Red Hat OpenShift GitOps 1.12

Understanding OpenShift GitOps

Introduction to OpenShift GitOps

Last Updated: 2024-05-13

Red Hat OpenShift GitOps 1.12 Understanding OpenShift GitOps

Introduction to OpenShift GitOps

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 an overview of OpenShift GitOps and its features.

Table of Contents

CHAPTER 1. WHAT IS GITOPS?

GitOps is a declarative way to implement continuous deployment for cloud native applications. You can use GitOps to create repeatable processes for managing OpenShift Container Platform clusters and applications across multi-cluster Kubernetes environments. GitOps handles and automates complex deployments at a fast pace, saving time during deployment and release cycles.

The GitOps workflow pushes an application through development, testing, staging, and production. GitOps either deploys a new application or updates an existing one, so you only need to update the repository; GitOps automates everything else.

GitOps is a set of practices that use Git pull requests to manage infrastructure and application configurations. In GitOps, the Git repository is the only source of truth for system and application configuration. This Git repository contains a declarative description of the infrastructure you need in your specified environment and contains an automated process to make your environment match the described state. Also, it contains the entire state of the system so that the trail of changes to the system state are visible and auditable. By using GitOps, you resolve the issues of infrastructure and application configuration sprawl.

GitOps defines infrastructure and application definitions as code. Then, it uses this code to manage multiple workspaces and clusters to simplify the creation of infrastructure and application configurations. By following the principles of the code, you can store the configuration of clusters and applications in Git repositories, and then follow the Git workflow to apply these repositories to your chosen clusters. You can apply the core principles of developing and maintaining software in a Git repository to the creation and management of your cluster and application configuration files.

CHAPTER 2. ABOUT RED HAT OPENSHIFT GITOPS

Red Hat OpenShift GitOps is an Operator that uses Argo CD as the declarative GitOps engine. It enables GitOps workflows across multicluster OpenShift and Kubernetes infrastructure. Using Red Hat OpenShift GitOps, administrators can consistently configure and deploy Kubernetes-based infrastructure and applications across clusters and development lifecycles. Red Hat OpenShift GitOps is based on the open source project Argo CD and provides a similar set of features to what the upstream offers, with additional automation, integration into Red Hat OpenShift Container Platform and the benefits of Red Hat's enterprise support, quality assurance and focus on enterprise security.

NOTE

Because Red Hat OpenShift GitOps releases on a different cadence from OpenShift Container Platform, the Red Hat OpenShift GitOps documentation is now available as separate documentation sets for each minor version of the product.

The Red Hat OpenShift GitOps documentation is available at https://docs.openshift.com/gitops/.

Documentation for specific versions is available using the version selector dropdown, or directly by adding the version to the URL, for example, https://docs.openshift.com/gitops/1.8.

In addition, the Red Hat OpenShift GitOps documentation is also available on the Red Hat Portal at https://access.redhat.com/documentation/en-us/red_hat_openshift_gitops/.

For additional information about the Red Hat OpenShift GitOps life cycle and supported platforms, refer to the Platform Life Cycle Policy.

Red Hat OpenShift GitOps ensures consistency in applications when you deploy them to different clusters in different environments, such as: development, staging, and production. Red Hat OpenShift GitOps organizes the deployment process around the configuration repositories and makes them the central element. It always has at least two repositories:

1. Application repository with the source code
2. Environment configuration repository that defines the desired state of the application

These repositories contain a declarative description of the infrastructure you need in your specified environment. They also contain an automated process to make your environment match the described state.

Red Hat OpenShift GitOps uses Argo CD to maintain cluster resources. Argo CD is an open-source declarative tool for the continuous deployment (CD) of applications. Red Hat OpenShift GitOps implements Argo CD as a controller so that it continuously monitors application definitions and configurations defined in a Git repository. Then, Argo CD compares the specified state of these configurations with their live state on the cluster.

Argo CD reports any configurations that deviate from their specified state. These reports allow administrators to automatically or manually resync configurations to the defined state. Therefore, Argo CD enables you to deliver global custom resources, like the resources that are used to configure OpenShift Container Platform clusters.

2.1. KEY FEATURES

Red Hat OpenShift GitOps helps you automate the following tasks:

- Ensure that the clusters have similar states for configuration, monitoring, and storage
- Apply or revert configuration changes to multiple OpenShift Container Platform clusters
- Associate templated configuration with different environments
- Promote applications across clusters, from staging to production

2.2. ADDITIONAL RESOURCES

- Extending the Kubernetes API with custom resource definitions
- Managing resources from custom resource definitions
- What is GitOps?

CHAPTER 3. GATHERING DIAGNOSTIC INFORMATION FOR SUPPORT

When you open a support case, you must provide debugging information about your cluster to the Red Hat Support team. You can use the **must-gather** tool to collect diagnostic information for project-level resources, cluster-level resources, and Red Hat OpenShift GitOps components.

> **NOTE**
>
> For prompt support, provide diagnostic information for both OpenShift Container Platform and Red Hat OpenShift GitOps.

3.1. ABOUT THE MUST-GATHER TOOL

The **oc adm must-gather** CLI command collects the information from your cluster that is most likely needed for debugging issues, including:

- Resource definitions
- Service logs

By default, the **oc adm must-gather** command uses the default plugin image and writes into **./must-gather.local**.

Alternatively, you can collect specific information by running the command with the appropriate arguments as described in the following sections:

- To collect data related to one or more specific features, use the **--image** argument with an image, as listed in a following section.

  Example command

  ```
  $ oc adm must-gather --image=registry.redhat.io/openshift-gitops-1/must-gather-rhel8:v1.10.0
  ```

- To collect the audit logs, use the **-- /usr/bin/gather_audit_logs** argument, as described in a following section.

  Example command

  ```
  $ oc adm must-gather -- /usr/bin/gather_audit_logs
  ```

  > **NOTE**
  >
  > Audit logs are not collected as part of the default set of information to reduce the size of the files.

When you run **oc adm must-gather**, a new pod with a random name is created in a new project on the cluster. The data is collected on that pod and saved in a new directory that starts with **must-gather.local**. This directory is created in the current working directory.

Example pod

```
NAMESPACE                   NAME              READY   STATUS      RESTARTS      AGE
...
openshift-must-gather-5drcj   must-gather-bklx4   2/2     Running    0           72s
openshift-must-gather-5drcj   must-gather-s8sdh   2/2     Running    0           72s
...
```

Optionally, you can run the **oc adm must-gather** command in a specific namespace by using the **--run-namespace** option.

Example command

```
$ oc adm must-gather --image=registry.redhat.io/openshift-gitops-1/must-gather-rhel8:v1.10.0
```

3.2. COLLECTING DEBUGGING DATA FOR RED HAT OPENSHIFT GITOPS

Use the **oc adm must-gather** CLI command to collect the following details about the cluster that is associated with Red Hat OpenShift GitOps:

- The subscription and namespace of the Red Hat OpenShift GitOps Operator.
- The namespaces where ArgoCD objects are available and the objects in those namespaces, such as **ArgoCD**, **Applications**, **ApplicationSets**, **AppProjects**, and **configmaps**.
- A list of the namespaces that are managed by the Red Hat OpenShift GitOps Operator, and resources from those namespaces.
- All GitOps-related custom resource objects and definitions.
- Operator and Argo CD logs.
- Warning and error-level events.

Prerequisites

- You have logged in to the OpenShift Container Platform cluster as an administrator.
- You have installed the OpenShift Container Platform CLI (**oc**).
- You have installed the Red Hat OpenShift GitOps Operator.

Procedure

1. Navigate to the directory where you want to store the debugging information.
2. Run the **oc adm must-gather** command with the Red Hat OpenShift GitOps **must-gather** image:

   ```
   $ oc adm must-gather --image=registry.redhat.io/openshift-gitops-1/must-gather-rhel8:
   <image_version_tag> 1
   ```

   1 The must-gather image for GitOps.

Example command

```
$ oc adm must-gather --image=registry.redhat.io/openshift-gitops-1/must-gather-rhel8:v1.10.0
```

The **must-gather** tool creates a new directory that starts with **./must-gather.local** in the current directory. For example, **./must-gather.local.4157245944708210399**.

3. Create a compressed file from the directory that was just created. For example, on a computer that uses a Linux operating system, run the following command:

```
$ tar -cvaf must-gather.tar.gz must-gather.local.4157245944708210399 1
```

1 Replace **must-gather-local.4157245944708210399** with the actual directory name.

4. Attach the compressed file to your support case on the Red Hat Customer Portal .

3.3. ADDITIONAL RESOURCES

- Gathering data about specific features