



# Red Hat OpenShift GitOps 1.12

## GitOps workloads on infrastructure nodes

Running GitOps control plane workloads on infrastructure nodes



## Red Hat OpenShift GitOps 1.12 GitOps workloads on infrastructure nodes

---

Running GitOps control plane workloads on infrastructure nodes

## 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<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.

## Abstract

This document provides instructions for running certain workloads on infrastructure nodes that are installed by OpenShift GitOps. It also discusses how to move the default workloads to the infrastructure nodes.

## Table of Contents

<b>CHAPTER 1. RUNNING GITOPS CONTROL PLANE WORKLOADS ON INFRASTRUCTURE NODES</b>	<b>.....</b>	<b>3</b>
1.1. MOVING GITOPS CONTROL PLANE WORKLOADS TO INFRASTRUCTURE NODES		3
1.2. MOVING THE GITOPS OPERATOR POD TO INFRASTRUCTURE NODES		4
1.3. ADDITIONAL RESOURCES		6



# CHAPTER 1. RUNNING GITOPS CONTROL PLANE WORKLOADS ON INFRASTRUCTURE NODES

You can use infrastructure nodes to isolate infrastructure workloads for two primary purposes:

- To prevent billing costs associated with the number of subscriptions
- To separate maintenance and management

You can use the OpenShift Container Platform to run GitOps control plane workloads on infrastructure nodes. This includes the Operator pod and the control plane workloads created by the Red Hat OpenShift GitOps Operator in the **openshift-gitops** namespace by default, including the default Argo CD instance in this namespace.

With GitOps control plane workloads, you can securely and declaratively isolate the infrastructure workloads by creating multiple isolated Argo CD instances in a cluster, with full control over what an Argo CD instance is capable of. In addition, you can manage these Argo CD instances declaratively across multiple developer namespaces. By using taints, you can ensure that only infrastructure components run on these nodes.



## NOTE

All other Argo CD instances installed in user namespaces are not eligible to run on infrastructure nodes.

## 1.1. MOVING GITOPS CONTROL PLANE WORKLOADS TO INFRASTRUCTURE NODES

You can move the GitOps control plane workloads installed by the Red Hat OpenShift GitOps to the infrastructure nodes. The following are the control plane workloads that you can move:

- **kam deployment**
- **cluster deployment** (backend service)
- **openshift-gitops-applicationset-controller deployment**
- **openshift-gitops-dex-server deployment**
- **openshift-gitops-redis deployment**
- **openshift-gitops-redis-ha-haproxy deployment**
- **openshift-gitops-repo-sever deployment**
- **openshift-gitops-server deployment**
- **openshift-gitops-application-controller statefulset**
- **openshift-gitops-redis-server statefulset**

### Procedure

1. Label existing nodes as infrastructure by running the following command:

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

2. Edit the **GitOpsService** custom resource (CR) to add the infrastructure node selector:

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

3. In the **GitOpsService** CR file, add **runOnInfra** field to the **spec** section and set it to **true**. This field moves the control plane workloads in **openshift-gitops** namespace to the infrastructure nodes:

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

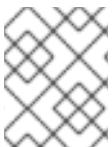
4. Optional: Apply taints and isolate the workloads on infrastructure nodes and prevent other workloads from scheduling on these nodes.

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

5. Optional: If you apply taints to the nodes, you can add tolerations in the **GitOpsService** CR:

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

To verify that the workloads are scheduled on infrastructure nodes in the Red Hat OpenShift GitOps namespace, click any of the pod names and ensure that the **Node selector** and **Tolerations** have been added.



#### NOTE

Any manually added **Node selectors** and **Tolerations** in the default Argo CD CR will be overwritten by the toggle and the tolerations in the **GitOpsService** CR.

## 1.2. MOVING THE GITOPS OPERATOR POD TO INFRASTRUCTURE NODES

You can move the GitOps Operator pod to the infrastructure nodes.

### Prerequisites



- You have installed the Red Hat OpenShift GitOps Operator on your OpenShift Container Platform cluster.
- You have access to the cluster with **cluster-admin** privileges.

## Procedure

1. Label an existing node as infrastructure node by running the following command:

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

- 1 Replace <node\_name> with the name of the node you want to label as infrastructure node.

### Example output

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

2. Edit the Red Hat OpenShift GitOps **Subscription** resource by running the following command:

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

3. Add **nodeSelector** and **tolerations** to the **spec.config** field in the **Subscription** resource:

### Example Subscription

```
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 This ensures that the operator pod is only scheduled on an infrastructure node.

- 2 This ensures that the pod is accepted by the infrastructure node.

### Example output

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

4. Verify that the GitOps Operator pod is running on the infrastructure node by running the following command:

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

## Example output

```
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 Ensure that the listed **<node\_name>** is the node with the **node-role.kubernetes.io/infra** label.

## 1.3. ADDITIONAL RESOURCES

- For more information about taints and tolerations, see [Controlling pod placement using node taints](#).
- For more information about infrastructure machine sets, see [Creating infrastructure machine sets](#).