



Red Hat OpenShift GitOps 1.12

GitOps CLI (argocd) reference

Configuring the GitOps CLI and logging in to the Argo CD server in the default mode.

Red Hat OpenShift GitOps 1.12 GitOps CLI (argocd) reference

Configuring the GitOps CLI and logging in to the Argo CD server in the default mode.

Legal Notice

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.

Abstract

This document provides information about how to configure the GitOps CLI and log in to the Argo CD server in the default mode. It also discusses about the basic GitOps argocd commands.

Table of Contents

CHAPTER 1. CONFIGURING THE GITOPS CLI	3
1.1. ENABLING TAB COMPLETION	3
1.2. ADDITIONAL RESOURCES	3
CHAPTER 2. LOGGING IN TO THE ARGO CD SERVER IN THE DEFAULT MODE	5
2.1. LOGGING IN TO THE ARGO CD SERVER	5
2.2. ADDITIONAL RESOURCES	6
CHAPTER 3. GITOPS ARGOCD CLI REFERENCE	7
3.1. BASIC SYNTAX	7
3.1.1. Default mode	7
3.1.2. Core mode	7
3.2. GLOBAL OPTIONS	8
3.3. UTILITY COMMANDS	12
3.3.1. argocd	12
3.3.2. version	12
3.3.3. help	13
3.3.4. completion	13
3.4. ADDITIONAL RESOURCES	13

CHAPTER 1. CONFIGURING THE GITOPS CLI



IMPORTANT

The Red Hat OpenShift GitOps **argocd** CLI tool is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#).

You can configure the GitOps **argocd** CLI to enable tab completion.

1.1. ENABLING TAB COMPLETION

After you install the GitOps **argocd** CLI, you can enable tab completion to automatically complete **argocd** commands or suggest options when you press Tab.



NOTE

Tab completions only exist for the Bash shell.

Prerequisites

- You must have the GitOps **argocd** CLI tool installed.
- You must have **bash-completion** installed on your local system.

Procedure

The following procedure enables tab completion for the Bash shell.

1. Save the Bash completion code to a file:

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

2. Copy the file to **/etc/bash_completion.d/**:

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

Alternatively, you can save the file to a local directory and source it from your **.bash_profile** file instead.

Tab completion is enabled when you open a new terminal.

1.2. ADDITIONAL RESOURCES

- [Installing the GitOps CLI](#)
- [Logging in to the Argo CD server in the default mode](#)

- [Basic GitOps argocd commands](#)

CHAPTER 2. LOGGING IN TO THE ARGO CD SERVER IN THE DEFAULT MODE



IMPORTANT

The Red Hat OpenShift GitOps **argocd** CLI tool is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#).

You can log in to the Argo CD server in the default mode using the GitOps **argocd** CLI and your Argo CD credentials to execute commands.

2.1. LOGGING IN TO THE ARGO CD SERVER

After you install and configure the GitOps **argocd** CLI, you must log in to the Argo CD server to execute commands in the default mode.

Prerequisites

- You must have the GitOps **argocd** CLI tool installed and configured.

Procedure

You must use your credentials to log in and remain logged in throughout the session.



IMPORTANT

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. Get the **admin** account password for the Argo CD server:

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

2. Get the Argo CD server URL:

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

3. Log in to the Argo CD server by using the **admin** account password and enclosing it in single quotes:



IMPORTANT

Enclosing the password in single quotes ensures that special characters, such as `$`, are not misinterpreted by the shell. Always use single quotes to enclose the literal value of the password.

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

Example

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

After a successful login, the session context will be displayed as follows:

Example output

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

2.2. ADDITIONAL RESOURCES

- [Installing the GitOps CLI](#)
- [Configuring the GitOps CLI](#)
- [Basic GitOps argocd commands](#)

CHAPTER 3. GITOPS ARGOCD CLI REFERENCE



IMPORTANT

The Red Hat OpenShift GitOps **argocd** CLI tool is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#).

This section lists the basic GitOps **argocd** CLI commands.

3.1. BASIC SYNTAX

The GitOps **argocd** CLI is a tool for configuring and managing Red Hat OpenShift GitOps and Argo CD resources from the command line.

3.1.1. Default mode

In the default mode, the **argocd** CLI client communicates with the Argo CD server component through API requests. To execute commands, you must log in to the Argo CD server using your Argo CD credentials and remain logged in throughout the session. 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.

Command syntax

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

3.1.2. Core mode

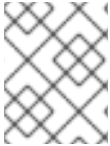
In this mode, the CLI communicates directly with the Kubernetes API server through the credentials set in the **kubeconfig** file. The default **kubeconfig** file is available at the **\$HOME/.kube/config** location. You can customize this file using the **KUBECONFIG** environment variable. To execute commands in the **core** mode, you can use the **--core** argument and don't need to log in to the Argo CD server for user authentication.

To specify the Repo server component name in the **<argocd-instance-name>-repo-server** format, you can either use the **--repo-server-name** command line option or set the **ARGOCD_REPO_SERVER_NAME** environment variable.

Command syntax

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

You can choose one of the following options to execute **argocd** commands in the **core** mode:



NOTE

If multiple Argo CD instances are in use, set the default namespace of the current context to the namespace of the ArgoCD instance you want to interact with.

- Default **kubeconfig** file with the default context:
argocd --core [command or options] [arguments...]

Example 1: Display a list of applications

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

Example 2: Display a list of applications

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

- Default **kubeconfig** file with a custom context:
argocd --core --kube-context [context] [command or options] [arguments...]

Example 1: Display a list of applications

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

Example 2: Display a list of applications

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

- A custom **kubeconfig** file with the default context:
KUBECONFIG=~/.kube/custom_config argocd --core [command or options] [arguments...]

Example: Display a list of applications

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

- A custom **kubeconfig** file with a custom context:
KUBECONFIG=~/.kube/custom_config argocd --core --kube-context [context] [command or options] [arguments...]

Example: Display a list of applications

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

3.2. GLOBAL OPTIONS

Global options are applicable to all the subcommands of the **argocd**.

Table 3.1. Global options

Option	Argument type	Description
--auth-token	string	Authentication token.
--client-crt	string	Client certificate file.
--client-crt-key	string	Client certificate key file.
--config	string	Path to Argo CD configuration file. Defaults to the /home/user/.config/argocd/config path.
--controller-name	string	Name of the Argo CD Application Controller component with argocd-application-controller as the default label. If the label of this component's name differs from the default, for example, when you are installing it through the Helm chart, set either the --controller-name option or the ARGOCD_APPLICATION_CONTROLLER_NAME environment variable to specify the name of the component.
--core	N/A	If set to true , the CLI communicates directly with the Kubernetes API server instead of using the Argo CD API server.
--grpc-web	N/A	Enable the gRPC-Web protocol for the Argo CD server. This is useful if, for example, the server is behind a proxy that does not support the HTTP/2 protocol.
--grpc-web-root-path	string	Enable the gRPC-Web protocol for the Argo CD server. This is useful if, for example, the server is behind a proxy that does not support the HTTP/2 protocol. Set web root.

Option	Argument type	Description
-H, --header	string	Configure an additional header to all requests made by the GitOps argocd CLI. You can add multiple headers by setting this option multiple times. This option also supports comma-separated headers.
-h, --help	N/A	Help for the GitOps argocd CLI.
--http-retry-max	integer	Set the maximum number of retries to establish an HTTP connection to the Argo CD server.
--insecure	N/A	Skip server certificate and domain verification.
--kube-context	string	Direct the command to the given kube context.
--logformat	string	Set the logging format to either text or JSON. Defaults to text .
--loglevel	string	Set the logging level. Defaults to info . The debug , info , warn , and error levels are available. The logging levels are listed in decreasing order of verbosity.
--plaintext	N/A	Disable Transport Layer Security (TLS) protocols.
--port-forward	N/A	Connect to a random Argo CD server port using port forwarding.
--port-forward-namespace	string	Namespace name to be used for port forwarding.

Option	Argument type	Description
--redis-haproxy-name	string	<p>Name of the Redis HA Proxy deployment with argocd-redis-ha-haproxy as the default label.</p> <p>If the label of this deployment's name differs from the default, for example, when you are installing HAProxy through the Helm chart, set either the --redis-haproxy-name option or the ARGOCD_REDIS_HAPROXY_NAME environment variable to specify the name of the deployment.</p>
--redis-name	string	<p>Name of the Redis deployment with argocd-redis as the default label.</p> <p>If the label of this deployment's name differs from the default, for example, when you are installing it through the Helm chart, set either the --redis-name option or the ARGOCD_REDIS_NAME environment variable to specify the name of the deployment.</p>
--repo-server-name	string	<p>Name of the Argo CD Repo server with argocd-repo-server as the default label.</p> <p>If the label of this Repo server's name differs from the default, for example, when you are installing it through the Helm chart, set either the --repo-server-name option or the ARGOCD_REPO_SERVER_NAME environment variable to specify the name of the deployment.</p>
--server	string	Argo CD server address.
--server-crt	string	Server certificate file.

Option	Argument type	Description
--server-name	string	<p>Name of the Argo CD API server with argocd-server as the default label.</p> <p>If the label of this Argo CD API server's name differs from the default, for example, when you are installing it through the Helm chart, set either the --server-name option or the ARGOCD_SERVER_NAME environment variable to specify the name of the deployment.</p>

3.3. UTILITY COMMANDS

3.3.1. argocd

Parent command for GitOps **argocd** CLI.

Example: Display all options

```
$ argocd
```

3.3.2. version

Print version information of the CLI.

Command syntax

argocd version [flags]

Example: Print the full version of client and server to stdout

```
$ argocd version
```

Example: Print only full version of the client, no connection to server will be made

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

Example: Print only full version of the server

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

Example: Print the full version of client and server in JSON format

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


Example: Print only client and server core version strings in YAML format

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

3.3.3. help

Print the help message about any command in the application.

Command syntax

argocd help [command] [flags]

Example: Get the help text for all the available commands

```
$ argocd help
```

Example: Get the help text for admin subcommand

```
$ argocd help admin
```

3.3.4. completion

Write **bash** or **zsh** shell completion code to standard output.

Command syntax

argocd completion SHELL [flags]

For **bash**, ensure you have Bash completions installed and enabled. Alternatively, write it to a file and source it in your **.bash_profile**.

Example: Access completion in your current shell

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

For **zsh**, ensure you have Bash completions installed and enabled.

Example: Add to your ~/.zshrc file and access completion in your current shell

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

3.4. ADDITIONAL RESOURCES

- [Installing the GitOps CLI](#)
- [Configuring the GitOps CLI](#)
- [Logging in to the Argo CD server in the default mode](#)
- [OpenShift GitOps CLI User Guide](#)
- [argocd command reference](#)

