

## Configuring NiFi Registry CR

Date published: 2024-06-11

Date modified: 2024-06-11



# Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

# Contents

<b>Configuring a NiFi Registry cluster.....</b>	<b>4</b>
<b>Configuring group, version, kind, and meta.....</b>	<b>4</b>
<b>Configuring images.....</b>	<b>4</b>
<b>Configuring persistence.....</b>	<b>5</b>
<b>Configuring LDAP authentication.....</b>	<b>5</b>
Generating a node certificate.....	6
<b>Example CR.....</b>	<b>7</b>

## Configuring a NiFi Registry cluster

Cloudera Flow Management - Kubernetes Operator can deploy NiFi Registry instances using the NiFiRegistry custom resource. NiFi Registry instances are configured through these CRs. No additional configuration is required after deployment.

A custom resource (CR) is a YAML file that describes your desired NiFi Registry deployments. This single file contains all configuration information required for the NiFi Registry instance, no additional configuration is required after deployment.

This documentation provides sample configuration code snippets that help you create a CR.

## Configuring group, version, kind, and meta

This is the initial section of your YAML file that you need to specify in all cases.

You need to add the following section on top of each NiFi Registry custom resource (CR) you write. It defines the group “cfm.cloudera.com”, the version “v1alpha1”, the kind “NiFiRegistry”, and the name of your cluster and the nodes. It can also specify the namespace in which resources will be deployed. It is expected that a single NiFi cluster is deployed in a given namespace. You can also specify namespace during deployment, if that is what you want, omit namespace from the CR.

```
apiVersion: cfm.cloudera.com/v1alpha
kind: NiFiRegistry
metadata:
  name: [***NIFI_REGISTRY_NAME***]
```

## Configuring images

This section describes the images used for running NiFi Registry. This provides a way of manually upgrading the NiFi version in an existing cluster or very quickly rolling out NiFi clusters with new versions.

A CFM NiFi Registry deployment includes two container images: cfm-nifiregistry-k8s and cfm-tini. The cfm-nifiregistry-k8s image is the actual registry image itself. The cfm-tini image is a small utility image used for aggregating logs.

Pulling images from Cloudera’s registries requires a pull secret containing your Cloudera credentials. Create this pull secret with

```
kubectl create secret docker-registry my-pull-secret \
--docker-username=[***CLOUDERA_USER***] \
--docker-password=[***CLOUDERA_PASSWORD***] \
--docker-server=container.repository.cloudera.com
```

```
spec:
  image:
    repository: [***CFM-NIFI-REGISTRY-K8S_REPOSITORY***]
    tag: 2.8.0-bXX
    imagePullPolicy: IfNotPresent
    pullSecret: my-pull-secret
  tiniImage:
    Repository: [***CFM-TINI_REPOSITORY***]
    tag: 2.8.0-bXX
    imagePullPolicy: IfNotPresent
```

```
pullSecret: [***PULL SECRET***]
```

- The default [\*\*\*CFM-NIFI-REGISTRY-K8S REPOSITORY\*\*\*] is container.repository.cloudera.com/cloudera/cfm-nifi-registry-k8s
- The default [\*\*\*CFM-TINI REPOSITORY\*\*\*] is container.repository.cloudera.com/cloudera/cfm-tini

if your Kubernetes cluster has no internet connection or you want to use a self-hosted repository, replace these with the relevant paths.

## Configuring persistence

Learn about configuring storage for NiFi Registry.

Cloudera Flow Management - Kubernetes Operator can configure persistent volumes for the following directories:

- flow\_storage
- data
- extension\_bundles

In the persistence spec, a default size and StorageClass can be defined which applies to each of the directories. The spec can be further configured to define specific sizes and StorageClasses for each directory if desired.

```
spec:
  persistence:
    size: 1Gi
    storageClass: default
    flowStorage:
      size: 3Gi
    data: {}
    extensionBundles:
      storageClass: some-storage-class
```

## Configuring LDAP authentication

Learn how to configure an LDAP server for user authentication in your NiFi or NiFi Registry cluster.

Cloudera Flow Management - Kubernetes Operator can configure NiFi to connect to an LDAP server for user authentication.

Prerequisites:

- Full LDAP URL, i.e. ldap://[\*\*\*LDAP SERVER URL\*\*\*]:[\*\*\*LDAP PORT\*\*\*]
- Desired authentication strategy
- Authentication credentials and key/trust stores if using LDAPS.
- User search filters

For LDAP servers protected with any authentication, a Secret must be created containing the correct authentication credentials and TLS resources (if applicable). The Secret must contain the following data fields:

- managerPassword
- keystore (if TLS is configured)
- keystorePassword (if TLS is configured)
- truststore (if TLS is configured)
- truststorePassword (if TLS is configured)

Create the secret using the kubect1 CLI utility:

```
kubect1 create secret generic my-ldap-creds \
  --from-literal=managerPassword=myMan@gerPassw0rd \
  --from-file=keystore=/path/to/keystore \
  --from-literal=keystorePassword=myKeystorePassword \
  --from-file=truststore=/path/to/truststore \
  --from-literal=truststorePassword=myTruststorePassword
```

The following example shows a connection to an LDAP server protected with basic authentication with TLS.

```
spec:
  security:
    initialAdminIdentity: mynifiadmin
    ldap:
      authenticationStrategy: SIMPLE
      managerDN: "cn=admin,dc=example,dc=org"
      secretName: my-openldap-creds
      referralStrategy: FOLLOW
      connectTimeout: 3 secs
      readTimeout: 10 secs
      url: ldap://my-ldap-url:389
      userSearchBase: "dc=example,dc=org"
      userSearchFilter: "(uid={0})"
      identityStrategy: USE_USERNAME
      authenticationExpiration: 12 hours
  tls:
    keystoreType: jks
    truststoreType: jks
    clientAuth: NONE
    protocol: TLSv1.2
```

By default, Cloudera Flow Management - Kubernetes Operator does not deploy a UserGroupProvider using the LDAP target. This means NiFi does not pull down any users, only queries the LDAP server for authentication. This impedes configuring user access, requiring the NiFi administrator to create each user manually.

The following example shows configuring user synchronization with the LDAP server:

```
spec:
  security:
    ldap:
      sync:
        interval: 30 min
        userObjectClass: inetOrgPerson
        userSearchScope: SUBTREE
        userIdentityAttribute: cn
        userGroupNameAttribute: ou
        userGroupNameReferencedGroupAttribute: ou
        groupSearchBase: "dc=example,dc=org"
        groupObjectClass: organizationalUnit
        groupSearchScope: OBJECT
        groupNameAttribute: ou
```

## Generating a node certificate

Node certificate generation is a required configuration if LDAP authentication is used. The Cloudera Flow Management - Kubernetes Operator provides automatic certificate generation for the NiFi Registry by way of cert-manager Certificates to provide TLS protection to the WebUI as well as between the Registry and NiFis in the cluster.

To configure NodeCertGen, a cert-manager [Issuer](#) or [ClusterIssuer](#) is required. A self-signed Issuer setup is good for development environments, while production environments should use a third-party authority or internal signing CAs.

```
spec:
  security:
    nodeCertGen:
      issuerRef:
        name: self-signed-ca-issuer
        kind: ClusterIssuer
```

### Related Information

[Issuers and ClusterIssuers](#)

## Example CR

This custom resource example configures a basic NiFi Registry instance with a single replica, no security, and a Route to connect to the UI.

```
apiVersion: cfm.cloudera.com/v1alpha1
kind: NifiRegistry
metadata:
  name: mynifiregistry
spec:
  image:
    repository: container.repository.cloudera.com/cloudera/cfm-nifi-k8s
    tag: [***NIFI REGISTRY TAG***]
  tiniImage:
    repository: container.repository.cloudera.com/cloudera/cfm-tini
    tag: [***CFM TINI TAG***]
  hostname: mynifiregistry.[***OPENSIFT ROUTER DOMAIN***]
  uiConnection:
    type: Route
```