Cloudera Flow Management Operator for Kubernetes 2.11.0

# NiFi Deployment

**Date published: 2024-06-11**
**Date modified: 2025-09-30**

**https://docs.cloudera.com/**

# Legal Notice

# Contents

# Deploying a NiFi instance in a Kubernetes cluser

With Cloudera Flow Management Operator for Kubernetes you can deploy NiFi instances to your Kubernetes cluster.

## About this task

You can deploy a NiFi cluster by creating a NiFi custom resource (CR) and deploying it to the Kubernetes cluster.

## Before you begin

- Ensure the Cloudera Flow Management Operator for Kubernetes has been installed and is running.
- You have created a NiFi CR YAML file that complies with the documentation provided by Cloudera.

## Procedure

1. Create a namespace for the NiFi if it does not already exist.

```
$ kubectl create namespace my-nifi
```

2. In *[***NIFI CLUSTER NAMESPACE***]*, create an image pull secret to access the installation artifacts.

```
kubectl create secret docker-registry [***SECRET NAME***] \
  --namespace [***NIFI CLUSTER NAMESPACE***] \
  --docker-server container.repository.cloudera.com \
  --docker-username [***USERNAME***] \
  --docker-password [***PASSWORD***]
```

   Replace:

   - *[***SECRET NAME***]* with the desired Kubernetes secret name.
   - *[***NIFI CLUSTER NAMESPACE***]* with the namespace you created.
   - *[***USERNAME***]* and *[***PASSWORD***]* with your Cloudera credentials.

   For example:

```
kubectl create secret docker-registry docker-pull-secret \
  --namespace cfm-operator-system \
  --docker-server container.repository.cloudera.com \
  --docker-username my-username \
  --docker-password my-password
```

3. Deploy NiFi to the Kubernetes cluster.

```
kubectl apply -f [***CR YAML PATH***] --namespace [***NIFI CLUSTER
 NAMESPACE***]
```

   Replace:

   - *[***CR YAML PATH***]* with the absolute or relative path to the CR YAML file you created for NiFi.
   - *[***NIFI CLUSTER NAMESPACE***]* with the namespace you created to deploy NiFi.

## What to do next

**Note:** The deployment process creates a default ‘cfm-operator.cfm-operator-system.svc’ user to give Cloudera Flow Management Operator for Kubernetes the permissions it needs to manage the NiFi cluster. Do not remove this user or any of its permissions, as it will render the operator unable to manage that NiFi Cluster.

### Related Information

Configuring a NiFi CR
NiFi CR example

# Deploying a NiFi instance in Kubernetes (air-gap)

With Cloudera Flow Management Operator for Kubernetes you can deploy NiFi clusters to your Kubernetes cluster. Complete these steps if your Kubernetes cluster does not have internet access, or if you want to install it from a self-hosted registry.

## About this task

You can deploy a NiFi cluster by creating a NiFi custom resource (CR) and deploying it to the Kubernetes cluster.

## Before you begin

- Ensure the Cloudera Flow Management Operator for Kubernetes has been installed and is running.
- A self-hosted Docker registry is required. Your registry must be accessible by your Kubernetes cluster.
- A machine with Internet connectivity is required. While the Kubernetes cluster does not need internet access, you will need a machine to pull the images from the Cloudera Docker registry.
- Access to docker or equivalent utility that you can use to pull and push images is required. The following steps use docker. Replace commands where necessary.
- Ensure that you have access to your Cloudera credentials (username and password). Credentials are required to access the Cloudera Archive and Cloudera Docker registry where installation artifacts are hosted.
- Ensure that you have access to a valid Cloudera license.
- Review the Helm chart reference before installation.

  The Helm chart accepts various configuration properties that you can set during installation. Using these properties you can customize your installation.
- You have created a NiFi CR YAML file that complies with the documentation provided by Cloudera.
- Obtain the Apache NiFi Docker image that is required for your installation scenario.

| Artifact | Location |
|---|---|
| Apache NiFi Docker images | container.repository.cloudera.com/cloudera/cfm-nifi-k8s:1.28.1.2.3.17.0-9 |
| | container.repository.cloudera.com/cloudera/cfm-nifi-k8s:2.6.0.4.3.4.0-234 |

## Procedure

1. Create a namespace for the NiFi if it does not already exist.

   ```
   kubectl create namespace [***NIFI CLUSTER NAMESPACE***]
   ```

   Replace *[***NIFI CLUSTER NAMESPACE***]* with the desired namespace for NiFi.

   ```
   $ kubectl create namespace my-nifi
   ```

2. Create a Kubernetes secret containing your Cloudera credentials.

   ```
   kubectl create secret docker-registry [***SECRET NAME***] \
     --namespace [***NIFI CLUSTER NAMESPACE***] \
     --docker-server [***CONTAINER REGISTRY***] \
     --docker-username [***USERNAME***] \
     --docker-password [***PASSWORD***]
   ```

Replace:

- *[***SECRET NAME***]* with the desired Kubernetes secret name.
- *[***USERNAME***]* and *[***PASSWORD***]* with your internal registry credentials.
- *[***NIFI CLUSTER NAMESPACE***]* with the Cloudera Flow Management Operator for Kubernetes installation namespace.
- *[***CONTAINER REGISTRY***]* with your internal registry URL.

3. Move the installation artifacts to a local registry using the `docker pull`, `docker tag`, and `docker push` commands.

```
docker pull container.repository.cloudera.com/cloudera/cfm-nifi-
k8s:[***CFM OPERATOR NIFI VERSION***] \
docker tag container.repository.cloudera.com/cloudera/cfm-nifi-k8s:[***CFM
 OPERATOR NIFI VERSION***] [***PRIVATE REGISTRY[:PORT]/PATH/TAG:CFM
 OPERATOR NIFI VERSION***] \
docker push [***PATH TO SELF-HOSTED REGISTRY***]/cfm-nifi-k8s:[***CFM
 OPERATOR NIFI VERSION***]
```

For example:

```
docker pull container.repository.cloudera.com/cloudera/cfm-nifi-
k8s:1.28.1.2.3.17.0-9 \
docker tag container.repository.cloudera.com/cloudera/cfm-nifi-k8
s:1.28.1.2.3.17.0-9 us-central1-docker.pkg.dev/nifi-testing/cfm-k8s/cfm-
nifi-k8s:3.0.0-b126-nifi_1.28.1.2.3.17.0-9 \
docker push us-central1-docker.pkg.dev/nifi-testing/cfm-k8s/cfm-nifi-
k8s:1.28.1.2.3.17.0-9
```

**Note:**

If Kubernetes is running on a different architecture than your local machine, you may need to specify a --platform option for your docker pull.

For more information on pulling, pushing, and tagging Docker images, see the Docker documentation.

4. In *[***NIFI CLUSTER NAMESPACE***]*, create an image pull secret to access the installation artifacts.

```
kubectl create secret docker-registry [***SECRET NAME***] \
  --namespace [***NIFI CLUSTER NAMESPACE***] \
  --docker-server container.repository.cloudera.com \
  --docker-username [***USERNAME***] \
  --docker-password [***PASSWORD***]
```

Replace:

- *[***SECRET NAME***]* with the desired Kubernetes secret name.
- *[***NIFI CLUSTER NAMESPACE***]* with the namespace you created.
- *[***USERNAME***]* and *[***PASSWORD***]* with your Cloudera credentials.

For example:

```
kubectl create secret docker-registry docker-pull-secret \
  --namespace cfm-operator-system \
  --docker-server container.repository.cloudera.com \
  --docker-username my-username \
  --docker-password my-password
```

5. Deploy NiFi to the Kubernetes cluster.

```
kubectl apply -f [***CR YAML PATH***] --namespace [***NIFI CLUSTER
 NAMESPACE***]
```

Replace:

- *[***CR YAML PATH***]* with the absolute or relative path to the CR YAML file you created for NiFi.
- *[***NIFI CLUSTER NAMESPACE***]* with the namespace you created to deploy NiFi.

### Related Information

Configuring a NiFi CR
NiFi CR example

# Enabling edit access with LDAP authentication

If you configured LDAP authentication for your NiFi cluster, you need to perform additional configuration to enable access to the canvas for the admin user.

### About this task

On initial deployment with LDAP user authentication, the specified initial admin identity does not have permissions to edit the canvas, resulting in grayed-out Flow controls. You can configure access from the NiFi web UI.

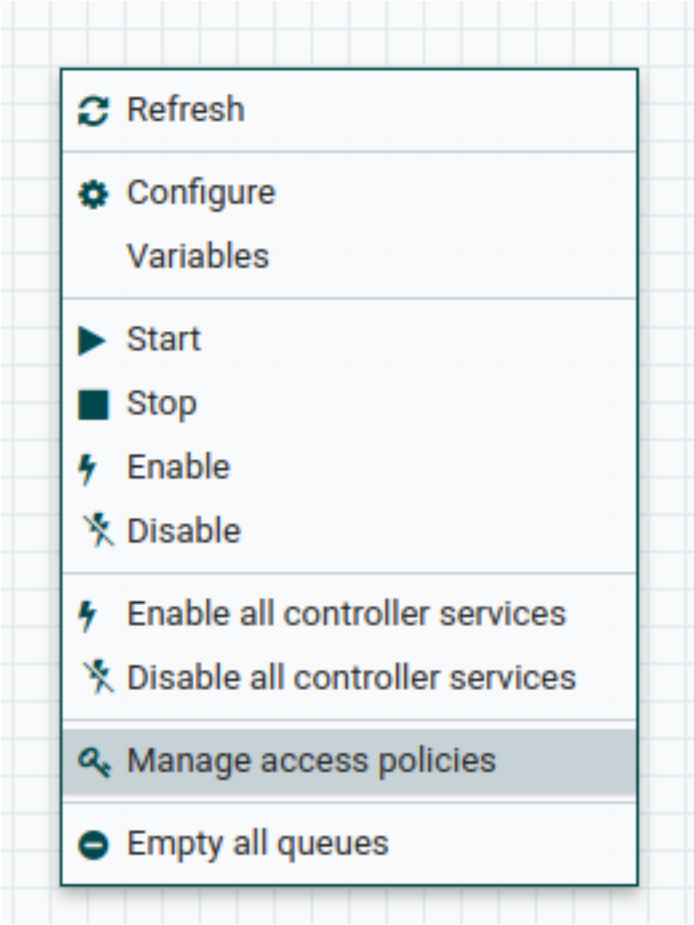
### Procedure

1. Access the NiFi UI by navigating to https://*[***NIFI HOST***]*:*[***NIFI PORT***]*/nifi in a web browser.
   The default *[***NIFI PORT***]* is 8443.
2. Log in using the admin user credentials.
3. Go to Global Menu Policies Access Policies
   You can grant rights by selecting a policy from the drop-down list and then clicking Create.
   Grant at least the following policy to the admin user:
   - view the UI

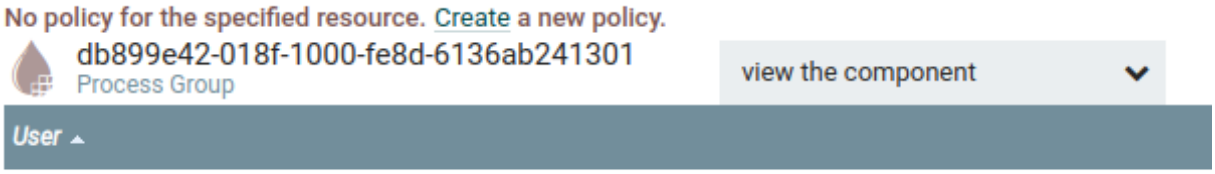
   The available global access policies are:

| Policy | Privilege |
|---|---|
| view the UI | Allows users to view the UI |
| access the controller | Allows users to view and modify the controller including Management Controller Services, Reporting Tasks, Registry Clients, Parameter Providers and nodes in the cluster |
| query provenance | Allows users to submit a provenance search and request even lineage |
| access restricted components | Allows users to create/modify restricted components assuming other permissions are sufficient. The restricted components may indicate which specific permissions are required. Permissions can be granted for specific restrictions or be granted regardless of restrictions. If permission is granted regardless of restrictions, the user can create/ modify all restricted components. |
| access all policies | Allows users to view and modify the policies for all components |
| access users/groups | Allows users to view and modify the users and user groups |
| retrieve site-to-site details | Allows other NiFi instances to retrieve Site-To-Site details |
| view system diagnostics | Allows users to view System Diagnostics |
| proxy user requests | Allows proxy machines to send requests on the behalf of others |
| access counters | Allows users to view and modify counters |

4. Grant the admin user edit permission by right-clicking on the canvas and selecting Manage access policies.




5. Select an access policy from the drop-down list and click Create to make the Add User button available for the policy. Add the admin user.




The available component-level access policies are:

| Policy | Privilege |
|---|---|
| view the component | Allows users to view component configuration details |
| modify the component | Allows users to modify component configuration details |
| view provenance | Allows users to view provenance events generated by this component |
| view the data | Allows users to view metadata and content for this component in FlowFile queues in outbound connections and through provenance events |

| Policy | Privilege |
|---|---|
| modify the data | Allows users to empty FlowFile queues in outbound connections and submit replays through provenance events |
| view the policies | Allows users to view the list of users who can view and modify a component |
| modify the policies | Allows users to modify the list of users who can view and modify a component |
| retrieve data via site-to-site | Allows a port to receive data from NiFi instances |
| send data via site-to-site | Allows a port to send data from NiFi instances |

6. Select view the component and click Create to make the Add User button available for the policy. Add the admin user.

   This provides the admin user write access.