Apache NiFi 3

Installing and Upgrading NiFi

Date of Publish: 2018-08-13

http://docs.hortonworks.com

Contents

NiFi Installation

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

Installing NiFi on Linux

To install NiFi on a Linux operating system:

1. Extract the NiFi files in the repo location listed in the *HDF Release Notes* to the location from which you want to run the application.

Installing NiFi on Linux

To install NiFi on a Linux operating system:

Procedure

1. Extract the NiFi files in the repo location listed in the *HDF Release Notes* to the location from which you want to run the application.

2. Proceed to the instructions for starting and stopping NiFi.

Related Information
HDF Release Notes

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
```

Installing NiFi on Windows

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 files in the repo location at http://public-repo-1.hortonworks.com/HDF/windows/3.x/updates/3.1.0.0/nifi-3.1.0.0-564.msi to the location from which you want to run the application.
2. Install NiFi to C:/nifi or some root folder.

Configuring the NiFi MSI
The MSI adds the Windows service for NiFi. The service is configured to be executed by either a local user in the computer, or a domain user in ActiveDirectory.

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 HDF 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.

Procedure

1. If the computer is a part of a domain, then **Local User** checkbox appears in the **HDF NiFi setup** window. Check the **Local User** checkbox to specify that Local user is used to execute 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

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 HDF NiFi setup window.

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 permission to 'Log on as a service.'

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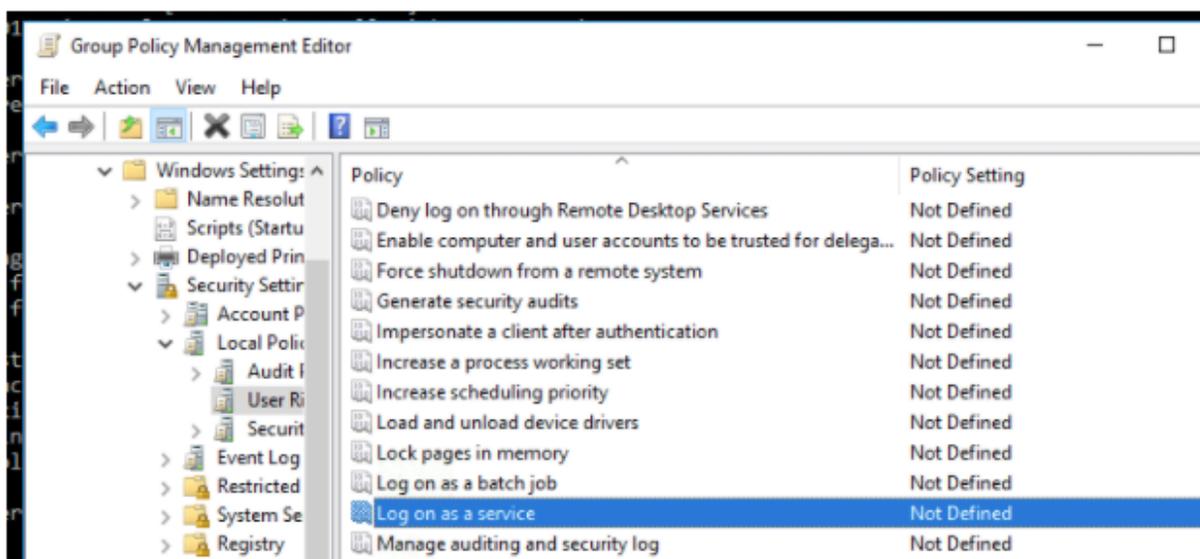
2. Navigate to a machine on which NiFi is installed and enter the following command:

   ```
   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 HDF NiFi setup window, uncheck 'Local User' checkbox then click **Install**.

   HDF_NiFi_setup.png

HDF NiFi setup

HDF NiFi package directory

C:\nifi   Browse

Java command

java

java.arg.2

-Xms512m

java.arg.3

-Xmx512m

nifi.database.directory

./database_repository

nifi.content.repository.directory.default

./content_repository

nifi.flowfile.repository.directory

./flowfile_repository

nifi.provenance.repository.directory.default

./provenance_repository

NiFi service username

nifi   ☑ Local User

NiFi service password

●●●●●●●●●●●●   ☐ Show

Install   Cancel   Reset

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.

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.

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:
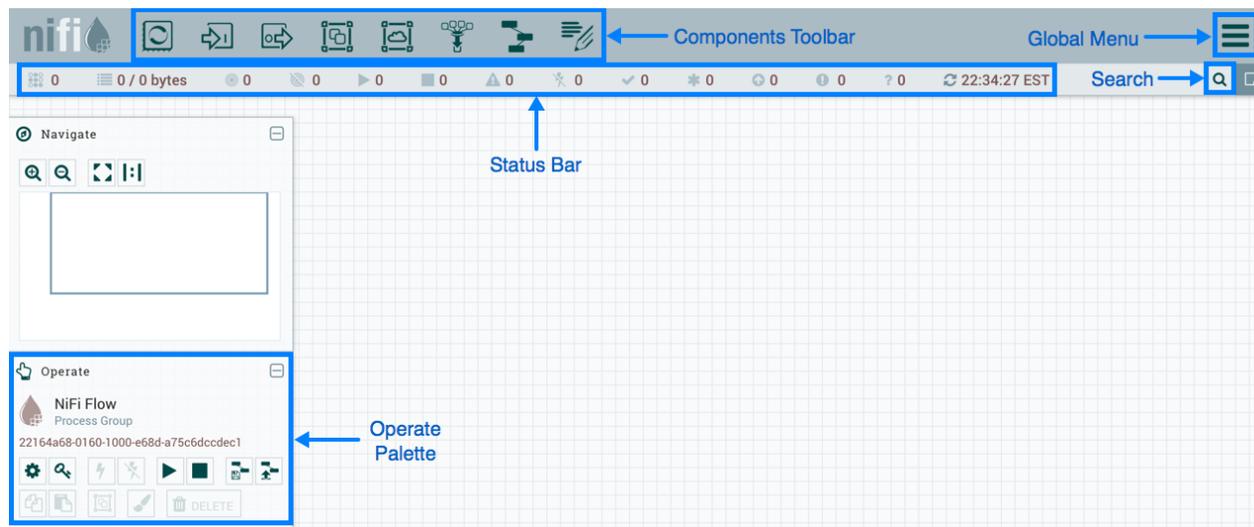
Docker Installation

You can use Docker to install NiFi to utilize the resource efficiency of a container. The NiFi Docker installation provides a Docker image and associated scripts for configuring an instance of Apache NiFi.

Note:

You can install a single instance of NiFi using the Docker image. Running containerized NIFI in a clustered mode is not supported.

Importing Docker

The first step to use the NiFi Docker installation is to import Docker. Enter the following command to import Docker:

```
docker import [OPTIONS] file|URL|- [REPOSITORY[:TAG]]
```

You can use the following options:

--change , -c	Apply Dockerfile instruction to the created image
--message , -m	Set commit message for imported image

Downloading NiFi using Docker

The second step to use the NiFi Docker installation is to download the NiFi image from the Docker hub.

For the location of the Docker hub, see the *HDF Release Notes*.

```
docker pull hortonworks/<hdf-nifi-image-name>
```

Related Information
HDF Release Notes

Running a Docker Container

You can run a Docker container in three environments:

Standalone Instance, Unsecured

The standalone unsecured instance runs a NiFi instance by itself. This is the minimum configuration needed to run a NiFi instance.

```
                        docker run --name nifi \
    -p 18080:8080 \
    -d \
    apache/nifi:latest
```

This will provide a running instance, exposing the instance UI to the host system on port 18080, viewable at http://localhost:18080/nifi.

Standalone Instance, Two-Way SSL

Use this configuration to run NiFi in a standalone instance with a transport layer security protocol (SSL).

In this configuration, you must provide the following:

- Certificates and the associated configuration information.

    Note:

    The AUTH environment variable is set to tls.
- A DN as provided by an accessing client certificate in the INITIAL_ADMIN_IDENTITY environment variable.

    This value will be used to seed the instance with an initial user with administrative privileges.
- A volume to provide certificates on the host system to the container instance.

```
                        docker run --name nifi \
    -v /User/dreynolds/certs/localhost:/opt/certs \
    -p 18443:8443 \
    -e AUTH=tls \
    -e KEYSTORE_PATH=/opt/certs/keystore.jks \
    -e KEYSTORE_TYPE=JKS \
    -e KEYSTORE_PASSWORD=QKZv1hSWAFQYZ+WU1jjF5ank+l4igeOfQRp+OSbkkrs \
    -e TRUSTSTORE_PATH=/opt/certs/truststore.jks \
    -e TRUSTSTORE_PASSWORD=rHkWR1gDNW3R9hgbeRsT3OM3Ue0zwGtQqcFKJD2EXWE \
    -e TRUSTSTORE_TYPE=JKS \
    -e INITIAL_ADMIN_IDENTITY='CN=Random User, O=Apache, OU=NiFi, C=US' \
    -d \
    apache/nifi:latest
```

Standalone Instance, LDAP

Use this configuration to run NiFi in a standalone instance with LDAP.

In this configuration, you must to provide the following:

U

- Certificates and the associated configuration information.

  Optionally, if the LDAP provider of interest is operating in LDAPS or START_TLS modes, you will need to provide certificates.

  > **Note:**
  >
  > The AUTH environment variable is set to ldap.

- A DN as provided by the configured LDAP server in the INITIAL_ADMIN_IDENTITY environment variable.

  This value will be used to seed the instance with an initial user with administrative privileges.

- A volume to provide certificates on the host system to the container instance.

Enter the following for a minimal connection to an LDAP server using SIMPLE authentication:

```
                    docker run --name nifi \
 -v /User/dreynolds/certs/localhost:/opt/certs \
 -p 18443:8443 \
 -e AUTH=tls \
 -e KEYSTORE_PATH=/opt/certs/keystore.jks \
 -e KEYSTORE_TYPE=JKS \
 -e KEYSTORE_PASSWORD=QKZv1hSWAFQYZ+WU1jjF5ank+l4igeOfQRp+OSbkkrs \
 -e TRUSTSTORE_PATH=/opt/certs/truststore.jks \
 -e TRUSTSTORE_PASSWORD=rHkWR1gDNW3R9hgbeRsT3OM3Ue0zwGtQqcFKJD2EXWE \
 -e TRUSTSTORE_TYPE=JKS \
 -e INITIAL_ADMIN_IDENTITY='cn=admin,dc=example,dc=org' \
 -e LDAP_AUTHENTICATION_STRATEGY='SIMPLE' \
 -e LDAP_MANAGER_DN='cn=admin,dc=example,dc=org' \
 -e LDAP_MANAGER_PASSWORD='password' \
 -e LDAP_USER_SEARCH_BASE='dc=example,dc=org' \
 -e LDAP_USER_SEARCH_FILTER='cn={0}' \
 -e LDAP_IDENTITY_STRATEGY='USE_DN' \
 -e LDAP_URL='ldap://ldap:389' \
 -d \
 apache/nifi:latest
```

You can add the following, optional environment variables to the above command when connecting to a secure LDAP server configured with START_TLS or LDAPS:

```
                    -e LDAP_TLS_KEYSTORE: ''
 -e LDAP_TLS_KEYSTORE_PASSWORD: ''
 -e LDAP_TLS_KEYSTORE_TYPE: ''
 -e LDAP_TLS_TRUSTSTORE: ''
 -e LDAP_TLS_TRUSTSTORE_PASSWORD: ''
 -e LDAP_TLS_TRUSTSTORE_TYPE: ''
```

Configuration Information

The following ports are specified by the Docker container for NiFi operation within the container and can be published to the host:

Function	Property	Port
HTTP Port	nifi.web.http.port	8080
HTTPS Port	nifi.web.https.port	8443
Remote Input Socket Port	nifi.remote.input.socket.port	10000

Upgrading NiFi

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.

Preserve Customizations Prior to 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. You can find the latest NiFi download location in the *HDF Release Notes*.

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-1.0.0.2.0.2.0-17/, install your new NiFi version in a path similar to /opt/nifi/nifi-1.2.0.3.0.2.0-76/.

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

HDF Release Notes

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	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.
	Hortonworks 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.

Configuration file	Necessary changes
	If you are upgrading to HDF 3.0.x, 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 he 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.
	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. **Attention:** 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:
	
	<pre>nifi.content.repository.directory.content1=
	nifi.content.repository.directory.conten2=</pre>
	Attention:
	
	You may experience data loss if content repositories are not accessible to new NiFi.
	
	Attention:
	
	You may experience data loss may if property names are wrong or the property points to wrong content repository.

Configuration file	Necessary changes
	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.

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.