OpenShift Container Platform 4.2

インストール

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

Last Updated: 2020-04-15

OpenShift Container Platform 4.2 インストール

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

法律上の通知

概要

本書では、OpenShift Container Platform 4.2 のインストール方法と、一部の設定プロセスの詳細について説明します。

目次

第1章 インストールログの収集

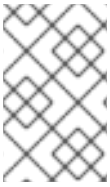
OpenShift Container Platform のインストールした場合のトラブルシューティングのために、ブートストラップおよびコントロールプレーン、またはマスター、マシンからログを収集できます。

前提条件

- OpenShift Container Platform クラスターのインストールを試みたが、インストールに失敗している。
- SSH キーをインストールプログラムに指定しており、そのキーは実行中の **ssh-agent** プロセスにある。

1.1. 失敗したインストールのログの収集

SSH キーをインストールプログラムに指定している場合、失敗したインストールについてのデータを収集することができます。



注記

実行中のクラスターからログを収集する場合とは異なるコマンドを使用して失敗したインストールについてのログを収集します。実行中のクラスターからログを収集する必要がある場合は、**oc adm must-gather** コマンドを使用します。

前提条件

- OpenShift Container Platform のインストールがブートストラッププロセスの終了前に失敗している。ブートストラップノードは実行中であり、SSH でアクセスできる必要がある。
- **ssh-agent** プロセスはコンピューター上でアクティブであり、**ssh-agent** プロセスとインストールプログラムの両方に同じ SSH キーを提供している。
- 独自にプロビジョニングしたインフラストラクチャーにクラスターのインストールを試行した場合には、コントロールプレーン、またはマスター、マシンの完全修飾ドメイン名があること。

手順

1. ブートストラップおよびコントロールプレーンマシンからインストールログを収集するために必要なコマンドを生成します。
   - 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している場合は、以下のコマンドを実行します。

     ```
     $ ./openshift-install gather bootstrap --dir=<directory> 1
     ```

     1 **installation_directory** は、インストールプログラムが作成する OpenShift Container Platform 定義ファイルを保存しているディレクトリーです。

     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の場合、インストールプログラムは、ホスト名または IP アドレスを指定しなくてもよいようにクラスターについての情報を保存します。

- 独自にプロビジョニングしたインフラストラクチャーを使用している場合は、以下のコマンドを実行します。

```
$ ./openshift-install gather bootstrap --dir=<directory> \ 1
  --bootstrap <bootstrap_address> \ 2
  --master <master_1_address> \ 3
  --master <master_2_address> \ 4
  --master <master_3_address>" 5
```

1. **installation_directory** は、インストールプログラムが作成する OpenShift Container Platform 定義ファイルを保存しているディレクトリーです。
2. **<bootstrap_address>** は、クラスターのブートストラップマシンの完全修飾ドメイン名または IP アドレスです。
3. 4. 5. クラスター内のそれぞれのコントロールプレーン、またはマスター、マシンについては、**<master_*_address>** をその完全修飾ドメイン名または IP アドレスに置き換えます。

注記

デフォルトクラスターには 3 つのコントロールプレーンマシンが含まれます。クラスターが使用する数にかかわらず、表示されるようにすべてのコントロールプレーンマシンを一覧表示します。

コマンド出力は以下の例のようになります。

```
INFO Pulling debug logs from the bootstrap machine
INFO Bootstrap gather logs captured here "<directory>/log-bundle-<timestamp>.tar.gz"
```

インストールの失敗についての Red Hat サポートケースを作成する場合は、圧縮したログをケースに含めるようにしてください。

1.2. ホストへの SSH アクセスによるログの手動収集

must-gather または自動化された収集方法が機能しない場合にログを手動で収集します。

前提条件

- ホストへの SSH アクセスがあること。

手順

1. 以下を実行し、**journalctl** コマンドを使用してブートストラップホストから **bootkube.service** サービスログを収集します。

```
$ journalctl -b -f -u bootkube.service
```

2. Podman ログを使用して、ブートストラップホストのコンテナーログを収集します。これは、ホストからすべてのコンテナーログを取得するためにループで表示されます。

```
$ for pod in $(sudo podman ps -a -q); do sudo podman logs $pod; done
```

3. または、以下を実行し、**tail** コマンドを使用してホストのコンテナーログを収集します。

```
# tail -f /var/lib/containers/storage/overlay-containers/*/userdata/ctr.log
```

4. 以下を実行し、**journalctl** コマンドを使用して **kubelet.service** および **crio.service** サービスログをマスターホストから収集します。

```
$ journalctl -b -f -u kubelet.service -u crio.service
```

5. 以下を実行し、**tail** コマンドを使用してマスターおよびワーカーホストコンテナーログを収集します。

```
$ sudo tail -f /var/log/containers/*
```

1.3. ホストへの SSH アクセスを使用しないログの手動収集

must-gather または自動化された収集方法が機能しない場合にログを手動で収集します。

ノードへの SSH アクセスがない場合は、システムジャーナルにアクセスし、ホストで生じていることを調査できます。

前提条件

- OpenShift Container Platform のインストールが完了している。
- API サービスが機能している。
- システム管理者権限がある。

手順

1. 以下を実行し、**/var/log** の下にある **journald** ユニットログにアクセスします。

```
$ oc adm node-logs --role=master -u kubelet
```

2. 以下を実行し、**/var/log** の下にあるホストファイルのパスにアクセスします。

```
$ oc adm node-logs --role=master --path=openshift-apiserver
```

第2章 インストール設定

2.1. 各種プラットフォームのインストール方法

各種のプラットフォームで各種のインストールを実行できます。

表2.1 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のオプション

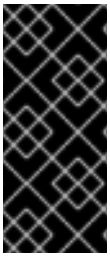
	AWS	Azure	GCP	OpenStack	ベアメタル	vSphere	IBM Z
デフォルト	X	X	X				
カスタム	X	X	X	X			
Network Operator	X	X	X				

表2.2 ユーザーによってプロビジョニングされるインフラストラクチャーのオプション

	AWS	Azure	GCP	OpenStack	ベアメタル	vSphere	IBM Z
カスタム	X		X		X	X	X
Network Operator					X	X	
ネットワークが制限されたインストール	X				X	X	

2.2. 接続するネットワークが制限された環境でのインストール用のミラーレジストリーの作成

クラスターをネットワークが制限された環境でプロビジョニングするインフラストラクチャーにインストールする前に、ミラーレジストリーを作成する必要があります。ネットワークが制限された環境でのインストールは、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ではなく、ユーザーがプロビジョニングするインフラストラクチャーでのみサポートされます。



重要

ミラーリポジトリーに設定するデータを取得するために、インターネットへのアクセスが必要です。この手順では、ご使用のネットワークとインターネットのどちらにもアクセスできる bastion ホストにミラーレジストリーを配置します。bastion ホストへのアクセスがない場合は、制限に最も適切に対応する方法で、ミラーレジストリーのコンテンツを制限されたネットワークで使用できるようにします。

2.2.1. ミラーレジストリーについて

インストールプログラムを生成するために必要な OpenShift Container Platform レジストリーおよびイメージのコンテンツをミラーリングできます。

ミラーレジストリーは、ネットワークが制限された環境でインストールを実行するために必要となる主要なコンポーネントです。このミラーは、インターネットと制限されたネットワークの両方にアクセスできる bastion ホストで、または制限を満たす他の方法を使用して実行できます。

OpenShift Container Platform がリリースペイロードの整合性を検証する方法により、ローカルレジストリーのイメージ参照は Quay.io 上で Red Hat によってホストされるものと同一になります。インストールのブートストラッププロセスでは、プルされるリポジトリーの種類を問わず、イメージには同じダイジェストがなければなりません。リリースペイロードが同一になるようにするには、イメージをローカルリポジトリーにミラーリングします。

2.2.2. bastion ホストの準備

ミラーレジストリーを作成する前に、bastion ホストを準備する必要があります。

2.2.2.1. CLI のインストール

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

重要

以前のバージョンの **oc** をインストールしている場合、これを使用して OpenShift Container Platform 4.2 のすべてのコマンドを実行することはできません。新規バージョンの **oc**をダウンロードし、インストールします。

手順

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

注記

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

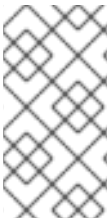
3. ファイルをファイルシステムに保存します。
4. 圧縮ファイルを展開します。
5. これを **PATH** にあるディレクトリーに配置します。

CLI のインストール後は、**oc** コマンドを使用して利用できます。

```
$ oc <command>
```

2.2.3. ミラーレジストリーの作成

OpenShift Container Platform のインストールに必要なミラーリングされたコンテンツをホストするためにレジストリーを作成します。ネットワークが制限された環境でのインストールの場合は、bastion ホストにミラーを配置する必要があります。



注記

以下の手順では、**/opt/registry** フォルダーにデータを保存し、**podman** コンテナーで実行される単純なレジストリーを作成します。Red Hat Quay などの異なるレジストリーソリューションを使用できます。以下の手順で、レジストリーが正常に機能することを確認します。

前提条件

- レジストリーホストとして使用する Red Hat Enterprise Linux (RHEL) サーバーがネットワーク上にある。
- レジストリーホストはインターネットにアクセスできる。

手順

bastion ホストで以下のアクションを実行します。

1. 必要なパッケージをインストールします。

```
# yum -y install podman httpd-tools
```

podman パッケージは、レジストリーを実行する際に使用するコンテナーパッケージを提供します。**httpd-tools** パッケージは、ユーザーの作成に使用する **htpasswd** ユーティリティーを提供します。

2. レジストリーのフォルダーを作成します。

```
# mkdir -p /opt/registry/{auth,certs,data}
```

これらのフォルダーはレジストリーコンテナー内にマウントされます。

3. レジストリーの証明書を指定します。既存の信頼される認証局がない場合は、自己署名の証明書を生成することができます。

```
$ cd /opt/registry/certs
# openssl req -newkey rsa:4096 -nodes -sha256 -keyout domain.key -x509 -days 365 -out
domain.crt
```

プロンプト時に、証明書に必要な値を指定します。

国名 (2 文字コード)	場所として 2 文字の ISO 国コードを指定します。ISO 3166 country codes 標準を参照してください。
State or Province Name (正式名)	都道府県名の正式名を入力します。

Locality Name (例: 市)	市の名前を入力します。
Organization Name (例: 会社)	会社名を入力します。
Organizational Unit Name (例: セクション)	部門名を入力します。
Common Name (例: 本人の名前またはサーバーのホスト名)	レジストリーホストのホスト名を入力します。ホスト名が DNS にあり、予想される IP アドレスに解決されていることを確認します。
Email Address	メールアドレスを入力します。詳細は、OpenSSL ドキュメントの req の説明を参照してください。

4. **bcrpt** 形式を使用するレジストリー用のユーザー名およびパスワードを生成します。

```
# htpasswd -bBc /opt/registry/auth/htpasswd <user_name> <password> 1
```

1 **<user_name>** および **<password>** をユーザー名およびパスワードに置き換えます。

5. レジストリーをホストする **mirror-registry** コンテナーを作成します。

```
# podman run --name mirror-registry -p <local_registry_host_port>:5000 \ 1
    -v /opt/registry/data:/var/lib/registry:z \
    -v /opt/registry/auth:/auth:z \
    -e "REGISTRY_AUTH=htpasswd" \
    -e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" \
    -e REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd \
    -v /opt/registry/certs:/certs:z \
    -e REGISTRY_HTTP_TLS_CERTIFICATE=/certs/domain.crt \
    -e REGISTRY_HTTP_TLS_KEY=/certs/domain.key \
    -d docker.io/library/registry:2
```

1 **<local_registry_host_port>** については、ミラーレジストリーがコンテンツの送信に使用するポートを指定します。

6. レジストリーに必要なポートを開きます。

```
# firewall-cmd --add-port=<local_registry_host_port>/tcp --zone=internal --permanent (1)
# firewall-cmd --add-port=<local_registry_host_port>/tcp --zone=public   --permanent (2)
# firewall-cmd --reload
```

(1) (2) **<local_registry_host_port>** については、ミラーレジストリーがコンテンツの送信に使用するポートを指定します。

7. 信頼された証明書の一覧に自己署名証明書を追加します。

```
# cp /opt/registry/certs/domain.crt /etc/pki/ca-trust/source/anchors/
# update-ca-trust
```

ミラープロセス中にレジストリーにログインするには、証明書を信頼する必要があります。

8. レジストリーが使用できることを確認します。

```
$ curl -u <user_name>:<password> -k https://<local_registry_host_name>:
<local_registry_host_port>/v2/_catalog (1)

{"repositories":[]}
```

(1) **<user_name>** および **<password>**については、レジストリーにユーザー名とパスワードを指定します。**<local_registry_host_name>**については、**registry.example.com** などのように、証明書に指定したレジストリードメイン名を指定します。**<local_registry_host_port>** については、ミラーレジストリーがコンテンツの送信に使用するポートを指定します。

コマンドの出力に空のリポジトリーが表示される場合、レジストリーは利用可能な状態です。

2.2.4. レジストリーのプルシークレットへの追加

OpenShift Container Platform クラスターをネットワークが制限された環境にインストールする前に、OpenShift Container Platform クラスターのプルシークレットを変更してローカルレジストリーを記述します。

前提条件

- ネットワークが制限された環境で使用するミラーレジストリーを設定していること。

手順

bastion ホストで以下の手順を実行します。

1. Red Hat OpenShift Cluster Manager サイトの Pull Secret ページから **registry.redhat.io** プルシークレットをダウンロードします。
2. ミラーレジストリーの base64 でエンコードされたユーザー名およびパスワードまたはトークンを生成します。

```
$ echo -n '<user_name>:<password>' | base64 -w0 (1)

BGVtbYk3ZHAtqXs=
```

1 **<user_name>** および **<password>** については、レジストリーに設定したユーザー名およびパスワードを指定します。

3. JSON 形式でプルシークレットのコピーを作成します。

```
$ cat ./pull-secret.text | jq .  > <path>/<pull-secret-file>1
```

1 プルシークレットを保存するフォルダーへのパスおよび作成する JSON ファイルの名前を指定します。

ファイルの内容は以下の例のようになります。

```
{
  "auths": {
    "cloud.openshift.com": {
      "auth": "b3BlbnNo...",
      "email": "you@example.com"
    },
    "quay.io": {
      "auth": "b3BlbnNo...",
      "email": "you@example.com"
    },
    "registry.connect.redhat.com": {
      "auth": "NTE3Njg5Nj...",
      "email": "you@example.com"
    },
    "registry.redhat.io": {
      "auth": "NTE3Njg5Nj...",
      "email": "you@example.com"
    }
  }
}
```

4. 新規ファイルを編集し、レジストリーについて記述するセクションをこれに追加します。

```
  "auths": {
...
    "<local_registry_host_name>:<local_registry_host_port>": { 1
      "auth": "<credentials>", 2
      "email": "you@example.com"
  },
...
```

1 **<local_registry_host_name>** については、証明書に指定したレジストリードメイン名を指定し、**<local_registry_host_port>** については、ミラーレジストリーがコンテンツを送るために使用するポートを指定します。

2 **<credentials>** については、生成したミラーレジストリーの base64 でエンコードされたユーザー名およびパスワードを指定します。

ファイルは以下の例のようになります。

```
{
  "auths": {
    "cloud.openshift.com": {
      "auth": "b3BlbnNo...",
      "email": "you@example.com"
    },
    "quay.io": {
      "auth": "b3BlbnNo...",
      "email": "you@example.com"
    },
    "registry.connect.redhat.com": {
      "auth": "NTE3Njg5Nj...",
      "email": "you@example.com"
    },
    "<local_registry_host_name>:<local_registry_host_port>": {
      "auth": "<credentials>",
      "email": "you@example.com"
    },
    "registry.redhat.io": {
      "auth": "NTE3Njg5Nj...",
      "email": "you@example.com"
    }
  }
}
```

2.2.5. OpenShift Container Platform イメージリポジトリーのミラーリング

クラスターのインストールまたはアップグレードに使用する OpenShift Container Platform イメージリポジトリーをミラーリングします。

前提条件

- ネットワークが制限された環境で使用するミラーレジストリーを設定し、設定した証明書および認証情報にアクセスできること。
- Red Hat OpenShift Cluster Manager のサイトの「Pull Secret」ページからプルシークレットをダウンロードしており、ミラーリポジトリーに認証を組み込むようにこれを変更している。

手順

bastion ホストで以下の手順を実行します。

1. 「OpenShift Container Platform ダウンロード」ページを確認し、インストールする必要のある OpenShift Container Platform のバージョンを判別します。
2. 必要な環境変数を設定します。

```
$ export OCP_RELEASE=<release_version> 1
$ export LOCAL_REGISTRY='<local_registry_host_name>:<local_registry_host_port>' 2
$ export LOCAL_REPOSITORY='<repository_name>' 3
$ export PRODUCT_REPO='openshift-release-dev' 4
$ export LOCAL_SECRET_JSON='<path_to_pull_secret>' 5
$ export RELEASE_NAME="ocp-release" 6
```

1. **<release_version>** については、インストールする OpenShift Container Platform のバージョン番号 (例: **4.2.0**) を指定します。

2. **<local_registry_host_name>** については、ミラーレジストリーのレジストリードメイン名を指定し、**<local_registry_host_port>** については、コンテンツの送信に使用するポートを指定します。

3. **<repository_name>**については、**ocp4/openshift4**などのレジストリーに作成するリポジトリーの名前を指定します。

4. ミラーリングするリポジトリー。実稼働環境のリリースの場合には、**openshift-release-dev**を指定する必要があります。

5. **<path_to_pull_secret>**については、作成したミラーレジストリーのプルシークレットおよびファイル名の絶対パスを指定します。

6. リリースミラー。実稼働環境のリリースについては、**ocp-release**を指定する必要があります。

3. リポジトリーをミラーリングします。

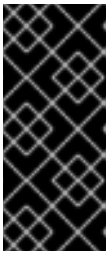
```
$ oc adm -a ${LOCAL_SECRET_JSON} release mirror \
    --from=quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE} \
    --to=${LOCAL_REGISTRY}/${LOCAL_REPOSITORY} \
    --to-release-image=${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}:${OCP_RELEASE}
```

このコマンドは、リリース情報をダイジェストとしてプルします。その出力には、クラスターのインストール時に必要な **imageContentSources** データが含まれます。

4. 直前のコマンドの出力の **imageContentSources** セクション全体を記録します。ミラーの情報はミラーリングされたリポジトリーに一意であり、インストール時に **imageContentSources** セクションを **install-config.yaml** ファイルに追加する必要があります。

5. ミラーリングしたコンテンツをベースとしているインストールプログラムを作成するには、これを展開し、リリースに固定します。

```
$ oc adm -a ${LOCAL_SECRET_JSON} release extract --command=openshift-install
"${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}:${OCP_RELEASE}"
```



重要

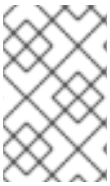
選択した OpenShift Container Platform バージョンに適したイメージを使用するには、ミラーリングされたコンテンツからインストールプログラムを展開する必要があります。

インターネット接続のあるマシンで、このステップを実行する必要があります。

2.2.6. 代替のレジストリーまたはミラーリングされたレジストリーでの Samples Operator イメージストリームの使用

Samples Operator によって管理される OpenShift namespace のほとんどのイメージストリームは、Red Hat レジストリーの registry.redhat.io にあるイメージを参照します。ミラーリングはこれらのイメージストリームには適用されません。**jenkins**、**jenkins-agent-maven**、および **jenkins-agent-**

nodejs イメージストリームは、インストールペイロードからのもので、Samples Operator によって管理されるため、これらのイメージストリームには追加のミラーリングの手順は必要ありません。



注記

cli、**installer**、 **must-gather**、および **tests** イメージストリームはインストールペイロードの一部ですが、Samples Operator によって管理されません。これらについては、この手順で扱いません。

前提条件

- **cluster-admin** ロールを持つユーザーとしてのクラスターへのアクセス。
- ミラーレジストリーのプルシークレットの作成。

手順

1. ミラーリングする特定のイメージストリームのイメージにアクセスします。

```
$ oc get is <imagestream> -n openshift -o json | jq .spec.tags[].from.name | grep
registry.redhat.io
```

2. ネットワークが制限された環境で必要とするイメージストリームに関連付けられた registry.redhat.io のイメージを定義されたミラーのいずれかにミラーリングします。

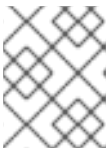
```
$ oc image mirror registry.redhat.io/rhscl/ruby-25-rhel7:latest ${MIRROR_ADDR}/rhscl/ruby-
25-rhel7:latest
```

3. クラスターのイメージ設定オブジェクトに、ミラーに必要な信頼される CA を追加します。

```
$ oc create configmap registry-config --from-
file=${MIRROR_ADDR_HOSTNAME}..5000=$path/ca.crt -n openshift-config
$ oc patch image.config.openshift.io/cluster --patch '{"spec":{"additionalTrustedCA":
{"name":"registry-config"}}}' --type=merge
```

4. Samples Operator 設定オブジェクトの **samplesRegistry** フィールドを、ミラー設定で定義されたミラーの場所の **hostname** の部分を含むように更新します。

```
$ oc get configs.samples.operator.openshift.io -n openshift-cluster-samples-operator
```



注記

これは、イメージストリームのインポートプロセスでミラーまたは検索メカニズムが使用されないので必要になります。

5. Samples Operator 設定オブジェクトの **skippedImagestreams** フィールドにミラーリングされないイメージストリームを追加します。または、サンプルイメージストリームのいずれもサポートする必要がない場合は、Samples Operator を Samples Operator 設定オブジェクトの **Removed** に設定します。

注記

省略されないミラーリングされないイメージがあるか、または Samples Operator が **Removed** に変更されない場合、 Samples Operator はイメージストリームのインポートが失敗し始めてから 2 時間後に **Degraded** ステータスを報告します。

OpenShift namespace のテンプレートの多くはイメージストリームを参照します。そのため、**Removed** を使用してイメージストリームとテンプレートの両方を除去すると、イメージストリームのいずれかが欠落しているためにテンプレートが正常に機能しない場合にテンプレートの使用を試行する可能性がなくなります。

次のステップ

- VMware vSphere、ベアメタル、または Amazon Web Servicesなど、ネットワークが制限された環境でプロビジョニングする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。

2.3. 利用可能なクラスターのカスタマイズ

OpenShift Container Platform クラスターのデプロイ後は、大半のクラスター設定およびカスタマイズが終了していることになります。数多くの**設定リソース**が利用可能です。

注記

クラスターを IBM Z にインストールする場合は、すべての特長および機能が利用可能である訳ではありません。詳細は、『リリースノート』を参照してください。

イメージレジストリー、ネットワーク設定、イメージビルドの動作およびアイデンティティープロバイダーなどのクラスターの主要な機能を設定するために設定リソースを変更します。

これらのリソースを使用して制御する設定の現在の記述については、 **oc explain** コマンドを使用します (例: **oc explain builds --api-version=config.openshift.io/v1**)。

2.3.1. クラスター設定リソース

すべてのクラスター設定リソースはグローバルにスコープが設定され (namescope が使用されない)、**cluster**という名前が付けられます。

リソース名	説明
apiserver.config.openshift.io	証明書および認証局などの api-server 設定を提供します。
authentication.config.openshift.io	クラスターのアイデンティティープロバイダーおよび認証設定を制御します。
build.config.openshift.io	クラスターのすべてのビルドについてのデフォルトおよび有効にされている設定を制御します。
console.config.openshift.io	ログアウト動作を含む Web コンソールインターフェースの動作を設定します。

リソース名	説明
featuregate.config.openshift.io	FeatureGates を有効にして、テクノロジープレビュー機能を使用できるようにします。
image.config.openshift.io	特定の イメージレジストリー が処理される方法を設定します (allowed、disallowed、insecure、CA details)。
ingress.config.openshift.io	ルートのデフォルトドメインなどのルーティングに関連する設定の詳細。
oauth.config.openshift.io	内部 OAuth サーバーフローに関連するアイデンティティープロバイダーおよび他の動作を設定します。
project.config.openshift.io	プロジェクトテンプレートを含む プロジェクトの作成方法 を設定します。
proxy.config.openshift.io	外部ネットワークアクセスを必要とするコンポーネントで使用されるプロキシーを定義します。注: すべてのコンポーネントがこの値を使用する訳ではありません。
scheduler.config.openshift.io	ポリシーやデフォルトノードセレクターなどのスケジューラーの動作を設定します。

2.3.2. Operator 設定リソース

これらの設定リソースは、**cluster**という名前のクラススコープのインスタンスです。これは、特定の Operator によって所有される特定コンポーネントの動作を制御します。

リソース名	説明
console.operator.openshift.io	ブランドのカスタマイズなどのコンソールの外観の制御
config.imageregistry.operator.openshift.io	パブリックルーティング、プロキシー設定、リソース設定、レプリカ数およびストレージタイプなどの 内部イメージレジストリー設定を設定します。
config.samples.operator.openshift.io	Samples Operator を設定して、クラスターにインストールされるイメージストリームとテンプレートのサンプルを制御します。

2.3.3. 追加の設定リソース

これらの設定リソースは、特定コンポーネントの単一インスタンスを表します。場合によっては、リソースの複数のインスタンスを作成して、複数のインスタンスを要求できます。他の場合には、Operator は特定の namespace の特定のリソースインスタンス名のみを使用できます。追加のリソースインスタンスの作成方法や作成するタイミングについての詳細は、コンポーネント固有のドキュメントを参照してください。

リソース名	インスタンス名	Namespace	説明
alertmanager.monitoring.coreos.com	main	openshift-monitoring	alertmanager デプロイメントパラメーターを制御します。
ingresscontroller.operator.openshift.io	default	openshift-ingress-operator	ドメイン、レプリカ数、証明書、およびコントローラーの配置などの Ingress Operator 動作を設定します。

2.3.4. 情報リソース

これらのリソースを使用して、クラスターについての情報を取得します。これらのリソースは直接編集しないでください。

リソース名	インスタンス名	説明
clusterversion.config.openshift.io	version	OpenShift Container Platform 4.2 では、実稼働クラスターの ClusterVersion リソースをカスタマイズすることはできません。その代わりとして、 クラスターの更新プロセスを実行します。
dns.config.openshift.io	cluster	クラスターの DNS 設定を変更することはできません。DNS Operator ステータスを表示できます。
infrastructure.config.openshift.io	cluster	クラスターはそのクラウドプロバイダーとの対話を可能にする設定の詳細。
network.config.openshift.io	cluster	インストール後にクラスターのネットワークを変更することはできません。ネットワークをカスタマイズするには、インストール時にネットワークをカスタマイズするプロセスを実行します。

2.4. ファイアウォールの設定

ファイアウォールを使用する場合、OpenShift Container Platform が機能するために必要なサイトにアクセスできるように設定する必要があります。一部のサイトにはアクセスを常に付与し、クラスターをホストするために Red Hat Insights、Telemetry サービス、クラウドを使用したり、特定のビルドストラテジーをホストする場合に追加のアクセスを付与する必要があります。

2.4.1. OpenShift Container Platform のファイアウォールの設定

OpenShift Container Platform をインストールする前に、ファイアウォールを、OpenShift Container Platform が必要とするサイトへのアクセスを付与するように設定する必要があります。

手順

1. 以下のレジストリー URL をホワイトリスト化します。

URL	関数
registry.redhat.io	コアコンテナーイメージを指定します。
***.quay.io**	コアコンテナーイメージを指定します。
sso.redhat.com	**https://cloud.redhat.com/openshift** サイトでは、**sso.redhat.com** からの認証を使用します。

2. ビルドに必要な言語またはフレームワークのリソースを提供するサイトをホワイトリストに入れます。

3. Telemetry を無効にしていない場合は、以下の URL へのアクセスを許可して Red Hat Insights にアクセスできるようにする必要があります。

URL	関数
cert-api.access.redhat.com	Telemetry で必須
api.access.redhat.com	Telemetry で必須
infogw.api.openshift.com	Telemetry で必須
https://cloud.redhat.com/api/ingress	Telemetry および **insights-operator** で必須

4. Amazon Web Services (AWS) 、Microsoft Azure、または Google Cloud Platform (GCP) を使用してクラスターをホストする場合、クラウドプロバイダー API およびそのクラウドの DNS を提供する URL へのアクセスを付与する必要があります。

クラウド	URL	機能
AWS	***.amazonaws.com**	AWS サービスおよびリソースへのアクセスに必要です。AWS ドキュメントの「AWS Service Endpoints」を参照し、使用するリージョンを許可するエンドポイントを判別します。
GCP	***.googleapis.com**	GCP サービスおよびリソースへのアクセスに必要です。GCP ドキュメントの「Cloud Endpoints」を参照し、API を許可するエンドポイントを判別します。
	accounts.google.com	GCP アカウントへのアクセスに必要です。

クラウド	URL	機能
Azure	**management.azure.com**	Azure サービスおよびリソースへのアクセスに必要です。Azure ドキュメントで「Azure REST API Reference」を参照し、API を許可するエンドポイントを判別します。

5. 以下の URL をホワイトリスト化します。

URL	機能
mirror.openshift.com	ミラーリングされたインストールのコンテンツおよびイメージへのアクセスに必要。
***.cloudfront.net**	クラスターに必要な Quay.io イメージを提供するために Quay CDN で必要。
***.apps.<cluster_name>.<base_domain>**	Ingress ワイルドカードをインストール時に設定しない限り、デフォルトのクラスタールートへのアクセスに必要。
quay-registry.s3.amazonaws.com	AWS で Quay イメージコンテンツにアクセスするために必要。
api.openshift.com	クラスターに更新が利用可能かどうかを確認するために必要。
art-rhcos-ci.s3.amazonaws.com	Red Hat Enterprise Linux CoreOS (RHCOS) イメージをダウンロードするために必要。
api.openshift.com	クラスタートークンに必須
cloud.redhat.com/openshift	クラスタートークンに必須