスターノードの SSH アクセス用のキーペアの生成

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

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

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

重要

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

注記

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

手順

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

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

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

2. 公開 SSH キーを表示します。

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

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

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

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

注記

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

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

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

出力例

```
Agent pid 31874
```

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

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

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

出力例

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

次のステップ

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

4.4. インストールプログラムの取得

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

前提条件

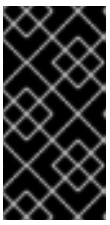
- 500 MB のローカルディスク領域がある Linux または macOS を実行するコンピューターが必要です。

手順

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

重要

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



重要

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

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

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

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

4.4.1. インストール設定ファイルの作成

Alibaba Cloud にインストールする OpenShift Container Platform クラスターをカスタマイズできます。

前提条件

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

手順

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

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

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

ディレクトリーを指定する場合:

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

注記

古い設定の再利用を回避するために、**~/.powervs** ディレクトリーは必ず削除してください。以下のコマンドを実行します。

```
$ rm -rf ~/.powervs
```

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. ターゲットとするプラットフォームとして **alibabacloud** を選択します。

iii. クラスターをデプロイするリージョンを選択します。

iv. 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。

v. クラスターの記述名を指定します。

vi. Red Hat OpenShift Cluster Manager からプルシークレット を貼り付けます。

2. クラスターを Alibaba Cloud にインストールするには、Cloud Credential Operator (CCO) が手動モードで動作する必要があります。**install-config.yaml** ファイルを変更して、**credentialsMode** パラメーターを **Manual** に設定します。

credentialsMode が Manual に設定された install-config.yaml 設定ファイルの例

```
apiVersion: v1
```

```
baseDomain: cluster1.example.com
credentialsMode: Manual (1)
compute:
- architecture: amd64
  hyperthreading: Enabled
 ...
```

1 この行を追加して、 **credentialsMode** を **Manual** に設定します。

3. **install-config.yaml** ファイルを変更します。利用可能なパラメーターの詳細は、「インストール設定パラメーター」のセクションを参照してください。
4. **install-config.yaml** ファイルをバックアップし、複数のクラスターをインストールするのに使用できるようにします。

重要

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

4.4.2. 必要なインストールマニフェストの生成

クラスターがマシンを設定するために必要な 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。

手順

1. インストールプログラムが含まれているディレクトリーから次のコマンドを実行して、マニフェストを生成します。

```
$ openshift-install create manifests --dir <installation_directory>
```

ここで、

<installation_directory>
インストールプログラムがファイルを作成するディレクトリーを指定します。

4.4.3. ccoctl ツールを使用した OpenShift Container Platform コンポーネントのクレデンシャルの作成

OpenShift Container Platform Cloud Credential Operator (CCO) ユーティリティーを使用して、Alibaba Cloud RAM ユーザーとクラスター内コンポーネントごとのポリシーの作成を自動化できます。

注記

デフォルトで、**ccoctl** はコマンドが実行されるディレクトリーにオブジェクトを作成します。オブジェクトを別のディレクトリーに作成するには、**--output-dir** フラグを使用します。この手順では、**<path_to_ccoctl_output_dir>** を使用してこの場所を参照します。

前提条件

以下が必要になります。

- **ccoctl** バイナリーを抽出して準備している。
- OpenShift Container Platform クラスターを作成するための十分な権限を持つ RAM ユーザーを作成している。
- その RAM ユーザーの AccessKeyID (**access_key_id**) と AccessKeySecret (**access_key_secret**) をローカルコンピューターの **~/.alibabacloud/credentials ファイル** に追加しました。

手順

1. 以下のコマンドを実行して、**$RELEASE_IMAGE** 変数を設定します。

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 以下のコマンドを実行して、OpenShift Container Platform リリースイメージから **CredentialsRequest** オブジェクトのリストを抽出します。

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --cloud=alibabacloud \
  --to=<path_to_directory_with_list_of_credentials_requests>/credrequests (1)
```

(1) **credrequests** は、**CredentialsRequest** オブジェクトのリストが格納されるディレクトリーです。ディレクトリーが存在しない場合、このコマンドはディレクトリーを作成します。

注記

このコマンドの実行には少し時間がかかる場合があります。

3. クラスターでクラスター機能を使用して 1 つ以上のオプションコンポーネントを無効にする場合は、無効なコンポーネントの **CredentialsRequest** カスタムリソースを削除します。

Alibaba Cloud 上の OpenShift Container Platform 4.13 の credrequests ディレクトリーの内容の例

```
0000_30_machine-api-operator_00_credentials-request.yaml (1)
0000_50_cluster-image-registry-operator_01-registry-credentials-request-alibaba.yaml (2)
0000_50_cluster-ingress-operator_00-ingress-credentials-request.yaml (3)
0000_50_cluster-storage-operator_03_credentials_request_alibaba.yaml (4)
```

(1) Machine API Operator CR が必要です。

(2) Image Registry Operator CR が必要です。

(3) Ingress Operator CR が必要です。

(4) Storage Operator CR はオプションのコンポーネントであり、クラスターで無効になって

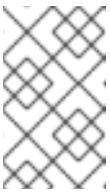
4. **ccoctl** ツールを使用して、**credrequests** ディレクトリーですべての **CredentialsRequest** オブジェクトを処理します。

   a. ツールを使用するには、次のコマンドを実行します。

```
$ ccoctl alibabacloud create-ram-users \
  --name <name> \
  --region=<alibaba_region> \
  --credentials-requests-dir=
<path_to_directory_with_list_of_credentials_requests>/credrequests \
  --output-dir=<path_to_ccoctl_output_dir>
```

   ここで、

   - **<name>** は、追跡用に作成されたクラウドリソースにタグを付けるために使用される名前です。
   - **<alibaba_region>** は、クラウドリソースが作成される Alibaba Cloud リージョンです。
   - **<path_to_directory_with_list_of_credentials_requests>/credrequests** は、コンポーネント **CredentialsRequest** オブジェクトのファイルを含むディレクトリーです。
   - **<path_to_ccoctl_output_dir>** は、生成されたコンポーネントクレデンシャルシークレットが配置されるディレクトリーです。



注記

クラスターで **TechPreviewNoUpgrade** 機能セットによって有効化されたテクノロジープレビュー機能を使用している場合は、**--enable-tech-preview** パラメーターを含める必要があります。

出力例

```
2022/02/11 16:18:26 Created RAM User: user1-alicloud-openshift-machine-api-
alibabacloud-credentials
2022/02/11 16:18:27 Ready for creating new ram policy user1-alicloud-openshift-
machine-api-alibabacloud-credentials-policy-policy
2022/02/11 16:18:27 RAM policy user1-alicloud-openshift-machine-api-alibabacloud-
credentials-policy-policy has created
2022/02/11 16:18:28 Policy user1-alicloud-openshift-machine-api-alibabacloud-
credentials-policy-policy has attached on user user1-alicloud-openshift-machine-api-
alibabacloud-credentials
2022/02/11 16:18:29 Created access keys for RAM User: user1-alicloud-openshift-
machine-api-alibabacloud-credentials
2022/02/11 16:18:29 Saved credentials configuration to: user1-
alicloud/manifests/openshift-machine-api-alibabacloud-credentials-credentials.yaml
...
```

注記

RAM ユーザーは、同時に最大 2 つの AccessKey を持つことができます。**ccoctl alibabacloud create-ram-users** を 3 回以上実行すると、以前に生成されたマニフェストシークレットが古くなり、新しく生成されたシークレットを再適用する必要があります。

b. OpenShift Container Platform シークレットが作成されていることを確認します。

```
$ ls <path_to_ccoctl_output_dir>/manifests
```

出力例:

```
openshift-cluster-csi-drivers-alibaba-disk-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-alibabacloud-credentials-credentials.yaml
```

RAM ユーザーとポリシーが Alibaba Cloud にクエリーを実行して作成されていることを確認できます。詳細については、RAM ユーザーとポリシーのリスト表示に関する Alibaba Cloud のドキュメントを参照してください。

5. 生成されたクレデンシャルファイルをターゲットマニフェストディレクトリーにコピーします。

```
$ cp ./<path_to_ccoctl_output_dir>/manifests/*credentials.yaml
./<path_to_installation>dir>/manifests/
```

ここで、

<path_to_ccoctl_output_dir>

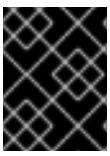
ccoctl alibabacloud create-ram-users コマンドによって作成されるディレクトリーを指定します。

<path_to_installation_dir>

インストールプログラムがファイルを作成するディレクトリーを指定します。

4.4.4. 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。



重要

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

4.4.4.1. 必須設定パラメーター

必須のインストール設定パラメーターは、以下の表で説明されています。

表4.1 必須パラメーター

パラメーター	説明	値
apiVersion	**install-config.yaml** コンテンツの API バージョン。現在のバージョンは **v1** です。インストールプログラムは、古い API バージョンもサポートしている場合があります。	文字列
baseDomain	クラウドプロバイダーのベースドメイン。ベースドメインは、OpenShift Container Platform クラスターコンポーネントへのルートを作成するために使用されます。クラスターの完全な DNS 名は、**<metadata.name>.<baseDomain>** 形式を使用する **baseDomain** と **metadata.name** パラメーターの値を組み合わせたものです。	**example.com** などの完全修飾ドメインまたはサブドメイン名。
metadata	Kubernetes リソース **ObjectMeta**。ここからは **name** パラメーターのみが消費されます。	オブジェクト
metadata.name	クラスターの名前。クラスターの DNS レコードはすべて **{{.metadata.name}}.{{.baseDomain}}** のサブドメインです。	**dev** などの小文字、ハイフン (**-**)、およびピリオド (**.**) が含まれる文字列。
platform	インストールを実行する特定のプラットフォームの設定: **alibabacloud**、**aws**、**baremetal**、**azure**、**gcp**、**ibmcloud**、**Nutanix**、**openstack**、**ovirt**、**powervs**、**vsphere**、または **{}**。**platform.<platform>** パラメーターに関する追加情報は、以下の表で特定のプラットフォームを参照してください。	オブジェクト

パラメーター	説明	値

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

4.4.4.2. ネットワーク設定パラメーター

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

IPv4 アドレスのみがサポートされます。

注記

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

表4.2 ネットワークパラメーター

パラメーター	説明	値
networking	クラスターのネットワークの設定。	オブジェクト  **注記** インストール後に **networking** オブジェクトで指定したパラメーターを変更することはできません。

パラメーター	説明	値
networking.networkType	インストールする Red Hat OpenShift Networking ネットワークプラグイン。	**OpenShiftSDN** または **OVNKubernetes** のいずれか。**OpenShiftSDN** は、すべての Linux ネットワークの Container Network Interface (CNI)プラグインです。**OVNKubernetes** は、Linux ネットワークと、Linux サーバーと Windows サーバーの両方を含む Linux ネットワークおよびハイブリッドネットワーク用の CNI プラグインです。デフォルトの値は **OVNkubernetes** です。
networking.clusterNetwork	Pod の IP アドレスブロック。 デフォルト値は **10.128.0.0/14** で、ホストの接頭辞は **/23** です。 複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。	オブジェクトの配列。以下に例を示します。 networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23
networking.clusterNetwork.cidr	**networking.clusterNetwork** を使用する場合に必須です。IP アドレスブロック。 IPv4 ネットワーク	CIDR (Classless Inter-Domain Routing) 表記の IP アドレスブロック。IPv4 ブロックの接頭辞長は **0** から **32** の間になります。
networking.clusterNetwork.hostPrefix	それぞれの個別ノードに割り当てるサブネット接頭辞長。たとえば、**hostPrefix** が **23** に設定される場合、各ノードに指定の **cidr** から **/23** サブネットが割り当てられます。**hostPrefix** 値の **23** は、510 (2^(32 - 23) - 2) Pod IP アドレスを提供します。	サブネット接頭辞。 デフォルト値は **23** です。
networking.serviceNetwork	サービスの IP アドレスブロック。デフォルト値は **172.30.0.0/16** です。 OpenShift SDN および OVN-Kubernetes ネットワークプラグインは、サービスネットワークの単一 IP アドレスブロックのみをサポートします。	CIDR 形式の IP アドレスブロックを持つ配列。以下に例を示します。 networking: serviceNetwork: - 172.30.0.0/16

パラメーター	説明	値
networking.machineNetwork	マシンの IP アドレスブロック。 複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。	オブジェクトの配列。以下に例を示します。 networking: machineNetwork: - cidr: 10.0.0.0/16
networking.machineNetwork.cidr	**networking.machineNetwork** を使用する場合に必須です。IP アドレスブロック。libvirt と IBM Power Virtual Server を除くすべてのプラットフォームのデフォルト値は **10.0.0.0/16** です。libvirt の場合、デフォルト値は **192.168.126.0/24** です。IBM Power Virtual Server の場合、デフォルト値は **192.168.0.0/24** です。	CIDR 表記の IP ネットワークブロック。 例: **10.0.0.0/16** **注記** 優先される NIC が置かれている CIDR に一致する **networking.machineNetwork** を設定します。

4.4.4.3. オプションの設定パラメーター

オプションのインストール設定パラメーターは、以下の表で説明されています。

表4.3 オプションのパラメーター

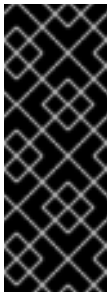
パラメーター	説明	値
additionalTrustBundle	ノードの信頼済み証明書ストアに追加される PEM でエンコードされた X.509 証明書バンドル。この信頼バンドルは、プロキシーが設定されている場合にも使用することができます。	文字列
capabilities	オプションのコアクラスターコンポーネントのインストールを制御します。オプションのコンポーネントを無効にすることで、OpenShift Container Platform クラスターのフットプリントを削減できます。詳細は、**インストール** の「クラスター機能ページ」を参照してください。	文字列配列

パラメーター	説明	値
capabilities.baselineCapabilitySet	有効にするオプション機能の初期セットを選択します。有効な値は **None**、**v4.11**、**v4.12**、**vCurrent** です。デフォルト値は **vCurrent** です。	文字列
capabilities.additionalEnabledCapabilities	オプションの機能のセットを、**baselineCapabilitySet** で指定したものを超えて拡張します。このパラメーターで複数の機能を指定できます。	文字列配列
cpuPartitioningMode	ワークロードパーティション設定を使用して、OpenShift Container Platform サービス、クラスター管理ワークロード、およびインフラストラクチャー Pod を分離し、予約された CPU セットで実行できます。ワークロードパーティショニングは、インストール中にのみ有効にできます。インストール後に無効にすることはできません。このフィールドはワークロードのパーティショニングを有効にしますが、特定の CPU を使用するようにワークロードを設定するわけではありません。詳細は、**スケーラビリティとパフォーマンス** セクションの **ワークロードパーティショニング** ページを参照してください。	**None** または **AllNodes**。デフォルト値は **None** です。
compute	コンピュートノードを形成するマシンの設定。	**MachinePool** オブジェクトの配列。
compute.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現在、さまざまなアーキテクチャーのクラスターはサポートされていません。すべてのプールは同じアーキテクチャーを指定する必要があります。有効な値はデフォルト **amd64** です。	String

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、同時マルチ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化されます。  **重要** 同時マルチ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.name	**compute** を使用する場合に必須です。マシンプールの名前。	**worker**
compute.platform	**compute** を使用する場合に必須です。このパラメーターを使用して、ワーカーマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は **controlPlane.platform** パラメーターの値に一致する必要があります。	**alibabacloud**、**aws**、**azure**、**gcp**、**ibmcloud**、**nutanix**、**openstack**、**ovirt**、**powervs**、**vsphere**、または **{}**
compute.replicas	プロビジョニングするコンピュートマシン (ワーカーマシンとしても知られる) の数。	**2** 以上の正の整数。デフォルト値は **3** です。
featureSet	機能セットのクラスターを有効にします。機能セットは、デフォルトで有効にされない OpenShift Container Platform 機能のコレクションです。インストール中に機能セットを有効にする方法の詳細は、「機能ゲートの使用による各種機能の有効化」を参照してください。	文字列。**TechPreviewNoUpgrade** など、有効にする機能セットの名前。
controlPlane	コントロールプレーンを形成するマシンの設定。	**MachinePool** オブジェクトの配列。

パラメーター	説明	値
controlPlane.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現在、さまざまなアーキテクチャーのクラスターはサポートされていません。すべてのプールは同じアーキテクチャーを指定する必要があります。有効な値はデフォルト **amd64** です。	String
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、同時マルチ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化されます。  **重要** 同時マルチスレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。	**Enabled** または **Disabled**
controlPlane.name	**controlPlane** を使用する場合に必須です。マシンプールの名前。	**master**
controlPlane.platform	**controlPlane** を使用する場合に必須です。このパラメーターを使用して、コントロールプレーンマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は **compute.platform** パラメーターの値に一致する必要があります。	**alibabacloud**、**aws**、**azure**、**gcp**、**ibmcloud**、**nutanix**、**openstack**、**ovirt**、**powervs**、**vsphere**、または **{}**
controlPlane.replicas	プロビジョニングするコントロールプレーンマシンの数。	サポートされる値は **3** のみです (これはデフォルト値です)。

パラメーター	説明	値
credentialsMode	Cloud Credential Operator (CCO) モード。CCO は、モードが指定されていない場合に指定される認証情報の機能を動的に判別しようとします。この場合、複数のモードがサポートされるプラットフォームで mint モードが優先されます。  **注記** すべてのクラウドプロバイダーですべての CCO モードがサポートされているわけではありません。CCO モードの詳細は、**Cluster Operators リファレンス** の **Cloud Credential Operator** を参照してください。  **注記** AWS アカウントでサービスコントロールポリシー (SCP) が有効になっている場合は、**credentialsMode** パラメーターを **Mint**、**Passthrough** または **Manual** に設定する必要があります。	**Mint**、**Passthrough**、**Manual**、または空の文字列 (**""**)。
imageContentSources	release-image コンテンツのソースおよびリポジトリー。	オブジェクトの配列。この表の以下の行で説明されているように、**source** およびオプションで **mirrors** が含まれます。
imageContentSources.source	**imageContentSources** を使用する場合に必須です。ユーザーが参照するリポジトリーを指定します (例: イメージプル仕様)。	文字列
imageContentSources.mirrors	同じイメージが含まれる可能性のあるリポジトリーを 1 つ以上指定します。	文字列の配列。

パラメーター	説明	値
publish	Kubernetes API、OpenShift ルートなどのクラスターのユーザーに表示されるエンドポイントをパブリッシュまたは公開する方法。	**Internal** または **External**。デフォルト値は **External** です。 このパラメーターを **Internal** に設定することは、クラウド以外のプラットフォームではサポートされません。  **重要** フィールドの値が **Internal** に設定されている場合、クラスターは機能しなくなります。詳細は、BZ#1953035 を参照してください。
sshKey	クラスターマシンへのアクセスを認証するための SSH キー。  **注記** インストールのデバッグまたは障害復旧を実行する必要のある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。	たとえば、**sshKey: ssh-ed25519 AAAA..** です。

1. すべてのクラウドプロバイダーですべての CCO モードがサポートされているわけではありません。CCO モードの詳細は、**認証と認可** コンテンツの「クラウドプロバイダーの認証情報の管理」を参照してください。

4.4.4.4. 追加の Alibaba Cloud 設定パラメーター

Alibaba Cloud の追加の設定パラメーターは、以下の表で説明されています。**alibabacloud** パラメーターは、Alibaba Cloud にインストールするときに使用される設定です。**defaultMachinePlatform** パラメーターは、独自のプラットフォーム設定を定義しないマシンプール用に Alibaba Cloud にインストールするときに使用されるデフォルト設定です。

これらのパラメーターは、指定されているコンピュートマシンとコントロールプレーンマシンの両方に適用されます。

注記

定義されている場合、パラメーター **compute.platform.alibabacloud** および **controlPlane.platform.alibabacloud** は、コンピューティングマシンとコントロールプレーンマシンの **platform.alibabacloud.defaultMachinePlatform** 設定をそれぞれ上書きします。

表4.4 オプションの Alibaba Cloud パラメーター

パラメーター	説明	値
compute.platform.alibabacloud.imageID	ECS インスタンスの作成に使用される imageID。ImageID はクラスターと同じリージョンに属している必要があります。	文字列。
compute.platform.alibabacloud.instanceType	InstanceType は、ECS インスタンスタイプを定義します。例: **ecs.g6.large**	文字列。
compute.platform.alibabacloud.systemDiskCategory	システムディスクのカテゴリーを定義します。例: **cloud_efficiency**、**cloud_essd**	文字列。
compute.platform.alibabacloud.systemDisksize	システムディスクのサイズをギビバイト (GiB) 単位で定義します。	integer
compute.platform.alibabacloud.zones	使用できるアベイラビリティーゾーンのリスト。例: **cn-hangzhou-h**、**cn-hangzhou-j**	文字列リスト。
controlPlane.platform.alibabacloud.imageID	ECS インスタンスの作成に使用される imageID。ImageID はクラスターと同じリージョンに属している必要があります。	文字列。
controlPlane.platform.alibabacloud.instanceType	InstanceType は、ECS インスタンスタイプを定義します。例: **ecs.g6.xlarge**	文字列。
controlPlane.platform.alibabacloud.systemDiskCategory	システムディスクのカテゴリーを定義します。例: **cloud_efficiency**、**cloud_essd**	文字列。

パラメーター	説明	値
controlPlane.platform.alibabacloud.systemDiskSize	システムディスクのサイズをギビバイト (GiB) 単位で定義します。	integer
controlPlane.platform.alibabacloud.zones	使用できるアベイラビリティーゾーンのリスト。例: **cn-hangzhou-h**、**cn-hangzhou-j**	文字列リスト。
platform.alibabacloud.region	必須。クラスターが作成される Alibaba Cloud リージョン。	文字列。
platform.alibabacloud.resourceGroupID	クラスターがインストールされている既存のリソースグループの ID。空の場合、インストールプログラムはクラスターの新しいリソースグループを作成します。	文字列。
platform.alibabacloud.tags	クラスター用に作成されたすべての Alibaba Cloud リソースに適用する追加のキーと値。	オブジェクト。
platform.alibabacloud.vpcID	クラスターをインストールする必要がある既存の VPC の ID。空の場合、インストールプログラムはクラスターの新しい VPC を作成します。	文字列。
platform.alibabacloud.vswitchIDs	クラスターリソースが作成される既存の VSwitches の ID リスト。既存の VSwitch は、既存の VPC も使用している場合にのみ使用できます。空の場合、インストールプログラムはクラスターの新しい VSwitch を作成します。	文字列リスト。
platform.alibabacloud.defaultMachinePlatform.imageID	コンピュートマシンとコントロールプレーンマシンの両方で、ECS インスタンスの作成に使用する必要があるイメージ ID。設定されている場合、イメージ ID はクラスターと同じリージョンに属している必要があります。	文字列。

パラメーター	説明	値
platform.alibabacloud.defaultMachinePlatform.instanceType	コンピュートマシンとコントロールプレーンマシンの両方で、ECS インスタンスの作成に使用される ECS インスタンスタイプ。例: **ecs.g6.xlarge**	文字列。
platform.alibabacloud.defaultMachinePlatform.systemDiskCategory	コンピュートマシンとコントロールプレーンマシンの両方におけるシステムディスクのカテゴリー。例: **cloud_efficiency**、**cloud_essd**。	文字列、たとえば ""、**cloud_efficiency**、**cloud_essd**。
platform.alibabacloud.defaultMachinePlatform.systemDiskSize	コンピュートマシンとコントロールプレーンマシンの両方で、ギビバイト (GiB) 単位のシステムディスクのサイズ。最小値は **120** です。	integer
platform.alibabacloud.defaultMachinePlatform.zones	コンピュートマシンとコントロールプレーンマシンの両方で、使用可能なアベイラビリティーゾーンのリスト。例: **cn-hangzhou-h**、**cn-hangzhou-j**	文字列リスト。
platform.alibabacloud.privateZoneID	クラスターの内部 API の DNS レコードを追加する既存のプライベートゾーンの ID。既存のプライベートゾーンは、既存の VPC も使用している場合にのみ使用できます。プライベートゾーンは、サブネットを含む VPC に関連付ける必要があります。プライベートゾーンを未設定のままにして、インストールプログラムがプライベートゾーンを作成するようにします。	文字列。

4.4.5. Alibaba Cloud 用にカスタマイズされた install-config.yaml ファイルのサンプル

インストール設定ファイル (**install-config.yaml**) をカスタマイズして、クラスターのプラットフォームに関する詳細を指定したり、必要なパラメーターの値を変更したりできます。

```
apiVersion: v1
baseDomain: alicloud-dev.devcluster.openshift.com
credentialsMode: Manual
compute:
```

```
- architecture: amd64
  hyperthreading: Enabled
  name: worker
  platform: {}
  replicas: 3
controlPlane:
  architecture: amd64
  hyperthreading: Enabled
  name: master
  platform: {}
  replicas: 3
metadata:
  creationTimestamp: null
  name: test-cluster 1
 networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  networkType: OVNKubernetes 2
  serviceNetwork:
  - 172.30.0.0/16
platform:
  alibabacloud:
    defaultMachinePlatform: 3
      instanceType: ecs.g6.xlarge
      systemDiskCategory: cloud_efficiency
      systemDiskSize: 200
    region: ap-southeast-1 4
    resourceGroupID: rg-acfnw6j3hyai 5
    vpcID: vpc-0xifdjerdibmaqvtjob2b 6
    vswitchIDs: 7
    - vsw-0xi8ycgwc8wv5rhviwdq5
    - vsw-0xiy6v3z2tedv009b4pz2
publish: External
pullSecret: '{"auths": {"cloud.openshift.com": {"auth": ... }' 8
sshKey: |
  ssh-rsa AAAA... 9
```

1. 必須。インストールプログラムにより、クラスター名の入力を求められます。
2. インストールするクラスターネットワークプラグイン。サポートされている値は **OVNKubernetes** と **OpenShiftSDN** です。デフォルトの値は **OVNkubernetes** です。
3. オプション。独自のプラットフォーム設定を定義しないマシンプールのパラメーターを指定します。
4. 必須。インストールプログラムにより、クラスターをデプロイするリージョンの入力を求められます。
5. オプション。クラスターをインストールする必要がある既存のリソースグループを指定します。
8. 必須。インストールプログラムは、プルシークレットの入力を求めます。

9 オプション。インストールプログラムは、クラスター内のマシンへのアクセスに使用する SSH キー値の入力を求めます。

6 7 オプション。これらは vswitchID 値の例です。

4.4.6. インストール時のクラスター全体のプロキシーの設定

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

前提条件

- 既存の **install-config.yaml** ファイルがある。
- クラスターがアクセスする必要のあるサイトを確認済みで、それらのいずれかがプロキシーをバイパスする必要があるかどうかを判別している。デフォルトで、すべてのクラスター Egress トラフィック (クラスターをホストするクラウドに関するクラウドプロバイダー API に対する呼び出しを含む) はプロキシーされます。プロキシーを必要に応じてバイパスするために、サイトを **Proxy** オブジェクトの **spec.noProxy** フィールドに追加している。

注記

Proxy オブジェクトの **status.noProxy** フィールドには、インストール設定の **networking.machineNetwork[].cidr**、**networking.clusterNetwork[].cidr**、および **networking.serviceNetwork[]** フィールドの値が設定されます。

Amazon Web Services (AWS)、Google Cloud、Microsoft Azure、および Red Hat OpenStack Platform (RHOSP)へのインストールの場合、**Proxy** オブジェクトの **status.noProxy** フィールドには、インスタンスメタデータのエンドポイント(**169.254.169.254**)も設定されます。

手順

1. **install-config.yaml** ファイルを編集し、プロキシー設定を追加します。以下に例を示します。

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

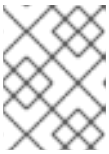
1 クラスター外の HTTP 接続を作成するために使用するプロキシー URL。URL スキームは **http** である必要があります。

2 クラスター外で HTTPS 接続を作成するために使用するプロキシー URL。

3 プロキシーから除外するための宛先ドメイン名、IP アドレス、または他のネットワーク CIDR のコンマ区切りのリスト。サブドメインのみと一致するように、ドメインの前に **.**

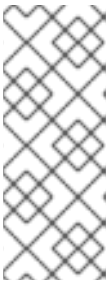
4 指定されている場合、インストールプログラムは HTTPS 接続のプロキシーに必要な 1 つ以上の追加の CA 証明書が含まれる **user-ca-bundle** という名前の設定マップを **openshift-config** namespace に生成します。次に Cluster Network Operator は、これらのコンテンツを Red Hat Enterprise Linux CoreOS (RHCOS) 信頼バンドルにマージする **trusted-ca-bundle** 設定マップを作成し、この設定マップは **Proxy** オブジェクトの **trustedCA** フィールドで参照されます。**additionalTrustBundle** フィールドは、プロキシーのアイデンティティー証明書が RHCOS 信頼バンドルからの認証局によって署名されない限り必要になります。

5 オプション: **trustedCA** フィールドの **user-ca-bundle** 設定マップを参照する **Proxy** オブジェクトの設定を決定するポリシー。許可される値は **Proxyonly** および **Always** です。**Proxyonly** を使用して、**http/https** プロキシーが設定されている場合にのみ **user-ca-bundle** 設定マップを参照します。**Always** を使用して、常に **user-ca-bundle** 設定マップを参照します。デフォルト値は **Proxyonly** です。



注記

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



注記

インストーラーがタイムアウトした場合は、インストーラーの **wait-for** コマンドを使用してデプロイメントを再起動してからデプロイメントを完了します。以下に例を示します。

```
$ ./openshift-install wait-for install-complete --log-level debug
```

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

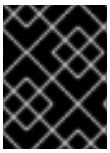
インストールプログラムは、指定の **install-config.yaml** ファイルのプロキシー設定を使用する **cluster** という名前のクラスター全体のプロキシーを作成します。プロキシー設定が指定されていない場合、**cluster Proxy** オブジェクトが依然として作成されますが、これには **spec** がありません。

注記

cluster という名前の **Proxy** オブジェクトのみがサポートされ、追加のプロキシーを作成することはできません。

4.5. 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する。
- ホスト上のクラウドプロバイダーアカウントに、クラスターをデプロイするための適切な権限があることを確認してください。アカウントの権限が正しくないと、インストールプロセスが失敗し、不足している権限を示すエラーメッセージが表示されます。

手順

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

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

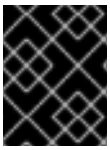
1 **<installation_directory>** に、カスタマイズした **./install-config.yaml** ファイルの場所を指定します。

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。

検証

クラスターのデプロイが正常に完了すると、次のようになります。

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

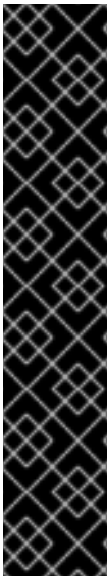


重要

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

出力例

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



重要

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

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

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

重要

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.13 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールします。

Linux への OpenShift CLI のインストール

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

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **Product Variant** ドロップダウンリストからアーキテクチャーを選択します。
3. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
4. **OpenShift v4.13 Linux Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。
5. アーカイブを展開します。

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

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

```
$ echo $PATH
```

検証

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

```
$ oc <command>
```

Windows への OpenShift CLI のインストール

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

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
3. **OpenShift v4.13 Windows Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。
4. ZIP プログラムでアーカイブを解凍します。
5. **oc** バイナリーを、**PATH** にあるディレクトリーに移動します。
   PATH を確認するには、コマンドプロンプトを開いて以下のコマンドを実行します。

```
C:\> path
```

検証

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

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

macOS への OpenShift CLI のインストール

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

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
3. **OpenShift v4.13 macOS Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。

注記

macOS arm64 の場合は、**OpenShift v4.13 macOS arm64 Client** エントリーを選択します。

4. アーカイブを展開し、解凍します。
5. **oc** バイナリーをパスにあるディレクトリーに移動します。

PATH を確認するには、ターミナルを開き、以下のコマンドを実行します。

```
$ echo $PATH
```

検証

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

```
$ oc <command>
```

4.7. CLI の使用によるクラスターへのログイン

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

前提条件

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

手順

1. **kubeadmin** 認証情報をエクスポートします。

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

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

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

```
$ oc whoami
```

出力例

```
system:admin
```

4.8. WEB コンソールを使用したクラスターへのログイン

kubeadmin ユーザーは、OpenShift Container Platform のインストール後はデフォルトで存在します。OpenShift Container Platform Web コンソールを使用し、**kubeadmin** ユーザーとしてクラスターにログインできます。

前提条件

- インストールホストにアクセスできる。

- クラスターのインストールを完了しており、すべてのクラスター Operator が利用可能である。

手順

1. インストールホストで **kubeadmin-password** ファイルから **kubeadmin** ユーザーのパスワードを取得します。

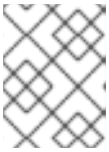
```
$ cat <installation_directory>/auth/kubeadmin-password
```

注記

または、インストールホストで **<installation_directory>/.openshift_install.log** ログファイルから **kubeadmin** パスワードを取得できます。

2. OpenShift Container Platform Web コンソールルートをリスト表示します。

```
$ oc get routes -n openshift-console | grep 'console-openshift'
```



注記

または、インストールホストで **<installation_directory>/.openshift_install.log** ログファイルからで OpenShift Container Platform ルートを取得できます。

出力例

```
console     console-openshift-console.apps.<cluster_name>.<base_domain>           console
https   reencrypt/Redirect   None
```

3. Web ブラウザーで前述のコマンドの出力で詳細に説明されたルートに移動し、**kubeadmin** ユーザーとしてログインします。

4.9. OPENSHIFT CONTAINER PLATFORM の TELEMETRY アクセス

OpenShift Container Platform 4.13 では、クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは OpenShift Cluster Manager Hybrid Cloud Console に登録されます。

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

関連情報

- Telemetry サービスの詳細は、リモートヘルスモニタリング を参照して ください。
- OpenShift Container Platform Web コンソールへのアクセスと理解の詳細については、Web コンソールへのアクセス を参照してください。
- OpenShift Container Platform Web コンソールへ のアクセスと理解の詳細については、Web コンソールへのアクセス を参照してください。

4.10. 次のステップ

- インストールの検証
- クラスターをカスタマイズします。
- リモートヘルスレポート

第5章 ネットワークをカスタマイズして ALIBABA CLOUD にクラスターをインストールする

OpenShift Container Platform 4.13 では、カスタマイズされたネットワーク設定オプションを使用して、Alibaba Cloud にクラスターをインストールできます。ネットワーク設定をカスタマイズすることにより、クラスターは環境内の既存の IP アドレスの割り当てと共存でき、既存の MTU および VXLAN 設定と統合できます。

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

重要

OpenShift Container Platform 上の Alibaba Cloud は、テクノロジープレビュー機能としてのみ利用できます。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行い、フィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

5.1. 前提条件

- OpenShift Container Platform のインストールおよび更新 プロセスの詳細を確認している。
- クラスターインストール方法の選択およびそのユーザー向けの準備 を確認している。
- ドメインを登録 している。
- ファイアウォールを使用する場合は、クラスターがアクセスを必要とするサイトを許可するようにファイアウォールを設定する必要がある。
- ご使用の環境でクラウド Resource Access Management (RAM) API にアクセスできない場合、または管理者レベルのクレデンシャルシークレットを **kube-system** namespace に保存したくない場合は、Resource Access Management (RAM) 認証情報を手動で作成および維持 することができます。

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

OpenShift Container Platform 4.13 では、クラスターをインストールするためにインターネットアクセスが必要になります。

インターネットへのアクセスは以下を実行するために必要です。

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

重要

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

5.3. クラスターノードの SSH アクセス用のキーペアの生成

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

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

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

重要

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

注記

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

手順

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

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

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

2. 公開 SSH キーを表示します。

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

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

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

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

注記

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

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

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

出力例

```
Agent pid 31874
```

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

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

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

出力例

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

次のステップ

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

5.4. インストールプログラムの取得

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

前提条件

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

手順

1. OpenShift Cluster Manager サイトの インフラストラクチャープロバイダー ページにアクセスします。Red Hat アカウントがある場合は、認証情報を使用してログインします。アカウントがない場合はこれを作成します。

2. インフラストラクチャープロバイダーを選択します。

3. インストールタイプのページに移動し、ホストオペレーティングシステムとアーキテクチャーに対応するインストールプログラムをダウンロードして、インストール設定ファイルを保存するディレクトリーにファイルを配置します。

重要

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

重要

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

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

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

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

5.5. ネットワーク設定フェーズ

OpenShift Container Platform をインストールする前に、ネットワーク設定をカスタマイズできる 2 つのフェーズがあります。

フェーズ 1

マニフェストファイルを作成する前に、**install-config.yaml** ファイルで以下のネットワーク関連のフィールドをカスタマイズできます。

- **networking.networkType**
- **networking.clusterNetwork**
- **networking.serviceNetwork**
- **networking.machineNetwork**
  詳細は、「インストール設定パラメーター」を参照してください。

注記

優先されるサブネットが配置されている Classless Inter-Domain Routing (CIDR) と一致するように **networking.machineNetwork** を設定します。

重要

CIDR 範囲 **172.17.0.0/16** は **libVirt** によって予約されています。クラスター内のネットワークに **172.17.0.0/16** CIDR 範囲と重複する他の CIDR 範囲を使用することはできません。

フェーズ 2

openshift-install create manifests を実行してマニフェストファイルを作成した後に、変更するフィールドのみでカスタマイズされた Cluster Network Operator マニフェストを定義できます。マニフェストを使用して、高度なネットワーク設定を指定できます。

フェーズ 2 では、**install-config.yaml** ファイルのフェーズ 1 で指定した値をオーバーライドすることはできません。ただし、フェーズ 2 でネットワークプラグインをカスタマイズできます。

5.5.1. インストール設定ファイルの作成

インストールする OpenShift Container Platform クラスターをカスタマイズできます。

前提条件

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

手順

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

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

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

ディレクトリーを指定する場合:

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

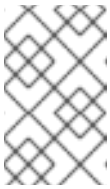
注記

古い設定の再利用を回避するために、~/**.powervs** ディレクトリーは必ず削除してください。以下のコマンドを実行します。

```
$ rm -rf ~/.powervs
```

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

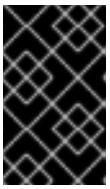
インストールのデバッグまたは障害復旧を実行する必要のある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。

ii. クラスターの記述名を入力します。

iii. Red Hat OpenShift Cluster Manager からプルシークレット 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は、「インストール設定パラメーター」のセクションを参照してください。

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



重要

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

5.5.2. 必要なインストールマニフェストの生成

クラスターがマシンを設定するために必要な 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。

手順

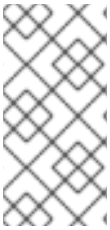
1. インストールプログラムが含まれているディレクトリーから次のコマンドを実行して、マニフェストを生成します。

```
$ openshift-install create manifests --dir <installation_directory>
```

ここで、

<installation_directory>

インストールプログラムがファイルを作成するディレクトリーを指定します。



注記

デフォルトで、**ccoctl** はコマンドが実行されるディレクトリーにオブジェクトを作成します。オブジェクトを別のディレクトリーに作成するには、**--output-dir** フラグを使用します。この手順では、**<path_to_ccoctl_output_dir>** を使用してこの場所を参照します。

前提条件

以下が必要になります。

- **ccoctl** バイナリーを抽出して準備している。

手順

1. 以下のコマンドを実行して、**$RELEASE_IMAGE** 変数を設定します。

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 以下のコマンドを実行して、OpenShift Container Platform リリースイメージから **CredentialsRequest** オブジェクトのリストを抽出します。

```
<1> `credrequests` is the directory where the list of `CredentialsRequest` objects is stored. This command creates the directory if it does not exist.
```

注記

このコマンドの実行には少し時間がかかる場合があります。

5.5.3. 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。

重要

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

5.5.3.1. 必須設定パラメーター

必須のインストール設定パラメーターは、以下の表で説明されています。

表5.1 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
apiVersion	**install-config.yaml** コンテンツの API バージョン。現在のバージョンは **v1** です。インストールプログラムは、古い API バージョンもサポートしている場合があります。	文字列
baseDomain	クラウドプロバイダーのベースドメイン。ベースドメインは、OpenShift Container Platform クラスターコンポーネントへのルートを作成するために使用されます。クラスターの完全な DNS 名は、**baseDomain** と **<metadata.name>.<baseDomain>** 形式を使用する **metadata.name** パラメーターの値の組み合わせです。	**example.com** などの完全修飾ドメインまたはサブドメイン名。
metadata	Kubernetes リソース **ObjectMeta**。ここからは **name** パラメーターのみが消費されます。	オブジェクト
metadata.name	クラスターの名前。クラスターの DNS レコードはすべて **{{.metadata.name}}.{{.baseDomain}}** のサブドメインです。	**dev** などの小文字、ハイフン (**-**)、およびピリオド (**.**) が含まれる文字列。
platform	インストールを実行する特定のプラットフォームの設定: **alibabacloud**、**aws**、**baremetal**、**azure**、**gcp**、**ibmcloud**、**Nutanix**、**openstack**、**ovirt**、**powervs**、**vsphere**、または **{}**。**platform.<platform>** パラメーターに関する追加情報は、以下の表で特定のプラットフォームを参照してください。	オブジェクト

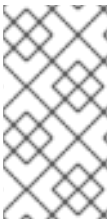
パラメーター	説明	値

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

5.5.3.2. ネットワーク設定パラメーター

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

IPv4 アドレスのみがサポートされます。



注記

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

表5.2 ネットワークパラメーター

パラメーター	説明	値
networking	クラスターのネットワークの設定。	オブジェクト  **注記** インストール後に **networking** オブジェクトで指定したパラメーターを変更することはできません。

パラメーター	説明	値
networking.networkType	インストールする Red Hat OpenShift Networking ネットワークプラグイン。	**OpenShiftSDN** または **OVNKubernetes** のいずれか。**OpenShiftSDN** は、すべての Linux ネットワークの Container Network Interface (CNI)プラグインです。**OVNKubernetes** は、Linux ネットワークと、Linux サーバーと Windows サーバーの両方を含む Linux ネットワークおよびハイブリッドネットワーク用の CNI プラグインです。デフォルトの値は **OVNkubernetes** です。
networking.clusterNetwork	Pod の IP アドレスブロック。 デフォルト値は **10.128.0.0/14** で、ホストの接頭辞は **/23** です。 複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。	オブジェクトの配列。以下に例を示します。 networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23
networking.clusterNetwork.cidr	**networking.clusterNetwork** を使用する場合に必須です。IP アドレスブロック。 IPv4 ネットワーク	CIDR (Classless Inter-Domain Routing) 表記の IP アドレスブロック。IPv4 ブロックの接頭辞長は **0** から **32** の間になります。
networking.clusterNetwork.hostPrefix	それぞれの個別ノードに割り当てるサブネット接頭辞長。たとえば、**hostPrefix** が **23** に設定される場合、各ノードに指定の **cidr** から **/23** サブネットが割り当てられます。**hostPrefix** 値の **23** は、510 (2^(32 - 23) - 2) Pod IP アドレスを提供します。	サブネット接頭辞。 デフォルト値は **23** です。
networking.serviceNetwork	サービスの IP アドレスブロック。デフォルト値は **172.30.0.0/16** です。 OpenShift SDN および OVN-Kubernetes ネットワークプラグインは、サービスネットワークの単一 IP アドレスブロックのみをサポートします。	CIDR 形式の IP アドレスブロックを持つ配列。以下に例を示します。 networking: serviceNetwork: - 172.30.0.0/16

パラメーター	説明	値
networking.machineNetwork	マシンの IP アドレスブロック。 複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。	オブジェクトの配列。以下に例を示します。 networking: machineNetwork: - cidr: 10.0.0.0/16
networking.machineNetwork.cidr	**networking.machineNetwork** を使用する場合に必須です。IP アドレスブロック。libvirt と IBM Power Virtual Server を除くすべてのプラットフォームのデフォルト値は **10.0.0.0/16** です。libvirt の場合、デフォルト値は **192.168.126.0/24** です。IBM Power Virtual Server の場合、デフォルト値は **192.168.0.0/24** です。	CIDR 表記の IP ネットワークブロック。 例: **10.0.0.0/16** **注記** 優先される NIC が置かれている CIDR に一致する **networking.machineNetwork** を設定します。

5.5.3.3. オプションの設定パラメーター

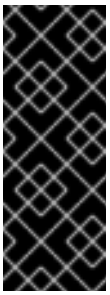
オプションのインストール設定パラメーターは、以下の表で説明されています。

表5.3 オプションのパラメーター

パラメーター	説明	値
additionalTrustBundle	ノードの信頼済み証明書ストアに追加される PEM でエンコードされた X.509 証明書バンドル。この信頼バンドルは、プロキシーが設定されている場合にも使用することができます。	文字列
capabilities	オプションのコアクラスターコンポーネントのインストールを制御します。オプションのコンポーネントを無効にすることで、OpenShift Container Platform クラスターのフットプリントを削減できます。詳細は、**インストール** の「クラスター機能ページ」を参照してください。	文字列配列
capabilities.baselineCapabilitySet	有効にするオプション機能の初期セットを選択します。有効な値は **None**、**v4.11**、**v4.12**、**vCurrent** です。デフォルト値は **vCurrent** です。	文字列

パラメーター	説明	値
capabilities.additionalEnabledCapabilities	オプションの機能のセットを、**baselineCapabilitySet** で指定したものを超えて拡張します。このパラメーターで複数の機能を指定できます。	文字列配列
cpuPartitioningMode	ワークロードパーティション設定を使用して、OpenShift Container Platform サービス、クラスター管理ワークロード、およびインフラストラクチャー Pod を分離し、予約された CPU セットで実行できます。ワークロードパーティショニングは、インストール中にのみ有効にできます。インストール後に無効にすることはできません。このフィールドはワークロードのパーティショニングを有効にしますが、特定の CPU を使用するようにワークロードを設定するわけではありません。詳細は、**スケーラビリティとパフォーマンス** セクションの **ワークロードパーティショニング** ページを参照してください。	**None** または **AllNodes**。デフォルト値は **None** です。
compute	コンピュートノードを形成するマシンの設定。	**MachinePool** オブジェクトの配列。
compute.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現在、さまざまなアーキテクチャーのクラスターはサポートされていません。すべてのプールは同じアーキテクチャーを指定する必要があります。有効な値はデフォルト **amd64** です。	String

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.name	**compute** を使用する場合に必須です。マシンプールの名前。	**worker**
compute.platform	**compute** を使用する場合に必須です。このパラメーターを使用して、ワーカーマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は **controlPlane.platform** パラメーターの値に一致する必要があります。	**alibabacloud**、**aws**、**azure**、**gcp**、**ibmcloud**、**nutanix**、**openstack**、**ovirt**、**powervs**、**vsphere**、または **{}**
compute.replicas	プロビジョニングするコンピュートマシン (ワーカーマシンとしても知られる) の数。	**2** 以上の正の整数。デフォルト値は **3** です。
featureSet	機能セットのクラスターを有効にします。機能セットは、デフォルトで有効にされない OpenShift Container Platform 機能のコレクションです。インストール中に機能セットを有効にする方法の詳細は、「機能ゲートの使用による各種機能の有効化」を参照してください。	文字列。**TechPreviewNoUpgrade** など、有効にする機能セットの名前。
controlPlane	コントロールプレーンを形成するマシンの設定。	**MachinePool** オブジェクトの配列。

パラメーター	説明	値
controlPlane.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現在、さまざまなアーキテクチャーのクラスターはサポートされていません。すべてのプールは同じアーキテクチャーを指定する必要があります。有効な値はデフォルト **amd64** です。	String
controlPlane: hyperthreading:	コントロールプレーンマシンで同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。  **重要** 同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。	**Enabled** または **Disabled**
controlPlane.name	**controlPlane** を使用する場合に必須です。マシンプールの名前。	**master**
controlPlane.platform	**controlPlane** を使用する場合に必須です。このパラメーターを使用して、コントロールプレーンマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は **compute.platform** パラメーターの値に一致する必要があります。	**alibabacloud**、**aws**、**azure**、**gcp**、**ibmcloud**、**nutanix**、**openstack**、**ovirt**、**powervs**、**vsphere**、または **{}**
controlPlane.replicas	プロビジョニングするコントロールプレーンマシンの数。	サポートされる値は **3** のみです (これはデフォルト値です)。

パラメーター	説明	値
credentialsMode	Cloud Credential Operator (CCO) モード。CCO は、モードが指定されていない場合に指定される認証情報の機能を動的に判別しようとします。この場合、複数のモードがサポートされるプラットフォームで mint モードが優先されます。  **注記** すべてのクラウドプロバイダーですべての CCO モードがサポートされているわけではありません。CCO モードの詳細は、**Cluster Operators リファレンス** の **Cloud Credential Operator** を参照してください。  **注記** AWS アカウントでサービスコントロールポリシー (SCP) が有効になっている場合は、**credentialsMode** パラメーターを **Mint**、**Passthrough** または **Manual** に設定する必要があります。	**Mint**、**Passthrough**、**Manual**、または空の文字列 (**""**)。
imageContentSources	release-image コンテンツのソースおよびリポジトリー。	オブジェクトの配列。この表の以下の行で説明されているように、**source** およびオプションで **mirrors** が含まれます。
imageContentSources.source	**imageContentSources** を使用する場合に必須です。ユーザーが参照するリポジトリーを指定します (例: イメージプル仕様)。	文字列
imageContentSources.mirrors	同じイメージが含まれる可能性のあるリポジトリーを 1 つ以上指定します。	文字列の配列。

パラメーター	説明	値
publish	Kubernetes API、OpenShift ルートなどのクラスターのユーザーに表示されるエンドポイントをパブリッシュまたは公開する方法。	**Internal** または **External**。デフォルト値は **External** です。 このパラメーターを **Internal** に設定することは、クラウド以外のプラットフォームではサポートされません。  **重要** フィールドの値が **Internal** に設定されている場合、クラスターは機能しなくなります。詳細は、BZ#1953035 を参照してください。
sshKey	クラスターマシンへのアクセスを認証するための SSH キー。  **注記** インストールのデバッグまたは障害復旧を実行する必要のある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。	たとえば、**sshKey: ssh-ed25519 AAAA..** です。

1. すべてのクラウドプロバイダーですべての CCO モードがサポートされているわけではありません。CCO モードの詳細は、**認証と認可** コンテンツの「クラウドプロバイダーの認証情報の管理」を参照してください。

5.5.4. Alibaba Cloud 用にカスタマイズされた install-config.yaml ファイルのサンプル

インストール設定ファイル (**install-config.yaml**) をカスタマイズして、クラスターのプラットフォームに関する詳細を指定したり、必要なパラメーターの値を変更したりできます。

```
apiVersion: v1
baseDomain: alicloud-dev.devcluster.openshift.com
credentialsMode: Manual
compute:
- architecture: amd64
  hyperthreading: Enabled
  name: worker
  platform: {}
  replicas: 3
controlPlane:
```

```
  architecture: amd64
  hyperthreading: Enabled
  name: master
  platform: {}
  replicas: 3
metadata:
  creationTimestamp: null
  name: test-cluster 1
 networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  networkType: OVNKubernetes 2
  serviceNetwork:
  - 172.30.0.0/16
platform:
  alibabacloud:
    defaultMachinePlatform: 3
      instanceType: ecs.g6.xlarge
      systemDiskCategory: cloud_efficiency
      systemDiskSize: 200
    region: ap-southeast-1 4
    resourceGroupID: rg-acfnw6j3hyai 5
    vpcID: vpc-0xifdjerdibmaqvtjob2b 6
    vswitchIDs: 7
    - vsw-0xi8ycgwc8wv5rhviwdq5
    - vsw-0xiy6v3z2tedv009b4pz2
publish: External
pullSecret: '{"auths": {"cloud.openshift.com": {"auth": ... }' 8
sshKey: |
  ssh-rsa AAAA... 9
```

1 必須。インストールプログラムにより、クラスター名の入力を求められます。

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

3 オプション。独自のプラットフォーム設定を定義しないマシンプールのパラメーターを指定します。

4 必須。インストールプログラムにより、クラスターをデプロイするリージョンの入力を求められます。

5 オプション。クラスターをインストールする必要がある既存のリソースグループを指定します。

8 必須。インストールプログラムは、プルシークレットの入力を求めます。

9 オプション。インストールプログラムは、クラスター内のマシンへのアクセスに使用する SSH キー値の入力を求めます。

6 7 オプション。これらは vswitchID 値の例です。

5.5.5. インストール時のクラスター全体のプロキシーの設定

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

前提条件

- 既存の **install-config.yaml** ファイルがある。
- クラスターがアクセスする必要のあるサイトを確認済みで、それらのいずれかがプロキシーをバイパスする必要があるかどうかを判別している。デフォルトで、すべてのクラスター Egress トラフィック (クラスターをホストするクラウドに関するクラウドプロバイダー API に対する呼び出しを含む) はプロキシーされます。プロキシーを必要に応じてバイパスするために、サイトを **Proxy** オブジェクトの **spec.noProxy** フィールドに追加している。

注記

Proxy オブジェクトの **status.noProxy** フィールドには、インストール設定の **networking.machineNetwork[].cidr**、**networking.clusterNetwork[].cidr**、および **networking.serviceNetwork[]** フィールドの値が設定されます。

Amazon Web Services (AWS)、Google Cloud、Microsoft Azure、および Red Hat OpenStack Platform (RHOSP)へのインストールの場合、**Proxy** オブジェクトの **status.noProxy** フィールドには、インスタンスメタデータのエンドポイント(**169.254.169.254**)も設定されます。

手順

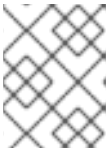
1. **install-config.yaml** ファイルを編集し、プロキシー設定を追加します。以下に例を示します。

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

1. クラスター外の HTTP 接続を作成するために使用するプロキシー URL。URL スキームは **http** である必要があります。
2. クラスター外で HTTPS 接続を作成するために使用するプロキシー URL。
3. プロキシーから除外するための宛先ドメイン名、IP アドレス、または他のネットワーク CIDR のコンマ区切りのリスト。サブドメインのみと一致するように、ドメインの前に **.** を付けます。たとえば、**.y.com** は **x.y.com** に一致しますが、**y.com** には一致しません。***** を使用し、すべての宛先のプロキシーをバイパスします。
4. 指定されている場合、インストールプログラムは HTTPS 接続のプロキシーに必要な 1 つ

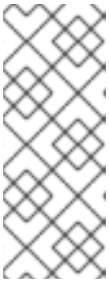
以上の追加の CA 証明書が含まれる **user-ca-bundle** という名前の設定マップを **openshift-config** namespace に生成します。次に Cluster Network Operator は、これらのコンテンツを Red Hat Enterprise Linux CoreOS (RHCOS) 信頼バンドルにマージする **trusted-ca-bundle** 設定マップを作成し、この設定マップは **Proxy** オブジェクトの **trustedCA** フィールドで参照されます。**additionalTrustBundle** フィールドは、プロキシーのアイデンティティー証明書が RHCOS 信頼バンドルからの認証局によって署名されない限り必要になります。

5 オプション： **trustedCA** フィールドの **user-ca-bundle** 設定マップを参照する **Proxy** オブジェクトの設定を決定するポリシー。許可される値は **Proxyonly** および **Always** です。**Proxyonly** を使用して、**http/https** プロキシーが設定されている場合にのみ **user-ca-bundle** 設定マップを参照します。**Always** を使用して、常に **user-ca-bundle** 設定マップを参照します。デフォルト値は **Proxyonly** です。



注記

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



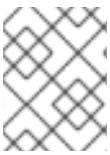
注記

インストーラーがタイムアウトした場合は、インストーラーの **wait-for** コマンドを使用してデプロイメントを再起動してからデプロイメントを完了します。以下に例を示します。

```
$ ./openshift-install wait-for install-complete --log-level debug
```

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

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



注記

cluster という名前の **Proxy** オブジェクトのみがサポートされ、追加のプロキシーを作成することはできません。

5.6. CLUSTER NETWORK OPERATOR (CNO) の設定

クラスターネットワークの設定は、Cluster Network Operator (CNO) 設定の一部として指定され、**cluster** という名前のカスタムリソース (CR) オブジェクトに保存されます。CR は **operator.openshift.io** API グループの **Network** API のフィールドを指定します。

CNO 設定は、**Network.config.openshift.io** API グループの **Network** API からクラスターのインストール時に以下のフィールドを継承し、これらのフィールドは変更できません。

clusterNetwork

Pod IP アドレスの割り当てに使用する IP アドレスプール。

serviceNetwork

サービスの IP アドレスプール。

defaultNetwork.type

OpenShift SDN や OVN-Kubernetes などのクラスターネットワークプラグイン。

defaultNetwork オブジェクトのフィールドを **cluster** という名前の CNO オブジェクトに設定することにより、クラスターのクラスターネットワークプラグイン設定を指定できます。

5.6.1. Cluster Network Operator 設定オブジェクト

Cluster Network Operator (CNO) のフィールドは以下の表で説明されています。

表5.4 Cluster Network Operator 設定オブジェクト

フィールド	型	説明
metadata.name	**string**	CNO オブジェクトの名前。この名前は常に **cluster** です。

spec.clusterNetwork	**array**	Pod ID アドレスの割り当て、サブネット接頭辞の長さのクラスター内の個別ノードへの割り当てに使用される IP アドレスのブロックを指定するリストです。以下に例を示します。
		
		`spec:`
		`  clusterNetwork:`
		`  - cidr: 10.128.0.0/19`
		`  hostPrefix: 23`
		`  - cidr: 10.128.32.0/19`
		`  hostPrefix: 23`
		
		マニフェストを作成する前に、このフィールドを **install-config.yaml** ファイルでのみカスタマイズすることができます。この値は、マニフェストファイルでは読み取り専用です。

spec.serviceNetwork	**array**	サービスの IP アドレスのブロック。OpenShift SDN および OVN-Kubernetes ネットワークプラグインは、サービスネットワークの単一 IP アドレスブロックのみをサポートします。以下に例を示します。
		
		`spec:`
		`  serviceNetwork:`
		`  - 172.30.0.0/14`
		
		マニフェストを作成する前に、このフィールドを **install-config.yaml** ファイルでのみカスタマイズすることができます。この値は、マニフェストファイルでは読み取り専用です。

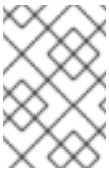
spec.defaultNetwork	**object**	クラスターネットワークのネットワークプラグインを設定します。
spec.kubeProxyConfig	**object**	このオブジェクトのフィールドは、kube-proxy 設定を指定します。OVN-Kubernetes クラスターネットワークプラグインを使用している場合、kube-proxy 設定は機能しません。

重要

複数のネットワークにオブジェクトをデプロイする必要があるクラスターの場合は、**install-config.yaml** ファイルで定義されている各ネットワークタイプの **clusterNetwork.hostPrefix** パラメーターに、必ず同じ値を指定してください。**clusterNetwork.hostPrefix** パラメーターにそれぞれ異なる値を設定すると、OVN-Kubernetes ネットワークプラグインに影響が及び、異なるノード間のオブジェクトトラフィックをプラグインが効果的にルーティングできなくなる可能性があります。

defaultNetwork オブジェクト設定
defaultNetwork オブジェクトの値は、以下の表で定義されます。

表5.5 defaultNetwork オブジェクト

フィールド	型	説明
type	**string**	**OpenShiftSDN** または **OVNKubernetes** のいずれか。Red Hat OpenShift Networking ネットワークプラグインは、インストール中に選択されます。この値は、クラスターのインストール後は変更できません。  **注記** OpenShift Container Platform は、デフォルトで OVN-Kubernetes ネットワークプラグインを使用します。
openshiftSDNConfig	**object**	このオブジェクトは、OpenShift SDN ネットワークプラグインに対してのみ有効です。
ovnKubernetesConfig	**object**	このオブジェクトは、OVN-Kubernetes ネットワークプラグインに対してのみ有効です。

OpenShift SDN ネットワークプラグインの設定
以下の表では、OpenShift SDN ネットワークプラグインの設定フィールドを説明します。

表5.6 openshiftSDNConfig オブジェクト

フィールド	型	説明
mode	**string**	OpenShift SDN のネットワーク分離モードを設定します。デフォルト値は **NetworkPolicy** です。 **Multitenant** および **Subnet** の値は、OpenShift Container Platform 3.x との後方互換性を維持するために利用できますが、その使用は推奨されていません。この値は、クラスターのインストール後は変更できません。

フィールド	型	説明
mtu	**integer**	VXLAN オーバーレイネットワークの最大転送単位 (MTU)。これは、プライマリーネットワークインターフェイスの MTU に基づいて自動的に検出されます。通常、検出された MTU を上書きする必要はありません。 自動検出した値が予想される値ではない場合は、ノード上のプライマリーネットワークインターフェイスの MTU が正しいことを確認します。このオプションを使用して、ノード上のプライマリーネットワークインターフェイスの MTU 値を変更することはできません。 クラスターで異なるノードに異なる MTU 値が必要な場合、この値をクラスター内の最小の MTU 値よりも **50** 小さく設定する必要があります。たとえば、クラスター内の一部のノードでは MTU が **9001** であり、MTU が **1500** のクラスターもある場合には、この値を **1450** に設定する必要があります。 クラスターインストール時またはインストール後のタスクとして値を設定できます。詳細は、OpenShift Container Platform Networking ドキュメントの "Changing the MTU for the cluster network" を参照してください。
vxlanPort	**integer**	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。この値は、クラスターのインストール後は変更できません。 別の VXLAN ネットワークの一部である既存ノードと共に仮想化環境で実行している場合は、これを変更する必要がある可能性があります。たとえば、OpenShift SDN オーバーレイを VMware NSX-T 上で実行する場合は、両方の SDN が同じデフォルトの VXLAN ポート番号を使用するため、VXLAN の別のポートを選択する必要があります。 Amazon Web Services (AWS) では、VXLAN にポート **9000** とポート **9999** 間の代替ポートを選択できます。

OVN-Kubernetes ネットワークプラグインの設定
次の表では、OVN-Kubernetes ネットワークプラグインの設定フィールドを説明します。

表5.7 ovnKubernetesConfig オブジェクト

フィールド	型	説明

フィールド	型	説明
mtu	**integer**	Geneve (Generic Network Virtualization Encapsulation) オーバーレイネットワークの MTU (maximum transmission unit)。これは、プライマリーネットワークインターフェイスの MTU に基づいて自動的に検出されます。通常、検出された MTU を上書きする必要はありません。 自動検出した値が予想される値ではない場合は、ノード上のプライマリーネットワークインターフェイスの MTU が正しいことを確認します。このオプションを使用して、ノード上のプライマリーネットワークインターフェイスの MTU 値を変更することはできません。 クラスターで異なるノードに異なる MTU 値が必要な場合、この値をクラスター内の最小の MTU 値よりも **100** 小さく設定する必要があります。たとえば、クラスター内の一部のノードでは MTU が **9001** であり、MTU が **1500** のクラスターもある場合には、この値を **1400** に設定する必要があります。
genevePort	**integer**	すべての Geneve パケットに使用するポート。デフォルト値は **6081** です。この値は、クラスターのインストール後は変更できません。
ipsecConfig	**object**	IPsec 暗号化を有効にするために空のオブジェクトを指定します。
policyAuditConfig	**object**	ネットワークポリシー監査ロギングをカスタマイズする設定オブジェクトを指定します。指定されていない場合は、デフォルトの監査ログ設定が使用されます。
gatewayConfig	**object**	オプション: Egress トラフィックのノードゲートウェイへの送信方法をカスタマイズするための設定オブジェクトを指定します。  **注記** Egress トラフィックの移行中は、Cluster Network Operator (CNO) が変更を正常にロールアウトするまで、ワークロードとサービストラフィックに多少の中断が発生することが予想されます。

フィールド	型	説明
v4InternalSubnet	既存のネットワーククインフラストラクチャーが **100.64.0.0/16** IPv4 サブネットと重複している場合は、OVN-Kubernetes による内部使用のために別の IP アドレス範囲を指定できます。IP アドレス範囲が、OpenShift Container Platform インストールで使用される他のサブネットと重複しないようにする必要があります。IP アドレス範囲は、クラスターに追加できるノードの最大数より大きくする必要があります。たとえば、**clusterNetwork.cidr** 値が **10.128.0.0/14** で、**clusterNetwork.hostPrefix** 値が **/23** の場合、ノードの最大数は **2^(23-14)=512** です。 このフィールドは、インストール後に変更できません。	デフォルト値は **100.64.0.0/16** です。

フィールド	型	説明
v6InternalSubnet	既存のネットワークインフラストラクチャーが **fd98::/48** IPv6 サブネットと重複する場合は、OVN-Kubernetes による内部使用のために別の IP アドレス範囲を指定できます。IP アドレス範囲が、OpenShift Container Platform インストールで使用される他のサブネットと重複しないようにする必要があります。IP アドレス範囲は、クラスターに追加できるノードの最大数より大きくする必要があります。 このフィールドは、インストール後に変更できません。	デフォルト値は **fd98::/48** です。

表5.8 policyAuditConfig オブジェクト

フィールド	型	説明
rateLimit	integer	ノードごとに毎秒生成されるメッセージの最大数。デフォルト値は、1秒あたり **20** メッセージです。
maxFileSize	integer	監査ログの最大サイズ (バイト単位)。デフォルト値は **50000000** (50MB) です。
maxLogFiles	integer	保持されるログファイルの最大数。

フィールド	型	説明
比較先	string	以下の追加の監査ログターゲットのいずれかになります。 **libc** ホスト上の journald プロセスの libc **syslog()** 関数。 **udp:<host>:<port>** syslog サーバー。**<host>:<port>** を syslog サーバーのホストおよびポートに置き換えます。 **unix:<file>** **<file>** で指定された Unix ドメインソケットファイル。 **null** 監査ログを追加のターゲットに送信しないでください。
syslogFacility	string	RFC5424 で定義される **kern** などの syslog ファシリティー。デフォルト値は **local0** です。

表5.9 gatewayConfig オブジェクト

フィールド	型	説明
routingViaHost	**boolean**	Pod からホストネットワークスタックへの Egress トラフィックを送信するには、このフィールドを **true** に設定します。インストールおよびアプリケーションがカーネルルーティングテーブルに手動設定されたルートに依存するなど非常に特化されている場合には、Egress トラフィックをホストネットワークスタックにルーティングすることを推奨します。デフォルトでは、Egress トラフィックは OVN で処理され、クラスターを終了するために処理され、トラフィックはカーネルルーティングテーブルの特殊なルートによる影響を受けません。デフォルト値は **false** です。 このフィールドで、Open vSwitch ハードウェアオフロード機能との対話が可能になりました。このフィールドを **true** に設定すると、egress トラフィックがホストネットワークスタックで処理されるため、パフォーマンス的に、オフロードによる利点は得られません。

IPsec が有効な OVN-Kubernetes 設定の例

```
defaultNetwork:
  type: OVNKubernetes
  ovnKubernetesConfig:
    mtu: 1400
    genevePort: 6081
    ipsecConfig: {}
```

kubeProxyConfig オブジェクト設定
kubeProxyConfig オブジェクトの値は以下の表で定義されます。

表5.10 kubeProxyConfig オブジェクト

フィールド	型	説明
iptablesSyncPeriod	**string**	**iptables** ルールの更新期間。デフォルト値は **30s** です。有効な接尾辞には、**s**、**m**、および **h** などが含まれ、これらについては、Go **time** パッケージ ドキュメントで説明されています。 **注記** OpenShift Container Platform 4.3 以降で強化されたパフォーマンスの向上により、**iptablesSyncPeriod** パラメーターを調整する必要はなくなりました。

proxyArguments.iptables-min-sync-period	**array**	**iptables** ルールを更新する前の最小期間。このフィールドにより、更新の頻度が高くなり過ぎないようにできます。有効な接尾辞には、**s**、**m**、および **h** などが含まれ、これらについては、Go **time** パッケージ で説明されています。デフォルト値:
		
		`kubeProxyConfig:`
		`  proxyArguments:`
		`  iptables-min-sync-period:`
		`  - 0s`

5.7. 高度なネットワーク設定の指定

ネットワークプラグインに高度なネットワーク設定を使用し、クラスターを既存のネットワーク環境に統合することができます。高度なネットワーク設定は、クラスターのインストール前にのみ指定することができます。

重要

インストールプロブラムで作成される OpenShift Container Platform マニフェストファイルを変更してネットワーク設定をカスタマイズすることは、サポートされていません。以下の手順のように、作成するマニフェストファイルを適用することがサポートされています。

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ている。

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、マニフェストを作成します。

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

1 **<installation_directory>** 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. **cluster-network-03-config.yml** という名前の、高度なネットワーク設定用のスタブマニフェストファイルを **<installation_directory>/manifests/** ディレクトリーに作成します。

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
```

3. 以下の例のように、**cluster-network-03-config.yml** ファイルで、クラスターの高度なネットワーク設定を指定します。

OpenShift SDN ネットワークプロバイダーに異なる VXLAN ポートを指定します。

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  defaultNetwork:
    openshiftSDNConfig:
      vxlanPort: 4800
```

OVN-Kubernetes ネットワークプロバイダーの IPsec を有効にします。

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  defaultNetwork:
    ovnKubernetesConfig:
      ipsecConfig: {}
```

4. 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、Ignition 設定ファイルの作成時に **manifests/** ディレクトリーを使用します。

5.8. OVN-KUBERNETES を使用したハイブリッドネットワークの設定

OVN-Kubernetes ネットワークプラグインを使用してハイブリッドネットワークを使用するようにクラスターを設定できます。これにより、異なるノードのネットワーク設定をサポートするハイブリッドクラスターが可能になります。

注記

この設定は、同じクラスター内で Linux ノードと Windows ノードの両方を実行するために必要です。

前提条件

- **install-config.yaml** ファイルで **networking.networkType** パラメーターの **OVNKubernetes** を定義していること。詳細は、選択したクラウドプロバイダーでの OpenShift Container Platform ネットワークのカスタマイズの設定に関するインストールドキュメントを参照してください。

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、マニフェストを作成します。

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

ここでは、以下のようになります。

<installation_directory>

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. **cluster-network-03-config.yml** という名前の、高度なネットワーク設定用のスタブマニフェストファイルを **<installation_directory>/manifests/** ディレクトリーに作成します。

```
$ cat <<EOF > <installation_directory>/manifests/cluster-network-03-config.yml
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
EOF
```

ここでは、以下のようになります。

<installation_directory>

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。

3. **cluster-network-03-config.yml** ファイルをエディターで開き、次の例のようにハイブリッドネットワークを使用して OVN-Kubernetes を設定します。

ハイブリッドネットワーク設定の指定

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  defaultNetwork:
    ovnKubernetesConfig:
      hybridOverlayConfig:
        hybridClusterNetwork: 1
        - cidr: 10.132.0.0/14
          hostPrefix: 23
        hybridOverlayVXLANPort: 9898 2
```

1. 追加のオーバーレイネットワーク上のノードに使用される CIDR 設定を指定します。**hybridClusterNetwork** CIDR は **clusterNetwork** CIDR と重複できません。

2. 追加のオーバーレイネットワークのカスタム VXLAN ポートを指定します。これは、vSphere にインストールされたクラスターで Windows ノードを実行するために必要であり、その他のクラウドプロバイダー用に設定することはできません。カスタムポートには、デフォルトの **4789** ポートを除くいずれかのオープンポートを使用できます。この要件の詳細は、Microsoft ドキュメントの Pod-to-pod connectivity between hosts is broken を参照してください。

注記

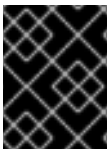
Windows Server Long-Term Servicing Channel (LTSC): Windows Server 2019 は、カスタムの VXLAN ポートの選択をサポートしないため、カスタムの **hybridOverlayVXLANPort** 値を持つクラスターではサポートされません。

4. **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。

5. 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。

5.9. クラスターのデプロイ

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



重要

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

前提条件

- 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する。
- ホスト上のクラウドプロバイダーアカウントに、クラスターをデプロイするための適切な権限があることを確認してください。アカウントの権限が正しくないと、インストールプロセスが失敗し、不足している権限を示すエラーメッセージが表示されます。

手順

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

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

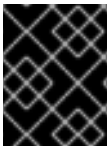
1. **<installation_directory>** に、カスタマイズした **./install-config.yaml** ファイルの場所を指定します。

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。

検証

クラスターのデプロイが正常に完了すると、次のようになります。

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

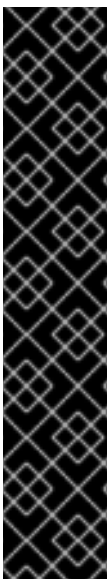


重要

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

出力例

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



重要

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

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

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

重要

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.13 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールします。

Linux への OpenShift CLI のインストール

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

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **Product Variant** ドロップダウンリストからアーキテクチャーを選択します。
3. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
4. **OpenShift v4.13 Linux Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。
5. アーカイブを展開します。

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

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

```
$ echo $PATH
```

検証

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

```
$ oc <command>
```

Windows への OpenShift CLI のインストール

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

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
3. **OpenShift v4.13 Windows Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。
4. ZIP プログラムでアーカイブを解凍します。
5. **oc** バイナリーを、**PATH** にあるディレクトリーに移動します。
PATH を確認するには、コマンドプロンプトを開いて以下のコマンドを実行します。

```
C:\> path
```

検証

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

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

macOS への OpenShift CLI のインストール

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

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
3. **OpenShift v4.13 macOS Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。

注記

macOS arm64 の場合は、**OpenShift v4.13 macOS arm64 Client** エントリーを選択します。

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

```
$ echo $PATH
```

検証

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

```
$ oc <command>
```

5.11. CLI の使用によるクラスターへのログイン

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

前提条件

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

手順

1. **kubeadmin** 認証情報をエクスポートします。

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

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

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

```
$ oc whoami
```

出力例

```
system:admin
```

5.12. WEB コンソールを使用したクラスターへのログイン

kubeadmin ユーザーは、OpenShift Container Platform のインストール後はデフォルトで存在します。OpenShift Container Platform Web コンソールを使用し、**kubeadmin** ユーザーとしてクラスターにログインできます。

前提条件

- インストールホストにアクセスできる。
- クラスターのインストールを完了しており、すべてのクラスター Operator が利用可能である。

手順

1. インストールホストで **kubeadmin-password** ファイルから **kubeadmin** ユーザーのパスワードを取得します。

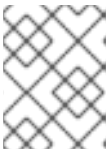
```
$ cat <installation_directory>/auth/kubeadmin-password
```

注記

または、インストールホストで **<installation_directory>/.openshift_install.log** ログファイルから **kubeadmin** パスワードを取得できます。

2. OpenShift Container Platform Web コンソールルートをリスト表示します。

```
$ oc get routes -n openshift-console | grep 'console-openshift'
```



注記

または、インストールホストで **<installation_directory>/.openshift_install.log** ログファイルからで OpenShift Container Platform ルートを取得できます。

出力例

```
console     console-openshift-console.apps.<cluster_name>.<base_domain>          console
https   reencrypt/Redirect   None
```

3. Web ブラウザーで前述のコマンドの出力で詳細に説明されたルートに移動し、**kubeadmin** ユーザーとしてログインします。

5.13. OPENSHIFT CONTAINER PLATFORM の TELEMETRY アクセス

OpenShift Container Platform 4.13 では、クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは OpenShift Cluster Manager Hybrid Cloud Console に登録されます。

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

関連情報

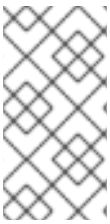
- Telemetry サービスの詳細は、リモートヘルスモニタリング を参照して ください。
- OpenShift Container Platform Web コンソールへのアクセスと理解の詳細については、Web コンソールへのアクセス を参照してください。
- OpenShift Container Platform Web コンソールへ のアクセスと理解の詳細については、Web コンソールへのアクセス を参照してください。

5.14. 次のステップ

- インストールの検証
- クラスターをカスタマイズします。
- リモートヘルスレポート

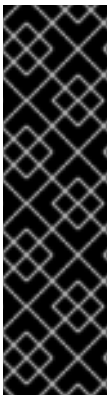
第6章 ALIBABA CLOUD 上のクラスターを既存の VPC にインストールする

OpenShift Container Platform バージョン 4.13 では、Alibaba Cloud Services 上の既存の Alibaba Virtual Private Cloud (VPC) にクラスターをインストールできます。インストールプログラムは必要なインフラストラクチャーをプロビジョニングし、その後カスタマイズできます。VPC 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' ファイルのパラメーターを変更します。



注記

OpenShift Container Platform インストール設定のスコープは意図的に狭められています。単純さを確保し、確実にインストールを実行できるように設計されているためです。インストールが完了した後にさらに多くの OpenShift Container Platform 設定タスクを実行することができます。



重要

OpenShift Container Platform 上の Alibaba Cloud は、テクノロジープレビュー機能としてのみ利用できます。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行い、フィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

6.1. 前提条件

- OpenShift Container Platform のインストールおよび更新 プロセスの詳細を確認している。
- クラスターインストール方法の選択およびそのユーザー向けの準備 を確認している。
- ドメインを登録 している。
- ファイアウォールを使用する場合は、クラスターがアクセスを必要とするサイトを許可するようにファイアウォールを設定する必要がある。
- ご使用の環境でクラウド Resource Access Management (RAM) API にアクセスできない場合、または管理者レベルのクレデンシャルシークレットを **kube-system** namespace に保存したくない場合は、Resource Access Management (RAM) 認証情報を手動で作成および維持 することができます。

6.2. カスタム VPC の使用

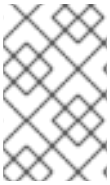
OpenShift Container Platform 4.13 では、Alibaba Cloud Platform の既存の Virtual Private Cloud (VPC) 内の既存のサブネットにクラスターをデプロイできます。OpenShift Container Platform を既存の Alibaba VPC にデプロイすることで、新しいアカウントの制限の制約を回避し、所属する組織の運用上の制約をより簡単に順守することができます。VPC を作成するために必要なインフラストラクチャーの作成パーミッションを取得できない場合は、このインストールオプションを使用します。vSwitch を使用してネットワークを設定する必要があります。

6.2.1. VPC を使用するための要件

VPC CIDR ブロックとマシンネットワーク CIDR の組み合わせは、空であってはなりません。vSwitch はマシンネットワーク内にある必要があります。

インストールプログラムでは、次のコンポーネントは作成されません。

- VPC
- vSwitch
- ルートテーブル
- NAT ゲートウェイ



注記

インストールプログラムでは、クラウド提供の DNS サーバーを使用する必要があります。カスタム DNS サーバーの使用はサポートされていないため、インストールが失敗します。

6.2.2. VPC 検証

指定した vSwitch が適切であることを確認するために、インストールプログラムは次のデータを確認します。

- 指定するすべての vSwitch が存在する必要があります。
- コントロールプレーンマシンとコンピュートマシンに 1 つ以上の vSwitch を提供しました。
- vSwitch の CIDR は、指定したマシン CIDR に属します。

6.2.3. パーミッションの区分

一部の個人は、クラウド内に他とは異なるリソースを作成できます。たとえば、インスタンス、バケット、ロードバランサーなどのアプリケーション固有のアイテムを作成できる場合がありますが、VPC や vSwitch などのネットワーク関連のコンポーネントは作成できません。

6.2.4. クラスター間の分離

OpenShift Container Platform を既存のネットワークにデプロイする場合、クラスターサービスの分離は以下の方法で軽減されます。

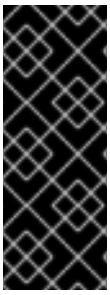
- 複数の OpenShift Container Platform クラスターを同じ VPC にインストールできます。
- ICMP Ingress はネットワーク全体で許可されます。
- TCP 22 Ingress (SSH) はネットワーク全体に対して許可されます。
- コントロールプレーンの TCP 6443 Ingress (Kubernetes API) はネットワーク全体に対して許可されます。
- コントロールプレーンの TCP 22623 Ingress (MCS) はネットワーク全体に対して許可されます。

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

OpenShift Container Platform 4.13 では、クラスターをインストールするためにインターネットアクセスが必要になります。

インターネットへのアクセスは以下を実行するために必要です。

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



重要

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

6.4. クラスターノードの SSH アクセス用のキーペアの生成

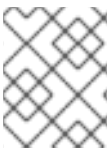
OpenShift Container Platform をインストールする際に、SSH パブリックキーをインストールプログラムに指定できます。キーは、Ignition 設定ファイルを介して Red Hat Enterprise Linux CoreOS (RHCOS) ノードに渡され、ノードへの SSH アクセスを認証するために使用されます。このキーは各ノードの **core** ユーザーの **~/.ssh/authorized_keys** リストに追加され、パスワードなしの認証が可能になります。

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

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

重要

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



注記

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

手順

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

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

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

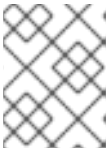
2. 公開 SSH キーを表示します。

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

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

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

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



注記

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

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

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

出力例

```
Agent pid 31874
```

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

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

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

出力例

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

次のステップ

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

6.5. インストールプログラムの取得

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

前提条件

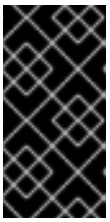
- 500 MB のローカルディスク領域がある Linux または macOS を実行するコンピューターが必要です。

手順

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

重要

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



重要

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

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

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

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

6.5.1. インストール設定ファイルの作成

Alibaba Cloud にインストールする OpenShift Container Platform クラスターをカスタマイズできます。

前提条件

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

手順

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

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

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

   ディレクトリーを指定する場合:

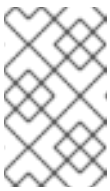
   - ディレクトリーに **execute** 権限があることを確認します。この権限は、インストールディレクトリーで Terraform バイナリーを実行するために必要です。
   - 空のディレクトリーを使用します。ブートストラップ X.509 証明書などの一部のインストールアセットは有効期限が短いため、インストールディレクトリーを再利用しないでください。別のクラスターインストールの個別のファイルを再利用する必要がある場合は、それらをディレクトリーにコピーすることができます。ただし、インストールアセットのファイル名はリリース間で変更される可能性があります。インストールファイルを以前のバージョン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

注記

古い設定の再利用を回避するために、~/**.powervs** ディレクトリーは必ず削除してください。以下のコマンドを実行します。

```
$ rm -rf ~/.powervs
```

   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. ターゲットとするプラットフォームとして **alibabacloud** を選択します。

iii. クラスターをデプロイするリージョンを選択します。

iv. 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。

v. クラスターの記述名を指定します。

vi. Red Hat OpenShift Cluster Manager からプルシークレット を貼り付けます。

2. クラスターを Alibaba Cloud にインストールするには、Cloud Credential Operator (CCO) が手動モードで動作する必要があります。**install-config.yaml** ファイルを変更して、**credentialsMode** パラメーターを **Manual** に設定します。

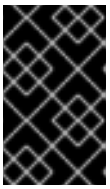
credentialsMode が Manual に設定された install-config.yaml 設定ファイルの例

```
apiVersion: v1
baseDomain: cluster1.example.com
credentialsMode: Manual 1
compute:
- architecture: amd64
  hyperthreading: Enabled
...
```

1 この行を追加して、 **credentialsMode** を **Manual** に設定します。

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

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



重要

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

6.5.2. インストール設定パラメーター

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

重要

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

6.5.2.1. 必須設定パラメーター

必須のインストール設定パラメーターは、以下の表で説明されています。

表6.1 必須パラメーター

パラメーター	説明	値
apiVersion	**install-config.yaml** コンテンツの API バージョン。現在のバージョンは **v1** です。インストールプログラムは、古い API バージョンもサポートしている場合があります。	文字列
baseDomain	クラウドプロバイダーのベースドメイン。ベースドメインは、OpenShift Container Platform クラスターコンポーネントへのルートを作成するために使用されます。クラスターの完全な DNS 名は、**<metadata.name>.<baseDomain>** 形式を使用する **baseDomain** と **metadata.name** パラメーターの値を組み合わせたものです。	**example.com** などの完全修飾ドメインまたはサブドメイン名。
metadata	Kubernetes リソース **ObjectMeta**。ここからは **name** パラメーターのみが消費されます。	オブジェクト
metadata.name	クラスターの名前。クラスターの DNS レコードはすべて **{{.metadata.name}}.{{.baseDomain}}** のサブドメインです。	**dev** などの小文字、ハイフン (**-**)、およびピリオド (**.**) が含まれる文字列。
platform	インストールを実行する特定のプラットフォームの設定: **alibabacloud**、**aws**、**baremetal**、**azure**、**gcp**、**ibmcloud**、**Nutanix**、**openstack**、**ovirt**、**powervs**、**vsphere**、または **{}**。**platform.<platform>** パラメーターに関する追加情報は、以下の表で特定のプラットフォームを参照してください。	オブジェクト

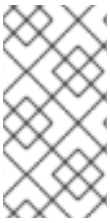
パラメーター	説明	値

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

6.5.2.2. ネットワーク設定パラメーター

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

IPv4 アドレスのみがサポートされます。



注記

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

表6.2 ネットワークパラメーター

パラメーター	説明	値
networking	クラスターのネットワークの設定。	オブジェクト  **注記** インストール後に **networking** オブジェクトで指定したパラメーターを変更することはできません。

パラメーター	説明	値
networking.networkType	インストールする Red Hat OpenShift Networking ネットワークプラグイン。	**OpenShiftSDN** または **OVNKubernetes** のいずれか。**OpenShiftSDN** は、すべての Linux ネットワークの Container Network Interface (CNI)プラグインです。**OVNKubernetes** は、Linux ネットワークと、Linux サーバーと Windows サーバーの両方を含む Linux ネットワークおよびハイブリッドネットワーク用の CNI プラグインです。デフォルトの値は **OVNkubernetes** です。
networking.clusterNetwork	Pod の IP アドレスブロック。 デフォルト値は **10.128.0.0/14** で、ホストの接頭辞は **/23** です。 複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。	オブジェクトの配列。以下に例を示します。 networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23
networking.clusterNetwork.cidr	**networking.clusterNetwork** を使用する場合に必須です。IP アドレスブロック。 IPv4 ネットワーク	CIDR (Classless Inter-Domain Routing) 表記の IP アドレスブロック。IPv4 ブロックの接頭辞長は **0** から **32** の間になります。
networking.clusterNetwork.hostPrefix	それぞれの個別ノードに割り当てるサブネット接頭辞長。たとえば、**hostPrefix** が **23** に設定される場合、各ノードに指定の **cidr** から **/23** サブネットが割り当てられます。**hostPrefix** 値の **23** は、510 (2^(32 - 23) - 2) Pod IP アドレスを提供します。	サブネット接頭辞。 デフォルト値は **23** です。
networking.serviceNetwork	サービスの IP アドレスブロック。デフォルト値は **172.30.0.0/16** です。 OpenShift SDN および OVN-Kubernetes ネットワークプラグインは、サービスネットワークの単一 IP アドレスブロックのみをサポートします。	CIDR 形式の IP アドレスブロックを持つ配列。以下に例を示します。 networking: serviceNetwork: - 172.30.0.0/16

パラメーター	説明	値
networking.machineNetwork	マシンの IP アドレスブロック。 複数の IP アドレスブロックを指定する場合は、ブロックが重複しないようにしてください。	オブジェクトの配列。以下に例を示します。 networking: machineNetwork: - cidr: 10.0.0.0/16
networking.machineNetwork.cidr	**networking.machineNetwork** を使用する場合に必須です。IP アドレスブロック。libvirt と IBM Power Virtual Server を除くすべてのプラットフォームのデフォルト値は **10.0.0.0/16** です。libvirt の場合、デフォルト値は **192.168.126.0/24** です。IBM Power Virtual Server の場合、デフォルト値は **192.168.0.0/24** です。	CIDR 表記の IP ネットワークブロック。 例: **10.0.0.0/16** **注記** 優先される NIC が置かれている CIDR に一致する **networking.machineNetwork** を設定します。

6.5.2.3. オプションの設定パラメーター

オプションのインストール設定パラメーターは、以下の表で説明されています。

表6.3 オプションのパラメーター

パラメーター	説明	値
additionalTrustBundle	ノードの信頼済み証明書ストアに追加される PEM でエンコードされた X.509 証明書バンドル。この信頼バンドルは、プロキシーが設定されている場合にも使用することができます。	文字列
capabilities	オプションのコアクラスターコンポーネントのインストールを制御します。オプションのコンポーネントを無効にすることで、OpenShift Container Platform クラスターのフットプリントを削減できます。詳細は、**インストール** の「クラスター機能ページ」を参照してください。	文字列配列

パラメーター	説明	値
capabilities.baselineCapabilitySet	有効にするオプション機能の初期セットを選択します。有効な値は **None**、**v4.11**、**v4.12**、**vCurrent** です。デフォルト値は **vCurrent** です。	文字列
capabilities.additionalEnabledCapabilities	オプションの機能のセットを、**baselineCapabilitySet** で指定したものを超えて拡張します。このパラメーターで複数の機能を指定できます。	文字列配列
cpuPartitioningMode	ワークロードパーティション設定を使用して、OpenShift Container Platform サービス、クラスター管理ワークロード、およびインフラストラクチャー Pod を分離し、予約された CPU セットで実行できます。ワークロードパーティショニングは、インストール中にのみ有効にできます。インストール後に無効にすることはできません。このフィールドはワークロードのパーティショニングを有効にしますが、特定の CPU を使用するようにワークロードを設定するわけではありません。詳細は、**スケーラビリティとパフォーマンス** セクションの **ワークロードパーティショニング** ページを参照してください。	**None** または **AllNodes**。デフォルト値は **None** です。
compute	コンピュートノードを形成するマシンの設定。	**MachinePool** オブジェクトの配列。
compute.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現在、さまざまなアーキテクチャーのクラスターはサポートされていません。すべてのプールは同じアーキテクチャーを指定する必要があります。有効な値はデフォルト **amd64** です。	String

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、同時マルチ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化されます。  **重要** 同時マルチ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.name	**compute** を使用する場合に必須です。マシンプールの名前。	**worker**
compute.platform	**compute** を使用する場合に必須です。このパラメーターを使用して、ワーカーマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は **controlPlane.platform** パラメーターの値に一致する必要があります。	**alibabacloud**、**aws**、**azure**、**gcp**、**ibmcloud**、**nutanix**、**openstack**、**ovirt**、**powervs**、**vsphere**、または **{}**
compute.replicas	プロビジョニングするコンピュートマシン (ワーカーマシンとしても知られる) の数。	**2** 以上の正の整数。デフォルト値は **3** です。
featureSet	機能セットのクラスターを有効にします。機能セットは、デフォルトで有効にされない OpenShift Container Platform 機能のコレクションです。インストール中に機能セットを有効にする方法の詳細は、「機能ゲートの使用による各種機能の有効化」を参照してください。	文字列。**TechPreviewNoUpgrade** など、有効にする機能セットの名前。
controlPlane	コントロールプレーンを形成するマシンの設定。	**MachinePool** オブジェクトの配列。

パラメーター	説明	値
controlPlane.architecture	プール内のマシンの命令セットアーキテクチャーを決定します。現在、さまざまなアーキテクチャーのクラスターはサポートされていません。すべてのプールは同じアーキテクチャーを指定する必要があります。有効な値はデフォルト **amd64** です。	String
controlPlane: hyperthreading:	コントロールプレーンマシンで同時マルチスレッドまたは **hyperthreading** を有効/無効にするかどうか。デフォルトでは、同時スレッドはマシンのコアのパフォーマンスを上げるために有効にされます。  **重要** 同時スレッドを無効にする場合は、容量計画においてマシンパフォーマンスの大幅な低下が考慮に入れられていることを確認します。	**Enabled** または **Disabled**
controlPlane.name	**controlPlane** を使用する場合に必須です。マシンプールの名前。	**master**
controlPlane.platform	**controlPlane** を使用する場合に必須です。このパラメーターを使用して、コントロールプレーンマシンをホストするクラウドプロバイダーを指定します。このパラメーターの値は **compute.platform** パラメーターの値に一致する必要があります。	**alibabacloud**、**aws**、**azure**、**gcp**、**ibmcloud**、**nutanix**、**openstack**、**ovirt**、**powervs**、**vsphere**、または **{}**
controlPlane.replicas	プロビジョニングするコントロールプレーンマシンの数。	サポートされる値は **3** のみです (これはデフォルト値です)。

パラメーター	説明	値
credentialsMode	Cloud Credential Operator (CCO) モード。CCO は、モードが指定されていない場合に指定される認証情報の機能を動的に判別しようとします。この場合、複数のモードがサポートされるプラットフォームで mint モードが優先されます。  **注記** すべてのクラウドプロバイダーですべての CCO モードがサポートされているわけではありません。CCO モードの詳細は、**Cluster Operators リファレンス** の **Cloud Credential Operator** を参照してください。  **注記** AWS アカウントでサービスコントロールポリシー (SCP) が有効になっている場合は、**credentialsMode** パラメーターを **Mint**、**Passthrough** または **Manual** に設定する必要があります。	**Mint**、**Passthrough**、**Manual**、または空の文字列 (**""**)。
imageContentSources	release-image コンテンツのソースおよびリポジトリー。	オブジェクトの配列。この表の以下の行で説明されているように、**source** およびオプションで **mirrors** が含まれます。
imageContentSources.source	**imageContentSources** を使用する場合に必須です。ユーザーが参照するリポジトリーを指定します (例: イメージプル仕様)。	文字列
imageContentSources.mirrors	同じイメージが含まれる可能性のあるリポジトリーを 1 つ以上指定します。	文字列の配列。

パラメーター	説明	値
publish	Kubernetes API、OpenShift ルートなどのクラスターのユーザーに表示されるエンドポイントをパブリッシュまたは公開する方法。	**Internal** または **External**。デフォルト値は **External** です。 このパラメーターを **Internal** に設定することは、クラウド以外のプラットフォームではサポートされません。  **重要** フィールドの値が **Internal** に設定されている場合、クラスターは機能しなくなります。詳細は、BZ#1953035 を参照してください。
sshKey	クラスターマシンへのアクセスを認証するための SSH キー。  **注記** インストールのデバッグまたは障害復旧を実行する必要のある実稼働用の OpenShift Container Platform クラスターでは、**ssh-agent** プロセスが使用する SSH キーを指定します。	たとえば、**sshKey: ssh-ed25519 AAAA..** です。

1. すべてのクラウドプロバイダーですべての CCO モードがサポートされているわけではありません。CCO モードの詳細は、**認証と認可** コンテンツの「クラウドプロバイダーの認証情報の管理」を参照してください。

6.5.2.4. 追加の Alibaba Cloud 設定パラメーター

Alibaba Cloud の追加の設定パラメーターは、以下の表で説明されています。**alibabacloud** パラメーターは、Alibaba Cloud にインストールするときに使用される設定です。**defaultMachinePlatform** パラメーターは、独自のプラットフォーム設定を定義しないマシンプール用に Alibaba Cloud にインストールするときに使用されるデフォルト設定です。

これらのパラメーターは、指定されているコンピュートマシンとコントロールプレーンマシンの両方に適用されます。

注記

定義されている場合、パラメーター **compute.platform.alibabacloud** および **controlPlane.platform.alibabacloud** は、コンピューティングマシンとコントロールプレーンマシンの **platform.alibabacloud.defaultMachinePlatform** 設定をそれぞれ上書きします。

表6.4 オプションの Alibaba Cloud パラメーター

パラメーター	説明	値
compute.platform.alibabacloud.imageID	ECS インスタンスの作成に使用される imageID。ImageID はクラスターと同じリージョンに属している必要があります。	文字列。
compute.platform.alibabacloud.instanceType	InstanceType は、ECS インスタンスタイプを定義します。例: **ecs.g6.large**	文字列。
compute.platform.alibabacloud.systemDiskCategory	システムディスクのカテゴリーを定義します。例: **cloud_efficiency**、**cloud_essd**	文字列。
compute.platform.alibabacloud.systemDisksize	システムディスクのサイズをギビバイト (GiB) 単位で定義します。	integer
compute.platform.alibabacloud.zones	使用できるアベイラビリティーゾーンのリスト。例: **cn-hangzhou-h**、**cn-hangzhou-j**	文字列リスト。
controlPlane.platform.alibabacloud.imageID	ECS インスタンスの作成に使用される imageID。ImageID はクラスターと同じリージョンに属している必要があります。	文字列。
controlPlane.platform.alibabacloud.instanceType	InstanceType は、ECS インスタンスタイプを定義します。例: **ecs.g6.xlarge**	文字列。
controlPlane.platform.alibabacloud.systemDiskCategory	システムディスクのカテゴリーを定義します。例: **cloud_efficiency**、**cloud_essd**	文字列。

パラメーター	説明	値
controlPlane.platform.alibabacloud.systemDiskSize	システムディスクのサイズをギビバイト (GiB) 単位で定義します。	integer
controlPlane.platform.alibabacloud.zones	使用できるアベイラビリティーゾーンのリスト。例: **cn-hangzhou-h**、**cn-hangzhou-j**	文字列リスト。
platform.alibabacloud.region	必須。クラスターが作成される Alibaba Cloud リージョン。	文字列。
platform.alibabacloud.resourceGroupID	クラスターがインストールされている既存のリソースグループの ID。空の場合、インストールプログラムはクラスターの新しいリソースグループを作成します。	文字列。
platform.alibabacloud.tags	クラスター用に作成されたすべての Alibaba Cloud リソースに適用する追加のキーと値。	オブジェクト。
platform.alibabacloud.vpcID	クラスターをインストールする必要がある既存の VPC の ID。空の場合、インストールプログラムはクラスターの新しい VPC を作成します。	文字列。
platform.alibabacloud.vswitchIDs	クラスターリソースが作成される既存の VSwitches の ID リスト。既存の VSwitch は、既存の VPC も使用している場合にのみ使用できます。空の場合、インストールプログラムはクラスターの新しい VSwitch を作成します。	文字列リスト。
platform.alibabacloud.defaultMachinePlatform.imageID	コンピュートマシンとコントロールプレーンマシンの両方で、ECS インスタンスの作成に使用する必要があるイメージ ID。設定されている場合、イメージ ID はクラスターと同じリージョンに属している必要があります。	文字列。

パラメーター	説明	値
platform.alibabacloud.defaultMachinePlatform.instanceType	コンピュートマシンとコントロールプレーンマシンの両方で、ECS インスタンスの作成に使用される ECS インスタンスタイプ。例: **ecs.g6.xlarge**	文字列。
platform.alibabacloud.defaultMachinePlatform.systemDiskCategory	コンピュートマシンとコントロールプレーンマシンの両方におけるシステムディスクのカテゴリー。例: **cloud_efficiency**、**cloud_essd**。	文字列、たとえば ""、**cloud_efficiency**、**cloud_essd**。
platform.alibabacloud.defaultMachinePlatform.systemDiskSize	コンピュートマシンとコントロールプレーンマシンの両方で、ギビバイト (GiB) 単位のシステムディスクのサイズ。最小値は **120** です。	integer
platform.alibabacloud.defaultMachinePlatform.zones	コンピュートマシンとコントロールプレーンマシンの両方で、使用可能なアベイラビリティーゾーンのリスト。例: **cn-hangzhou-h**、**cn-hangzhou-j**	文字列リスト。
platform.alibabacloud.privateZoneID	クラスターの内部 API の DNS レコードを追加する既存のプライベートゾーンの ID。既存のプライベートゾーンは、既存の VPC も使用している場合にのみ使用できます。プライベートゾーンは、サブネットを含む VPC に関連付ける必要があります。プライベートゾーンを未設定のままにして、インストールプログラムがプライベートゾーンを作成するようにします。	文字列。

6.5.3. Alibaba Cloud 用にカスタマイズされた install-config.yaml ファイルのサンプル

インストール設定ファイル (**install-config.yaml**) をカスタマイズして、クラスターのプラットフォームに関する詳細を指定したり、必要なパラメーターの値を変更したりできます。

```
apiVersion: v1
baseDomain: alicloud-dev.devcluster.openshift.com
credentialsMode: Manual
compute:
```

```
- architecture: amd64
  hyperthreading: Enabled
  name: worker
  platform: {}
  replicas: 3
controlPlane:
  architecture: amd64
  hyperthreading: Enabled
  name: master
  platform: {}
  replicas: 3
metadata:
  creationTimestamp: null
  name: test-cluster 1
 networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  networkType: OVNKubernetes 2
  serviceNetwork:
  - 172.30.0.0/16
platform:
  alibabacloud:
    defaultMachinePlatform: 3
      instanceType: ecs.g6.xlarge
      systemDiskCategory: cloud_efficiency
      systemDiskSize: 200
    region: ap-southeast-1 4
    resourceGroupID: rg-acfnw6j3hyai 5
    vpcID: vpc-0xifdjerdibmaqvtjob2b 6
    vswitchIDs: 7
    - vsw-0xi8ycgwc8wv5rhviwdq5
    - vsw-0xiy6v3z2tedv009b4pz2
publish: External
pullSecret: '{"auths": {"cloud.openshift.com": {"auth": ... }' 8
sshKey: |
  ssh-rsa AAAA... 9
```

1. 必須。インストールプログラムにより、クラスター名の入力を求められます。
2. インストールするクラスターネットワークプラグイン。サポートされている値は **OVNKubernetes** と **OpenShiftSDN** です。デフォルトの値は **OVNkubernetes** です。
3. オプション。独自のプラットフォーム設定を定義しないマシンプールのパラメーターを指定します。
4. 必須。インストールプログラムにより、クラスターをデプロイするリージョンの入力を求められます。
5. オプション。クラスターをインストールする必要がある既存のリソースグループを指定します。
8. 必須。インストールプログラムは、プルシークレットの入力を求めます。

9 オプション。インストールプログラムは、クラスター内のマシンへのアクセスに使用する SSH キー値の入力を求めます。

6 7 オプション。これらは vswitchID 値の例です。

6.5.4. 必要なインストールマニフェストの生成

クラスターがマシンを設定するために必要な 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。

手順

1. インストールプログラムが含まれているディレクトリーから次のコマンドを実行して、マニフェストを生成します。

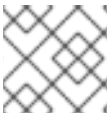
```
$ openshift-install create manifests --dir <installation_directory>
```

ここで、

<installation_directory>
インストールプログラムがファイルを作成するディレクトリーを指定します。

6.5.5. Cloud Credential Operator ユーティリティーの設定

Cloud Credential Operator (CCO) が手動モードで動作しているときにクラスターの外部からクラウドクレデンシャルを作成および管理するには、CCO ユーティリティー (**ccoctl**) バイナリーを抽出して準備します。



注記

ccoctl ユーティリティーは、Linux 環境で実行する必要がある Linux バイナリーです。

前提条件

- クラスター管理者のアクセスを持つ OpenShift Container Platform アカウントを使用できる。
- OpenShift CLI (**oc**) がインストールされている。

手順

1. 以下のコマンドを実行して、OpenShift Container Platform リリースイメージを取得します。

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 以下のコマンドを実行して、OpenShift Container Platform リリースイメージから CCO コンテナーイメージを取得します。

```
$ CCO_IMAGE=$(oc adm release info --image-for='cloud-credential-operator'
$RELEASE_IMAGE -a ~/.pull-secret)
```

注記

$RELEASE_IMAGE のアーキテクチャーが、**ccoctl**ツールを使用する環境のアーキテクチャーと一致していることを確認してください。

3. 以下のコマンドを実行して、OpenShift Container Platform リリースイメージ内の CCO コンテナーイメージから **ccoctl** バイナリーを抽出します。

```
$ oc image extract $CCO_IMAGE --file="/usr/bin/ccoctl" -a ~/.pull-secret
```

4. 次のコマンドを実行して、権限を変更して **ccoctl** を実行可能にします。

```
$ chmod 775 ccoctl
```

検証

- **ccoctl** が使用できることを確認するには、help ファイルを表示します。コマンドを実行するときは、相対ファイル名を使用します。以下に例を示します。

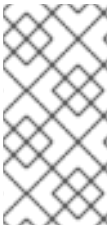
```
$ ./ccoctl.rhel9
```

出力例

```
OpenShift credentials provisioning tool

Usage:
  ccoctl [command]

Available Commands:
  alibabacloud Manage credentials objects for alibaba cloud
  aws          Manage credentials objects for AWS cloud
  gcp          Manage credentials objects for Google cloud
  help         Help about any command
  ibmcloud     Manage credentials objects for IBM Cloud
  nutanix      Manage credentials objects for Nutanix

Flags:
  -h, --help   help for ccoctl

Use "ccoctl [command] --help" for more information about a command.
```

6.5.6. ccoctl ツールを使用した OpenShift Container Platform コンポーネントのクレデンシャルの作成

OpenShift Container Platform Cloud Credential Operator (CCO) ユーティリティーを使用して、Alibaba Cloud RAM ユーザーとクラスター内コンポーネントごとのポリシーの作成を自動化できます。



注記

デフォルトで、**ccoctl** はコマンドが実行されるディレクトリーにオブジェクトを作成します。オブジェクトを別のディレクトリーに作成するには、**--output-dir** フラグを使用します。この手順では、**<path_to_ccoctl_output_dir>** を使用してこの場所を参照します。

前提条件

以下が必要になります。

- **ccoctl** バイナリーを抽出して準備している。
- OpenShift Container Platform クラスターを作成するための十分な権限を持つ RAM ユーザーを作成している。
- その RAM ユーザーの AccessKeyID (**access_key_id**) と AccessKeySecret (**access_key_secret**) をローカルコンピューターの **~/.alibabacloud/credentials ファイル** に追加しました。

手順

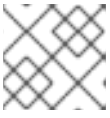
1. 以下のコマンドを実行して、**$RELEASE_IMAGE** 変数を設定します。

```
$ RELEASE_IMAGE=$(./openshift-install version | awk '/release image/ {print $3}')
```

2. 以下のコマンドを実行して、OpenShift Container Platform リリースイメージから **CredentialsRequest** オブジェクトのリストを抽出します。

```
$ oc adm release extract \
  --from=$RELEASE_IMAGE \
  --credentials-requests \
  --cloud=alibabacloud \
  --to=<path_to_directory_with_list_of_credentials_requests>/credrequests (1)
```

(1) **credrequests** は、**CredentialsRequest** オブジェクトのリストが格納されるディレクトリーです。ディレクトリーが存在しない場合、このコマンドはディレクトリーを作成します。



注記

このコマンドの実行には少し時間がかかる場合があります。

3. クラスターでクラスター機能を使用して 1 つ以上のオプションコンポーネントを無効にする場合は、無効なコンポーネントの **CredentialsRequest** カスタムリソースを削除します。

Alibaba Cloud 上の OpenShift Container Platform 4.12 の credrequests ディレクトリーの内容の例

```
0000_30_machine-api-operator_00_credentials-request.yaml (1)
0000_50_cluster-image-registry-operator_01-registry-credentials-request-alibaba.yaml (2)
0000_50_cluster-ingress-operator_00-ingress-credentials-request.yaml (3)
```

```
0000_50_cluster-storage-operator_03_credentials_request_alibaba.yaml 4
```

1. Machine API Operator CR が必要です。
2. Image Registry Operator CR が必要です。
3. Ingress Operator CR が必要です。
4. Storage Operator CR はオプションのコンポーネントであり、クラスターで無効になっている場合があります。

4. **ccoctl** ツールを使用して、**credrequests** ディレクトリーですべての **CredentialsRequest** オブジェクトを処理します。

   a. ツールを使用するには、次のコマンドを実行します。

```
$ ccoctl alibabacloud create-ram-users \
  --name <name> \
  --region=<alibaba_region> \
  --credentials-requests-dir=
<path_to_directory_with_list_of_credentials_requests>/credrequests \
  --output-dir=<path_to_ccoctl_output_dir>
```

   ここで、

   - **<name>** は、追跡用に作成されたクラウドリソースにタグを付けるために使用される名前です。
   - **<alibaba_region>** は、クラウドリソースが作成される Alibaba Cloud リージョンです。
   - **<path_to_directory_with_list_of_credentials_requests>/credrequests** は、コンポーネント **CredentialsRequest** オブジェクトのファイルを含むディレクトリーです。
   - **<path_to_ccoctl_output_dir>** は、生成されたコンポーネントクレデンシャルシークレットが配置されるディレクトリーです。

注記

クラスターで **TechPreviewNoUpgrade** 機能セットによって有効化されたテクノロジープレビュー機能を使用している場合は、**--enable-tech-preview** パラメーターを含める必要があります。

出力例

```
2022/02/11 16:18:26 Created RAM User: user1-alicloud-openshift-machine-api-
alibabacloud-credentials
2022/02/11 16:18:27 Ready for creating new ram policy user1-alicloud-openshift-
machine-api-alibabacloud-credentials-policy-policy
2022/02/11 16:18:27 RAM policy user1-alicloud-openshift-machine-api-alibabacloud-
credentials-policy-policy has created
2022/02/11 16:18:28 Policy user1-alicloud-openshift-machine-api-alibabacloud-
credentials-policy-policy has attached on user user1-alicloud-openshift-machine-api-
alibabacloud-credentials
```

```
2022/02/11 16:18:29 Created access keys for RAM User: user1-alicloud-openshift-machine-api-alibabacloud-credentials
2022/02/11 16:18:29 Saved credentials configuration to: user1-alicloud/manifests/openshift-machine-api-alibabacloud-credentials-credentials.yaml
...
```

注記

RAM ユーザーは、同時に最大 2 つの AccessKey を持つことができます。**ccoctl alibabacloud create-ram-users** を 3 回以上実行すると、以前に生成されたマニフェストシークレットが古くなり、新しく生成されたシークレットを再適用する必要があります。

b. OpenShift Container Platform シークレットが作成されていることを確認します。

```
$ ls <path_to_ccoctl_output_dir>/manifests
```

出力例:

```
openshift-cluster-csi-drivers-alibaba-disk-credentials-credentials.yaml
openshift-image-registry-installer-cloud-credentials-credentials.yaml
openshift-ingress-operator-cloud-credentials-credentials.yaml
openshift-machine-api-alibabacloud-credentials-credentials.yaml
```

RAM ユーザーとポリシーが Alibaba Cloud にクエリーを実行して作成されていることを確認できます。詳細については、RAM ユーザーとポリシーのリスト表示に関する Alibaba Cloud のドキュメントを参照してください。

5. 生成されたクレデンシャルファイルをターゲットマニフェストディレクトリーにコピーします。

```
$ cp ./<path_to_ccoctl_output_dir>/manifests/*credentials.yaml ./<path_to_installation>dir>/manifests/
```

ここで、

<path_to_ccoctl_output_dir>

ccoctl alibabacloud create-ram-users コマンドによって作成されるディレクトリーを指定します。

<path_to_installation_dir>

インストールプログラムがファイルを作成するディレクトリーを指定します。

6.6. 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する。
- ホスト上のクラウドプロバイダーアカウントに、クラスターをデプロイするための適切な権限があることを確認してください。アカウントの権限が正しくないと、インストールプロセスが失敗し、不足している権限を示すエラーメッセージが表示されます。

手順

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

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

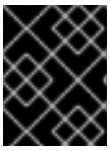
1 **<installation_directory>** に、カスタマイズした **./install-config.yaml** ファイルの場所を指定します。

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。

検証

クラスターのデプロイが正常に完了すると、次のようになります。

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

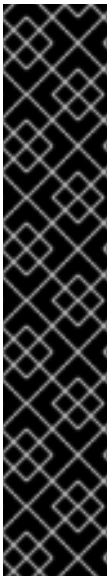


重要

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

出力例

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



重要

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

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

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

重要

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.13 のすべてのコマンドを実行することはできません。新規バージョンの **oc** をダウンロードし、インストールします。

Linux への OpenShift CLI のインストール

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

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **Product Variant** ドロップダウンリストからアーキテクチャーを選択します。
3. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
4. **OpenShift v4.13 Linux Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。
5. アーカイブを展開します。

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

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

```
$ echo $PATH
```

検証

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

```
$ oc <command>
```

Windows への OpenShift CLI のインストール

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

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
3. **OpenShift v4.13 Windows Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。
4. ZIP プログラムでアーカイブを解凍します。
5. **oc** バイナリーを、**PATH** にあるディレクトリーに移動します。
   PATH を確認するには、コマンドプロンプトを開いて以下のコマンドを実行します。

```
C:\> path
```

検証

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

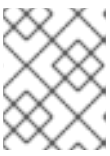
```
C:\> oc <command>
```

macOS への OpenShift CLI のインストール

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

手順

1. Red Hat カスタマーポータルの OpenShift Container Platform ダウンロードページ に移動します。
2. **バージョン** ドロップダウンリストから適切なバージョンを選択します。
3. **OpenShift v4.13 macOS Client** エントリーの横にある **Download Now** をクリックして、ファイルを保存します。



注記

macOS arm64 の場合は、**OpenShift v4.13 macOS arm64 Client** エントリーを選択します。

4. アーカイブを展開し、解凍します。
5. **oc** バイナリーをパスにあるディレクトリーに移動します。

PATH を確認するには、ターミナルを開き、以下のコマンドを実行します。

```
$ echo $PATH
```

検証

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

```
$ oc <command>
```

6.8. CLI の使用によるクラスターへのログイン

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

前提条件

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

手順

1. **kubeadmin** 認証情報をエクスポートします。

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

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

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

```
$ oc whoami
```

出力例

```
system:admin
```

6.9. WEB コンソールを使用したクラスターへのログイン

kubeadmin ユーザーは、OpenShift Container Platform のインストール後はデフォルトで存在します。OpenShift Container Platform Web コンソールを使用し、**kubeadmin** ユーザーとしてクラスターにログインできます。

前提条件

- インストールホストにアクセスできる。

- クラスターのインストールを完了しており、すべてのクラスター Operator が利用可能である。

手順

1. インストールホストで **kubeadmin-password** ファイルから **kubeadmin** ユーザーのパスワードを取得します。

```
$ cat <installation_directory>/auth/kubeadmin-password
```

注記

または、インストールホストで **<installation_directory>/.openshift_install.log** ログファイルから **kubeadmin** パスワードを取得できます。

2. OpenShift Container Platform Web コンソールルートをリスト表示します。

```
$ oc get routes -n openshift-console | grep 'console-openshift'
```

注記

または、インストールホストで **<installation_directory>/.openshift_install.log** ログファイルからで OpenShift Container Platform ルートを取得できます。

出力例

```
console     console-openshift-console.apps.<cluster_name>.<base_domain>           console
https   reencrypt/Redirect   None
```

3. Web ブラウザーで前述のコマンドの出力で詳細に説明されたルートに移動し、**kubeadmin** ユーザーとしてログインします。

6.10. OPENSHIFT CONTAINER PLATFORM の TELEMETRY アクセス

OpenShift Container Platform 4.13 では、クラスターの健全性および正常に実行された更新についてのメトリクスを提供するためにデフォルトで実行される Telemetry サービスにもインターネットアクセスが必要です。クラスターがインターネットに接続されている場合、Telemetry は自動的に実行され、クラスターは OpenShift Cluster Manager Hybrid Cloud Console に登録されます。

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

関連情報

- Telemetry サービスの詳細は、リモートヘルスモニタリング を参照してください。
- OpenShift Container Platform Web コンソールへのアクセスと理解の詳細については、Web コンソールへのアクセス を参照してください。

6.11. 次のステップ

- インストールの検証
- クラスターのカスタマイズ
- リモートヘルスレポート

第7章 ALIBABA CLOUD でのクラスターのアンインストール

Alibaba Cloud にデプロイしたクラスターを削除できます。

7.1. 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

installer-provisioned infrastructure を使用するクラスターは、クラウドから削除できます。

注記

アンインストール後に、とくに user-provisioned infrastructure (UPI) クラスターで適切に削除されていないリソースがあるかどうかについて、クラウドプロバイダーを確認します。インストーラーが作成されなかったり、インストーラーがアクセスできない場合には、リソースがある可能性があります。

前提条件

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

手順

1. クラスターをインストールするために使用したコンピューターのインストールプログラムが含まれるディレクトリーから、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。