



# Red Hat OpenShift GitOps 1.12

## Argo CD applications

Creating and deploying applications on the OpenShift cluster by using the Argo CD dashboard, oc tool, or GitOps CLI.



## Red Hat OpenShift GitOps 1.12 Argo CD applications

---

Creating and deploying applications on the OpenShift cluster by using the Argo CD dashboard, oc tool, or GitOps CLI.

## 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 creating and deploying your applications to the OpenShift cluster by using the Argo CD dashboard, oc tool, or GitOps CLI. It also discusses how to verify the self-healing behavior in Argo CD.

---

## Table of Contents

<b>CHAPTER 1. DEPLOYING A SPRING BOOT APPLICATION WITH ARGO CD .....</b>	<b>3</b>
1.1. CREATING AN APPLICATION BY USING THE ARGO CD DASHBOARD	3
1.2. CREATING AN APPLICATION BY USING THE OC TOOL	4
1.3. VERIFYING ARGO CD SELF-HEALING BEHAVIOR	4
<b>CHAPTER 2. CREATING AN APPLICATION BY USING THE GITOPS CLI .....</b>	<b>6</b>
2.1. CREATING AN APPLICATION IN THE DEFAULT MODE BY USING THE GITOPS CLI	6
2.2. CREATING AN APPLICATION IN CORE MODE BY USING THE GITOPS CLI	7
2.3. ADDITIONAL RESOURCES	9



# CHAPTER 1. DEPLOYING A SPRING BOOT APPLICATION WITH ARGO CD

With Argo CD, you can deploy your applications to the OpenShift Container Platform cluster either by using the Argo CD dashboard or by using the **oc** tool.

## 1.1. CREATING AN APPLICATION BY USING THE ARGO CD DASHBOARD

Argo CD provides a dashboard which allows you to create applications.

### Prerequisites

- You have logged in to the OpenShift Container Platform cluster as an administrator.
- You have installed the Red Hat OpenShift GitOps Operator on your OpenShift Container Platform cluster.
- You have logged in to Argo CD instance.

### Procedure

1. In the Argo CD dashboard, click **NEW APP** to add a new Argo CD application.
2. For this workflow, create a **spring-petclinic** application with the following configurations:

**Application Name**

**spring-petclinic**

**Project**

**default**

**Sync Policy**

**Automatic**

**Repository URL**

<https://github.com/redhat-developer/openshift-gitops-getting-started>

**Revision**

**HEAD**

**Path**

**app**

**Destination**

<https://kubernetes.default.svc>

**Namespace**

**spring-petclinic**

3. Click **CREATE** to create your application.
4. Open the **Administrator** perspective of the web console and expand **Administration** → **Namespaces**.

5. Search for and select the namespace, then enter **argocd.argoproj.io/managed-by=openshift-gitops** in the **Label** field so that the Argo CD instance in the **openshift-gitops** namespace can manage your namespace.

## 1.2. CREATING AN APPLICATION BY USING THE **oc** TOOL

You can create Argo CD applications in your terminal by using the **oc** tool.

### Prerequisites

- You have installed the Red Hat OpenShift GitOps Operator on your OpenShift Container Platform cluster.
- You have logged in to an Argo CD instance.

### Procedure

1. Download [the sample application](#):

```
$ git clone git@github.com:redhat-developer/openshift-gitops-getting-started.git
```

2. Create the application:

```
$ oc create -f openshift-gitops-getting-started/argo/app.yaml
```

3. Run the **oc get** command to review the created application:

```
$ oc get application -n openshift-gitops
```

4. Add a label to the namespace your application is deployed in so that the Argo CD instance in the **openshift-gitops** namespace can manage it:

```
$ oc label namespace spring-petclinic argocd.argoproj.io/managed-by=openshift-gitops
```

## 1.3. VERIFYING ARGO CD SELF-HEALING BEHAVIOR

Argo CD constantly monitors the state of deployed applications, detects differences between the specified manifests in Git and live changes in the cluster, and then automatically corrects them. This behavior is referred to as self-healing.

You can test and observe the self-healing behavior in Argo CD.

### Prerequisites

- You have installed the Red Hat OpenShift GitOps Operator on your OpenShift Container Platform cluster.
- You have logged in to an Argo CD instance.
- The sample **app-spring-petclinic** application is deployed and configured.

### Procedure



1. In the Argo CD dashboard, verify that your application has the **Synced** status.
2. Click the **app-spring-petclinic** tile in the Argo CD dashboard to view the application resources that are deployed to the cluster.
3. In the OpenShift Container Platform web console, navigate to the **Developer** perspective.
4. Modify the Spring PetClinic deployment and commit the changes to the **app/** directory of the Git repository. Argo CD will automatically deploy the changes to the cluster.
  - a. Fork the [OpenShift GitOps getting started repository](#).
  - b. In the **deployment.yaml** file, change the **failureThreshold** value to **5**.
  - c. In the deployment cluster, run the following command to verify the changed value of the **failureThreshold** field:

```
$ oc edit deployment spring-petclinic -n spring-petclinic
```

- a. Run the following command to modify the deployment:
- ```
$ oc scale deployment spring-petclinic --replicas 2 -n spring-petclinic
```
- b. In the OpenShift Container Platform web console, notice that the deployment scales up to two pods and immediately scales down again to one pod. Argo CD detected a difference from the Git repository and auto-healed the application on the OpenShift Container Platform cluster.
6. In the Argo CD dashboard, click the **app-spring-petclinic** tile → **APP DETAILS** → **EVENTS**. The **EVENTS** tab displays the following events: Argo CD detecting out of sync deployment resources on the cluster and then resyncing the Git repository to correct it.

## CHAPTER 2. CREATING AN APPLICATION BY USING THE GITOPS CLI

With Argo CD, you can create your applications on an OpenShift Container Platform cluster by using the GitOps **argocd** CLI.

### 2.1. CREATING AN APPLICATION IN THE DEFAULT MODE BY USING THE GITOPS CLI

You can create applications in the default mode by using the GitOps **argocd** CLI.

#### Prerequisites

- You have installed the Red Hat OpenShift GitOps Operator on your OpenShift Container Platform cluster.
- You have installed the OpenShift CLI (**oc**).
- You have installed the Red Hat OpenShift GitOps **argocd** CLI.
- You have logged in to Argo CD instance.

#### Procedure

1. Get the **admin** account password for the Argo CD server:

```
$ ADMIN_PASSWD=$(oc get secret openshift-gitops-cluster -n openshift-gitops -o
jsonpath='{.data.admin\.password}' | base64 -d)
```

2. Get the Argo CD server URL:

```
$ SERVER_URL=$(oc get routes openshift-gitops-server -n openshift-gitops -o
jsonpath='{.status.ingress[0].host}')
```

3. Log in to the Argo CD server by using the **admin** account password and enclosing it in single quotes:



#### IMPORTANT

Enclosing the password in single quotes ensures that special characters, such as **\$**, are not misinterpreted by the shell. Always use single quotes to enclose the literal value of the password.

```
$ argocd login --username admin --password ${ADMIN_PASSWD} ${SERVER_URL}
```

#### Example

```
$ argocd login --username admin --password '<password>' openshift-gitops.openshift-
gitops.apps-crc.testing
```

4. Verify that you are able to run **argocd** commands in the default mode by listing all applications:

```
$ argocd app list
```

If the configuration is correct, then existing applications will be listed with the following header:

### Sample output

```
NAME CLUSTER NAMESPACE PROJECT STATUS HEALTH SYNCPOLICY
CONDITIONS REPO PATH TARGET
```

5. Create an application in the default mode:

```
$ argocd app create app-spring-petclinic \
  --repo https://github.com/redhat-developer/openshift-gitops-getting-started.git \
  --path app \
  --revision main \
  --dest-server https://kubernetes.default.svc \
  --dest-namespace spring-petclinic \
  --directory-recurse \
  --sync-policy automated \
  --self-heal \
  --sync-option Prune=true \
  --sync-option CreateNamespace=true
```

6. Label the **spring-petclinic** destination namespace to be managed by the **openshift-gitops** Argo CD instance:

```
$ oc label ns spring-petclinic "argocd.argoproj.io/managed-by=openshift-gitops"
```

7. List the available applications to confirm that the application is created successfully and repeat the command until the application has the **Healthy** and **Synced** statuses:

```
$ argocd app list
```

## 2.2. CREATING AN APPLICATION IN CORE MODE BY USING THE GITOPS CLI

You can create applications in **core** mode by using the GitOps **argocd** CLI.

### Prerequisites

- You have installed the Red Hat OpenShift GitOps Operator on your OpenShift Container Platform cluster.
- You have installed the OpenShift CLI (**oc**).
- You have installed the Red Hat OpenShift GitOps **argocd** CLI.

### Procedure

1. Log in to the OpenShift Container Platform cluster by using the **oc** CLI tool:

```
$ oc login -u <username> -p <password> <server_url>
```

## Example

```
$ oc login -u kubeadmin -p '<password>' https://api.crc.testing:6443
```

2. Check whether the context is set correctly in the **kubeconfig** file:

```
$ oc config current-context
```

3. Set the default namespace of the current context to **openshift-gitops**:

```
$ oc config set-context --current --namespace openshift-gitops
```

4. Set the following environment variable to override the Argo CD component names:

```
$ export ARGOCD_REPO_SERVER_NAME=openshift-gitops-repo-server
```

5. Verify that you are able to run **argocd** commands in **core** mode by listing all applications:

```
$ argocd app list --core
```

If the configuration is correct, then existing applications will be listed with the following header:

## Sample output

```
NAME CLUSTER NAMESPACE PROJECT STATUS HEALTH SYNCPOLICY
CONDITIONS REPO PATH TARGET
```

6. Create an application in **core** mode:

```
$ argocd app create app-spring-petclinic --core \
  --repo https://github.com/redhat-developer/openshift-gitops-getting-started.git \
  --path app \
  --revision main \
  --dest-server https://kubernetes.default.svc \
  --dest-namespace spring-petclinic \
  --directory-recurse \
  --sync-policy automated \
  --self-heal \
  --sync-option Prune=true \
  --sync-option CreateNamespace=true
```

7. Label the **spring-petclinic** destination namespace to be managed by the **openshift-gitops** Argo CD instance:

```
$ oc label ns spring-petclinic "argocd.argoproj.io/managed-by=openshift-gitops"
```

8. List the available applications to confirm that the application is created successfully and repeat the command until the application has the **Healthy** and **Synced** statuses:

```
$ argocd app list --core
```

## 2.3. ADDITIONAL RESOURCES

- [Installing the GitOps CLI](#)
- [Basic GitOps argocd commands](#)