

Cloudera Flow Management 1.0.1

CFM Component Installation and Upgrade

Date published: 2020-05-11

Date modified: 2020-05-11

The Cloudera logo is displayed in a bold, orange, sans-serif font. The word "CLOUDERA" is written in all caps, with a stylized 'E' that has a horizontal bar extending to the right.

<https://docs.cloudera.com/>

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

Installing NiFi Manually.....	4
Installing NiFi on Linux.....	4
Installing NiFi as a Service.....	4
Starting and Stopping NiFi on Linux.....	4
Launching the User Interface.....	5
 Installing NiFi Registry Manually.....	 5
 Upgrading NiFi Manually.....	 6
Getting ready to upgrade.....	6
Preserve Customizations Prior to Upgrade.....	6
Preserve your custom processors.....	7
Preserve your custom NAR files.....	7
Install the new NiFi version.....	7
Update the Configuration Files for Your New NiFi Installation.....	7
Migrating a dataflow with sensitive properties.....	9
Restarting the dataflow after upgrade.....	10
 Upgrading NiFi Registry Manually.....	 10
Step 1. Install the New Version of NiFi Registry.....	10
Step 2. Update the Configuration Files.....	11

Installing NiFi Manually

If you want to use NiFi for flow management and you do not need the extra components supplied by CFM, you can install NiFi as a stand-alone application on a Linux operating system.

Installing NiFi on Linux

Install NiFi on a Linux operating system.

Procedure

1. Extract the NiFi files in the repo location listed in the *CFM Release Notes* to the location from which you want to run the application.
2. Proceed to the instructions for starting and stopping NiFi.

Installing NiFi as a Service

You can also install NiFi as a service.

Procedure

1. Navigate to the NiFi installation directory.
2. Enter:

```
bin/nifi.sh install
```

Results

The NiFi service is installed with the default name `nifi`.

You can specify a custom name by specifying that name during your install command.

For example, to install NiFi as a service with the name `dataflow`, enter:

```
bin/nifi.sh install dataflow
```

Starting and Stopping NiFi on Linux

Once you have downloaded and installed NiFi, you can start it by using the command appropriate for how you want to interact with NiFi. You can start NiFi in the foreground, background, or as a service.

Starting NiFi in the foreground:

Starting NiFi in the foreground will leave the application running until you press `Ctrl-C`. At that time, it will initiate shutdown of the application.

1. From a terminal window, navigate to the NiFi installation directory.
2. Enter:

```
bin/nifi.sh run
```

Starting NiFi in the background:

If you start NiFi in the background, to check the status and see if NiFi is currently running, execute the command `bin/nifi.sh status`. To shut down NiFi, execute the command `bin/nifi.sh stop`.

1. From a terminal window, navigate to the NiFi installation directory.
2. Enter:

```
bin/nifi.sh start
```

Starting NiFi as a service:

Once installed, you can start and stop the NiFi service by using the appropriate commands, such as `sudo service nifi start` and `sudo service nifi stop`. Additionally, you can check the running status with `sudo service nifi status`.

1. From a terminal window, enter:

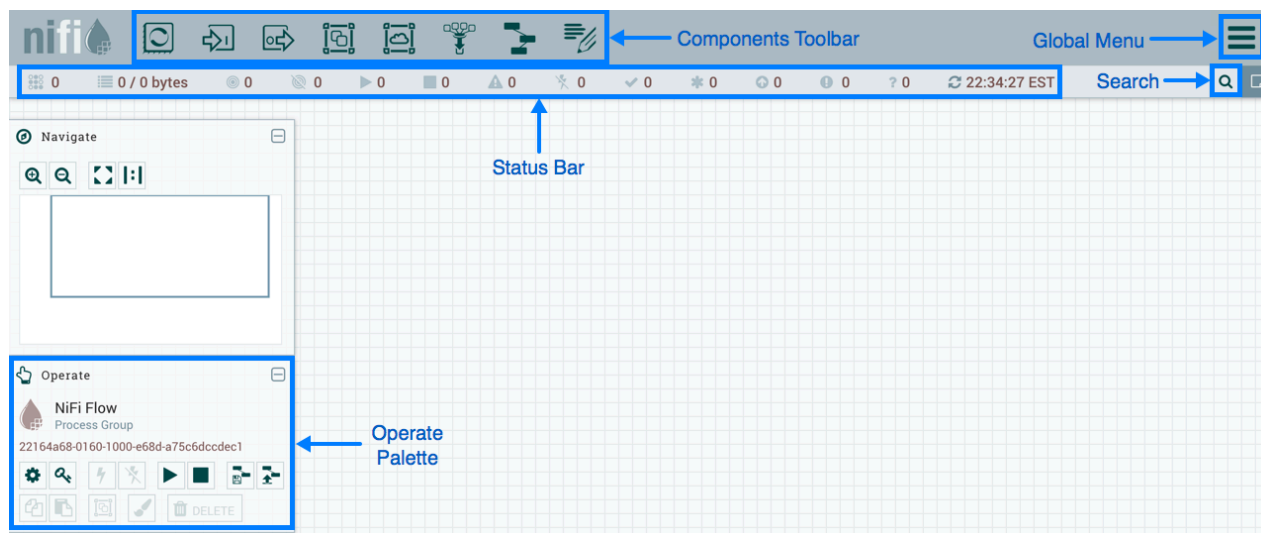
```
sudo service nifi start
```

Launching the User Interface

After you have started NiFi, you can bring up the User Interface (UI) to create and monitor dataflows.

Open a web browser and navigate to `http://localhost:8080/nifi`.

The toolbars at the top of the UI are very important to create your first dataflow:



Installing NiFi Registry Manually

You can install NiFi Registry as a stand-alone application on a Linux operating system.

Procedure

1. Decompress and untar the installation files into the desired installation directory
 - From the `<installdir>/bin` directory, execute the following commands by typing `./nifi-registry.sh <command>`:
2. Make any desired edits in files found under `<installdir>/conf`

3. From the `<installdir>/bin` directory, execute the following commands by typing `./nifi-registry.sh <command>`:

- `start`: starts NiFi Registry in the background
- `stop`: stops NiFi Registry that is running in the background
- `status`: provides the current status of NiFi Registry
- `run`: runs NiFi Registry in the foreground and waits for a Ctrl-C to initiate shutdown of NiFi Registry
- `install`: installs NiFi Registry as a service that can then be controlled via
 - `service nifi-registry start`
 - `service nifi-registry stop`
 - `service nifi-registry status`

Results

When NiFi Registry first starts up, the following directories are created:

- `flow_storage`
- `database`
- `work`
- `logs`
- `run`

Upgrading NiFi Manually

If you are upgrading a cluster, All Nodes must be upgraded and stopped before starting new Apache NiFi nodes. Your cluster cannot contain nodes with different versions.

Getting ready to upgrade

Before you upgrade Apache NiFi, it is helpful to identify your upgrade path, review changes to the NiFi version to which you are upgrading, and prepare for any changes needed to your environment before or immediately after upgrade.

Interoperability and support information

If you are upgrading NiFi on a Linux operating system, refer to the *CFM Support Matrix* for interoperability and support information for your version of CFM.

Review migration guidance

Identify the version of Apache NiFi you are running, and the version of Apache NiFi to which you want to upgrade. Review the *Apache NiFi migration guidance* to be aware of changes made between versions and the impact they may have on your existing dataflows.

Clear queues and shutdown NiFi

Before you begin your NiFi upgrade, stop all the source processors to prevent the ingestion of new data, allow your NiFi flowfile processing queues to empty completely, and then stop NiFi.

Preserve Customizations Prior to Upgrade

If you have created any custom processors or customized any of the default NiFi Archive files that you would like to persist after the upgrade, you should preserve them during the upgrade.

Preserve your custom processors

If you have written custom processors, you can preserve them during the upgrade if you store them in a central location.

Procedure

1. Create a second library directory, called `custom_lib`.
2. Move your custom processors to this new lib directory.
3. Add a new line to the `nifi.properties` file to specify this new lib directory:

```
nifi.nar.library.directory=./lib
nifi.nar.library.directory.custom=/opt/configuration_resources/custom_lib
```

Preserve your custom NAR files

If you have customized any of the default NiFi Archive (NAR) files, upgrading NiFi overwrites these changes. To preserve the customization:

Procedure

1. Identify and save the changes you made to the default NAR files.
2. Perform your NiFi upgrade.
3. Implement the same NAR file changes in your new NiFi instance.
4. Start your dataflow.

Install the new NiFi version

Install the new NiFi into a directory parallel to the existing NiFi installation.

Procedure

1. Download the latest version of Apache NiFi from the [Cloudera Flow Management Downloads Page](#).
2. Uncompress the NiFi .tar file into a directory parallel to your existing NiFi directory:

```
tar -xvzf file-name
```

For example, if your existing NiFi installation is installed in `/opt/nifi/nifi-1.2.0.3.0.2.0-76/`, install your new NiFi version in a path similar to `/opt/nifi/nifi-1.11.4.1.1.0.0-119/`.

3. Create or update a symlink: `/opt/nifi/current # /opt/nifi/<nifi build dir>`
4. If you are upgrading a NiFi cluster, repeat these steps on each node in the cluster.

Update the Configuration Files for Your New NiFi Installation

Use the configuration files from your existing NiFi installation to manually update the corresponding properties your new NiFi deployment.

About this task

Do not copy configuration files from your existing NiFi version to the new NiFi version. The newer configuration files may introduce new properties that would be lost if you copy and paste configuration files.





Procedure

1. Use the following table to update the configuration files located in <installation-directory>/conf.
2. Double check all configured properties for typos.

What to do next

Table 1: Configuration File Changes

Configuration file	Necessary changes
state-management.xml	<p>For the “local-provider” value, verify the location of “state/local” directory.</p> <p>If you have retained the default location (./state/local), copy complete directory tree to new NiFi. The existing NiFi should be stopped if you are copying this directory because it may be constantly writing to this directory while running.</p> <p>Configuration best practices recommend that you move the state to an external directory like /opt/nifi/configuration-resources/ to facilitate future upgrades.</p> <p>If this is a NiFi cluster, the “cluster-provider” ZooKeeper “Connect String” should be using the same external ZooKeeper as the existing NiFi installation.</p> <p>Hortonworks does not support using the NiFi embedded ZooKeeper.</p> <p>If you are using a NiFi cluster, ensure that the new “cluster-provider” ZooKeeper “Root Node” values exactly match your previous values.</p>
login-identity-providers.xml	<p>If you used a provider value in your existing the NiFi installation, copy the <provider> ... </provider> configuration from the existing NiFi installation to this file in your new NiFi installation.</p>
logback.xml	<p>If you added any custom logging modification to the existing NiFi, make those same changes to the new NiFi installation.</p>
bootstrap.conf	<p>Use the existing NiFi bootstrap.conf file to update properties in new NiFi.</p> <p>If you are upgrading to CFM 1.1.0, you should add the following line: java.arg.100=-Djavax.security.auth.useSubjectCredsOnly=true</p>
bootstrap-notification-services.xml	<p>Update the values based on values from the existing NiFi file.</p>
authorizers.xml	<p>Copy the <authorizer> </authorizer> configured in the current NiFi to new the NiFi file.</p> <p>If you are using “file-provider”, ensure that you copy the users.xml and authorizations.xml files from the current to the new NiFi.</p> <p>Configuration best practices recommend creating a separate location outside of the NiFi base directory for storing such configuration files: for example, /opt/nifi/configuration-resources/. If you are storing these files in a separate directory, you do not need to move them. Instead, ensure that the new NiFi is pointing to the same files.</p>
nifi.properties	<p>Use the current file to populate the same properties in the new NiFi nifi.properties file.</p> <p>This file contains the majority of NiFi configuration settings, so ensure that you have copied the values correctly.</p> <p>If you followed NiFi best practices, the following properties should be pointing to external directories outside of the base NiFi installation path.</p> <p>If the properties point to directories inside the NiFi base installation path, you must copy the target directories to the new NiFi. Stop your current NiFi installation before you do this.</p> <p>nifi.flow.configuration.file= If you have retained the default value, (./conf/flow.xml.gz), copy flow.xml.gz from the current to the new NiFi base install conf directory.</p> <p>Alternately, you can copy to an external location and update the property value to point there.</p>

Configuration file	Necessary changes
	<p>nifi.flow.configuration.archive.dir= Same applies as above if you want to retain archived copies of the flow.xml.gz.</p>
	<p>nifi.database.directory= Best practices recommends that you use an external location for each repository. Point new NiFi at same external database repository location.</p>
	<p>nifi.flowfile.repository.directory= Best practices recommends that you use an external location for each repository. Point new NiFi at same external database repository location.</p> <p> Attention: You may experience data loss if flowfile repositories are not accessible to new NiFi.</p>
	<p>nifi.content.repository.directory.default= Best practices recommends that you use an external location for each repository. Point new NiFi at same external database repository location.</p> <p>Your current NiFi may have multiple content repos defined. Make sure exact same property names are used and point to appropriate matching content repo locations. For example:</p> <pre>nifi.content.repository.directory.content1= nifi.content.repository.directory.content2=</pre> <p> Attention: You may experience data loss if content repositories are not accessible to new NiFi.</p> <p> Attention: You may experience data loss may if property names are wrong or the property points to wrong content repository.</p>
	<p>nifi.provenance.repository.directory.default= Best practices recommends that you use an external location for each repository. Point new NiFi at same external database repository location.</p> <p>Your current NiFi might define multiple content repositories. . Ensure that your new installation uses the same property names and point to appropriate matching content repository locations.</p> <p>For example:</p> <pre>nifi.provenance.repository.directory.provenance1= nifi.provenance.repository.directory.provenance2=</pre> <p> Note: You may not be able to query old events if you have not moved the provenance repositories or did not update the properties correctly.</p>

Migrating a dataflow with sensitive properties

When you set a value for `nifi.sensitive.props.key` in `nifi.properties`, that key is used to encrypt sensitive properties like component passwords in the flow. If you need to change the sensitive properties key value, you can use the Encrypt-Config tool in the NiFi Toolkit to migrate the sensitive properties key to a new value and update the `flow.xml.gz` file.

About this task

The Encrypt-Config tool performs the following actions:

- Reads the existing flow.xml.gz and decrypts the sensitive values using the current key.
- Encrypts all the sensitive values with a specified new key.
- Updates the nifi.properties and flow.xml.gz files or creates new versions of them.

Run the following command:

```
$ ./nifi-toolkit-<version>/bin/encrypt-config.sh
-f /path/to/nifi/nifi-<source-version>/conf/flow.xml.gz
-g /path/to/nifi/nifi-<target-version>/conf/flow.xml.gz
-s <new-password>
-n /path/to/nifi/nifi-<source-version>/conf/nifi.properties
-o /path/to/nifi/nifi-<target-version>/conf/nifi.properties
-x
```

Where:

- -f – Specifies the source flow.xml.gz.
- -g – Specifies the destination flow.xml.gz.
- -s – Specifies the new sensitive properties key.
- -n – Specifies the source nifi.properties.
- -o – Specifies the destination nifi.properties.
- -x – Tells the Encrypt-Config tool to only process the sensitive properties.

For more information, see the *Encrypt-Config Tool* section in the *NiFi Toolkit Guide*.

Restarting the dataflow after upgrade

After you have installed and configured the new NiFi cluster, complete the upgrade process by configuring your new NiFi cluster to take over flow management tasks.

Procedure

1. Set the directory and file ownership of your new NiFi instance to match the current owner and permissions of the existing NiFi instance.
2. If you have not already done so, stop your existing NiFi instance.



Attention:

Do not start your new NiFi installation while the existing NiFi installation is still running. You may experience data loss if both NiFi configurations are pointing at any of the same external configuration files, repositories, or similar.

3. Start your new NiFi instance as the same user as for your existing NiFi instance.
4. If your existing NiFi instance was configured to run as a service, update any symlinks or service scripts to point to new NiFi version executables.

Upgrading NiFi Registry Manually

Upgrading NiFi Registry includes installing the new version of NiFi Registry and updating the configuration files.

Review the upgrade steps before you begin the upgrade process.

Step 1. Install the New Version of NiFi Registry

Install the new version of NiFi Registry into a directory parallel to the existing NiFi Registry installation.

Procedure

1. Go to the [Cloudera Flow Management Downloads](#) page.
2. From the Installation Type drop-down menu, choose NiFi Registry.
3. Click DOWNLOAD NOW! #

The download button appear on the page.



Cloudera Flow Management - NiFi Registry

CFM NiFi registry 1.1.0 (TAR)

Older Versions

- [CFM NiFi registry 1.0 \(TAR\)](#)

4. Uncompress the NiFi Registry .tar file into a directory parallel to your existing NiFi Registry directory:

```
tar -xvzf <file-name>
```

For example, if your existing NiFi Registry installation is installed in /opt/nifi-registry/nifi-registry-0.3.0.1.0.1.0-12/, install your new NiFi Registry version in /opt/nifi-registry-0.6.0.1.1.0.0-119/.

Related Information

[Step 2. Update the Configuration Files](#)

Step 2. Update the Configuration Files

Use the information in the configuration files from your previous NiFi Registry installation to manually update the corresponding properties in your new NiFi Registry deployment.

About this task

You must copy the users.xml and authorizations.xml files from the previous NiFi Registry conf directory to the new NiFi Registry conf directory.

Update the other configuration files according to the steps here and in the *Apache NiFi Registry Migration Guidance* document. Then, restart NiFi Registry to apply the updates.



Important: Besides the users.xml and authorizations.xml, do not copy any other configuration file from your previous NiFi Registry conf folder to the new conf folder. The new configuration files might contain new properties that would be lost if you replace the file.

Procedure

1. If you are using a file-provider, perform the following steps:
 - a) Copy the following files from the previous NiFi Registry conf folder:
 - users.xml
 - authorizations.xml
 - b) Go to the new NiFi Registry <installation-directory>/conf folder.

Name	Size	Kind
▼ nifi-registry-0.6.0.1.1.0.0-119	--	Folder
▶ docs	--	Folder
▶ bin	--	Folder
▼ conf	--	Folder
logback.xml	6 KB	XML Document
providers.xml	5 KB	XML Document
nifi-registry.properties	4 KB	Java Properties
bootstrap.conf	2 KB	Configuration file
identity-providers.xml	6 KB	XML Document
authorizers.xml	19 KB	XML Document
registry-aliases.xml	1 KB	XML Document
▶ lib	--	Folder
▶ ext	--	Folder
NOTICE	16 KB	TextEdit Document
LICENSE	87 KB	TextEdit Document
README	3 KB	TextEdit Document

- c) Paste the users.xml and authorizations.xml files.
2. Open the nifi-registry.properties file and make the following updates:
 - a) Populate the properties with the same values you used in the nifi-registry.properties file of your previous NiFi Registry version.
 - b) Populate any new properties.



Important: This file contains the majority of NiFi Registry configuration settings, so ensure that you have copied the values correctly.

3. Open the providers.xml file and make the following updates:
 - a) Replace the following section with the corresponding section in the previous NiFi Registry providers.xml file.

```
<flowPersistenceProvider>
  <class>persistence-provider-qualified-class-name</class>
  <property name="property-1">property-value-1</property>
  <property name="property-2">property-value-2</property>
  <property name="property-n">property-value-n</property>
</flowPersistenceProvider>
```

- b) Replace the following section with the corresponding section in the previous NiFi Registry providers.xml file.

```
<extensionBundlePersistenceProvider>
  <class>persistence-provider-qualified-class-name</class>
  <property name="property-1">property-value-1</property>
```

```
<property name="property-2">property-value-2</property>
<property name="property-n">property-value-n</property>
</extensionBundlePersistenceProvider>
```

4. If you use an identity provider, then open the `identity-providers.xml` file and make the following update:
 - a) Replace the `<provider> ... </provider>` section with the corresponding section in the previous NiFi Registry `identity-providers.xml` file.
5. Open the `authorizers.xml` file and make the following update:
 - a) Replace the following section with the corresponding section in the previous NiFi Registry `authorizers.xml` file.

```
<authorizer>
  <identifier>managed-authorizer</identifier>
  <class>org.apache.nifi.registry.security.authorization.Standard
ManagedAuthorizer</class>
  <property name="Access Policy Provider">file-access-policy-pro
vider</property>
</authorizer>
```

6. Save and close the configuration files.
7. Review the *Apache NiFi Registry Migration Guidance* document and complete all version-specific tasks.
8. Restart NiFi Registry.

From the `<installdir>/bin` directory, run the following command: `./nifi-registry.sh start`

Related Information

[Apache NiFi Registry Migration Guidance](#)