

1.x.0 to 1.x.0 Upgrade

Release Notes & Migration Guidance

Prior to upgrade you should review the [Release Notes](#) carefully to ensure that you understand the changes made and the impact they may have on your existing data flows and/or environment.

The instructions below are general steps to follow when upgrading from a 1.x.0 release to another. Check the [Migration Guidance](#) page for specific items that you should be aware of when moving between specific NiFi versions.

Important:

- You cannot have nodes with different NiFi versions in the same cluster.
- Running multiple NiFi nodes on the same host machine is highly discouraged.

Before you begin

Before you begin, you can perform some tasks to improve the efficiency and success of your upgrade.

Preserving your custom processors

If you have written any custom NARs, first test and validate them in a separate new NiFi release sandbox prior to upgrading. The verified NARs can be preserved during the upgrade by storing them in a centralized location as follows:

1. Create a second library directory, called `custom_lib`.
2. Move your custom NARs 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
```

Preserving your modified NARs

If you have modified any of the default NAR files, an upgrade will overwrite these changes. To preserve the customization:

1. Identify and save the changes you have made to the default NAR files in your existing instance.
2. Perform your upgrade to the new NiFi release.
3. Redo the changes you have made to the NAR files in your upgraded NiFi instance.

Installation

1. [Download](#) and extract the new NiFi tar.gz file.

2. For each node in your cluster, install the new NiFi in a directory parallel to your existing NiFi installation. Assuming you have installed into an `opt/` directory, for example:

Host Machine - Node 1

```
|--> opt/  
    |--> existing-nifi  
    |--> new-nifi
```

Host Machine - Node 2

```
|--> opt/  
    |--> existing-nifi  
    |--> new-nifi
```

Host Machine - Node 3

```
|--> opt/  
    |--> existing-nifi  
    |--> new-nifi
```

Note: Make sure that all file and directory ownerships for your new NiFi directories match what you set on the existing directories.

3. On your existing NiFi installation:

- Stop all processors responsible for ingesting new data.
- Allow NiFi to run until no more FlowFiles are queued anywhere in the dataflow(s).
- Shutdown your existing NiFi instance(s).

4. Use the configuration files from your existing NiFi installation (`<installation-directory>/conf`) to update the corresponding properties in your new NiFi deployment.

Configuration file	Necessary changes
authorizers.xml	<p>Copy the <code><authorizer>...</authorizer></code> configured in the existing NiFi to new the NiFi file.</p> <p>If you are using the default "file-provider" authorizer, ensure that you copy the <code>users.xml</code> and <code>authorizations.xml</code> files from the existing 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. 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. For example: <code>/opt/nifi/configuration-resources/</code></p>
bootstrap-notification-services.xml	Update the values based on values from the existing NiFi file.
bootstrap.conf	Use the existing NiFi <code>bootstrap.conf</code> file to update properties in the new NiFi.
logback.xml	If you added any custom logging modification to the existing NiFi, make those same changes to the new NiFi.
login-identity-providers.xml	If you used a provider value in the existing NiFi, copy the <code><provider>...</provider></code> configuration from the existing NiFi to this file in your new NiFi.
nifi.properties	<p>Use the existing file to populate the same properties in the new NiFi <code>nifi.properties</code> 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 below properties point to directories inside the NiFi base installation path, you must copy the target directories to the new NiFi. Stop your existing NiFi installation before you do this.</p> <p><code>nifi.flow.configuration.file=</code></p> <p>If you have retained the default value, (<code>./conf/flow.xml.gz</code>), copy <code>flow.xml.gz</code> from the existing 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> <p><code>nifi.flow.configuration.archive.dir=</code></p> <p>Same applies as above if you want to retain archived copies of the <code>flow.xml.gz</code>.</p> <p><code>nifi.database.directory=</code></p> <p>Best practices recommends that you use an external location for each repository. Point the new NiFi at the same external database repository location.</p> <p><code>nifi.flowfile.repository.directory=</code></p> <p>Best practices recommends that you use an external location for each repository. Point the new NiFi at the same external flowfile repository location.</p> <p>Warning: You may experience data loss if flowfile repositories are not accessible to the new NiFi.</p>

	<pre>nifi.content.repository.directory.default=</pre> <p>Best practices recommends that you use an external location for each repository. Point the new NiFi at the same external content repository location.</p> <p>Your existing NiFi may have multiple content repos defined. Make sure the exact same property names are used and point to the appropriate matching content repo locations. For example:</p> <pre>nifi.content.repository.directory.content1= nifi.content.repository.directory.content2=</pre> <p>Warning: You may experience data loss if content repositories are not accessible to new NiFi.</p> <p>Warning: You may experience data loss may if property names are wrong or the property points to wrong content repository.</p>
	<pre>nifi.provenance.repository.directory.default=</pre> <p>Best practices recommends that you use an external location for each repository. Point new NiFi at same external provenance repository location.</p> <p>Your existing NiFi may have multiple provenance repos defined. Make sure exact same property names are used and point to appropriate matching provenance repo locations. 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 provenance repos are not moved correctly or properties are not updated correctly.</p>
<p>state-management.xml</p>	<p>For the "local-provider" value, verify the location of "state/local" directory.</p> <p>If you have retained the default location (<code>./state/local</code>), copy the complete directory tree to the 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 <code>/opt/nifi/configuration-resources/</code> to facilitate easier upgrading later.</p> <p>For a NiFi cluster, the "cluster-provider" ZooKeeper "Connect String" should be using the same external ZooKeeper as the existing NiFi installation.</p> <p>For a NiFi cluster, make sure the "cluster-provider" ZooKeeper "Root Node" matches exactly the value you used in the existing NiFi.</p> <p>If you are also setting up a new external ZooKeeper, see the ZooKeeper Migrator section of the NiFi System Administrator's Guide for instructions on how to move ZooKeeper information from one cluster to another and migrating ZooKeeper node ownership.</p>

Note:

- Do not copy configuration files from your old NiFi version to the new NiFi version. The newer configuration files may introduce new properties that would be lost by doing so.
- Double check all configured properties for typos.
- If you are encrypting sensitive component properties in your dataflow via the sensitive properties key in `nifi.properties` file, make sure the same key is used when copying over your `flow.xml.gz`. If you need to change the key, use the [flow migration procedure](#) and the [NiFi Encrypt-Config Tool](#) to produce a new `flow.xml.gz` file from the existing one.

5. Start each of your new NiFi instances.

6. Verify that:

- All your data flows have returned to a running state. Some processors may have new properties that need to be configured, in which case they will be stopped and marked "invalid".
- All your expected controller services and reporting tasks are running again. Address any controller services or reporting tasks that are marked "invalid".

7. After confirming your new NiFi instances are stable and working as expected, the old instances can be removed.

Note: If original NiFi was setup to run as a service, update any symlinks or service scripts to point to the new NiFi version executables.