



# Red Hat OpenShift GitOps 1.12

## 了解 OpenShift GitOps

OpenShift GitOps 简介





## 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

本文档概述 OpenShift GitOps 及其功能。

---

## 目录

<b>第 1 章 什么是 GITOPS?</b> .....	<b>3</b>
<b>第 2 章 关于 RED HAT OPENSIFT GITOPS</b> .....	<b>4</b>
2.1. 主要特性	4
2.2. 其他资源	5
<b>第 3 章 为支持收集诊断信息</b> .....	<b>6</b>
3.1. 关于 MUST-GATHER 工具	6
3.2. 为 RED HAT OPENSIFT GITOPS 收集调试数据	7
3.3. 其他资源	8



## 第1章 什么是 GITOPS?

GitOps 是为云原生应用程序实施持续部署的一种声明方式。您可以使用 GitOps 创建可重复进程，用于在多集群 Kubernetes 环境间管理 OpenShift Container Platform 集群和应用程序。GitOps 以快速的速度处理和自动化复杂部署，节省部署和发行周期期间的的时间。

GitOps 工作流通过开发、测试、临时和生产环境来推送应用程序。GitOps 部署新应用程序或更新现有应用程序，因此您只需要更新存储库，GitOps 会自动执行所有操作。

GitOps 是一组使用 Git 拉取请求来管理基础架构和应用程序配置的实践。GitOps 中的 Git 存储库是系统和应用程序配置的唯一来源。此 Git 存储库包含指定环境中所需的基础架构声明描述，并包含自动流程，以使您的环境与上述状态匹配。它还包含该系统的完整状态，以便可查看并可审核更改到系统状态。通过使用 GitOps，您可以处理基础架构和应用程序配置 sprawl 的问题。

GitOps 将基础架构和应用程序定义定义为代码。然后，它会使用此代码来管理多个工作区和集群来简化基础架构和应用程序配置的创建过程。根据代码原则，您可以在 Git 存储库中存储集群和应用程序的配置，然后按照 Git 工作流将这些存储库应用到所选集群中。您可以将在 Git 存储库中开发和维护软件的核心原则应用到创建和管理集群和应用程序配置文件。

## 第 2 章 关于 RED HAT OPENSIFT GITOPS

Red Hat OpenShift GitOps 是一个使用 Argo CD 作为声明性 GitOps 引擎的 Operator。它启用了多集群 OpenShift 和 Kubernetes 基础架构的 GitOps 工作流。使用 Red Hat OpenShift GitOps，管理员可以在集群和开发生命周期中一致地配置和部署基于 Kubernetes 的基础架构和应用程序。Red Hat OpenShift GitOps 基于开源项目 [Argo CD](#)，为上游提供的功能提供类似的功能，并提供额外的自动化功能，集成到 Red Hat OpenShift Container Platform 中，以及 Red Hat OpenShift Container Platform 的好处，以及 Red Hat Enterprise 支持、质量保证并专注于企业安全性。



### 注意

因为 Red Hat OpenShift GitOps 的发行节奏与 OpenShift Container Platform 不同，所以 Red Hat OpenShift GitOps 文档现在作为每个次版本的单独文档提供。

Red Hat OpenShift GitOps 文档包括在 <https://docs.openshift.com/gitops/>。

特定版本的文档使用版本选择器下拉列表，或者直接将版本添加到 URL，例如 <https://docs.openshift.com/gitops/1.8>。

另外，Red Hat OpenShift GitOps 文档也包括在 Red Hat Portal [https://access.redhat.com/documentation/zh-cn/red\\_hat\\_openshift\\_gitops/](https://access.redhat.com/documentation/zh-cn/red_hat_openshift_gitops/) 中。

如需有关 Red Hat OpenShift GitOps 生命周期和支持的平台的信息，请参阅[平台生命周期政策](#)。

当应用程序部署到不同环境中的不同集群时，Red Hat OpenShift GitOps 可确保应用程序的一致性，如开发、临时和生产环境。Red Hat OpenShift GitOps 整理与配置仓库相关的部署过程，并将其作为核心元素。它总会保持至少有两个软件仓库：

1. 源代码的应用程序仓库
2. 定义应用程序所需状态的环境配置仓库

这些软件仓库包含您指定环境中所需的基础架构声明信息。它们还包含让您的环境与上述状态匹配的自动过程。

Red Hat OpenShift GitOps 使用 Argo CD 来维护集群资源。Argo CD 是一个开源声明工具，用于应用程序的持续部署(CD)。Red Hat OpenShift GitOps 将 Argo CD 实现作为一个控制器，以便持续监控 Git 存储库中定义的应用程序定义和配置。然后，Argo CD 将这些配置的指定状态与集群中的实时状态进行比较。

Argo CD 报告与指定状态不同的配置。报告允许管理员自动或者手动将配置重新同步到定义的状态。因此，ArgoCD 可让您提供全局自定义资源，如用于配置 OpenShift Container Platform 集群的资源。

### 2.1. 主要特性

Red Hat OpenShift GitOps 可帮助您自动执行以下任务：

- 确保集群具有类似的配置、监控和存储状态
- 对多个 OpenShift Container Platform 集群应用或恢复配置更改
- 将模板配置与不同环境关联
- 在集群间（从调试到生产阶段）推广应用程序。

## 2.2. 其他资源

- [使用自定义资源定义来扩展 Kubernetes API](#)
- [管理自定义资源定义中的资源](#)
- [什么是 GitOps?](#)

## 第 3 章 为支持收集诊断信息

当创建一个支持问题单时，您必须为红帽支持团队提供有关集群的调试信息。您可以使用 **must-gather** 工具来收集项目级别资源、集群级别资源和 Red Hat OpenShift GitOps 组件的诊断信息。



### 注意

为了获得快速支持，请提供 OpenShift Container Platform 和 Red Hat OpenShift GitOps 的诊断信息。

### 3.1. 关于 MUST-GATHER 工具

**oc adm must-gather** CLI 命令可收集最有助于解决问题的集群信息，包括：

- 资源定义
- 服务日志

默认情况下，**oc adm must-gather** 命令使用默认的插件镜像，并写入 **./must-gather.local**。

另外，您可以使用适当的参数运行命令来收集具体信息，如以下部分所述：

- 要收集与一个或多个特定功能相关的数据，请使用 **--image** 参数和镜像，如以下部分所述。

#### 示例命令

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

- 要收集审计日志，请使用 **-- /usr/bin/gather\_audit\_logs** 参数，如以下部分所述。

#### 示例命令

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



### 注意

作为默认信息集合的一部分，不会收集审计日志来减小文件的大小。

当您运行 **oc adm must-gather** 时，集群的新项目中会创建一个带有随机名称的新 pod。在该 pod 上收集数据，并保存至以 **must-gather.local** 开头的一个新目录中。此目录在当前工作目录中创建。

#### Pod 示例

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
...					

另外，您可以使用 **--run-namespace** 选项在特定命名空间中运行 **oc adm must-gather** 命令。

#### 示例命令

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

## 3.2. 为 RED HAT OPENSIFT GITOPS 收集调试数据

使用 **oc adm must-gather** CLI 命令收集有关与 Red Hat OpenShift GitOps 关联的集群的以下详情：

- Red Hat OpenShift GitOps Operator 的订阅和命名空间。
- ArgoCD 对象可用的命名空间，以及这些命名空间中的对象，如 **ArgoCD**, **Applications**, **ApplicationSets**, **AppProjects**, 和 **configmaps**。
- 由 Red Hat OpenShift GitOps Operator 管理的命名空间列表，以及来自这些命名空间中的资源。
- 所有与 GitOps 相关的自定义资源对象和定义。
- Operator 和 Argo CD 日志。
- 警告和错误级事件。

### 先决条件

- 以管理员身份登录到 OpenShift Container Platform 集群。
- 已安装 OpenShift Container Platform CLI (**oc**)。
- 已安装 Red Hat OpenShift GitOps Operator。

### 流程

1. 进入到要存储调试信息的目录。
2. 使用 Red Hat OpenShift GitOps **must-gather** 镜像运行 **oc adm must-gather** 命令：

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

- 1** GitOps 的 **must-gather** 镜像。

### 示例命令

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

**must-gather** 工具会创建一个以当前目录中 **./must-gather.local** 开头的新目录。例如，**./must-gather.local.4157245944708210399**。

3. 从刚才创建的目录中创建压缩文件。例如，在使用 Linux 操作系统的计算机上运行以下命令：

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

- 1** 将 **must-gather-local.4157245944708210399** 替换为实际目录名称。

4. 在[红帽客户门户](#)中为您的问题单附上压缩文件。

### 3.3. 其他资源

- [收集有关特定功能的数据](#)