

Managing NiFi in Cloudera Manager

Date published: 2019-06-26

Date modified: 2023-08-29



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

Using the Actions menu.....	4
Managing NiFi nodes.....	5
Adding a NiFi node.....	5
Removing a NiFi node.....	6
Property enhancements.....	6
XML property enhancements.....	6
Bootstrap and Properties file enhancements.....	7

Using the Actions menu

You can manage the NiFi service in Cloudera Manager on CDP Private Cloud Base using the Actions menu available on the service page. This menu provides a list of commands relevant to the role type you are viewing. These commands typically include stopping, starting, or restarting the instance, and may include additional commands depending on your role.

About this task

Follow these steps to access the Actions menu.

Procedure

1. From Cloudera Manager, click the Clusters tab in the left-hand navigation.
2. Click NiFi in the list of services to display the NiFi service page.
3. Click the Actions drop-down next to the service name.

The following commands are available:

Start

This action starts the NiFi service.

Restart

This action runs two other actions, 'Stop NiFi' and 'Start NiFi', one after another.

Rolling Restart

This action consists of three actions, and it runs a specific restart of the NiFi service. The Rolling restart action first runs 'Stop NiFi', then it runs 'Start NiFi', and finally it runs 'NifiNodeWaitUntilHealthy'. The goal of the last action is to wait until the NiFi service is started and works correctly. It is needed in the rolling restart because the 'Start NiFi' action just runs the start NiFi service command and does not wait until the NiFi service actually starts and also does not check if the service works correctly.

Migrate File-based Authorizations to Ranger

This action migrates policies from `authorizations.xml` to Ranger. For more information, see *Migrating NiFi File-Based authorization to Ranger*.

Create required NiFi objects

This action calls the `CreateRequiredNiFiObjects.sh` script, which is used to create NiFi objects like: `ReportLineageToAtlas`, `StandardRestrictedSSLContextService`, or `MonitorDiskUsage`. This script also adds component policies to Ranger, which are described in the `ranger-component-policies.xml` file. The action is automatically executed after the first NiFi start. If needed, you can rerun it later manually.

Remove Ranger repository

This action removes the related NiFi repository/service from Ranger.



Important: Use this action with caution as it causes NiFi to fail. If you ran the command accidentally, 1. stop NiFi, 2. create a marker file, and 3. start NiFi again. After completing these actions, Ranger NiFi repository/service, all related policies, and all NiFi objects will be created/added.

Reset File-based Authorizer Users and Policies

This action automatically removes the `users.xml` and `authorizations.xml` configuration files on each host with NiFi. It also creates backups of these two configurations in the same directory, but with

the file names changed (users.xml{current-date}, authorizations.xml{current-date}). This action helps you to regenerate users.xml and authorizations.xml config files on all NiFi hosts.

Create Marker File

This action creates the /var/lib/nifi/marker.txt file on a random NiFi host. You need this file to understand exactly which NiFi instance should create a NiFi Ranger repository/service and basic policies during the NiFi start. Without this file, Ranger repository/service is not created and NiFi service fails. The action runs before 'Start NiFi' only during the first NiFi start (after installation).

Create First Run File

This action creates the /var/lib/nifi/first_run.txt file on all NiFi hosts. You can use this file to check whether it is the first start of NiFi. For example during the NiFi start process (controller.sh), it is checked that it is not the first start of NiFi and groups are updated for some policies in the NiFi Ranger repository/service, if needed. After the /var/lib/nifi/first_run.txt file is created, it is removed automatically.

Run Post Upgrade Changes

This action runs the run_post_upgrade_changes.sh script. This script adds the required changes for a new CFM version after an upgrade. It also automatically runs during the first run after an upgrade, if the CFM version has changed.



Important: Do not execute this action manually, only in specific cases when you need to fix things in the scope of run_post_upgrade_changes.sh.

Wait until the NiFi Node is healthy

This action runs the 'nifi_wait_until_healthy' script, and the system waits until the NiFi service is started and works correctly.

Stop

This action stops the NiFi service.

Related Information

[Migrating NiFi File-Based authorization to Ranger](#)

Managing NiFi nodes

Cloudera Manager helps you to handle NiFi clusters within the CDP Private Cloud Base environment. It simplifies the process of adding and removing NiFi nodes, providing a centralized and real-time view of the instances across your cluster. Through the Admin Console, you have access to a single, unified platform to facilitate configuration changes and streamline cluster management.

Adding a NiFi node

Follow these steps to add a NiFi node to your cluster.

About this task

The new NiFi node(s) will be added to your cluster. You should now be able to use the new instance seamlessly.

Procedure

1. Go to your cluster's homepage on the Cloudera Manager UI.
2. Open the NiFi service page by clicking NiFi in the list of services.
3. Click Instances.

4. Click Add Role Instance to initiate the process of adding a NiFi node.
The Assign Roles page is displayed.
5. Assign the NiFi roles to the relevant hosts and click Continue to proceed.
The Review Change page is displayed allowing you to verify the configurations and assignments.
6. If needed, update the NiFi properties according to your specific requirements.
7. Click Finish to finalize the configuration process.
8. Click Actions Restart to restart the cluster.

Results

The new NiFi node(s) will be added to your cluster. You should now be able to use the new instance seamlessly.

Removing a NiFi node

Follow these steps to remove a NiFi node from your cluster.

About this task

After successful completion, the NiFi node will be removed from your cluster, and everything should continue to function correctly without the instance you removed.

Procedure

1. Go to your cluster's homepage on the Cloudera Manager UI.
2. Open the NiFi service page by clicking NiFi in the list of services.
3. Click Instances.
4. Select the NiFi Roles associated with the node you wish to remove.
5. Click Actions for selected Decommission to initiate the decommissioning process.
6. Once decommissioning is successfully completed, click Actions for selected once more and select Delete to remove the decommissioned roles.
7. Click Actions Restart to restart the cluster.

Results

After successful completion, the NiFi node will be removed from your cluster, and everything should continue to function correctly without the instance you removed.

Property enhancements

You can use the Cloudera Manager Safety Valves feature to extend CFM components beyond the default properties exposed in Cloudera Manager. The following sections describe how to configure different types of property files generated and used by NiFi and NiFi Registry.

XML property enhancements

In order to configure xml property files, dot notation is leveraged in order to represent the actual schema for a given property file. This allows users to enhance or overwrite xml based properties in Cloudera Manager using. The structure for naming is as follows:

```
xml.<properties-type>.<entity>.<identifier>.class
```

```
xml.<properties-type>.<entity>.<identifier>.property.<property-value>
```

Supported properties types for NiFi are authorizers (authorizers.xml) and loginIdentityProviders (used in login-identity-providers.xml). For NiFi Registry supported types are authorizers and identityProviders.

The below property key/value example creates a user group provider entry into the authorizers file for NiFi:

```
Name: xml.authorizers.userGroupProvider.file-user-group-provider.class
Value: org.apache.nifi.authorization.FileUserGroupProvider

Name: xml.authorizers.userGroupProvider.file-user-group-provider.property
.Initial User Identity 2
Value: CN=localhost, OU=NIFI
```

This translates to the following entry in the generated authorizers.xml file:

```
<authorizers>
....
  <userGroupProvider>
    <identifier>file-user-group-provider</identifier>
    <class>org.apache.nifi.authorization.FileUserGroupProvider</class>
    <property name="Initial User Identity 2">CN=localhost, OU=NIFI</prop
erty>
  </userGroupProvider>
...
</authorizers>
```

Properties names that have spaces are supported and do not need to be escaped.

Bootstrap and Properties file enhancements

Both Bootstrap and Properties files can be enhanced using their respective safety valves (which are used to generate those respective files). The key and value can be entered using the property editor or as hadoop property styled XML.

Below is an example of Safety Valve enhancements for nifi.properties, which decreases default flow election wait time:

NiFi Node Advanced Configuration Snippet (Safety Valve) for staging/nifi.properties.xml

NiFi Node Default Group

Name: nifi.cluster.flow.election.max.wait.time

Value: 1 min

Description: Description

☐ Final

[View as XML](#)

When entering safety valves be aware that if a property shares the same name as one already defined in CM it will override that value.

The NiFi Section for default values can be overridden using safety valves. These can be pasted in as needed using the View as XML link for nifi.properties.xml safety valve. See the *NiFi Documentation* for available property values and usages.

The NiFi Registry section for default values can be overridden using safety valves. These can be pasted in as needed using the View as XML link for nifi.registry.properties.xml safety valve. See the *NiFi Registry Documentation* for available property values.