



Red Hat OpenShift GitOps 1.13

GitOps CLI (argocd) リファレンス

GitOps CLI を設定し、デフォルトモードで Argo CD サーバーにログインする

Red Hat OpenShift GitOps 1.13 GitOps CLI (argocd) リファレンス

GitOps CLI を設定し、デフォルトモードで Argo CD サーバーにログインする

法律上の通知

Copyright © 2024 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、GitOps CLI を設定し、デフォルトモードで Argo CD サーバーにログインする方法を説明します。また、基本的な GitOps argocd コマンドも説明します。

目次

第1章 GITOPS CLI の設定	3
1.1. タブ補完の有効化	3
1.2. 関連情報	3
第2章 デフォルトモードで ARGO CD サーバーへのログイン	5
2.1. ARGO CD サーバーへのログイン	5
2.2. 関連情報	6
第3章 GITOPS ARGOCD CLI リファレンス	7
3.1. 基本的な構文	7
3.2. グローバルオプション	8
3.3. ユーティリティーコマンド	12
3.4. 関連情報	13

第1章 GITOPS CLI の設定



重要

Red Hat OpenShift GitOps **argocd** CLI ツールはテクノロジープレビュー機能のみです。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。

GitOps **argocd** CLI を設定してタブ補完を有効にできます。

1.1. タブ補完の有効化

GitOps **argocd** CLI をインストールした後、タブ補完を有効にして、Tab キーを押したときに **argocd** コマンドを自動的に補完したり、オプションを提案したりすることができます。



注記

タブ補完は Bash シェルにのみ存在します。

前提条件

- GitOps **argocd** CLI ツールがインストールされている。
- ローカルシステムに **bash-completion** がインストールされている。

手順

次の手順では、Bash シェルのタブ補完を有効にします。

1. Bash 補完コードをファイルに保存します。

```
$ argocd completion bash > argocd_bash_completion
```

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

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

または、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 **.bash_profile** 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。

1.2. 関連情報

- [GitOps CLI のインストール](#)
- [デフォルトモードで Argo CD サーバーへのログイン](#)

- [基本的な GitOps argocd コマンド](#)

第2章 デフォルトモードで ARGO CD サーバーへのログイン

重要

Red Hat OpenShift GitOps **argocd** CLI ツールはテクノロジープレビュー機能のみです。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。

GitOps **argocd** CLI と Argo CD 認証情報を使用してデフォルトモードで Argo CD サーバーにログインし、コマンドを実行できます。

2.1. ARGO CD サーバーへのログイン

GitOps **argocd** CLI をインストールして設定した後、Argo CD サーバーにログインして、デフォルトモードでコマンドを実行する必要があります。

前提条件

- GitOps **argocd** CLI ツールをインストールして設定している。

手順

ログインするには認証情報を使用し、セッション中はログインしたままにする必要があります。

重要

If the login session times out, you can use the ``relogin`` command to log in again. When done using the ``argocd`` commands, you can log out using the ``logout`` command.

1. Argo CD サーバーの **admin** アカウントのパスワードを取得します。

```
$ ADMIN_PASSWD=$(oc get secret openshift-gitops-cluster -n openshift-gitops -o jsonpath='{.data.admin\.password}' | base64 -d)
```

2. Argo CD サーバーの URL を取得します。

```
$ SERVER_URL=$(oc get routes openshift-gitops-server -n openshift-gitops -o jsonpath='{.status.ingress[0].host}')
```

3. **admin** アカウントのパスワードを使用して Argo CD サーバーにログインし、一重引用符で囲みます。



重要

パスワードを一重引用符で囲むと、\$などの特殊文字がシェルによって誤って解釈されなくなります。パスワードのリテラル値を囲むには常に一重引用符を使用してください。

```
$ argocd login --username admin --password ${ADMIN_PASSWD} ${SERVER_URL}
```

例

```
$ argocd login --username admin --password '<password>' openshift-gitops.openshift-  
gitops.apps-crc.testing
```

ログインが成功すると、セッションコンテキストが次のように表示されます。

出力例

```
'admin:login' logged in successfully  
Context '<server_url>' updated
```

2.2. 関連情報

- [GitOps CLI のインストール](#)
- [GitOps CLI の設定](#)
- [基本的な GitOps argocd コマンド](#)

第3章 GITOPS ARGOCD CLI リファレンス



重要

Red Hat OpenShift GitOps **argocd** CLI ツールはテクノロジープレビュー機能のみです。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。

このセクションでは、基本的な GitOps **argocd** CLI コマンドをリストします。

3.1. 基本的な構文

GitOps **argocd** CLI は、コマンドラインから Red Hat OpenShift GitOps と Argo CD リソースを設定および管理するためのツールです。

3.1.1. デフォルトモード

デフォルトモードでは、**argocd** CLI クライアントは API 要求を通じて Argo CD サーバーコンポーネントと通信します。コマンドを実行するには、Argo CD の認証情報を使用して Argo CD サーバーにログインし、セッション中はログインしたままにする必要があります。ログインセッションがタイムアウトした場合は、**relogin** コマンドを使用して再度ログインできます。**argocd** コマンドの使用が完了したら、**logout** コマンドを使用してログアウトできます。

コマンド構文

argocd [command or options] [arguments...]

3.1.2. コアモード

このモードでは、CLI は **kubeconfig** ファイルに設定された認証情報を通じて Kubernetes API サーバーと直接通信します。デフォルトの **kubeconfig** ファイルは、**\$HOME/.kube/config** にあります。**KUBECONFIG** 環境変数を使用してこのファイルをカスタマイズできます。**core** モードでコマンドを実行するには、**--core** 引数を使用でき、ユーザー認証のために Argo CD サーバーにログインする必要はありません。

<argocd-instance-name>-repo-server 形式で Repo サーバーコンポーネント名を指定するには、**--repo-server-name** コマンドラインオプションを使用するか、**ARGOCD_REPO_SERVER_NAME** 環境変数を設定します。

コマンド構文

KUBECONFIG=~/.kube/config argocd --core [command or options] [arguments...]

core モードで **argocd** コマンドを実行するには、次のいずれかのオプションを選択できます。



注記

複数の Argo CD インスタンスが使用されている場合は、現在のコンテキストのデフォルトの namespace を、対話する ArgoCD インスタンスの namespace に設定します。

- デフォルトのコンテキストを持つデフォルトの **kubeconfig** ファイル:
argocd --core [command or options] [arguments...]

例 1: アプリケーションのリストを表示する

```
$ argocd --core app list --repo-server-name openshift-gitops-repo-server
```

例 2: アプリケーションのリストを表示する

```
$ ARGOCD_REPO_SERVER_NAME=openshift-gitops-repo-server argocd --core app list
```

- カスタムコンテキストを持つデフォルトの **kubeconfig** ファイル:
argocd --core --kube-context [context] [command or options] [arguments...]

例 1: アプリケーションのリストを表示する

```
$ argocd --core --kube-context kubeadmin-local app list --repo-server-name openshift-gitops-repo-server
```

例 2: アプリケーションのリストを表示する

```
$ ARGOCD_REPO_SERVER_NAME=openshift-gitops-repo-server argocd --core --kube-context kubeadmin-local app list
```

- デフォルトのコンテキストを持つカスタム **kubeconfig** ファイル:
KUBECONFIG=~/.kube/custom_config argocd --core [command or options] [arguments...]

例: アプリケーションのリストを表示する

```
$ KUBECONFIG=~/.kube/custom_config argocd --core app list --repo-server-name openshift-gitops-repo-server
```

- カスタムコンテキストを持つカスタム **kubeconfig** ファイル:
KUBECONFIG=~/.kube/custom_config argocd --core --kube-context [context] [command or options] [arguments...]

例: アプリケーションのリストを表示する

```
$ KUBECONFIG=~/.kube/custom_config argocd --kube-context kubeadmin-local --core app list --repo-server-name openshift-gitops-repo-server
```

3.2. グローバルオプション

グローバルオプションは、**argocd** のすべてのサブコマンドに適用できます。

表3.1 グローバルオプション

オプション	引数のタイプ	説明
--auth-token	string	認証トークン。
--client-crt	string	クライアント証明書ファイル。
--client-crt-key	string	クライアント証明書キーファイル。
--config	string	Argo CD 設定ファイルへのパス。 デフォルトは /home/user/.config/argocd/config パス。
--controller-name	string	デフォルトのラベルとして argocd-application-controller を持つ Argo CD Application Controller コンポーネントの名前。 このコンポーネントの名前のラベルがデフォルトと異なる場合 (たとえば、Helm チャートを通じてインストールする場合)、 --controller-name オプションまたは ARGOCD_APPLICATION_CONTROLLER_NAME 環境変数のいずれかを設定して、コンポーネントの名前を指定します。
--core	該当なし	true に設定すると、CLI は Argo CD API サーバーを使用する代わりに、Kubernetes API サーバーと直接通信します。
--grpc-web	該当なし	Argo CD サーバーの gRPC-Web プロトコルを有効にします。これは、たとえば、サーバーが HTTP/2 プロトコルをサポートしていないプロキシの背後にある場合に便利です。
--grpc-web-root-path	string	Argo CD サーバーの gRPC-Web プロトコルを有効にします。これは、たとえば、サーバーが HTTP/2 プロトコルをサポートしていないプロキシの背後にある場合に便利です。Web ルートを設定します。

オプション	引数のタイプ	説明
-H, --header	string	GitOps argocd CLI によって行われたすべてのリクエストに追加のヘッダーを設定します。このオプションを複数回設定することで、複数のヘッダーを追加できます。このオプションは、コンマ区切りのヘッダーもサポートします。
-h, --help	該当なし	GitOps argocd CLI のヘルプ。
--http-retry-max	integer	Argo CD サーバーへの HTTP 接続を確立するための最大再試行回数を設定します。
--insecure	該当なし	サーバー証明書とドメイン検証をスキップします。
--kube-context	string	指定された kube コンテキストにコマンドを送信します。
--logformat	string	ログ形式をテキストまたは JSON に設定します。デフォルトは text です。
--loglevel	string	ログレベルを設定します。デフォルトは info です。 debug 、 info 、 warn 、 error レベルが利用可能です。ログレベルは、詳細度の降順でリストされます。
--plaintext	該当なし	Transport Layer Security (TLS) プロトコルを無効にします。
--port-forward	該当なし	ポート転送を使用してランダムな Argo CD サーバーポートに接続します。
--port-forward-namespace	string	ポート転送に使用する namespace の名前。

オプション	引数のタイプ	説明
--redis-haproxy-name	string	<p>argocd-redis-ha-haproxy をデフォルトのラベルとする Redis HA Proxy デプロイメントの名前。</p> <p>このデプロイメントの名前のラベルがデフォルトと異なる場合 (たとえば、Helm チャートを通じて HAProxy をインストールする場合) は、--redis-haproxy-name オプションまたは ARGOCD_REDIS_HAPROXY_NAME 環境変数のいずれかを設定して、デプロイメントの名前を指定します。</p>
--redis-name	string	<p>デフォルトのラベルとして argocd-redis を持つ Redis デプロイメントの名前。</p> <p>このデプロイメントの名前のラベルがデフォルトと異なる場合 (たとえば、Helm チャートを通じてインストールする場合) は、--redis-name オプションまたは ARGOCD_REDIS_NAME 環境変数のいずれかを設定して、デプロイメントの名前を指定します。</p>
--repo-server-name	string	<p>デフォルトのラベルとして argocd-repo-server を持つ Argo CD Repo サーバーの名前。</p> <p>この Repo サーバーの名前のラベルがデフォルトと異なる場合 (たとえば、Helm チャートを通じてインストールする場合) は、--repo-server-name オプションまたは ARGOCD_REPO_SERVER_NAME 環境変数のいずれかを設定して、デプロイメントの名前を指定します。</p>
--server	string	Argo CD サーバーのアドレス。
--server-crt	string	サーバーの証明書ファイル。

オプション	引数のタイプ	説明
--server-name	string	<p>デフォルトのラベルとして argocd-server を持つ Argo CD API サーバーの名前。</p> <p>この Argo CD API サーバーの名前のラベルがデフォルトと異なる場合 (たとえば、Helm チャートを通じてインストールする場合は、-server-name オプションまたは ARGOCD_SERVER_NAME 環境変数のいずれかを設定して、デプロイメントの名前を指定します。</p>

3.3. ユーティリティーコマンド

3.3.1. argocd

GitOps **argocd** CLI の親コマンド。

例: すべてのオプションの表示

```
$ argocd
```

3.3.2. version

CLI のバージョン情報を出力します。

コマンド構文

argocd version [flags]

例: クライアントとサーバーの完全なバージョンを標準出力に出力する

```
$ argocd version
```

例: クライアントのフルバージョンのみを出力し、サーバーへの接続は行われない

```
$ argocd version --client
```

例: サーバーのフルバージョンのみを出力する

```
$ argocd version --server <server_url>
```

例: クライアントとサーバーの完全なバージョンを JSON 形式で出力する

```
$ argocd version -o json
```


例: クライアントとサーバーのコアバージョン文字列のみを YAML 形式で出力する

```
$ argocd version --short -o yaml
```

3.3.3. help

アプリケーション内の任意のコマンドに関するヘルプメッセージを出力します。

コマンド構文

argocd help [command] [flags]

例: 利用可能なすべてのコマンドのヘルプテキストを取得する

```
$ argocd help
```

例: **admin** サブコマンドのヘルプテキストを取得する

```
$ argocd help admin
```

3.3.4. completion

bash または **zsh** シェルの補完コードを標準出力に書き込みます。

コマンド構文

argocd completion SHELL [flags]

bash の場合は、Bash 補完がインストールされ、有効になっていることを確認してください。または、ファイルに書き込んで、**.bash_profile** でソース化します。

例: 現在のシェルでのアクセス補完

```
# source <(argocd completion bash)
```

zsh の場合は、Bash 補完がインストールされ、有効になっていることを確認してください。

例: **~/zshrc** ファイルに追加し、現在のシェルで補完にアクセスする

```
source <(argocd completion zsh)
compdef _argocd argocd
```

3.4. 関連情報

- [GitOps CLI のインストール](#)
- [GitOps CLI の設定](#)
- [デフォルトモードで Argo CD サーバーへのログイン](#)
- [OpenShift GitOps CLI ユーザーガイド](#)
- [argocd コマンドリファレンス](#)

