Cloudera Flow Management 4.10.0

CFM Component Installation and Upgrade

Date published: 2019-06-26
Date modified: 2025-05-22

https://docs.cloudera.com/

Legal Notice

Contents

Installing and starting NiFi on Linux

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. Download the NiFi file listed in the *Standalone components* section of the *Download locations* documentation. Place the file in the location from which you want to run the application.
2. Make any desired edits in files found under <installdir>/conf
3. Proceed to the instructions for starting and stopping NiFi.

Related Information

Download locations

Installing NiFi as a service

Provides steps to 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, run the command `/bin/nifi.sh` status. To shut down NiFi, run 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 st art 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
```

Results

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

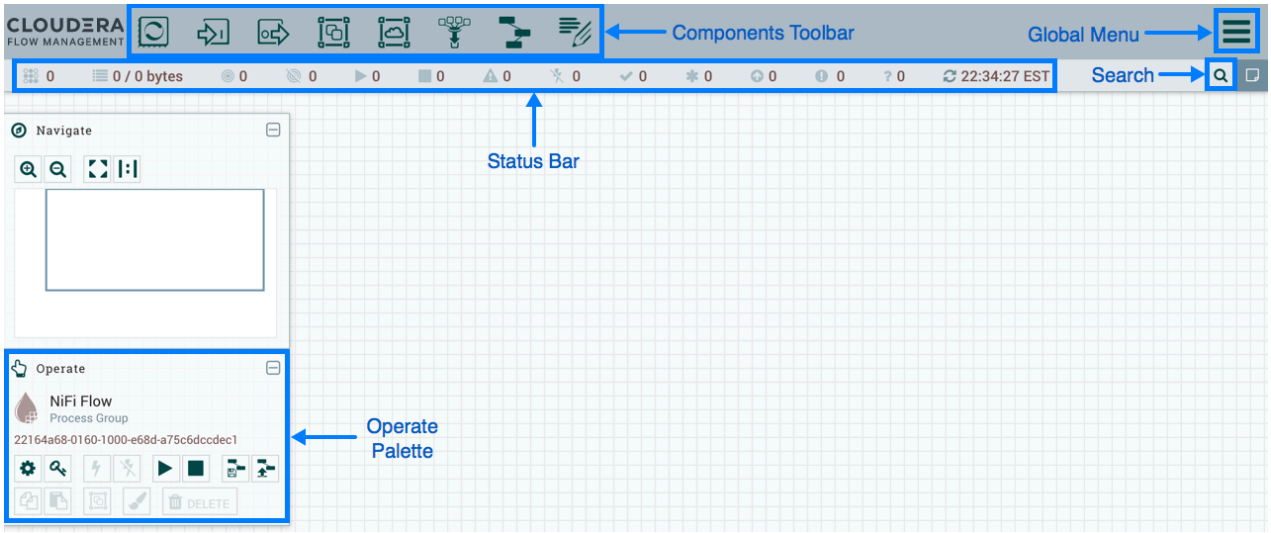
- content_repository
- database_repository
- flowfile_repository
- provenance_repository
- work directory
- logs directory
- Within the conf directory, the flow.xml.gz file is created
- flow_storage
- database
- work
- logs
- run

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. NiFi is unsecured by default and the default port is 8080.

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



Installing and starting NiFi Registry manually on Linux

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.
2. Make any desired edits in files found under <installdir>/conf
3. From the <installdir>/bin directory, run 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

Installing NiFi on Windows

Installing NiFi on a Windows involves meeting your system requirements and performing the installation, configuring either a local or domain user, and starting and stopping NiFi on Windows.

Related Information

Download locations

Installing NiFi using the MSI file

To install NiFi on a Windows machine, review the prerequisites, download and extract the MSI file, and place it in your installation directory.

About this task

You can install a single instance of NiFi or MiNiFi on a Windows machine. Running NIFI on Windows in a clustered mode is not supported.

Before you begin

Before you begin your NiFi installation, be sure you meet the following requirements:

- Install JDK 8.0 64 bit.
- Install Java to C:/java instead of C:/Program Files.
- Recent Windows versions mark everything in C:\Program Files as read only.
- Set the JAVA_HOME environment variable using the 8.3 style name conventions.

  For example: C:\Program\jdk1.8.0.
- Ensure JAVA_HOME is pointing to a 64-bit JRE/JDK.
- Ensure your system meets the minimum memory requirement for Windows which is 4GB.

Procedure

1. Extract the NiFi MSI files to the location from which you want to run the application.

   You can find the MSI file in the *Windows Files* section of the *Download locations* documentation.
2. Install NiFi to C:/nifi or a similar root folder.

Related Information

Download locations

Using a local user for NiFi Windows Service

Using a local user for your NiFi Windows Service requires that you specify the appropriate user name and password during your CFM NiFi setup.

About this task

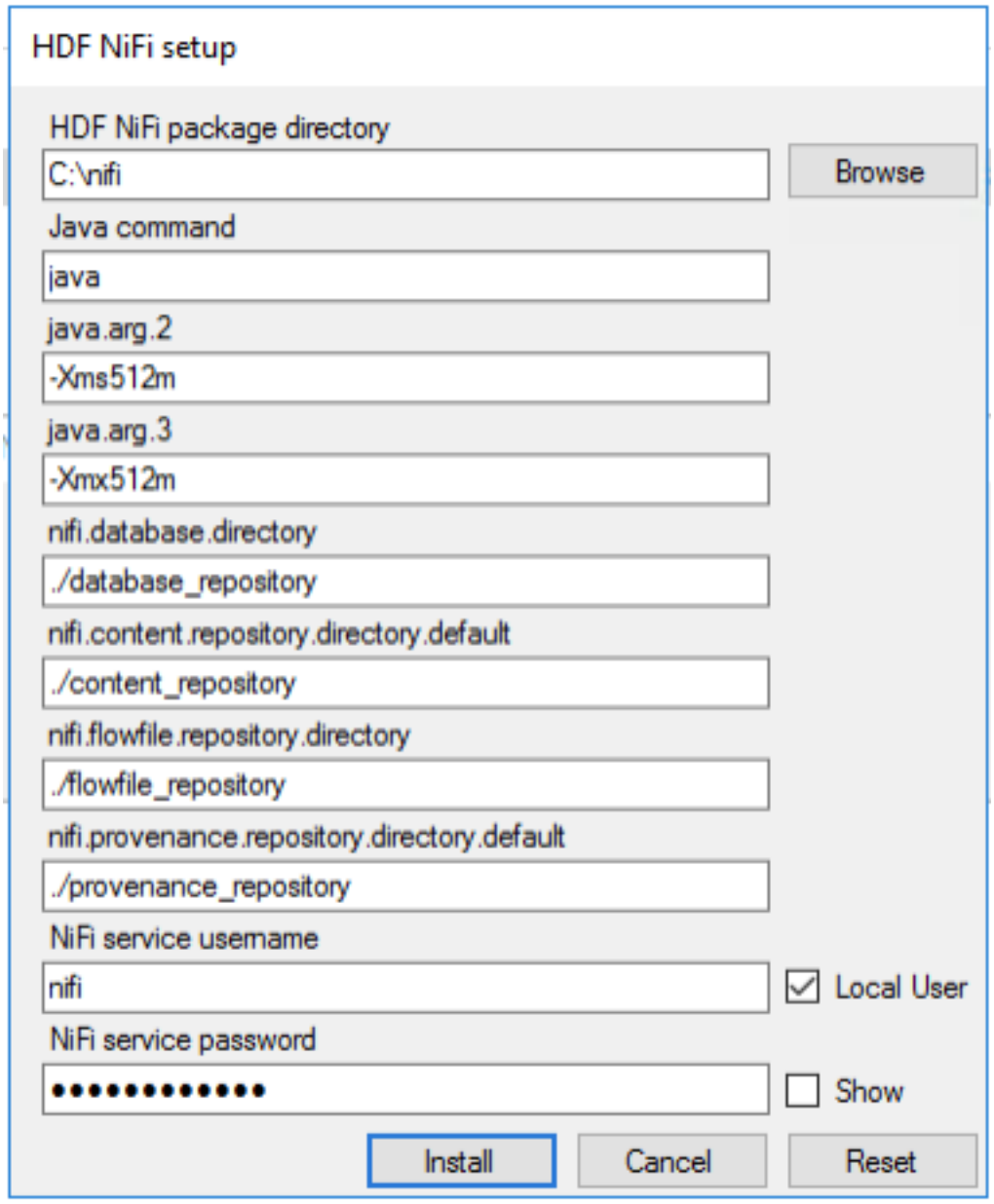
There is no prerequisite to use a Local user for the Windows service. The installer automatically sets up the user.

Note:

If you configure a local user for your NiFi Windows Service, do not configure a domain user.

Procedure

1. If the computer is a part of a domain, then the Local User checkbox appears in the CFM NiFi setup window. Check the Local User checkbox to specify that Local user is used to run the installed service.



   If a user specified at NiFi service username does not exist, the installer creates one with the specified NiFi service password. If the user already exists, the installer updates its password with the specified password.

2. The installer also grants the following privileges to the specified user:

   - SeCreateSymbolicLinkPrivilege
   - SeServiceLogonRight

Related Information

Download locations

Using a domain user for NiFi Windows Service

When you are using a domain user for your NiFi Windows service, you must set and configure the appropriate permissions, Install and set up the ActiveDirectory PowerShell module, and configure the appropriate user name and password in the CFM NiFi setup window.

About this task

Note:

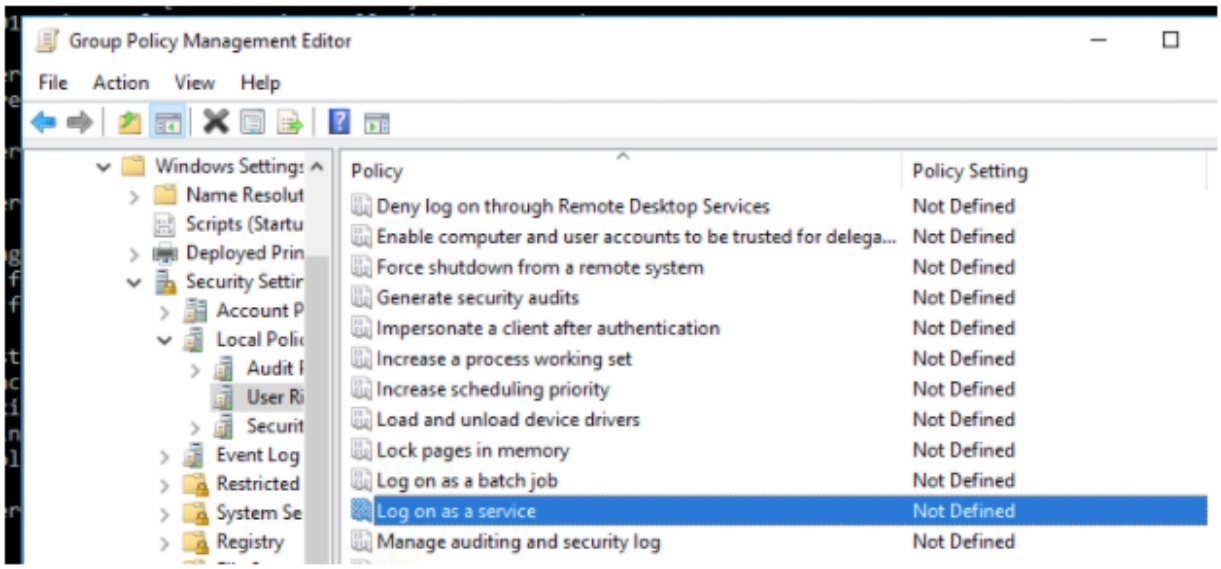
If you configure a domain user for your NiFi Windows Service, do not configure a local user.

Before you begin

- The computer must be part of the domain.
- The specified user must exist in the domain, and a correct password must be provided.
- ActiveDirectory PowerShell module must be available.

Procedure

1. In the Group Policy Management Editor, set your permissions to Log on as a service.



2. Navigate to a machine on which NiFi is installed and enter the following command:

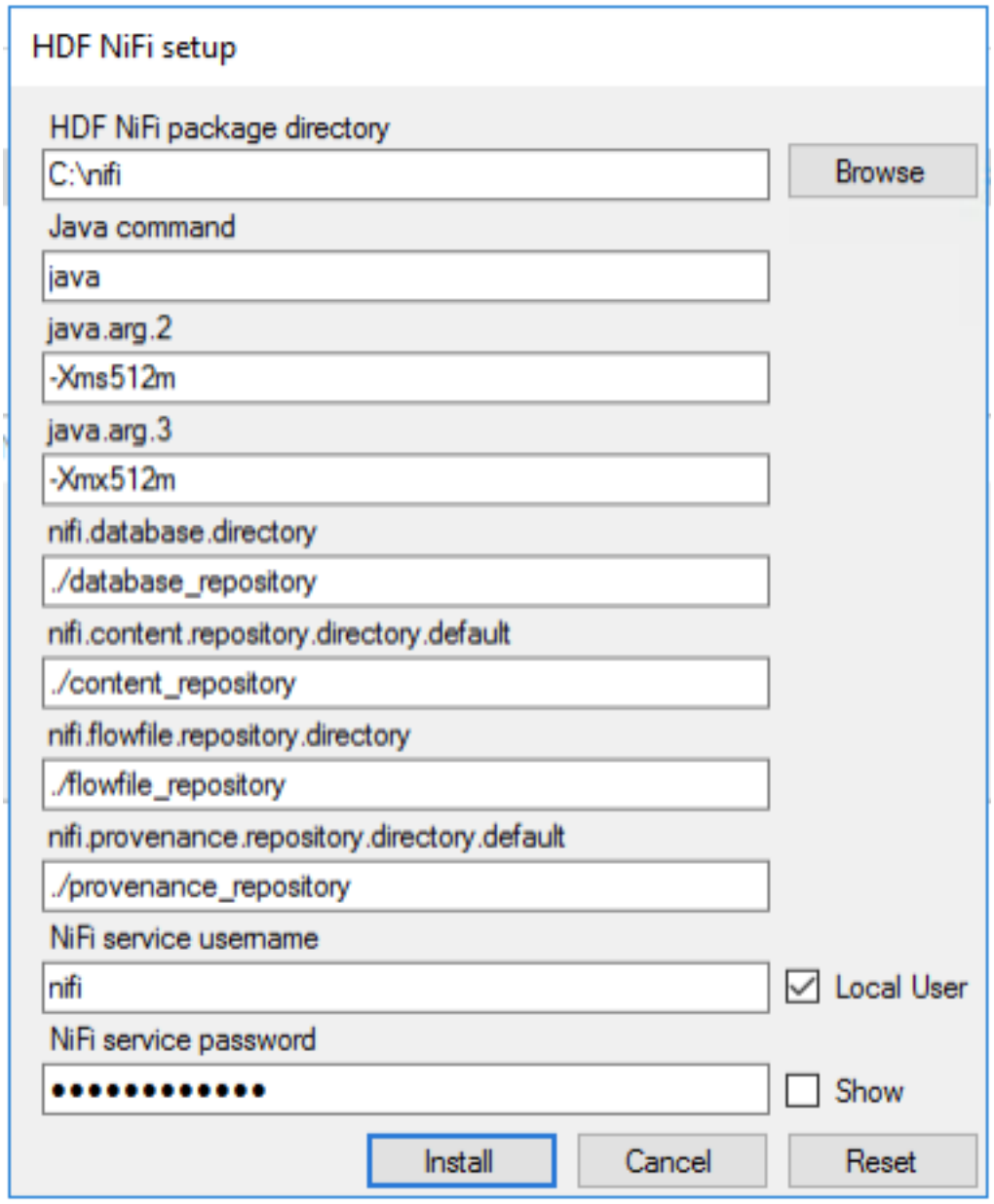
```
gpupdate
```

   The `gpupdate` command is a machine-wide command and can be executed from any directory on the NiFi machine.

3. Install the ActiveDirectory PowerShell module by entering the following in the PowerShell console:

```
Add-WindowsFeature RSAT-AD-PowerShell
```

4. In the CFM NiFi setup window, uncheck the Local User checkbox and click Install.



What to do next

After installation, you can update Java options at nifi-install-dir\conf\bootstrap.conf file. Repository locations are at nifi-install-dir\conf\nifi.properties file.

Related Information

Download locations

Starting and stopping NiFi on Windows

You can start NiFi by running a .bat command from the terminal, and you can stop NiFi using keyboard commands.

Starting NiFi

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

2. Enter:

```
bin\run-nifi.bat
```

Stopping NiFi

1. Select the window that was launched when you started NiFi.
2. Hold the Ctrl key while pressing C.

Related Information

Download locations

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.

Before you 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 shut down 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.

Related Information

Apache NiFi Migration Guidance

CFM Support Matrix

Preserving 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
```

Preserving your custom NAR files

If you have customized any of the default NiFi Archive (NAR) files, upgrading NiFi overwrites these changes. You can 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.

Installing 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 *Download location* documentation.
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-1.11.4.2.0.1.0-71/.
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.

Related Information

Download locations

Updating 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	For the `local-provider` value, verify the location of state/local directory. 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. Configuration best practices recommend that you move the state to an external directory like /opt/nifi/configuration-resources/ to facilitate future upgrades.
	If this is a NiFi cluster, the cluster-provider ZooKeeper Connect String should be using the same external ZooKeeper as the existing NiFi installation. Cloudera does not support using the NiFi embedded ZooKeeper.
	If you are using a NiFi cluster, ensure that the new cluster-provider ZooKeeper Root Node values exactly match your previous values.
login-identity-providers.xml	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.
logback.xml	If you added any custom logging modification to the existing NiFi, make those same changes to the new NiFi installation.
bootstrap.conf	Use the existing NiFi bootstrap.conf file to update properties in new NiFi.
	If you are upgrading to CFM 1.1.0, you should add the following line: java.arg.100=-Djavax.security.auth.useSubjectCredsOnly=true
bootstrap-notification-services.xml	Update the values based on values from the existing NiFi file.
authorizers.xml	Copy the <authorizer> …. </authorizer> configured in the current NiFi to new the NiFi file.
	If you are using file-provider, ensure that you copy the users.xml and authorizations.xml files from the current to the new NiFi. 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.
nifi.properties	Use the current file to populate the same properties in the new NiFi nifi.properties file. This file contains the majority of NiFi configuration settings, so ensure that you have copied the values correctly.
	If you followed NiFi best practices, the following properties should be pointing to external directories outside of the base NiFi installation path. 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.
	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. Alternately, you can copy to an external location and update the property value to point there.
	nifi.flow.configuration.archive.dir= Same applies as above if you want to retain archived copies of the flow.xml.gz.
	nifi.database.directory= Best practices recommends that you use an external location for each repository. Point new NiFi at same external database repository location.

Configuration file	Necessary changes
	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. **Warning:** You may experience data loss if flowfile repositories are not accessible to new NiFi.

	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.
	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:
	`nifi.content.repository.directory.content1=`
	`nifi.content.repository.directory.conten2=`
	Warning:
	You may experience data loss if content repositories are not accessible to new NiFi.
	You may experience data loss may if property names are wrong or the property points to wrong content repository.

	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.
	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.
	For example:
	`nifi.provenance.repository.directory.provenance1=`
	`nifi.provenance.repository.directory.provenance2=`
	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.

Migrating dataflows 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 Regsitry includes installing the new version of NiFi Registry and updating the configuration files.

Review the upgrade steps before you begin the upgrade process.

Installing 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. Download the latest version of Apache NiFi Registry from the *Download locations* documentation.
2. 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/nifiregistry/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/.

What to do next

Once you are done installing the new version of NiFi Registry, proceed to *Updating the configuration files*.

Related Information

Updating the configuration files
Download locations

Updating 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.

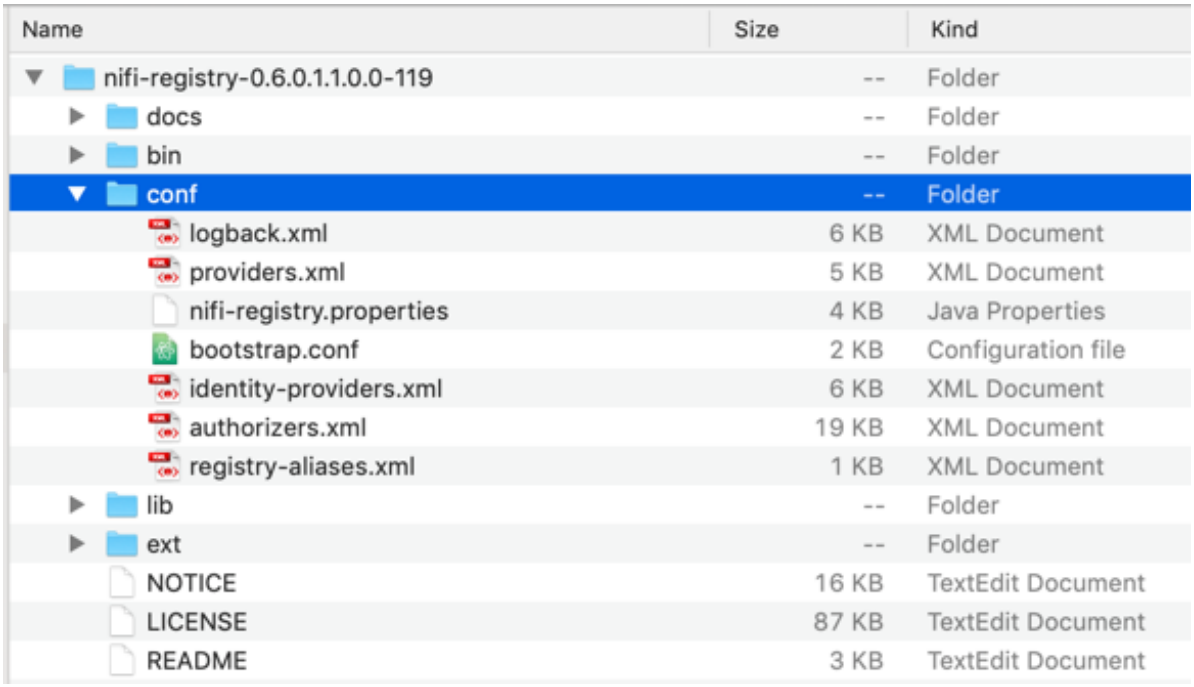
About this task

To complete 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.



   c) Paste the users.xml and authorizations.xmlfiles.

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