



Red Hat OpenShift GitOps 1.12

Argo Rollouts

使用 Argo Rollouts 进行进度交付

使用 Argo Rollouts 进行进度交付

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

本文档提供有关使用 Argo Rollouts 来封装声明性推出部署策略所需的所有定义。它还讨论了如何管理和自动作为 GitOps workflow 的一部分进行管理和自动化部署。

目录

第 1 章 使用 ARGO ROLLOUTS 进行渐进式部署交付	3
1.1. 先决条件	3
1.2. ARGO ROLLOUTS 的好处	3
1.3. 关于 ROLLOUTMANAGER 自定义资源和规格	4
1.4. 创建 ROLLOUTMANAGER 自定义资源	4
1.5. 删除 ROLLOUTMANAGER 自定义资源	5
1.6. 其他资源	6

第 1 章 使用 ARGO ROLLOUTS 进行渐进式部署交付



重要

Argo Rollouts 仅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](#)。

渐进式交付是以一种可控的渐进式方式发布产品更新的过程。渐进式交付在初始期间使产品更新的新版本只对一部分用户可用，从而达到降低版本风险的目的。这个过程涉及持续观察和分析新版本，以验证其行为是否与要求和预期集匹配。在持续验证新版本的过程中，会使新版本对更多的用户可用。

OpenShift Container Platform 通过使用路由来分隔不同服务间的网络数据来实现渐进式交付的功能，但这通常需要人工的干预和管理。

使用 Argo Rollouts，您可以使用自动化和指标分析来支持渐进式部署交付，并驱动应用程序的新版本自动推出部署或回滚。Argo Rollouts 提供了高级部署功能，并启用与入口控制器和服务网格的集成。您可以使用 Argo Rollouts 管理代表已部署应用程序的不同版本的多个副本集。根据您的部署策略，您可以通过优化其现有流量控制能力并逐步将流量移到新版本，从而在更新过程中处理这些版本的流量。您可以将 Argo Rollouts 与 Prometheus 等指标供应商相结合，根据参数集进行基于指标和策略驱动的推出部署和回滚。

1.1. 先决条件

- 您可以使用 **cluster-admin** 权限访问集群。
- 访问 OpenShift Container Platform web 控制台。
- 在集群中安装了 Red Hat OpenShift GitOps 1.9.0 或更新的版本。

1.2. ARGO ROLLOUTS 的好处

在传统基础架构中管理和协调高级部署策略通常涉及较长时间的维护窗口。使用 OpenShift Container Platform 和 Red Hat OpenShift GitOps 等工具实现自动化可以减少这些窗口，但设置这些策略仍具有挑战性。使用 Argo Rollouts，您可以通过允许应用程序团队以声明性方式定义其 rollout 策略来简化此过程。团队不再需要定义多个部署和服务，或为测试的流量控制和集成创建自动化。使用 Argo Rollouts，您可以封装声明性推出部署策略所需的所有定义，自动化和管理流程。

在 Red Hat OpenShift GitOps 中使用 Argo Rollouts 作为默认工作负载提供以下优点：

- 作为 GitOps 工作流的一部分的自动化进度交付
- 高级部署功能
- 优化现有的高级部署策略，如 blue-green 或 canary
- 部署可以实现零停机时间更新
- 细粒度、加权流量转换
- 能够在没有达到生产环境的新流量的情况下进行测试
- 自动回滚和提升

- 手工调整
- 可自定义的指标查询和分析业务关键性能指标 (KPI)
- 与 ingress 控制器和 Red Hat OpenShift Service Mesh 集成用于高级流量路由
- 与指标供应商集成以进行部署策略分析
- 使用多个供应商

使用 Argo Rollouts，用户可以在最终用户环境中更轻松地采用进度交付。这在不需要团队了解流量管理器 and 复杂基础架构的情况下提供了结构和指南。通过自动推出部署，Red Hat OpenShift GitOps Operator 为最终用户环境提供安全性，并有助于有效地管理资源、成本和时间。使用带有安全性和自动化部署的 Argo CD 的现有用户会在过程中早期收到反馈，并避免会影响到用户环境的的问题。

1.3. 关于 ROLLOUTMANAGER 自定义资源和规格

要使用 Argo Rollouts，您必须在集群中安装 Red Hat OpenShift GitOps Operator，然后在您选择的命名空间中创建 **RolloutManager** 自定义资源 (CR) 并将其提交到 Operator。您可以为单个或多个命名空间限定 **RolloutManager** CR。Operator 创建一个带有以下命名空间范围的支持资源的 **argo-rollouts** 实例：

- Argo Rollouts 控制器
- Argo Rollouts 指标服务
- Argo Rollouts 服务帐户
- Argo Rollouts 角色
- Argo Rollouts 角色绑定
- Argo Rollouts secret

您可以在 **RolloutsManager** CR 的 spec 中指定命令参数、环境变量、自定义镜像名称等。**RolloutManager** CR spec 定义 Argo Rollouts 的所需状态。

示例：RolloutManager CR

```
apiVersion: argoproj.io/v1beta1
kind: RolloutManager
metadata:
  name: argo-rollout
  labels:
    example: basic
spec: {}
```

1.3.1. Argo Rollouts 控制器

使用 Argo Rollouts 控制器资源，您可以管理命名空间中的渐进式应用程序交付。Argo Rollouts 控制器资源监控集群的事件，并在任何与 Argo Rollouts 相关的资源有变化时做出反应。控制器读取所有推出部署详情，并将集群引入相同的状态，如 rollout 定义中所述。

1.4. 创建 ROLLOUTMANAGER 自定义资源

要使用 Red Hat OpenShift GitOps 中的 Argo Rollouts 管理部署的进度交付，您必须在您选择的命名空间中创建并配置 **RolloutManager** 自定义资源 (CR)。默认情况下，任何新的 **argo-rollouts** 实例都只能管理部署它的命名空间中的资源，但您可以根据需要在多个命名空间中使用 Argo Rollouts。

先决条件

- 在集群中安装了 Red Hat OpenShift GitOps 1.9.0 或更新的版本。

流程

1. 以集群管理员身份登录 OpenShift Container Platform Web 控制台。
2. 在 **Administrator** 视角中，点 **Operators** → **Installed Operators**。
3. 从 **Project** 下拉菜单中选择您要创建和配置 **RolloutManager** 自定义资源 (CR) 的项目。
4. 从安装的 Operator 中选择 **OpenShift GitOps Operator**。
5. 在 **Details** 选项卡中，在 **Provided APIs** 部分下，点 **RolloutManager** 窗格中的 **Create instance**。
6. 在 **Create RolloutManager** 页面中，选择 **YAML** 视图，并使用默认 YAML 或根据您的要求编辑它：

示例：RolloutManager CR

```
apiVersion: argoproj.io/v1beta1
kind: RolloutManager
metadata:
  name: argo-rollout
  labels:
    example: basic
spec: {}
```

7. 点 **Create**。
8. 在 **RolloutManager** 选项卡中，在 **RolloutManagers** 部分下，验证 RolloutManager 实例的 **Status** 字段显示为 **Phase: Available**。
9. 在左侧导航窗格中，验证命名空间范围的支持资源的创建：
 - 点 **Workloads** → **Deployments** 来验证 **argo-rollouts** 部署是否可用，**Status** 显示为 **1 of 1 pods** running。
 - 点 **Workloads** → **Secrets** 来验证 **argo-rollouts-notification-secret** secret 是否可用。
 - 点 **Networking** → **Services** 来验证 **argo-rollouts-metrics** 服务是否可用。
 - 点 **User Management** → **Roles** 来验证 **argo-rollouts** 角色和 **argo-rollouts-aggregate-to-admin**、**argo-rollouts-aggregate-to-edit** 以及 **argo-rollouts-aggregate-to-view** 集群角色是否可用。
 - 点 **User Management** → **RoleBindings** 来验证 **argo-rollouts** 角色绑定是否可用。

1.5. 删除 ROLLOUTMANAGER 自定义资源

卸载 Red Hat OpenShift GitOps Operator 不会删除在安装过程中创建的资源。在卸载 Red Hat OpenShift GitOps Operator 前，您必须手动删除 **RolloutManager** 自定义资源(CR)。

先决条件

- 在集群中安装了 Red Hat OpenShift GitOps 1.9.0 或更新的版本。
- 命名空间中已存在 **RolloutManager** CR。

流程

1. 以集群管理员身份登录 OpenShift Container Platform Web 控制台。
2. 在 **Administrator** 视角中，点 **Operators** → **Installed Operators**。
3. 点 **Project** 下拉菜单，再选择包含 **RolloutManager** CR 的项目。
4. 从安装的 Operator 中选择 **OpenShift GitOps Operator**。
5. 点 **RolloutManager** 选项卡，以查找 **RolloutManagers** 部分下的 **RolloutManager** 实例。
6. 点实例。
7. 从下拉菜单中选择 **Actions** → **Delete RolloutManager**，然后点 **Delete** 在对话框中进行确认。
8. 在 **RolloutManager** 选项卡中，在 **RolloutManagers** 部分下，验证 **RolloutManager** 实例已不再可用。
9. 在左侧导航窗格中，验证删除命名空间范围的支持资源：
 - 点 **Workloads** → **Deployments** 来验证 **argo-rollouts** 部署已被删除。
 - 点 **Workloads** → **Secrets** 来验证 **argo-rollouts-notification-secret** secret 已被删除。
 - 点 **Networking** → **Services** 来验证 **argo-rollouts-metrics** 服务已被删除。
 - 点 **User Management** → **Roles** 来验证 **argo-rollouts** 角色和 **argo-rollouts-aggregate-to-admin**、**argo-rollouts-aggregate-to-edit** 以及 **argo-rollouts-aggregate-to-view** 集群角色已被删除。
 - 点 **User Management** → **RoleBindings** 来验证 **argo-rollouts** 角色绑定已被删除。

1.6. 其他资源

- [安装 Red Hat OpenShift GitOps](#)
- [卸载 Red Hat OpenShift GitOps](#)
- [Canary 部署](#)
- [蓝绿部署](#)
- [RolloutManager 自定义资源规格](#)
- [使用 Argo Rollouts 的蓝绿和 Canary 部署](#)
- [Argo Rollouts 技术预览限制](#)

