Red Hat OpenShift GitOps 1.11

GitOps workloads on infrastructure nodes

Running GitOps control plane workloads on infrastructure nodes
Running GitOps control plane workloads on infrastructure nodes
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

CHAPTER 1. RUNNING GITOPS CONTROL PLANE WORKLOADS ON INFRASTRUCTURE NODES  .........  3  
1.1. MOVING GITOPS WORKLOADS TO INFRASTRUCTURE NODES  3  
1.2. ADDITIONAL RESOURCES  4
You can use infrastructure nodes to prevent additional billing cost against subscription counts.

You can use the OpenShift Container Platform to run certain workloads on infrastructure nodes installed by the Red Hat OpenShift GitOps Operator. This comprises the workloads that are installed by the Red Hat OpenShift GitOps Operator by default in the `openshift-gitops` namespace, including the default Argo CD instance in that namespace.

**NOTE**

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

### 1.1. MOVING GITOPS WORKLOADS TO INFRASTRUCTURE NODES

You can move the default workloads installed by the Red Hat OpenShift GitOps to the infrastructure nodes. The workloads that can be moved are:

- `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:

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

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

   ```bash
   $ 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 workloads in `openshift-gitops` namespace to the infrastructure nodes:

   ```yaml
   apiVersion: pipelines.openshift.io/v1alpha1
   ```
4. Optional: Apply taints and isolate the workloads on infrastructure nodes and prevent other workloads from scheduling on these nodes.

```bash
$ 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:

```yaml
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. ADDITIONAL RESOURCES

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