



Red Hat OpenShift GitOps 1.12

Observability (可观察性)

使用可观察性功能查看 Argo CD 日志并监控 Argo CD 和应用程序资源的性能和健康状况

Red Hat OpenShift GitOps 1.12 Observability (可观察性)

使用可观察性功能查看 Argo CD 日志并监控 Argo CD 和应用程序资源的性能和健康状况

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

本文档详细介绍了如何将 OpenShift Logging 与 OpenShift GitOps 搭配使用，并监控 Argo CD 实例、应用程序健康状态和 Argo CD 自定义资源工作负载的性能。

目录

| | |
|----------------------------|----------|
| 第1章 日志记录 | 3 |
| 1.1. 查看 ARGO CD 日志 | 3 |
| 第2章 监控 | 4 |
| 2.1. 使用 GITOPS 仪表盘监控 | 4 |
| 2.2. 监控 ARGO CD 实例 | 4 |
| 2.3. 监控 GITOPS OPERATOR 性能 | 5 |
| 2.4. 监控应用程序资源和部署的健康状况信息 | 7 |
| 2.5. 监控 ARGO CD 自定义资源工作负载 | 9 |

第 1 章 日志记录

1.1. 查看 ARGO CD 日志

您可以使用 Red Hat OpenShift 的 logging 子系统查看 Argo CD 日志。logging 子系统可视化 Kibana 仪表板上的日志。OpenShift Logging Operator 默认启用 Argo CD 的日志记录。

1.1.1. 存储和检索 Argo CD 日志

您可以使用 Kibana 仪表板存储和检索 Argo CD 日志。

先决条件

- Red Hat OpenShift GitOps Operator 已安装在 OpenShift Container Platform 集群中。
- Red Hat OpenShift 的 logging 子系统安装有 OpenShift Container Platform 集群上的默认配置。

流程

1. 在 OpenShift Container Platform web 控制台中，进入  菜单 → **Observability** → **Logging** 来查看 Kibana 仪表板。
2. 创建索引模式。
 - a. 要显示所有索引，请将索引模式定义为 `*`，然后点 **Next step**。
 - b. 为 **Time Filter field name** 选择 `@timestamp`。
 - c. 单击 **Create index pattern**。
3. 在 Kibana 仪表板的导航面板中，点 **Discover** 选项卡。
4. 创建过滤器以检索 Argo CD 的日志。以下步骤创建一个过滤器，用于检索 `openshift-gitops` 命名空间中的所有 pod 的日志：
 - a. 点 **Add a filter** 。
 - b. 选择 `kubernetes.namespace_name` 字段。
 - c. 选择 `is operator`。
 - d. 选择 `openshift-gitops` 值。
 - e. 点击 **Save**。
5. 可选：添加额外的过滤器来缩小搜索范围。例如，要检索特定 pod 的日志，您可以使用 `kubernetes.pod_name` 作为字段创建另一个过滤器。
6. 在 Kibana 仪表板中查看过滤的 Argo CD 日志。

1.1.2. 其他资源

- [使用 Web 控制台为 Red Hat OpenShift 安装 logging 子系统](#)

第 2 章 监控

2.1. 使用 GITOPS 仪表板监控

您可以使用 Red Hat OpenShift GitOps 监控仪表板访问 GitOps 实例的图形视图，以观察集群中每个实例的行为和使用。

可用的 GitOps 仪表板有三个：

- **GitOps Overview**：查看集群中安装的所有 GitOps 实例的概述，包括应用程序、健康和同步状态、应用程序和同步活动的数量。
- **GitOps 组件**：查看应用程序控制器、repo-server、server 和其他 GitOps 组件的详细信息，如 CPU 或内存。
- **GitOps gRPC Services**：查看 Red Hat OpenShift GitOps 中不同组件间与 gRPC 服务活动相关的指标。

2.1.1. 访问 GitOps 监控仪表板

监控仪表板由 Operator 自动部署。您可以从 OpenShift Container Platform Web 控制台的 **Administrator** 视角访问 GitOps 监控仪表板。



注意

不支持禁用或更改仪表板的内容。

先决条件

- 访问 OpenShift Container Platform web 控制台。
- Red Hat OpenShift GitOps Operator 安装在默认命名空间 **openshift-gitops-operator** 中。
- 集群监控在 **openshift-gitops-operator** 命名空间中启用。
- 您已在定义的命名空间中安装了 Argo CD 应用程序，如 **openshift-gitops**。

流程

1. 在 web 控制台的 **Administrator** 视角中，进入 **Observe** → **Dashboards**。
2. 在 **Dashboard** 下拉列表中选择所需的 GitOps 仪表板：**GitOps (Overview)**、**GitOps / 组件** 或 **GitOps / gRPC Services**。
3. 可选：从 **Namespace**、**Cluster** 和 **Interval** 下拉列表中选择特定的命名空间、集群和间隔。
4. 在 GitOps 仪表板中查看所需的 GitOps 指标。

2.2. 监控 ARGO CD 实例

默认情况下，Red Hat OpenShift GitOps Operator 会自动检测您定义的命名空间中已安装的 Argo CD 实例，如 **openshift-gitops**，并将其连接到集群的监控堆栈，以便为不同步应用程序提供警报。

2.2.1. 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。
- 您已在 OpenShift Container Platform 集群中安装了 Red Hat OpenShift GitOps Operator。
- 您已在定义的命名空间中安装了 Argo CD 应用程序，如 **openshift-gitops**。

2.2.2. 使用 Prometheus 指标监控 Argo CD 健康状况

您可以通过针对它运行 Prometheus 指标查询来监控 Argo CD 应用程序的健康状况。

流程

1. 在 web 控制台的 **Developer** 视角中，选择安装 Argo CD 应用程序的命名空间，并导航到 **Observe** → **Metrics**。
2. 在 **Select query** 下拉列表中，选择 **Custom query**。
3. 要检查 Argo CD 应用程序的健康状态，在 **Expression** 字段中输入 Prometheus Query Language (PromQL) 查询，如下例所示：

Example

```
sum(argocd_app_info{dest_namespace=~"<your_defined_namespace>",health_status!=""})
by (health_status) ❶
```

- ❶ 将 `<your_defined_namespace>` 变量替换为您定义的命名空间的实际名称，如 **openshift-gitops**。

2.3. 监控 GITOPS OPERATOR 性能

Red Hat OpenShift GitOps Operator 会提供有关其性能的指标。使用获取这些指标的 OpenShift 监控堆栈，您可以监控和分析 Operator 的性能。Operator 会公开以下指标，您可以使用 OpenShift Container Platform Web 控制台查看这些指标：

表 2.1. GitOps Operator 性能指标

| 指标名称 | 类型 | 描述 |
|--|----|--|
| active_argocd_instances_total | 量表 | 在给定时间，当前由 Operator 管理的活跃 Argo CD 实例总数。 |
| active_argocd_instances_by_phase | 量表 | 给定阶段中活跃的 Argo CD 实例数量，如 pending 或 available。 |
| active_argocd_instance_reconciliation_count | 计数 | 在任意给定时间为给定命名空间中的实例发生的协调总数。 |

| 指标名称 | 类型 | 描述 |
|--|----|---|
| <code>controller_runtime_reconcile_time_seconds_per_instance_bucket</code> | 计数 | 实例的给定持续时间下完成的协调周期数量。例如， <code>controller_runtime_reconcile_time_seconds_per_instance_bucket{le="0.5"}</code> 显示给定实例在 0.5 秒下完成的协调数量。 |
| <code>controller_runtime_reconcile_time_seconds_per_instance_count</code> | 计数 | 为给定实例观察到的协调周期总数。 |
| <code>controller_runtime_reconcile_time_seconds_per_instance_sum</code> | 计数 | 给定实例观察到的协调所花费的时间总量。 |



注意

量表是一个可以向上或下移的值。counter 是只能向上的值。

2.3.1. 访问 GitOps Operator 指标

您可以从 OpenShift Container Platform Web 控制台的 **Administrator** 视角访问 Operator 指标，以跟踪 Operator 的性能。

先决条件

- 访问 OpenShift Container Platform web 控制台。
- Red Hat OpenShift GitOps Operator 安装在默认的 `openshift-gitops-operator` 命名空间中。
- 集群监控在 `openshift-gitops-operator` 命名空间中启用。

流程

1. 在 web 控制台的 **Administrator** 视角中，进入 **Observe** → **Metrics**。
2. 在 **Expression** 字段中输入指标。您可以从以下指标中选择：
 - `active_argocd_instances_total`
 - `active_argocd_instances_by_phase`
 - `active_argocd_instance_reconciliation_count`
 - `controller_runtime_reconcile_time_seconds_per_instance_bucket`
 - `controller_runtime_reconcile_time_seconds_per_instance_count`

- `controller_runtime_reconcile_time_seconds_per_instance_sum`

3. (可选) : 根据属性过滤指标。例如, 根据 **Available** 阶段过滤 `active_argocd_instances_by_phase` 指标 :

Example

```
active_argocd_instances_by_phase{phase="Available"}
```

4. (可选) : 点击 **Add query** 来输入多个查询。
5. 点 **Run queries** 启用并观察 GitOps Operator 指标。

2.3.2. 其他资源

- [在 Web 控制台中安装 Red Hat OpenShift GitOps Operator](#)

2.4. 监控应用程序资源和部署的健康状况信息

OpenShift Container Platform Web 控制台的 **Developer** 视角中的 **Red Hat OpenShift GitOps Environments** 页面显示应用程序环境成功部署的列表, 以及指向每个部署的修订版本的链接。

OpenShift Container Platform Web 控制台的 **Developer** 视角中的 **Application environments** 页面显示应用程序资源的健康状况, 如路由、同步状态、部署配置和部署历史记录。

OpenShift Container Platform Web 控制台的 **Developer** 视角中的环境页面与 Red Hat OpenShift GitOps Application Manager 命令行界面(CLI) **kam** 分离。您不必使用 **kam** 为环境生成应用程序环境清单, 以便在 OpenShift Container Platform Web 控制台的 **Developer** 视角中显示。您可以使用自己的清单, 但环境仍必须由命名空间表示。另外, 仍然需要特定的标签和注解。

2.4.1. 环境标签和注解的设置

本节在 OpenShift Container Platform Web 控制台的 **Developer** 视角中提供在 **Environments** 页面中显示环境应用程序所需的环境标签和注解的引用设置。

环境标签

环境应用程序清单必须包含 `labels.openshift.gitops/environment` 和 `destination.namespace` 字段。您必须为 `<environment_name>` 变量和环境应用程序清单的名称设置相同的值。

环境应用程序清单的规格

```
spec:
  labels:
    openshift.gitops/environment: <environment_name>
  destination:
    namespace: <environment_name>
# ...
```

环境应用程序清单示例

```
apiVersion: argoproj.io/v1beta1
kind: Application
metadata:
  name: dev-env 1
```

```

namespace: openshift-gitops
spec:
  labels:
    openshift.gitops/environment: dev-env
  destination:
    namespace: dev-env
# ...

```

- 1 环境应用清单的名称。设置的值与 `<environment_name>` 变量的值相同。

环境注解

环境命名空间清单必须包含 `annotations.app.openshift.io/vcs-uri` 和 `annotations.app.openshift.io/vcs-ref` 字段来指定应用程序的版本控制器代码源。您必须为 `<environment_name>` 变量和环境命名空间清单的名称设置相同的值。

环境命名空间清单的规格

```

apiVersion: v1
kind: Namespace
metadata:
  annotations:
    app.openshift.io/vcs-uri: <application_source_url>
    app.openshift.io/vcs-ref: <branch_reference>
  name: <environment_name> 1
# ...

```

- 1 环境命名空间清单的名称。设置的值与 `<environment_name>` 变量的值相同。

环境命名空间清单示例

```

apiVersion: v1
kind: Namespace
metadata:
  annotations:
    app.openshift.io/vcs-uri: https://example.com/<your_domain>/<your_gitops.git>
    app.openshift.io/vcs-ref: main
  labels:
    argocd.argoproj.io/managed-by: openshift-gitops
  name: dev-env
# ...

```

2.4.2. 检查健康信息

Red Hat OpenShift GitOps Operator 将在 `openshift-gitops` 命名空间中安装 GitOps 后端服务。

先决条件

- Red Hat OpenShift GitOps Operator 从 **OperatorHub** 安装。
- 确保您的应用程序由 Argo CD 同步。

流程

1. 点 **Developer** 视角下的 **Environments**。 **Environments** 页面中显示应用程序列表及其 **环境状态**。
2. 将鼠标悬停在 **Environment status** 列下的图标上，以查看所有环境的同步状态。
3. 点击列表中的应用程序名称查看特定应用程序的详情。
4. 在 **Application environments** 页面中，如果 **Overview** 选项卡下的 **Resources** 部分显示图标，将鼠标悬停在图标上来获取状态详情。
 - 一个分离的心型图标表示资源问题已降低应用程序的性能。
 - 一个黄色的符号表示资源问题中带有应用程序的健康状态的延迟数据。
5. 要查看应用程序的部署历史记录，请点 **Deployment History** 选项卡。该页面包括 **Last deployment**、**Description**（提交消息）、**Environment**、**Author** 和 **Revision** 等详情。

2.5. 监控 ARGO CD 自定义资源工作负载

使用 Red Hat OpenShift GitOps，您可以监控特定 Argo CD 实例的 Argo CD 自定义资源工作负载的可用性。通过监控 Argo CD 自定义资源工作负载，您可以通过为它们启用警报来获取有关 Argo CD 实例状态的最新信息。当相应 Argo CD 实例的组件工作负载 pod（如 application-controller、repo-server 或服务器）无法正常分配，且就绪副本数和所需副本数之间有偏差时，Operator 会触发警报。

您可以启用和禁用用于监控 Argo CD 自定义资源工作负载的设置。

2.5.1. 先决条件

- 您可以使用具有 **cluster-admin** 角色的用户访问集群。
- 您已在 OpenShift Container Platform 集群中安装了 Red Hat OpenShift GitOps Operator。
- 监控堆栈在 **openshift-monitoring** 项目中配置。另外，Argo CD 实例位于您可以通过 Prometheus 监控的命名空间中。
- **kube-state-metrics** 服务在集群中运行。
- 可选：如果您要为用户定义的项目中已存在 Argo CD 实例的监控，请确保 [为集群中的用户定义的项目启用了监控](#)。



注意

如果要为默认 **openshift-monitoring** 堆栈监视的命名空间中的 Argo CD 实例启用监控，例如，任何不是以 **openshiftbang** 开头的命名空间，您必须在集群中启用用户工作负载监控。此操作可让监控堆栈获取创建的 PrometheusRule。

2.5.2. 为 Argo CD 自定义资源工作负载启用监控

默认情况下，Argo CD 自定义资源工作负载的监控配置被设置为 **false**。

在 Red Hat OpenShift GitOps 中，您可以为特定的 Argo CD 实例启用工作负载监控。因此，Operator 会创建一个 **PrometheusRule** 对象，其中包含由特定 Argo CD 实例管理的所有工作负载的警报规则。当相应组件的副本数从特定时间达到所需状态时，这些警报规则会触发警报。Operator 不会覆盖用户对 **PrometheusRule** 对象所做的更改。

流程

1. 在给定的 Argo CD 实例中，将 `.spec.monitoring.enabled` 字段值设置为 `true`：

Argo CD 自定义资源示例

```

apiVersion: argoproj.io/v1beta1
kind: ArgoCD
metadata:
  name: example-argocd
  labels:
    example: repo
spec:
  # ...
  monitoring:
    enabled: true
  # ...

```

2. 验证 Operator 创建的 PrometheusRule 中是否包含警报规则：

警报规则示例

```

apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  name: argocd-component-status-alert
  namespace: openshift-gitops
spec:
  groups:
    - name: ArgoCDComponentStatus
      rules:
        # ...
        - alert: ApplicationSetControllerNotReady 1
          annotations:
            message: >-
              applicationSet controller deployment for Argo CD instance in
              namespace "default" is not running
          expr: >-
            kube_statefulset_status_replicas{statefulset="openshift-gitops-application-controller
            statefulset",
              namespace="openshift-gitops"} !=
            kube_statefulset_status_replicas_ready{statefulset="openshift-gitops-application-
            controller statefulset",
              namespace="openshift-gitops"}
          for: 1m
          labels:
            severity: critical

```

- 1** PrometheusRule 中的警报规则，用于检查 Argo CD 实例创建的工作负载是否如预期运行。

2.5.3. 禁用 Argo CD 自定义资源工作负载的监控

您可以为特定的 Argo CD 实例禁用工作负载监控。禁用工作负载监控会删除创建的 PrometheusRule。

流程

- 在给定的 Argo CD 实例中，将 `.spec.monitoring.enabled` 字段值设置为 `false` :

Argo CD 自定义资源示例

```
apiVersion: argoproj.io/v1beta1
kind: ArgoCD
metadata:
  name: example-argocd
  labels:
    example: repo
spec:
  # ...
  monitoring:
    enabled: false
  # ...
```

2.5.4. 其他资源

- [为用户定义的项目启用监控](#)