

Hortonworks DataFlow

Installing NiFi

(February 28, 2018)

Hortonworks DataFlow: Installing NiFi

Copyright © 2012-2018 Hortonworks, Inc. Some rights reserved.



Except where otherwise noted, this document is licensed under
Creative Commons Attribution ShareAlike 4.0 License.
<http://creativecommons.org/licenses/by-sa/4.0/legalcode>

Table of Contents

1. NiFi Installation	1
1.1. Installing NiFi on Linux	1
1.1.1. Installing NiFi as a Service	1
1.1.2. Starting and Stopping NiFi	1
1.2. Installing NiFi on Windows	2
1.2.1. Configuring the NiFi MSI	3
1.2.2. Starting and Stopping NiFi	5
1.3. Launching the User Interface	6
2. Docker Installation	7
2.1. Importing Docker	7
2.2. Downloading NiFi using Docker	7
2.3. Running a Docker Container	7
2.3.1. Standalone Instance, Unsecured	7
2.3.2. Standalone Instance, Two-Way SSL	8
2.3.3. Standalone Instance, LDAP	8
2.4. Configuration Information	9

List of Figures

1.1. Log_on_as_service.png 4
1.2. HDF_NiFi_setup.png 5

1. 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 \[1\]](#)
- [Installing NiFi on Windows \[2\]](#)
- [Launching the User Interface \[6\]](#)

1.1. Installing NiFi on Linux

To install NiFi on a Linux operating system:

1. Extract the NiFi files in the repo location listed in the [Release Notes](#) to the location from which you want to run the application.

1.1.1. Installing NiFi as a Service

You can also install NiFi as a service.

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

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

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
```

1.1.2. Starting and Stopping NiFi

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
```

1.2. Installing NiFi on Windows



Note

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.

Prerequisites

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.

Steps

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.

1.2.1. Configuring the NiFi MSI

The MSI adds the Windows service for MiNiFi. 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 \[3\]](#)
- [Using a Domain User for NiFi Windows Service \[4\]](#)

1.2.1.1. Using a Local User for NiFi Windows Service

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

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

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: [masked] Show

[Install] [Cancel] [Reset]

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.

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

- SeCreateSymbolicLinkPrivilege
- SeServiceLogonRight

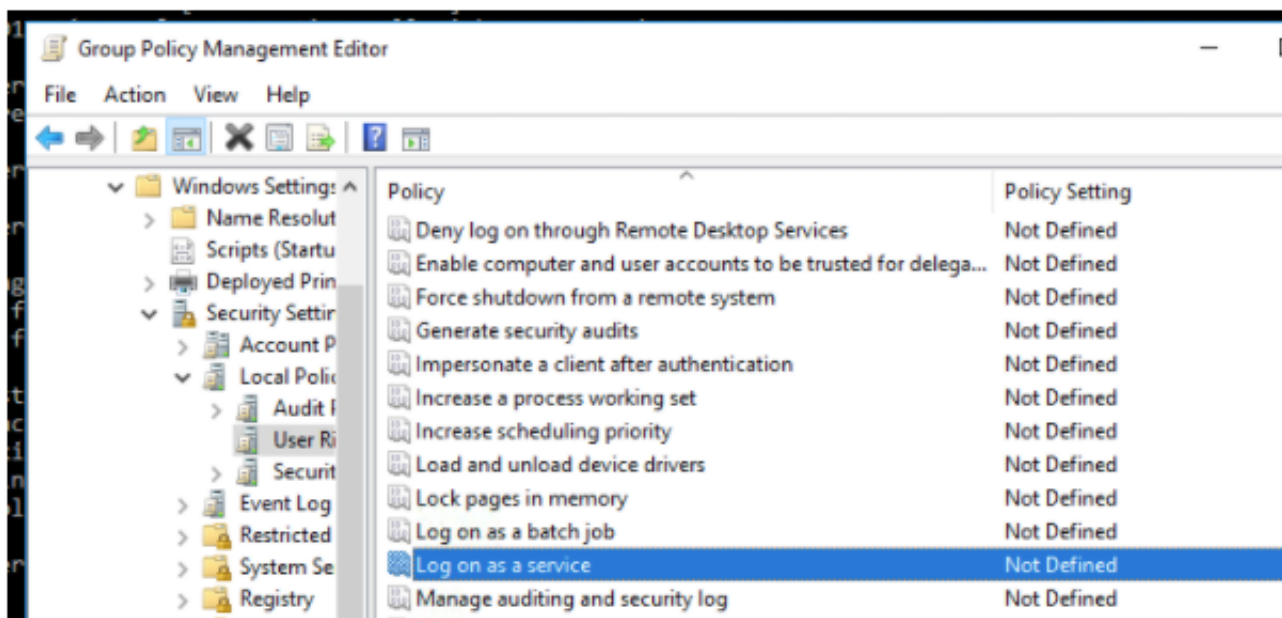
1.2.1.2. Using a Domain User for NiFi Windows Service

Prerequisites

- 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.

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

Figure 1.1. 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**.

Figure 1.2. HDF_NiFi_setup.png

HDF NiFi setup

HDF NiFi package directory
C:\nifi

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

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.

1.2.2. Starting and Stopping NiFi

Once you have downloaded and installed NiFi, you can start NiFi in the foreground or as a service on Windows.

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.

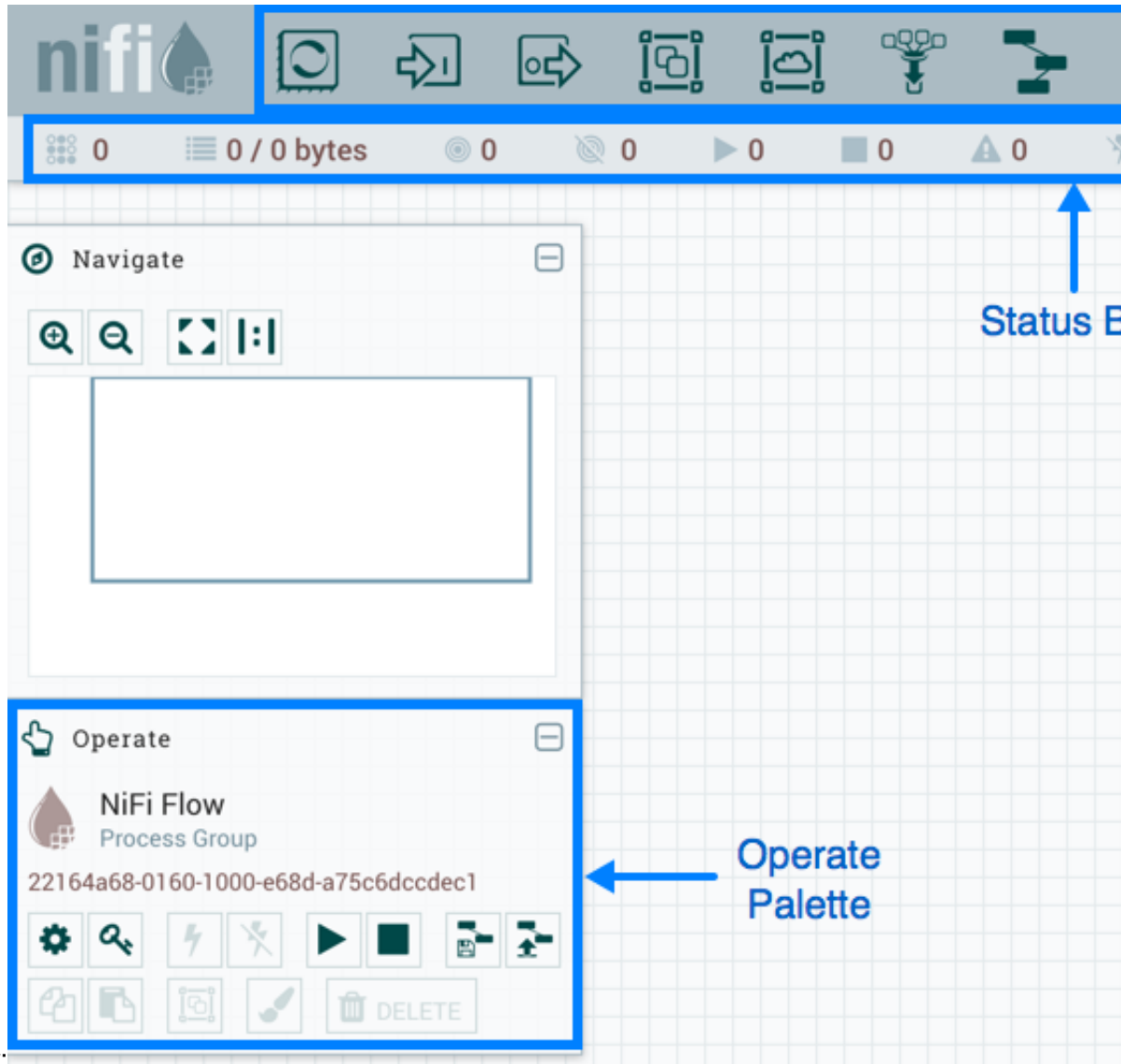
- 2. Hold the `Ctrl` key while pressing `C`.

1.3. 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:

2. 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.

2.1. 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

2.2. 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 [HDF Repository Locations](#).

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

2.3. Running a Docker Container

You can run a Docker container in three environments:

- [Standalone Instance, Unsecured \[7\]](#)
- [Standalone Instance, Two-Way SSL \[8\]](#)
- [Standalone Instance, LDAP \[8\]](#)

2.3.1. 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`.

2.3.2. 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
```

2.3.3. Standalone Instance, LDAP

Use this configuration to run NiFi in a standalone instance with a lightweight access security protocol (LDAP).

In this configuration, you must provide the following:

- 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: ''
```

2.4. 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