



Red Hat OpenShift GitOps 1.12

GitOps CLI (argocd)参考

配置 GitOps CLI 并以默认模式登录到 Argo CD 服务器。

Red Hat OpenShift GitOps 1.12 GitOps CLI (argocd)参考

配置 GitOps CLI 并以默认模式登录到 Argo CD 服务器。

法律通告

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.

摘要

本文档提供有关如何配置 GitOps CLI 并以默认模式登录到 Argo CD 服务器的信息。它还讨论了基本的 GitOps `argocd` 命令。

目录

第 1 章 配置 GITOPS CLI	3
1.1. 启用 TAB 自动完成功能	3
1.2. 其他资源	3
第 2 章 以默认模式登录到 ARGO CD 服务器	4
2.1. 登录到 ARGO CD 服务器	4
2.2. 其他资源	5
第 3 章 GITOPS ARGOCD CLI 参考	6
3.1. 基本语法	6
3.2. 全局选项	7
3.3. 工具命令	10
3.4. 其他资源	11

第 1 章 配置 GITOPS CLI



重要

Red Hat OpenShift GitOps **argocd** CLI 工具只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议 (SLA) 支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅[技术预览功能支持范围](#)。

您可以配置 GitOps **argocd** CLI 来启用 tab 自动完成功能。

1.1. 启用 TAB 自动完成功能

安装 GitOps **argocd** CLI 后，您可以启用 tab 自动完成功能，以便在按 Tab 键时自动完成 **argocd** 命令或建议选项。



注意

Tab 补全仅针对 Bash shell 存在。

先决条件

- 已安装 GitOps **argocd** CLI 工具。
- 需要在本地系统中安装了 **bash-completion**。

流程

以下流程为 Bash shell 启用 tab 自动完成功能。

1. 将 Bash 完成代码保存到一个文件中：

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

2. 将文件复制到 **/etc/bash_completion.d/**：

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

或者，您可以将文件保存到一个本地目录，并从 **.bash_profile** 文件中 source 这个文件。

开新终端时 tab 自动完成功能将被启用。

1.2. 其他资源

- [安装 GitOps CLI](#)
- [以默认模式登录到 Argo CD 服务器](#)
- [基本 GitOps argocd 命令](#)

第 2 章 以默认模式登录到 ARGO CD 服务器



重要

Red Hat OpenShift GitOps **argocd** CLI 工具只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议 (SLA) 支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅[技术预览功能支持范围](#)。

您可以使用 GitOps **argocd** CLI 和 Argo CD 凭证以默认模式登录到 Argo CD 服务器。

2.1. 登录到 ARGO CD 服务器

安装和配置 GitOps **argocd** CLI 后，您必须登录到 Argo CD 服务器，以便在默认模式下执行命令。

先决条件

- 已安装并配置了 GitOps **argocd** CLI 工具。

流程

您必须使用您的凭证登录并在整个会话中保持登录。



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. 获取 Argo CD 服务器的 **admin** 帐户密码：

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

2. 获取 Argo CD 服务器 URL：

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

3. 使用 **admin** 帐户密码并用单引号括起来来登录到 Argo CD 服务器：



重要

用单引号括起密码可确保 shell 不会误解特殊字符，如 **\$**。始终使用单引号括起密码的字面值。

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

Example


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

成功登录后，会话上下文将显示如下：

输出示例

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

2.2. 其他资源

- [安装 GitOps CLI](#)
- [配置 GitOps CLI](#)
- [基本 GitOps argocd 命令](#)

第 3 章 GITOPS ARGOCD CLI 参考



重要

Red Hat OpenShift GitOps **argocd** CLI 工具只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议 (SLA) 支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅[技术预览功能支持范围](#)。

本节列出了基本的 GitOps **argocd** CLI 命令。

3.1. 基本语法

GitOps **argocd** CLI 是一个用于从命令行配置和管理 Red Hat OpenShift GitOps 和 Argo CD 资源的工具。

3.1.1. 默认模式

在默认模式中，**argocd** CLI 客户端通过 API 请求与 Argo CD 服务器组件通信。要执行命令，您必须使用 Argo CD 凭证登录到 Argo CD 服务器，并在整个会话中保持登录。如果登录会话超时，您可以使用 **relogin** 命令再次登录。使用 **argocd** 命令完成后，您可以使用 **logout** 命令注销。

命令语法

ArgoCD [command or options] [arguments...]

3.1.2. 核心模式

在这个模式中，CLI 通过 **kubeconfig** 文件中设置的凭证直接与 Kubernetes API 服务器通信。默认 **kubeconfig** 文件位于 **\$HOME/.kube/config** 位置。您可以使用 **KUBECONFIG** 环境变量自定义此文件。要在 **core** 模式中执行命令，您可以使用 **--core** 参数，不需要登录到 Argo CD 服务器以进行用户身份验证。

要以 **<argocd-instance-name>-repo-server** 格式指定 Repo 服务器组件名称，您可以使用 **--repo-server-name** 命令行选项或设置 **ARGOCD_REPO_SERVER_NAME** 环境变量。

命令语法

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

您可以选择在 **core** 模式中执行 **argocd** 命令之一：



注意

如果使用多个 Argo CD 实例，请将当前上下文的默认命名空间设置为您要与之交互的 ArgoCD 实例的命名空间。

- 默认 **kubeconfig** 文件带有默认上下文：
ArgoCD --core [command or options] [arguments...]

示例 1：显示应用程序列表

■

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

示例 2：显示应用程序列表

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

- 带有自定义上下文的默认 **kubeconfig** 文件：

ArgoCD --core --kube-context [context] [command or options] [arguments...]

示例 1：显示应用程序列表

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

示例 2：显示应用程序列表

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

- 带有默认上下文的自定义 **kubeconfig** 文件：

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

示例：显示应用程序列表

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

- 带有自定义上下文的自定义 **kubeconfig** 文件：

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

示例：显示应用程序列表

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

3.2. 全局选项

全局选项可用于 **argocd** 的所有子命令。

表 3.1. 全局选项

选项	参数类型	描述
--auth-token	string	身份验证令牌。
--client-crt	string	客户端证书文件。
--client-crt-key	string	客户端证书密钥文件。

选项	参数类型	描述
--config	string	Argo CD 配置文件的路径。默认为 /home/user/.config/argocd/config 路径。
--controller-name	string	带有 argocd-application-controller 作为默认标签的 Argo CD Application Controller 组件的名称。 如果此组件名称的标签与默认值不同，例如，当您通过 Helm Chart 安装它时，请设置 --controller-name 选项或 ARGOCD_APPLICATION_CONTROLLER_NAME 环境变量来指定组件的名称。
--core	N/A	如果设置为 true ，CLI 直接与 Kubernetes API 服务器通信，而不使用 Argo CD API 服务器。
--grpc-web	N/A	为 Argo CD 服务器启用 gRPC-Web 协议。例如，当服务器位于不支持 HTTP/2 协议的代理后面，这很有用。
--grpc-web-root-path	string	为 Argo CD 服务器启用 gRPC-Web 协议。例如，当服务器位于不支持 HTTP/2 协议的代理后面，这很有用。设置 Web root。
-h,--header	string	将额外标头配置为由 GitOps argocd CLI 发出的所有请求。您可以通过多次设置这个选项来添加多个标头。这个选项还支持以逗号分隔的标头。
-h, --help	N/A	GitOps argocd CLI 的帮助信息。
--http-retry-max	整数	设置与 Argo CD 服务器建立 HTTP 连接的最大重试次数。
--insecure	N/A	跳过服务器证书和域验证。
--kube-context	string	将命令定向到给定的 kube 上下文。
--logformat	string	将日志记录格式设置为 text 或 JSON 。默认为 文本 。

选项	参数类型	描述
--loglevel	string	设置日志记录级别。默认为 info 。提供了 debug 、 info 、 warn 和 error 级别。日志记录级别按照详细程度降序列出。
--plaintext	N/A	禁用传输层安全(TLS)协议。
--port-forward	N/A	使用端口转发连接到随机 Argo CD 服务器端口。
--port-forward-namespace	string	用于端口转发的命名空间名称。
--redis-haproxy-name	string	<p>使用 argocd-redis-ha-haproxy 作为默认标签的 Redis HA 代理部署的名称。</p> <p>如果此部署的名称的标签与默认值不同，例如，当您通过 Helm Chart 安装 HAProxy 时，请设置 --redis-haproxy-name 选项或 ARGOCD_REDIS_HAPROXY_NAME 环境变量来指定部署的名称。</p>
--redis-name	string	<p>使用 argocd-redis 作为默认标签的 Redis 部署的名称。</p> <p>如果此部署的名称的标签与默认值不同，例如，当您通过 Helm Chart 安装它时，请设置 --redis-name 选项或 ARGOCD_REDIS_NAME 环境变量来指定部署的名称。</p>
--repo-server-name	string	<p>使用 argocd-repo-server 作为默认标签的 Argo CD Repo 服务器的名称。</p> <p>如果此 Repo 服务器名称的标签与默认的不同，例如，当您通过 Helm Chart 安装它时，请设置 --repo-server-name 选项或 ARGOCD_REPO_SERVER_NAME 环境变量来指定部署的名称。</p>
--server	string	Argo CD 服务器地址。
--server-crt	string	服务器证书文件。

选项	参数类型	描述
--server-name	string	<p>带有 argocd-server 作为默认标签的 Argo CD API 服务器的名称。</p> <p>如果此 Argo CD API 服务器名称的标签与默认值不同，例如，当您通过 Helm Chart 安装它时，请设置 -server-name 选项或 ARGOCD_SERVER_NAME 环境变量来指定部署的名称。</p>

3.3. 工具命令

3.3.1. argocd

GitOps **argocd** CLI 的父命令。

示例：显示所有选项

```
$ argocd
```

3.3.2. version

打印 CLI 的版本信息。

命令语法

ArgoCD 版本 [flags]

示例：将客户端和服务器的完整版本输出到 stdout

```
$ argocd version
```

示例：仅打印客户端的完整版本，不会进行与服务器的连接

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

示例：只打印服务器的完整版本

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

示例：以 JSON 格式打印客户端和服务器的完整版本

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

示例：以 YAML 格式仅打印客户端和服务器的内核版本字符串

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

3.3.3. 帮助

打印应用程序中任何命令的帮助信息。

命令语法

ArgoCD help [command] [flags]

示例：获取所有可用命令的帮助文本

```
$ argocd help
```

示例：获取 **admin** 子命令的帮助文本

```
$ argocd help admin
```

3.3.4. completion

将 **bash** 或 **zsh** shell 完成代码写入标准输出。

命令语法

ArgoCD completion SHELL [flags]

对于 **bash**，请确保已安装并启用了 Bash 完成。或者，将其写入一个文件，并将其保存在 **.bash_profile** 中。

示例：访问当前 shell 中的完成

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

对于 **zsh**，请确保已安装并启用了 Bash 完成。

示例：添加到 **~/.zshrc** 文件，并在当前 shell 中访问完成

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

3.4. 其他资源

- [安装 GitOps CLI](#)
- [配置 GitOps CLI](#)
- [以默认模式登录到 Argo CD 服务器](#)
- [OpenShift GitOps CLI 用户指南](#)
- [ArgoCD 命令参考](#)

