



# Red Hat OpenShift GitOps 1.12

## 基础架构节点上的 GitOps 工作负载

在基础架构节点上运行 GitOps control plane 工作负载



## Red Hat OpenShift GitOps 1.12 基础架构节点上的 GitOps 工作负载

---

在基础架构节点上运行 GitOps control plane 工作负载

## 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

本文档提供在 OpenShift GitOps 安装的基础架构节点上运行某些工作负载的说明。它还讨论如何将默认工作负载移到基础架构节点。

---

## 目录

<b>第 1 章 在基础架构节点上运行 GITOPS CONTROL PLANE 工作负载 .....</b>	<b>3</b>
1.1. 将 GITOPS CONTROL PLANE 工作负载移到基础架构节点	3
1.2. 将 GITOPS OPERATOR POD 移到基础架构节点	4
1.3. 其他资源	6



# 第 1 章 在基础架构节点上运行 GITOPS CONTROL PLANE 工作负载

对于两个主要目的，您可以使用基础架构节点隔离基础架构工作负载：

- 要防止与订阅数相关的计费成本
- 单独的维护和管理

您可以使用 OpenShift Container Platform 在基础架构节点上运行 GitOps control plane 工作负载。默认情况下，这包括 Operator pod 和由 **openshift-gitops** 命名空间中的 Red Hat OpenShift GitOps Operator 创建的 control plane 工作负载，包括此命名空间中的默认 Argo CD 实例。

使用 GitOps control plane 工作负载，您可以通过在集群中创建多个隔离的 Argo CD 实例来安全地并声明性地隔离基础架构工作负载，并完全控制 Argo CD 实例的功能。另外，您可以在多个开发人员命名空间中以声明性方式管理这些 Argo CD 实例。通过使用污点，您可以确保只有基础架构组件在这些节点上运行。



## 注意

在用户命名空间中安装的所有其他 Argo CD 实例都无权在基础架构节点上运行。

## 1.1. 将 GITOPS CONTROL PLANE 工作负载移到基础架构节点

您可以将 Red Hat OpenShift GitOps 安装的 GitOps control plane 工作负载移到基础架构节点。以下是您可以移动的 control plane 工作负载：

- **kam deployment**
- **集群部署**（后端服务）
- **openshift-gitops-applicationset-controller 部署**
- **openshift-gitops-dex-server 部署**
- **openshift-gitops-redis 部署**
- **openshift-gitops-redis-ha-haproxy 部署**
- **openshift-gitops-repo-sever 部署**
- **openshift-gitops-server 部署**
- **openshift-gitops-application-controller statefulset**
- **openshift-gitops-redis-server statefulset**

### 流程

1. 运行以下命令，将现有节点标记为基础架构：

```
$ oc label node <node-name> node-role.kubernetes.io/infra=
```

2. 编辑 **GitOpsService** 自定义资源(CR)以添加基础架构节点选择器：

■

```
$ oc edit gitopsservice -n openshift-gitops
```

- 在 **GitOpsService** CR 文件中，将 **runOnInfra** 字段添加到 **spec** 部分，并将其设置为 **true**。此字段将 **openshift-gitops** 命名空间中的 control plane 工作负载移到基础架构节点：

```
apiVersion: pipelines.openshift.io/v1alpha1
kind: GitopsService
metadata:
  name: cluster
spec:
  runOnInfra: true
```

- 可选：在基础架构节点上应用污点并隔离工作负载，并防止其他工作负载调度到这些节点上。

```
$ oc adm taint nodes -l node-role.kubernetes.io/infra
infra=reserved:NoSchedule infra=reserved:NoExecute
```

- 可选：如果您将污点应用到节点，您可以在 **GitOpsService** CR 中添加容限：

```
spec:
  runOnInfra: true
  tolerations:
  - effect: NoSchedule
    key: infra
    value: reserved
  - effect: NoExecute
    key: infra
    value: reserved
```

要验证工作负载是否已调度到 Red Hat OpenShift GitOps 命名空间中的基础架构节点上，请点击任何 pod 名称，并确保已添加了 **Node selector** 和 **Tolerations**。



### 注意

在默认 Argo CD CR 中手动添加节点选择器和 Tolerations 都会被 **GitOpsService** CR 中的切换和容限覆盖。

## 1.2. 将 GITOPS OPERATOR POD 移到基础架构节点

您可以将 GitOps Operator pod 移到基础架构节点。

### 先决条件

- 您已在 OpenShift Container Platform 集群中安装了 Red Hat OpenShift GitOps Operator。
- 您可以使用 **cluster-admin** 权限访问集群。

### 流程

- 运行以下命令，将现有节点标记为基础架构节点：

```
$ oc label node <node_name> node-role.kubernetes.io/infra= 1
```



- 1 将 `<node_name>` 替换为您要标记为基础架构节点的节点的名称。

### 输出示例

```
node/<node_name> labeled
```

2. 运行以下命令来编辑 Red Hat OpenShift GitOps **Subscription** 资源：

```
$ oc -n openshift-gitops-operator edit subscription openshift-gitops-operator
```

3. 在 **Subscription** 资源的 `spec.config` 字段中添加 `nodeSelector` 和 `tolerations`：

### 订阅示例

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: openshift-gitops-operator
  namespace: openshift-gitops-operator
spec:
  config:
    nodeSelector: 1
    node-role.kubernetes.io/infra: ""
    tolerations: 2
    - key: node-role.kubernetes.io/infra
      operator: Exists
      effect: NoSchedule
```

- 1 这样可确保 Operator pod 仅调度到基础架构节点上。

- 2 这样可确保基础架构节点接受 pod。

### 输出示例

```
subscription.operators.coreos.com/openshift-gitops-operator edited
```

4. 运行以下命令，验证 GitOps Operator pod 是否在基础架构节点上运行：

```
$ oc -n openshift-gitops-operator get po -owide
```

### 输出示例

```
NAME                                READY STATUS RESTARTS AGE IP
NODE      NOMINATED NODE  READINESS GATES
openshift-gitops-operator-controller-manager-abcd 2/2 Running 0 11m
94.142.44.126 <node_name> <none> <none> 1
```

- 1 确保列出的 `<node_name>` 是带有 `node-role.kubernetes.io/infra` 标签的节点。

### 1.3. 其他资源

- 如需有关污点和容限的更多信息，[请参阅使用节点污点控制 pod 放置。](#)
- 有关基础架构机器集的更多信息，[请参阅 创建基础架构机器集。](#)