Red Hat OpenShift Local 2.23

スタートガイド

Red Hat OpenShift Local の使用および開発に関するクイックスタートガイド

Last Updated: 2023-09-25

Red Hat OpenShift Local 2.23 スタートガイド

Red Hat OpenShift Local の使用および開発に関するクイックスタートガイド

Fabrice Flore-Thebault
ffloreth@redhat.com

法律上の通知

概要

本ガイドでは、Red Hat OpenShift Local を使用して速度を高める方法を説明します。本書の手順と例では、ホストワークステーション (Microsoft Windows、MacOS、または Red Hat Enterprise Linux) から Red Hat OpenShift Container Platform 4 を使用してコンテナー化されたアプリケーションを開発する最初の手順を説明します。

目次

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、Red Hat CTO である Chris Wright のメッセージ を参照してください。

第1章 RED HAT OPENSHIFT LOCAL の紹介

1.1. RED HAT OPENSHIFT LOCAL について

Red Hat OpenShift Local は、最小限の OpenShift Container Platform 4 クラスターと Podman コンテナーランタイムをローカルコンピューターに提供します。このようなランタイムは、開発およびテストを目的に、最小限の環境を提供します。Red Hat OpenShift Local は、主に開発者のデスクトップ上での実行を目的としています。ヘッドレスまたはマルチ開発者のセットアップなど、その他の OpenShift Container Platform のユースケースでは、完全な OpenShift インストーラー を使用します。

OpenShift Container Platform の詳細な紹介は、OpenShift Container Platform ドキュメント を参照してください。

Red Hat OpenShift Local には、目的のコンテナーランタイムを使用して Red Hat OpenShift Local インスタンスと対話するための **crc** コマンドラインインターフェイス (CLI) が含まれています。

1.2. 実稼働環境の OPENSHIFT CONTAINER PLATFORM インストールとの相違点

Red Hat OpenShift Local 用の OpenShift プリセットは、通常の OpenShift Container Platform インストールを提供しますが、以下の顕著な違いがあります。

- **OpenShift Container Platform クラスターは一時的なクラスターであり、実稼働環境での使用を目的としていません。**
- **Red Hat OpenShift Local では、新しい OpenShift Container Platform バージョンへのアップグレードパスはサポートされていません。** OpenShift Container Platform バージョンをアップグレードすると、再現が困難な問題が発生する可能性があります。
- コントロールプレーンとワーカーノードの両方として動作する単一のノードを使用します。
- デフォルトでは Cluster Monitoring Operator を無効にします。この無効な Operator が原因で、Web コンソールの該当部分が機能しなくなります。
- OpenShift Container Platform クラスターは、**インスタンス** と呼ばれる仮想マシンで実行します。これにより、特に外部ネットワーク関連でさらに相違点が生じる可能性があります。

Red Hat OpenShift Local が提供する OpenShift Container Platform クラスターには、以下のカスタマイズができないクラスター設定も含まれています。これらの設定は変更しないでください。

- ***.crc.testing** ドメイン。
- 内部クラスター通信に使用されるアドレスの範囲。
  - クラスターは **172** アドレス範囲を使用します。この設定が原因で、プロキシーが同じアドレス空間で実行されている場合などに問題が発生する可能性があります。

第2章 RED HAT OPENSHIFT LOCAL のインストール

2.1. 最小システム要件

Red Hat OpenShift Local の最小ハードウェアおよびオペレーティングシステムの要件は以下のとおりです。

2.1.1. ハードウェア要件

Red Hat OpenShift Local は、次のアーキテクチャーでサポートされています。

表2.1 プリセットとアーキテクチャーの互換性

プリセット	AMD64	Intel 64	M1
OpenShift Container Platform	はい	はい	はい
MicroShift	はい	はい	はい
Podman container runtime	はい	はい	はい

Red Hat OpenShift Local は、ネストされた仮想化をサポートしていません。

必要なコンテナーランタイムに応じて、Red Hat OpenShift Local には次のシステムリソースが必要です。

2.1.1.1. OpenShift Container Platform の場合

- 物理 CPU コア 4 個
- 空きメモリー 9 GB
- ストレージ領域の 35 GB

2.1.1.2. MicroShift の場合

- 物理 CPU コア 2 個
- 空きメモリー 4 GB
- ストレージ領域の 35 GB

注記

OpenShift Container Platform および MicroShift プリセットを Red Hat OpenShift Local インスタンスで実行するには、これらの最小限のリソースが必要です。一部のワークロードでは、より多くのリソースが必要になる場合があります。Red Hat OpenShift Local インスタンスにより多くのリソースを割り当てる方法は、インスタンスの設定 を参照してください。

2.1.1.3. Podman コンテナーランタイムの場合

- 物理 CPU コア 2 個
- 空きメモリー 2 GB
- ストレージ領域の 35 GB

2.1.2. オペレーティングシステム要件

Red Hat OpenShift Local には、サポートされるオペレーティングシステムの最小バージョンが必要です。

2.1.2.1. Microsoft Windows の要件

- Microsoft Windows では、Red Hat OpenShift Local に Windows 10 Fall Creators Update (バージョン 1709) 以降が必要です。Red Hat OpenShift Local は、Microsoft Windows の以前のバージョンでは動作しません。Microsoft Windows 10 Home Edition はサポートされません。

2.1.2.2. MacOS の要件

- MacOS を使用する場合、Red Hat OpenShift Local に MacOS 11 Big Sur 以降が必要です。Red Hat OpenShift Local は、以前のバージョンの MacOS では機能しません。

2.1.2.3. Linux の要件

- Linux では、Red Hat OpenShift Local は Red Hat Enterprise Linux/CentOS 8 および 9 マイナーリリースと安定した Fedora リリースのそれぞれ最新バージョン 2 つのみでサポートされます。
- Red Hat Enterprise Linux を使用する場合は、Red Hat OpenShift Local を実行するマシンが Red Hat カスタマーポータルに登録されている 必要があります。
- Ubuntu 18.04 LTS 以降および Debian 10 以降はサポートされておらず、ホストマシンの手動設定が必要になる場合があります。
- Linux ディストリビューションに必要なパッケージをインストールするには、必要なソフトウェアパッケージ を参照してください。

2.2. LINUX に必要なソフトウェアパッケージ

Red Hat OpenShift Local では、**libvirt** および **NetworkManager** パッケージが Linux 上で実行する必要があります。Linux ディストリビューションでこれらのパッケージをインストールするのに使用されるコマンドを確認するには、以下の表を参照してください。

表2.2 ディストリビューションによるパッケージのインストールコマンド

Linux ディストリビューション	インストールコマンド
Fedora/Red Hat Enterprise Linux/CentOS	**sudo dnf install NetworkManager**
Debian/Ubuntu	**sudo apt install qemu-kvm libvirt-daemon libvirt-daemon-system network-manager**

2.3. RED HAT OPENSHIFT LOCAL のインストール

Red Hat OpenShift Local は、Red Hat Enterprise Linux のポータブル実行可能ファイルとして利用できます。Microsoft Windows および MacOS では、ガイド付きインストーラーを使用して Red Hat OpenShift Local を使用できます。

前提条件

- ホストマシンが最小システム要件を満たしている。詳細は、最小システム要件 を参照してください。

手順

1. ご使用のプラットフォーム用の Red Hat OpenShift Local の最新リリースをダウンロードします。
2. Microsoft Windows で、アーカイブの内容をデプロイメントします。
3. MacOS または Microsoft Windows の場合は、ガイド付きインストーラーを実行し、手順に従います。

注記

Microsoft Windows では、Red Hat OpenShift Local をローカルの **C:** ドライブにインストールする必要があります。ネットワークドライブから Red Hat OpenShift Local を実行できません。

Red Hat Enterprise Linux の場合は、アーカイブが **~/Downloads** ディレクトリーにあるものとし、以下のステップを実行します。

a. アーカイブの内容をデプロイメントします。

```
$ cd ~/Downloads
$ tar xvf crc-linux-amd64.tar.xz
```

b. **~/bin** ディレクトリーが存在しない場合は、作成します。また、**crc** 実行可能ファイルをコピーします。

```
$ mkdir -p ~/bin
$ cp ~/Downloads/crc-linux-*-amd64/crc ~/bin
```

c. **~/bin** ディレクトリーを **$ PATH** に追加します。

```
$ export PATH=$PATH:$HOME/bin
$ echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
```

2.4. 使用状況データ収集について

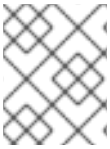
Red Hat OpenShift Local は、開発を支援するために、オプションの匿名使用データ収集を使用する前にプロンプトを表示します。個人を特定できる情報は収集されません。使用状況データ収集の同意は、いつでも承認または取り消すことができます。

関連情報

- 収集されるデータの詳細は、Red Hat Telemetry data collection notice を参照してください。
- 使用状況データ収集の同意を付与または取り消すには、使用状況データ収集の設定 を参照してください。

2.5. 使用状況データ収集の設定

使用状況データ収集の同意は、次の設定コマンドを使用していつでも承認または取り消すことができます。



注記

テレメトリー同意の設定を変更しても、実行中のインスタンスは変更されません。変更は、次に **crc start** コマンドを実行したときに有効になります。

手順

- テレメトリーを手動で有効にするには、以下のコマンドを実行します。

```
$ crc config set consent-telemetry yes
```

- テレメトリーを手動で無効にするには、以下のコマンドを実行します。

```
$ crc config set consent-telemetry no
```

関連情報

- 収集されるデータの詳細は、Red Hat Telemetry data collection notice を参照してください。

2.6. RED HAT OPENSHIFT LOCAL のアップグレード

Red Hat OpenShift Local 実行可能ファイルの新しいバージョンでは、以前のバージョンと互換性のない状態を防ぐために手動の設定が必要になります。

手順

1. Red Hat OpenShift Local の最新リリースをダウンロード します。
2. 既存の Red Hat OpenShift Local インスタンスを削除します。

```
$ crc delete
```

警告

crc delete コマンドを実行すると、Red Hat OpenShift Local ローカルインスタンスに保存されているデータが失われます。このコマンドを実行する前に、インスタンスに保存されている必要な情報を保存してください。

3. 以前の **crc** 実行可能ファイルを、最新リリースの実行ファイルに置き換えます。バージョンを確認して、新しい **crc** 実行可能ファイルが使用中であることを確認します。

```
$ crc version
```

4. 新しい Red Hat OpenShift Local ローカルリリースをセットアップします。

```
$ crc setup
```

5. 新しい Red Hat OpenShift Local インスタンスを開始します。

```
$ crc start
```

第3章 RED HAT OPENSHIFT LOCAL の使用

3.1. プリセットについて

Red Hat OpenShift Local プリセットは、マネージドコンテナーランタイムと、インスタンスがそのランタイムの実行に必要なシステムリソースの下限を表します。Red Hat OpenShift Local は、以下のプリセットを提供します。

openshift

事前設定済みの最小限の OpenShift Container Platform 4.13 クラスター

microshift

MicroShift

podman

Podman コンテナーランタイム

Microsoft Windows および MacOS では、Red Hat OpenShift Local ガイド付きインストーラーが目的のプリセットの入力を求めます。Linux では、OpenShift Container Platform プリセットがデフォルトで選択されています。この選択は、**crc setup** コマンドを実行する前に、**crc config** コマンドを使用して変更できます。選択したプリセットは、Microsoft Windows および MacOS のシステムトレイから、またはサポートされているすべてのオペレーティングシステムのコマンドラインから変更できます。一度にアクティブにできるプリセットは 1 つだけです。

関連情報

- 各プリセットの最小システム要件の詳細は、最小システム要件 を参照してください。
- 選択したプリセットの変更の詳細は、選択したプリセットの変更 を参照してください。

3.2. RED HAT OPENSHIFT LOCAL のセットアップ

crc setup コマンドは操作を実行し、Red Hat OpenShift Local インスタンスのホストマシンの環境を設定します。

crc setup コマンドは、**~/.crc** ディレクトリーが存在しない場合は作成します。

警告

新規バージョンを設定する場合は、新しい Red Hat OpenShift Local リリースを設定する前に、インスタンスに加えられた変更をすべて取得します。

前提条件

- Linux または MacOS の場合は、ユーザーアカウントに **sudo** コマンドを使用できるようになっている。Microsoft Windows の場合は、ユーザーアカウントが管理者権限で昇格できるようになっている。

注記

crc の実行ファイルは、**root** ユーザーまたは管理者として実行しないでください。**crc** 実行ファイルは常にユーザーアカウントで実行します。

手順

1. (オプション) Linux では、OpenShift Container Platform プリセットがデフォルトで選択されています。Podman コンテナーランタイムプリセットを選択するには、以下を実行します。

```
$ crc config set preset podman
```

2. Red Hat OpenShift Local 用にホストマシンをセットアップします。

```
$ crc setup
```

関連情報

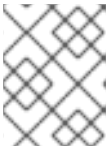
- 使用可能なコンテナーランタイムプリセットの詳細は、プリセットについて を参照してください。

3.3. インスタンスの開始

crc start コマンドは、Red Hat OpenShift Local インスタンスと設定済みコンテナーランタイムを開始します。

前提条件

- ネットワーク関連の問題を回避するには、VPN に接続されておらず、ネットワーク接続が信頼できる状態にしている。
- **crc setup** コマンドを使用してホストマシンを設定している。詳細は、Red Hat OpenShift Local のセットアップ を参照してください。
- Microsoft Windows の場合は、ユーザーアカウントが管理者権限で昇格できるようになっている。
- OpenShift プリセットの場合は、有効な OpenShift ユーザープルシークレットがある。Red Hat Hybrid Cloud Console の Red Hat OpenShift Local ページの Pull Secret セクションからプルシークレットをコピーするか、ダウンロードします。



注記

ユーザーのプルシークレットにアクセスするには、Red Hat アカウントが必要です。

手順

1. Red Hat OpenShift Local インスタンスを開始します。

```
$ crc start
```

2. OpenShift プリセットの場合は、プロンプトが表示されたら、ユーザーにプルシークレットを指定します。

注記

クラスターは、要求を提供する前に必要なコンテナーおよび Operator を起動するのに 4 分以上かかります。

関連情報

- インスタンスに割り当てられているデフォルトのリソースを変更するには、インスタンスの設定 を参照してください。
- **crc start** 時にエラーが表示される場合は、Red Hat OpenShift Local のトラブルシューティングセクション で潜在的な解決策を確認してください。

3.4. OPENSHIFT クラスターへのアクセス

OpenShift Container Platform Web コンソールまたは OpenShift CLI (**oc**) を使用して、Red Hat OpenShift Local インスタンスで実行されている OpenShift Container Platform クラスターにアクセスします。

3.4.1. OpenShift Web コンソールへのアクセス

Web ブラウザーを使用して、OpenShift Container Platform コンソールにアクセスします。

kubeadmin または **developer** ユーザーのいずれかを使用してクラスターにアクセスします。プロジェクトまたは OpenShift アプリケーションを作成するために、**developer** ユーザーを使用し、アプリケーションのデプロイメントに使用します。**kubeadmin** ユーザーは、新しいユーザーの作成やロールの設定などの管理作業にのみ使用してください。

前提条件

- Red Hat OpenShift Local は、OpenShift プリセットを使用するように設定されている。詳細は、選択したプリセットの変更 を参照してください。
- 実行中の Red Hat OpenShift Local インスタンス。詳細は、インスタンスの起動 を参照してください。

手順

1. デフォルトの Web ブラウザーで OpenShift Container Platform Web コンソールにアクセスするには、以下のコマンドを実行します。

```
$ crc console
```

2. **crc start** コマンドの出力でパスワードが出力された **developer** ユーザーとしてログインします。また、次のコマンドを実行すると、**developer** および **kubeadmin** ユーザーのパスワードを確認できます。

```
$ crc console --credentials
```

Red Hat OpenShift Local によって管理されている OpenShift Container Platform クラスターにアクセスできない場合は、Red Hat OpenShift Local のトラブルシューティング を参照してください。

関連情報

- OpenShift Container Platform ドキュメント では、プロジェクトとアプリケーションの作成について説明します。

3.4.2. OpenShift CLI による OpenShift クラスターへのアクセス

OpenShift CLI (**oc**) を使用して、Red Hat OpenShift Local によって管理される OpenShift Container Platform クラスターにアクセスします。

前提条件

- Red Hat OpenShift Local は、OpenShift プリセットを使用するように設定されている。詳細は、選択したプリセットの変更 を参照してください。
- 実行中の Red Hat OpenShift Local インスタンス。詳細は、インスタンスの起動 を参照してください。

手順

1. **crc oc-env** コマンドを実行して、キャッシュされた **oc** 実行可能ファイルを **$PATH** に追加します。

```
$ crc oc-env
```

2. 出力されたコマンドを実行します。
3. **developer** ユーザーとしてログインします。

```
$ oc login -u developer https://api.crc.testing:6443
```

注記

crc start コマンドは、**developer** ユーザーのパスワードを出力します。**crc console --credentials** コマンドを実行して表示することもできます。

4. **oc** を使用して OpenShift Container Platform クラスターと対話できるようになりました。たとえば、OpenShift Container Platform クラスター Operator が利用可能であることを確認するには、**kubeadmin** ユーザーとしてログインし、以下のコマンドを実行します。

```
$ oc config use-context crc-admin
$ oc whoami
kubeadmin
$ oc get co
```

注記

Red Hat OpenShift Local は、デフォルトで Cluster Monitoring Operator を無効にします。

Red Hat OpenShift Local によって管理されている OpenShift Container Platform クラスターにアクセスできない場合は、Red Hat OpenShift Local のトラブルシューティング を参照してください。

関連情報

- OpenShift Container Platform ドキュメント では、プロジェクトとアプリケーションの作成について説明します。

3.4.3. 内部 OpenShift レジストリーへのアクセス

Red Hat OpenShift Local インスタンスで実行している OpenShift Container Platform クラスターには、デフォルトで内部コンテナーイメージレジストリーが含まれています。この内部コンテナーイメージレジストリーは、ローカル開発コンテナーイメージの公開ターゲットとして使用できます。内部 OpenShift Container Platform レジストリーにアクセスするには、以下の手順に従います。

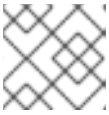
前提条件

- Red Hat OpenShift Local は、OpenShift プリセットを使用するように設定されている。詳細は、選択したプリセットの変更 を参照してください。
- 実行中の Red Hat OpenShift Local インスタンス。詳細は、インスタンスの起動 を参照してください。
- 動作する OpenShift CLI (**oc**) コマンド。詳細は、OpenShift CLI を使用した OpenShift クラスターのアクセス を参照してください。

手順

1. クラスターにログインしているユーザーを確認します。

```
$ oc whoami
```



注記

デモの目的で、現在のユーザーは **kubeadmin** であると想定されます。

2. トークンでそのユーザーとしてレジストリーにログインします。

```
$ oc registry login --insecure=true
```

3. 新しいプロジェクトを作成します。

```
$ oc new-project demo
```

4. コンテナーイメージの例をミラーリングします。

```
$ oc image mirror registry.access.redhat.com/ubi8/ubi:latest=default-route-openshift-image-registry.apps-crc.testing/demo/ubi8:latest --insecure=true --filter-by-os=linux/amd64
```

5. イメージストリームを取得し、プッシュされたイメージが表示されていることを確認します。

```
$ oc get is
```

6. イメージストリームでイメージルックアップを有効にします。

```
$ oc set image-lookup ubi8
```

この設定により、内部レジストリーに完全な URL を指定せずに、イメージストリームをイメージのソースに指定できます。

7. 最近プッシュされたイメージを使用して Pod を作成します。

```
$ oc run demo --image=ubi8 --command -- sleep 600s
```

3.5. ODO を使用したサンプルアプリケーションのデプロイ

odo を使用してコマンドラインから OpenShift プロジェクトおよびアプリケーションを作成できます。この手順では、Red Hat OpenShift Local インスタンスで実行している OpenShift Container Platform クラスターにサンプルアプリケーションをデプロイします。

前提条件

- **odo** がインストールされている。詳細は、**odo** ドキュメントの **odo** のインストール を参照してください。
- Red Hat OpenShift Local は、OpenShift プリセットを使用するように設定されている。詳細は、選択したプリセットの変更 を参照してください。
- Red Hat OpenShift Local ローカルインスタンスが実行している。詳細は、インスタンスの起動 を参照してください。

手順

1. Red Hat OpenShift Local が管理している実行中の OpenShift Container Platform クラスターに **developer** ユーザーとしてログインします。

```
$ odo login -u developer -p developer
```

2. アプリケーションのプロジェクトを作成します。

```
$ odo project create sample-app
```

3. コンポーネントのディレクトリーを作成します。

```
$ mkdir sample-app
$ cd sample-app
```

4. Node.js アプリケーションの例を複製します。

```
$ git clone https://github.com/openshift/nodejs-ex
$ cd nodejs-ex
```

5. **nodejs** コンポーネントをアプリケーションに追加します。

```
$ odo create nodejs
```

6. URL を作成し、ローカル設定ファイルにエントリーを追加します。

```
$ odo url create --port 8080
```

7. 変更をプッシュします。

```
$ odo push
```

これで、コンポーネントはアクセス可能な URL でクラスターにデプロイされます。

8. URL をリスト表示し、コンポーネントに必要な URL を確認します。

```
$ odo url list
```

9. 生成された URL を使用してデプロイされたアプリケーションを表示します。

関連情報

- **odo** の使用の詳細は、**odo** ドキュメント を参照してください。

3.6. インスタンスの停止

crc stop コマンドは、実行中の Red Hat OpenShift Local インスタンスおよびコンテナーランタイムを停止します。クラスターのシャットダウン中、停止プロセスには数分かかります。

手順

- Red Hat OpenShift Local インスタンスおよびコンテナーランタイムを停止します。

```
$ crc stop
```

3.7. インスタンスの削除

crc delete コマンドは、既存の Red Hat OpenShift Local インスタンスを削除します。

手順

- Red Hat OpenShift Local インスタンスを削除します。

```
$ crc delete
```

警告

crc delete コマンドを実行すると、Red Hat OpenShift Local ローカルインスタンスに保存されているデータが失われます。このコマンドを実行する前に、インスタンスに保存されている必要な情報を保存してください。

第4章 RED HAT OPENSHIFT LOCAL の設定

4.1. RED HAT OPENSHIFT LOCAL 設定について

crc config コマンドを使用して、**crc** 実行可能ファイルと Red Hat OpenShift Local インスタンスの両方を設定します。**crc config** コマンドには、設定で機能するサブコマンドが必要です。利用可能なサブコマンドは、**get**、**set**、**unset**、および **view** です。**get**、**set**、および **unset** サブコマンドは名前付きの設定可能なプロパティーで動作します。**crc config --help** コマンドを実行して、利用可能なプロパティーをリスト表示します。

crc config コマンドを使用して、crc **start** および **crc setup** コマンドの起動チェックの動作を設定することもできます。デフォルトでは、起動はエラーを確認し、条件が満たされない場合に実行を停止します。**skip-check** を **true** に設定して、チェックをスキップします。

4.2. RED HAT OPENSHIFT LOCAL 設定の表示

Red Hat OpenShift Local の実行ファイルは、設定可能なプロパティーと現在の Red Hat OpenShift Local 設定を表示するコマンドを提供します。

手順

- 利用可能な設定可能なプロパティーを表示するには、以下を実行します。

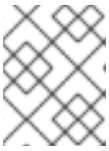
```
$ crc config --help
```

- 設定可能なプロパティーの値を表示するには、以下を実行します。

```
$ crc config get <property>
```

- 現在の設定を完了するには、以下を実行します。

```
$ crc config view
```



注記

crc config view コマンドは、設定がデフォルト値で設定されている場合に情報を返しません。

4.3. 選択したプリセットの変更

目的のプリセットを選択することで、Red Hat OpenShift Local インスタンスに使用されるコンテナーランタイムを変更できます。

Microsoft Windows および MacOS では、システムトレイまたはコマンドラインインターフェイスを使用して、選択したプリセットを変更できます。Linux では、コマンドラインインターフェイスを使用します。

重要

既存の Red Hat OpenShift Local インスタンスのプリセットを変更できません。プリセットの変更は、Red Hat OpenShift Local インスタンスが作成されたときにのみ適用されます。プリセットの変更を有効にするには、既存のインスタンスを削除して、新しいインスタンスを開始する必要があります。

手順

- コマンドラインから選択したプリセットを変更します。

```
$ crc config set preset <name>
```

有効なプリセット名は次のとおりです。

表4.1 プリセット名

名前	プリセット
openshift	OpenShift Container Platform
microshift	MicroShift
podman	Podman container runtime

関連情報

- 各プリセットの最小システム要件の詳細は、最小システム要件 を参照してください。

4.4. インスタンスの設定

cpus プロパティーおよび **memory** プロパティーを使用して、Red Hat OpenShift Local インスタンスで利用可能なデフォルトの仮想 CPU 数およびメモリー容量を設定します。

または、**--cpus** フラグおよび **--memory** フラグを使用して、それぞれ **crc start** コマンドに --cpus フラグおよび --memory フラグを使用して割り当てることができます。

重要

既存の Red Hat OpenShift Local インスタンスの設定を変更できません。設定の変更を有効にするには、実行中のインスタンスを停止して再起動する必要があります。

手順

- インスタンスで使用可能な vCPU の数を設定するには、以下を行います

```
$ crc config set cpus <number>
```

cpus プロパティーのデフォルト値は **4** です。割り当てる vCPU の数は、デフォルト以上である必要があります。

- 必要な数の vCPU でインスタンスを起動するには、以下を実行します。

```
$ crc start --cpus <number>
```

- インスタンスで使用可能なメモリーを設定するには、以下を実行します。

```
$ crc config set memory <number-in-mib>
```

注記

利用可能なメモリーの値は、メガバイト (MiB) で設定されます。メモリーの 1 つ (GiB) は 1024 MiB と等しくなります。

memory プロパティーのデフォルト値は **9216** です。割り当てるメモリー量は、デフォルト以上である必要があります。

- 必要な量のメモリーでインスタンスを起動するには、以下を実行します。

```
$ crc start --memory <number-in-mib>
```

第5章 ネットワーク

5.1. DNS 設定の詳細

5.1.1. 一般的な DNS 設定

Red Hat OpenShift Local によって管理される OpenShift Container Platform クラスターは、2 DNS ドメイン名 (**crc.testing** および **apps-crc.testing**) を使用します。**crc.testing** ドメインは、コア OpenShift Container Platform サービス用です。**apps-crc.testing** ドメインは、クラスターにデプロイされた OpenShift アプリケーションにアクセスするためのものです。

たとえば、OpenShift Container Platform API サーバーは、**console-openshift-console.apps-crc.testing** として OpenShift Container Platform コンソールにアクセスしている間に **api.crc.testing** として公開されます。これらの DNS ドメインは、Red Hat OpenShift Local インスタンス内で実行される **dnsmasq** DNS コンテナーによって提供されます。

crc setup コマンドは、これらのドメインを解決できるように、システムの DNS 設定を検出して調整します。**crc start** を起動する際に DNS が適切に設定されていることを確認するには、追加のチェックが行われます。

5.1.2. Linux 上の DNS

Linux では、ディストリビューションによっては、Red Hat OpenShift Local は以下の DNS 設定が必要となります。

5.1.2.1. NetworkManager + systemd-resolved

この設定は、Fedora 33 以降、Ubuntu Desktop editions でデフォルトで使用されます。

- Red Hat OpenShift Local コンテナーでは NetworkManager でネットワークを管理する必要があります。
- Red Hat OpenShift Local コンテナーは、**testing** ドメインの要求を **192.168.130.11** DNS サーバーに転送するように **systemd-resolved** を設定します。**192.168.130.11** は、Red Hat OpenShift Local インスタンスの IP です。
- **systemd-resolved** 設定は、**/etc/NetworkManager/dispatcher.d/99-crc.sh** の NetworkManager の dispatcher スクリプトで行います。

```
#!/bin/sh

export LC_ALL=C

systemd-resolve --interface crc --set-dns 192.168.130.11 --set-domain ~testing

exit 0
```

注記

systemd-resolved は、Red Hat Enterprise Linux および CentOS 8.3 でサポート対象外のテクノロジープレビューとしても提供されています。**systemd-resolved** を使用するように ホストを設定 したら、実行中のクラスターを停止して、**crc setup** を再実行します。

5.1.2.2. NetworkManager + dnsmasq

この設定は、Fedora 32 以前、Red Hat Enterprise Linux、CentOS ではデフォルトで使用されます。

- Red Hat OpenShift Local コンテナーでは NetworkManager でネットワークを管理する必要があります。
- NetworkManager は、**/etc/NetworkManager/conf.d/crc-nm-dnsmasq.conf** 設定ファイルで **dnsmasq** を使用します。
- この **dnsmasq** インスタンスの設定ファイルは **/etc/NetworkManager/dnsmasq.d/crc.conf** です。

```
server=/crc.testing/192.168.130.11
server=/apps-crc.testing/192.168.130.11
```

  - NetworkManager の **dnsmasq** インスタンスは、**crc.testing** ドメインおよび **apps-crc.testing** ドメインのリクエストを **192.168.130.11** DNS サーバーに転送します。

5.2. 予約された IP サブネット

Red Hat OpenShift Local によって管理される OpenShift Container Platform クラスターは、内部で使用するための IP サブネットを予約しますが、ホストネットワークと競合しないようにします。以下の IP サブネットが利用可能であることを確認します。

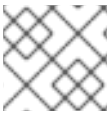
予約された IP サブネット

- **10.217.0.0/22**
- **10.217.4.0/23**
- **192.168.126.0/24**

また、ホストハイパーバイザーは、ホストオペレーティングシステムに応じて別の IP サブネットを予約します。MacOS および Microsoft Windows では追加のサブネットは予約されません。Linux 用の追加の予約サブネットは **192.168.130.0/24** です。

5.3. プロキシーの背後で RED HAT OPENSHIFT LOCAL を開始

環境変数や設定可能なプロパティーを使用して、定義されたプロキシーの背後で Red Hat OpenShift Local を起動できます。



注記

SOCKS プロキシーは OpenShift Container Platform ではサポートされません。

前提条件

- **crc oc-env** を使用していない場合は、クラスターと対話するときに、**no_proxy** 環境変数の一部として **.testing** ドメインをエクスポートする。埋め込み **oc** 実行可能ファイルには手動設定が必要ありません。埋め込み **oc** 実行可能ファイルの使用の詳細は、OpenShift CLI による OpenShift クラスターへのアクセス を参照してください。

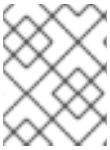
手順

1. **http_proxy** および **https_proxy** 環境変数を使用するか、以下のように **crc config set** コマンドを使用してプロキシーを定義します。

```
$ crc config set http-proxy http://proxy.example.com:<port>
$ crc config set https-proxy http://proxy.example.com:<port>
$ crc config set no-proxy <comma-separated-no-proxy-entries>
```

2. プロキシーがカスタム CA 証明書ファイルを使用する場合は、以下のように設定します。

```
$ crc config set proxy-ca-file <path-to-custom-ca-file>
```



注記

Red Hat OpenShift Local 設定に指定されたプロキシー関連の値は、環境変数を介して設定される値よりも優先されます。

5.4. リモートサーバーでの RED HAT OPENSHIFT LOCAL のセットアップ

Red Hat OpenShift Local が提供する OpenShift Container Platform クラスターを実行するようにリモートサーバーを設定します。

この手順では、Red Hat Enterprise Linux、Fedora、または CentOS サーバーを使用することを前提としています。この手順のすべてのコマンドをリモートサーバーで実行します。

警告

この手順は、ローカルネットワーク上でのみ実行してください。安全でないサーバーをインターネット上に公開することは、多くのセキュリティー上の問題があります。

前提条件

- Red Hat OpenShift Local が、リモートサーバーにインストールされ、設定されている。詳細は、Red Hat OpenShift Local のインストール および Red Hat OpenShift Local の設定 を参照してください。
- Red Hat OpenShift Local は、リモートサーバーで OpenShift プリセットを使用するように設定されている。詳細は、選択したプリセットの変更 を参照してください。
- ユーザーアカウントにリモートサーバーに対する **sudo** パーミッションがある。

手順

1. クラスターを起動します。

```
$ crc start
```

この手順の間、クラスターが稼働していることを確認してください。

2. **haproxy** パッケージおよびその他のユーティリティーをインストールします。

```
$ sudo dnf install haproxy /usr/sbin/semanage
```

3. クラスターとの通信を許可するようにファイアウォールを変更します。

```
$ sudo systemctl enable --now firewalld
$ sudo firewall-cmd --add-service=http --permanent
$ sudo firewall-cmd --add-service=https --permanent
$ sudo firewall-cmd --add-service=kube-apiserver --permanent
$ sudo firewall-cmd --reload
```

4. SELinux の場合は、HAProxy が TCP ポート 6443 でリッスンして、このポートで **kube-apiserver** を提供できるようにします。

```
$ sudo semanage port -a -t http_port_t -p tcp 6443
```

5. デフォルトの **haproxy** 設定のバックアップを作成します。

```
$ sudo cp /etc/haproxy/haproxy.cfg{,.bak}
```

6. クラスターで使用するように **haproxy** を設定します。

```
$ export CRC_IP=$(crc ip)
$ sudo tee /etc/haproxy/haproxy.cfg &>/dev/null <<EOF
global
    log /dev/log local0

defaults
    balance roundrobin
    log global
    maxconn 100
    mode tcp
    timeout connect 5s
    timeout client 500s
    timeout server 500s

listen apps
    bind 0.0.0.0:80
    server crcvm $CRC_IP:80 check

listen apps_ssl
    bind 0.0.0.0:443
    server crcvm $CRC_IP:443 check

listen api
    bind 0.0.0.0:6443
    server crcvm $CRC_IP:6443 check
EOF
```

7. **haproxy** サービスを起動します。

```
$ sudo systemctl start haproxy
```

5.5. リモートの RED HAT OPENSHIFT LOCAL ローカルインスタンスへの接続

dnsmasq を使用して、Red Hat OpenShift Local によって管理される OpenShift Container Platform クラスターを実行しているリモートサーバーにクライアントマシンを接続します。

この手順では、Red Hat Enterprise Linux、Fedora、または CentOS クライアントを使用することを前提としています。この手順の全コマンドをクライアント上で実行します。

重要

ローカルネットワーク上でのみ公開されるサーバーに接続します。

前提条件

- リモートサーバーが、クライアントが接続するように設定されている。詳細は、リモートサーバーでの Red Hat OpenShift Local のセットアップ を参照してください。
- サーバーの外部 IP アドレスを把握している。
- クライアントの **$PATH** に 最新の OpenShift CLI (**oc**) が指定されている。

手順

1. **dnsmasq** パッケージをインストールします。

```
$ sudo dnf install dnsmasq
```

2. NetworkManager での DNS 解決に対する **dnsmasq** の使用を有効にします。

```
$ sudo tee /etc/NetworkManager/conf.d/use-dnsmasq.conf &>/dev/null <<EOF
[main]
dns=dnsmasq
EOF
```

3. Red Hat OpenShift Local の DNS エントリーを **dnsmasq** 設定に追加します。

```
$ sudo tee /etc/NetworkManager/dnsmasq.d/external-crc.conf &>/dev/null <<EOF
address=/apps-crc.testing/SERVER_IP_ADDRESS
address=/api.crc.testing/SERVER_IP_ADDRESS
EOF
```

注記

/etc/NetworkManager/dnsmasq.d/crc.conf の既存のエントリーをコメントアウトします。これらのエントリーは、Red Hat OpenShift Local インスタンスを実行して作成し、リモートクラスターのエントリーと競合します。

4. NetworkManager サービスを再読み込みします。

```
$ sudo systemctl reload NetworkManager
```

5. **oc** を使用して **developer** ユーザーとしてリモートクラスターにログインします。

```
$ oc login -u developer -p developer https://api.crc.testing:6443
```

リモートの OpenShift Container Platform Web コンソールは https://console-openshift-console.apps-crc.testing から入手できます。

第6章 管理タスク

6.1. モニタリングの開始

Red Hat OpenShift Local は、Red Hat OpenShift Local が一般的なノートブックで実行できるように、デフォルトでクラスターモニタリングを無効にしています。モニタリングは、Red Hat Hybrid Cloud コンソール にクラスターを一覧表示するロールを担っています。以下の手順で、クラスターのモニタリングを有効にします。

前提条件

- Red Hat OpenShift Local インスタンスに追加のメモリーを割り当てる。コア機能には 14 GiB 以上のメモリー (値は **14336**) が推奨されます。ワークロードを増やすには、より多くのメモリーが必要です。詳細は、インスタンスの設定 を参照してください。

手順

1. **enable-cluster-monitoring** 設定可能プロパティーを **true** に設定します。

```
$ crc config set enable-cluster-monitoring true
```

2. インスタンスを起動します。

```
$ crc start
```

警告

クラスターモニタリングを無効にできません。モニタリングを削除するには、**enable-cluster-monitoring** 設定可能プロパティーを **false** に設定し、既存の Red Hat OpenShift Local インスタンスを削除します。

6.2. OPERATOR オーバーライドの有効化

Red Hat OpenShift Local が一般的なノートブックで実行されることを検証するには、リソース負荷の高いサービスの一部がデフォルトで無効になります。これらのサービスは、Operator オーバーライドリストから目的の Operator を手動で削除することで有効にできます。

前提条件

- 実行中の Red Hat OpenShift Local 仮想マシンと動作する **oc** コマンド。詳細は、**oc** を使用した OpenShift クラスターへのアクセス を参照してください。
- **oc** を介して **kubeadmin** ユーザーとしてログインする必要があります。

手順

1. アンマネージドの Operator をリストし、目的の Operator の数値インデックスをメモします。

- Linux または MacOS の場合:

```
$ oc get clusterversion version -ojsonpath='{range .spec.overrides[*]}{.name}{"\n"}{end}' |
nl -v 0
```

- PowerShell を使用する Microsoft Windows の場合:

```
PS> oc get clusterversion version -ojsonpath='{range .spec.overrides[*]}{.name}{"\n"}
{end}' | % {$nl++;"`t$($nl-1) `t $_"};$nl=0
```

2. 特定された数値インデックスを使用して、目的の Operator を開始します。

```
$ oc patch clusterversion/version --type='json' -p '[{"op":"remove",
"path":"/spec/overrides/<unmanaged-operator-index>"}]'
clusterversion.config.openshift.io/version patched
```

第7章 RED HAT OPENSHIFT LOCAL のトラブルシューティング

注記

Red Hat OpenShift Local の目的は、開発およびテストの目的で OpenShift Container Platform 環境を提供します。特定の OpenShift アプリケーションのインストール時に生じる問題は、Red Hat OpenShift Local のスコープ外にあります。該当するプロジェクトに、このような問題を報告します。

7.1. OPENSHIFT クラスターへのシェルアクセスの取得

トラブルシューティングまたはデバッグの目的でクラスターにアクセスするには、以下の手順に従います。

注記

OpenShift Container Platform クラスターへの直接アクセスは、通常の使用には必要ではなく、強く推奨されません。

前提条件

- クラスターへの OpenShift CLI (**oc**) アクセスを有効にし、**kubeadmin** ユーザーとしてログインする。詳細な手順は、OpenShift CLI による OpenShift クラスターへのアクセス を参照してください。

手順

1. **oc get nodes** コマンドを実行して、目的のノードを特定します。出力は以下のようになります。

```
$ oc get nodes
NAME                 STATUS   ROLES          AGE    VERSION
crc-shdl4-master-0   Ready    master,worker   7d7h   v1.14.6+7e13ab9a7
```

2. **oc debug nodes/<node>** を実行します。ここでの **<node>** は直前の手順で出力されるノードの名前です。

7.2. 期限切れの証明書のトラブルシューティング

リリースされた各 **crc** 実行可能ファイルの OpenShift Container Platform のシステムバンドルは、リリースから 1 年後に期限切れになります。この有効期限は、OpenShift Container Platform クラスターに埋め込まれた証明書によるものです。**crc start** コマンドは、必要に応じて証明書の更新プロセスをトリガーします。証明書の更新では、クラスターの起動時間に最大 5 分後に追加できます。

この追加の起動時間、または証明書の更新プロセスで失敗した場合は、以下の手順を使用します。

手順

自動的に更新できない期限切れの証明書エラーを解決するには、以下を実行します。

1. 最新の Red Hat OpenShift Local リリースをダウンロードし 、**$PATH** に **crc** 実行可能ファイルを設定します。

2. **crc delete** コマンドを使用して、証明書エラーでクラスターを削除します。

```
$ crc delete
```

警告

crc delete コマンドを実行すると、Red Hat OpenShift Local ローカルインスタンスに保存されているデータが失われます。このコマンドを実行する前に、インスタンスに保存されている必要な情報を保存してください。

3. 新しいリリースを設定します。

```
$ crc setup
```

4. 新しいインスタンスを開始します。

```
$ crc start
```

7.3. バンドルバージョンの不一致のトラブルシューティング

作成された Red Hat OpenShift Local インスタンスには、バンドル情報とインスタンスデータが含まれています。新規の Red Hat OpenShift Local リリースの設定時には、バンドル情報およびインスタンスデータは更新されません。この情報は、以前のインスタンスデータのカスタマイズにより更新されません。これにより、**crc start** コマンドの実行時にエラーが発生します。

```
$ crc start
...
FATA Bundle 'crc_hyperkit_4.2.8.crcbundle' was requested, but the existing VM is using
'crc_hyperkit_4.2.2.crcbundle'
```

手順

1. インスタンスを起動する前に **crc delete** コマンドを実行します。

```
$ crc delete
```

警告

crc delete コマンドを実行すると、Red Hat OpenShift Local ローカルインスタンスに保存されているデータが失われます。このコマンドを実行する前に、インスタンスに保存されている必要な情報を保存してください。

7.4. 不明な問題のトラブルシューティング

クリーンな状態で Red Hat OpenShift Local を再起動することで、ほとんどの問題を解決します。これには、インスタンスの停止、削除、**crc setup** コマンドによる変更の取り消し、変更の再適用、およびインスタンスの再起動が含まれます。

前提条件

- **crc setup** コマンドを使用してホストマシンを設定している。詳細は、Red Hat OpenShift Local のセットアップ を参照してください。
- **crc start** コマンドを使用して Red Hat OpenShift Local を起動している。詳細は、 インスタンスの起動 を参照してください。
- 最新の Red Hat OpenShift Local リリースを使用している。Red Hat OpenShift Local 1.2.0 よりも前のバージョンを使用すると、期限切れの x509 証明書に関連するエラーが発生する可能性があります。詳細は、期限切れの証明書のトラブルシューティング を参照してください。

手順

Red Hat OpenShift Local のトラブルシューティングを行うには、以下のステップを実行します。

1. Red Hat OpenShift Local インスタンスを停止します。

```
$ crc stop
```

2. Red Hat OpenShift Local インスタンスを削除します。

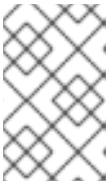
```
$ crc delete
```

警告

crc delete コマンドを実行すると、Red Hat OpenShift Local ローカルインスタンスに保存されているデータが失われます。このコマンドを実行する前に、インスタンスに保存されている必要な情報を保存してください。

3. **crc setup** コマンドで残りの変更をクリーンアップします。

```
$ crc cleanup
```



注記

crc cleanup コマンドは、既存の Red Hat OpenShift Local コンテナーインスタンスを削除し、**crc setup** コマンドで作成した DNS エントリーへの変更に戻ります。MacOS では、**crc cleanup** コマンドがシステムトレイも削除します。

4. 変更を適用するためにホストマシンを設定します。

```
$ crc setup
```

5. Red Hat OpenShift Local インスタンスを開始します。

```
$ crc start
```

注記

クラスターは、要求を提供する前に必要なコンテナーおよび Operator を起動するのに 4 分以上かかります。

この手順で問題が解決しない場合は、以下の手順を実行します。

1. 発生した問題については、未解決の問題を検索 します。
2. 発生した問題に対処する既存の問題がない場合は、問題を作成し、**~/.crc/crc.log** ファイルを作成された問題に割り当てます。**~/.crc/crc.log** ファイルには詳細なデバッグとトラブルシューティング情報があり、発生した問題を診断するのに役立ちます。