Cloudera Flow Management Operator 2.8.0

Configuring NiFi Registry CR

Date published: 2024-06-11
Date modified: 2024-06-11

CLOUDERA

https://docs.cloudera.com/

Legal Notice

Contents

Configuring a NiFi Registry cluster

CFM Operator can deploy NiFi Registry instances using the NiFiRegistry CR. 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

Group, Version, Kind, 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 to the top of each NiFi Registry CR you write. It defines the group "cfm.clou dera.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/c
  fm-nifiregistry-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

CFM 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

CFM Operator can configure NiFi Registry to connect to an LDAP server for user authentication.

Configuring LDAP authentication for NiFi and NiFi Registry are identical procedures. For mode information, see Configuring LDAP under *Configuring a NiFi cluster*.

Generating a node cerificate

Node certificate generation is a required configuration if LDAP authentication is used. The CFM 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

The following 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.[***OPENSHIFT ROUTER DOMAIN***]
    uiConnection:
        type: Route
```