



Red Hat build of MicroShift 4.12

CLI ツール

MicroShift コマンドラインツールの使用方法

Red Hat build of MicroShift 4.12 CLI ツール

MicroShift コマンドラインツールの使用方法

法律上の通知

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

このドキュメントでは、MicroShift のストレージの使用に関する情報を提供します。oc、kubectl などのオプションの CLI ツールのインストールと設定について詳細に説明します。また、CLI コマンドのリファレンスとその使用例も説明します。

目次

第1章 RED HAT BUILD OF MICROSHIFT の CLI ツール	3
第2章 GETTING STARTED WITH THE OPENSIFT CLI	4
2.1. OPENSIFT CLI のインストール	4
第3章 OPENSIFT CLI の設定	7
3.1. タブ補完の有効化	7
第4章 OC ツールの使用	9
4.1. OPENSIFT CLI について	9
4.2. RED HAT BUILD OF MICROSHIFT での OPENSIFT CLI の使用	9
4.3. ヘルプの表示	10
4.4. RED HAT BUILD OF MICROSHIFT での OC コマンドエラー	11
第5章 OC および KUBECTL コマンドの使用	12
5.1. OC バイナリー	12
5.2. KUBECTL バイナリー	13
第6章 OPENSIFT CLI コマンドリファレンス	14
6.1. RED HAT BUILD OF MICROSHIFT の OC コマンドリスト	14

第1章 RED HAT BUILD OF MICROSHIFT の CLI ツール

ユーザーは、Red Hat build of MicroShift を使用する際に、アプリケーションとクラスターの両方をビルド、デプロイ、および管理します。

Red Hat build of MicroShift では、これらのタスクを簡素化するさまざまなコマンドラインインターフェイス (CLI) ツールを使用でき、ユーザーはターミナルからさまざまな管理および開発操作を実行できます。これらのツールでは、デプロイメントの管理だけでなく、システムの各コンポーネントを操作する簡単なコマンドを利用できます。

OpenShift Container Platform と Kubernetes に慣れている場合は、組み込みの **microshift** コマンドタイプと Linux CLI ツールに加えて、コマンドのサブセットを使用できる OpenShift CLI (**oc**) ツールを必要に応じて使用できます。

関連情報

- [MicroShift の **oc** ツールのインストール](#)。
- [OpenShift CLI \(oc\)](#): OpenShift Container Platform ドキュメントで提供されている **oc** の完全な説明。マルチノードのデプロイメント、プロジェクト、開発者ツールに重点を置いたコマンドは、Red Hat build of MicroShift ではサポートされていません。
- [Red Hat Enterprise Linux \(RHEL\)](#): 特定のユースケースに合わせて RHEL ドキュメントを参照してください。

第2章 GETTING STARTED WITH THE OPENSIFT CLI

OpenShift CLI (**oc**) ツールを使用するには、Red Hat build of MicroShift インストールとは別にダウンロードしてインストールする必要があります。

2.1. OPENSIFT CLI のインストール

バイナリーをダウンロードするか、Homebrew を使用して、OpenShift CLI (**oc**) をインストールできます。

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

OpenShift CLI (**oc**) をインストールして、コマンドラインインターフェイスから Red Hat build of MicroShift と対話できます。**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** をインストールした場合は、それを使用して Red Hat build of MicroShift 4.12 のすべてのコマンドを実行することができません。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 v4.12 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
```

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 v4.12 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
```

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 v4.12 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。
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
```

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>
```

2.1.2. Homebrew を使用した OpenShift CLI のインストール

macOS の場合は、[Homebrew](#) パッケージマネージャーを使用して OpenShift CLI (**oc**) をインストールできます。

前提条件

- Homebrew (**brew**) がインストールされている。

手順

- 以下のコマンドを実行して `openshift-cli` パッケージをインストールします。

```
$ brew install openshift-cli
```

2.1.3. RPM を使用した OpenShift CLI のインストール

Red Hat Enterprise Linux (RHEL) で、Red Hat アカウントに有効な Red Hat build of MicroShift サブスクリプションがある場合は、OpenShift CLI (**oc**) を RPM としてインストールできます。

前提条件

- `root` または `sudo` の権限がある。

手順

- Red Hat Subscription Manager に登録します。

```
# subscription-manager register
```

- 最新のサブスクリプションデータをプルします。

```
# subscription-manager refresh
```

- 利用可能なサブスクリプションを一覧表示します。

```
# subscription-manager list --available --matches '*OpenShift*'
```

- 前のコマンドの出力で、Red Hat build of MicroShift サブスクリプションのプール ID を見つけて、サブスクリプションを登録済みシステムにアタッチします。

```
# subscription-manager attach --pool=<pool_id>
```

- Red Hat build of MicroShift 4.12 で必要なりポジトリを有効にします。

```
# subscription-manager repos --enable="rhocp-4.12-for-rhel-8-x86_64-rpms"
```



注記

OpenShift CLI (**oc**) を Red Hat Enterprise Linux (RHEL) 9 の RPM としてインストールすることはできません。バイナリーをダウンロードして、`{op-system-base} 9` 用の OpenShift CLI をインストールする必要があります。

- openshift-clients** パッケージをインストールします。

```
# yum install openshift-clients
```

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>
```

第3章 OPENSIFT CLI の設定

oc を使用するために、必要に応じて設定します。

3.1. タブ補完の有効化

Bash または Zsh シェルのタブ補完を有効にできます。

3.1.1. Bash のタブ補完を有効にする

OpenShift CLI (**oc**) ツールをインストールした後に、タブ補完を有効にして **oc** コマンドの自動補完を実行するか、Tab キーを押す際にオプションの提案が表示されるようにできます。次の手順では、Bash シェルのタブ補完を有効にします。

前提条件

- OpenShift CLI (**oc**) がインストールされている。
- **bash-completion** パッケージがインストールされている。

手順

1. Bash 補完コードをファイルに保存します。

```
$ oc completion bash > oc_bash_completion
```

2. ファイルを **/etc/bash_completion.d/** にコピーします。

```
$ sudo cp oc_bash_completion /etc/bash_completion.d/
```

さらにファイルをローカルディレクトリーに保存した後に、これを **.bashrc** ファイルから取得できるようにすることができます。

タブ補完は、新規ターミナルを開くと有効にされます。

3.1.2. Zsh のタブ補完を有効にする

OpenShift CLI (**oc**) ツールをインストールした後に、タブ補完を有効にして **oc** コマンドの自動補完を実行するか、Tab キーを押す際にオプションの提案が表示されるようにできます。次の手順では、Zsh シェルのタブ補完を有効にします。

前提条件

- OpenShift CLI (**oc**) がインストールされている。

手順

- **oc** のタブ補完を **.zshrc** ファイルに追加するには、次のコマンドを実行します。

```
$ cat >> ~/.zshrc<<EOF
if [ $commands[oc] ]; then
  source <(oc completion zsh)
```

```
| compdef _oc oc  
| fi  
| EOF
```

タブ補完は、新規ターミナルを開くと有効にされます。

第4章 oc ツールの使用

OpenShift Container Platform および Kubernetes に慣れている場合は、オプションの OpenShift CLI (**oc**) ツールを使用できます。

4.1. OPENSIFT CLI について

OpenShift コマンドラインインターフェイス (CLI) である **oc** コマンドを使用すると、ターミナルから Red Hat build of MicroShift プロジェクトをデプロイおよび管理できます。OpenShift CLI は以下の状況に適しています。

- プロジェクトソースコードの直接使用
- Red Hat build of MicroShift 操作のスクリプト作成
- 帯域幅リソースによって制限されているプロジェクトの管理

4.2. RED HAT BUILD OF MICROSHIFT での OPENSIFT CLI の使用

oc CLI を使用して Red Hat build of MicroShift で一般的なタスクを完了する方法は、次のセクションを参照してください。

4.2.1. Pod の表示

現在のプロジェクトの Pod を表示するには、**oc get pods** コマンドを使用します。



注記

Pod 内で **oc** を実行し、namespace を指定しない場合は、Pod の namespace がデフォルトで使用されます。

```
$ oc get pods -o wide
```

出力例

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
NOMINATED NODE						
cakephp-ex-1-build	0/1	Completed	0	5m45s	10.131.0.10	ip-10-0-141-74.ec2.internal
<none>						
cakephp-ex-1-deploy	0/1	Completed	0	3m44s	10.129.2.9	ip-10-0-147-65.ec2.internal
<none>						
cakephp-ex-1-ktz97	1/1	Running	0	3m33s	10.128.2.11	ip-10-0-168-105.ec2.internal
<none>						

4.2.2. Pod ログの表示

特定の Pod のログを表示するには、**oc logs** コマンドを使用します。

```
$ oc logs cakephp-ex-1-deploy
```

出力例

```
--> Scaling cakephp-ex-1 to 1
--> Success
```

4.2.3. サポートされる API のリソースの一覧表示

サーバー上でサポートされる API リソースの一覧を表示するには、**oc api-resources** コマンドを使用します。

```
$ oc api-resources
```

出力例

NAME	SHORTNAMES	APIGROUP	NAMESPACED	KIND
bindings			true	Binding
componentstatuses	cs		false	ComponentStatus
configmaps	cm		true	ConfigMap
...				

4.3. ヘルプの表示

次の方法で、CLI コマンドおよび Red Hat build of MicroShift リソースに関するヘルプを取得できます。

- **oc help --flag** を使用して、特定の CLI コマンドに関する情報を取得します。

例: oc create コマンドについてのヘルプの表示

```
$ oc create --help
```

出力例

```
Create a resource by filename or stdin

JSON and YAML formats are accepted.

Usage:
  oc create -f FILENAME [flags]

...
```

- 特定リソースに関する説明およびフィールドを表示するには、**oc explain** コマンドを使用します。

例: Pod リソースのドキュメントの表示

```
$ oc explain pods
```

出力例

```
KIND: Pod
VERSION: v1
```

DESCRIPTION:

Pod is a collection of containers that can run on a host. This resource is created by clients and scheduled onto hosts.

FIELDS:

apiVersion <string>

APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: <https://git.k8s.io/community/contributors/devel/api-conventions.md#resources>

...

4.4. RED HAT BUILD OF MICROSHIFT での OC コマンドエラー

すべての OpenShift Container Platform CLI ツール (**oc**) コマンドが Red Hat build of MicroShift デプロイメントに関連しているわけではありません。**oc** を使用して、サポートされていない API に対してリクエストの呼び出しを行うと、通常 **oc** バイナリーは、リソースが見つからないとのエラーメッセージを生成します。

出力例

たとえば、次の **new-project** コマンドを実行すると、

```
$ oc new-project test
```

次のエラーメッセージが生成される可能性があります。

```
Error from server (NotFound): the server could not find the requested resource (get
projectrequests.project.openshift.io)
```

また、**get projects** コマンドを実行すると、次のような別のエラーが生成される場合があります。

```
$ oc get projects
error: the server doesn't have a resource type "projects"
```

第5章 OC および KUBECTL コマンドの使用

Kubernetes のコマンドラインインターフェイス (CLI) **kubectl** は、Kubernetes クラスターに対してコマンドを実行するのに使用されます。Red Hat build of MicroShift は認定済みの Kubernetes ディストリビューションであるため、Red Hat build of MicroShift に同梱されているサポート対象の **kubectl** バイナリーを使用するか、**oc** バイナリーを使用して拡張機能を取得できます。

5.1. OC バイナリー

oc バイナリーは **kubectl** バイナリーと同じ機能を提供しますが、次のような追加の Red Hat build of MicroShift 機能をネイティブにサポートするように拡張されています。

- **Route リソース**
Route リソースオブジェクトは、Red Hat build of MicroShift ディストリビューションに固有のものであり、標準の Kubernetes プリミティブに基づいて構築されています。
- **追加コマンド**
追加コマンドの **oc new-app** などは、既存のソースコードまたは事前にビルドされたイメージを使用して新規アプリケーションを起動することを容易にします。



重要

以前のバージョンの **oc** バイナリーをインストールした場合は、それを使用して Red Hat build of MicroShift 4.12 のすべてのコマンドを実行することができません。最新の機能が必要な場合は、Red Hat build of MicroShift サーバーのバージョンに対応する **oc** バйнаリーの最新バージョンをダウンロードしてインストールする必要があります。

セキュリティ以外の API の変更は、古い **oc** バイナリーの更新を可能にするために、2 つ以上のマイナーリリース (例: 4.1 から 4.2、そして 4.3 へ) が必要です。新機能を使用するには新規の **oc** バイナリーが必要になる場合があります。4.3 サーバーには、4.2 **oc** バイナリーが使用できない機能が追加されている場合や、4.3 **oc** バイナリーには 4.2 サーバーでサポートされていない追加機能が含まれる場合があります。

表5.1 互換性に関する表

	X.Y (oc クライアント)	X.Y+N footnote:versionpolicy[N は1以上の数値です] (oc クライアント)
X.Y (サーバー)	1	3
X.Y+N footnote:versionpolicy[N] (サーバー)	2	1

1 完全に互換性があります。

2 **oc** クライアントは、サーバー機能にアクセスできない場合があります。

3 **oc** クライアントは、アクセスされるサーバーと互換性のないオプションおよび機能を提供する可能性があります。

5.2. KUBECTL バイナリー

kubectl バイナリーは、標準の Kubernetes 環境を使用する新規 Red Hat build of MicroShift ユーザー、または **kubectl** CLI を優先的に使用するユーザーの既存ワークフローおよびスクリプトをサポートする手段として提供されます。**kubectl** の既存ユーザーは引き続きバイナリーを使用し、Red Hat build of MicroShift クラスターに変更を加えなくても Kubernetes のプリミティブと対話できます。

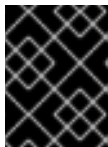
kubectl バイナリーは、**oc** バイナリーをダウンロードする場合にアーカイブに含まれます。

詳細は、[kubectl のドキュメント](#) を参照してください。

第6章 OPENSIFT CLI コマンドリファレンス

このリファレンスは、OpenShift CLI (**oc**) コマンドの説明とコマンド例を示しています。これらのコマンドを使用するには、**cluster-admin** または同等のパーミッションが必要です。

oc adm -h を実行して、すべての管理者コマンドを表示するか、**oc <command> --help** を実行して、特定のコマンドに関する追加情報を取得します。



重要

oc <command> --help を使用すると、**oc** コマンドの詳細が一覧表示されます。すべての **oc** コマンドが Red Hat build of MicroShift の使用に適用されるわけではありません。

6.1. RED HAT BUILD OF MICROSHIFT の OC コマンドリスト

以下に、Red Hat build of MicroShift ノードの管理、デプロイ、監視に使用できる **oc** コマンドの例をいくつか示します。

6.1.1. oc apply

設定をファイル名または標準入力 (stdin) 別のリソースに適用します。

使用例

```
# Apply the configuration in pod.json to a pod
oc apply -f ./pod.json
```

```
# Apply resources from a directory containing kustomization.yaml - e.g. dir/kustomization.yaml
oc apply -k dir/
```

```
# Apply the JSON passed into stdin to a pod
cat pod.json | oc apply -f -
```

```
# Apply the configuration from all files that end with '.json' - i.e. expand wildcard characters in file names
oc apply -f '*.json'
```

```
# Note: --prune is still in Alpha
# Apply the configuration in manifest.yaml that matches label app=nginx and delete all other resources that are not in the file and match label app=nginx
oc apply --prune -f manifest.yaml -l app=nginx
```

```
# Apply the configuration in manifest.yaml and delete all the other config maps that are not in the file
oc apply --prune -f manifest.yaml --all --prune-whitelist=core/v1/ConfigMap
```

6.1.2. oc delete

ファイル名、stdin、リソースおよび名前、またはリソースおよびラベルセクター別にリソースを削除します。

使用例

```
# Delete a pod using the type and name specified in pod.json
```

```
oc delete -f ./pod.json

# Delete resources from a directory containing kustomization.yaml - e.g. dir/kustomization.yaml
oc delete -k dir

# Delete resources from all files that end with '.json' - i.e. expand wildcard characters in file names
oc delete -f '*.json'

# Delete a pod based on the type and name in the JSON passed into stdin
cat pod.json | oc delete -f -

# Delete pods and services with same names "baz" and "foo"
oc delete pod,service baz foo

# Delete pods and services with label name=myLabel
oc delete pods,services -l name=myLabel

# Delete a pod with minimal delay
oc delete pod foo --now

# Force delete a pod on a dead node
oc delete pod foo --force

# Delete all pods
oc delete pods --all
```

6.1.3. oc get

1つ以上のリソースを表示します。

使用例

```
# List all pods in ps output format
oc get pods

# List all pods in ps output format with more information (such as node name)
oc get pods -o wide

# List a single replication controller with specified NAME in ps output format
oc get replicationcontroller web

# List deployments in JSON output format, in the "v1" version of the "apps" API group
oc get deployments.v1.apps -o json

# List a single pod in JSON output format
oc get -o json pod web-pod-13je7

# List a pod identified by type and name specified in "pod.yaml" in JSON output format
oc get -f pod.yaml -o json

# List resources from a directory with kustomization.yaml - e.g. dir/kustomization.
oc get -k dir/

# Return only the phase value of the specified pod
oc get -o template pod/web-pod-13je7 --template={{.status.phase}}
```

```
# List resource information in custom columns
oc get pod test-pod -o custom-
columns=CONTAINER:.spec.containers[0].name,IMAGE:.spec.containers[0].image

# List all replication controllers and services together in ps output format
oc get rc,services

# List one or more resources by their type and names
oc get rc/web service/frontend pods/web-pod-13je7

# List status subresource for a single pod.
oc get pod web-pod-13je7 --subresource status
```