**Introduction**

This tutorial provides basic concepts of the Apache NiFi. It was designed for beginners as well as professionals who want to learn the basics and programming concepts of Apache NiFi.

**What is Nifi?**

**Apache NiFi** is an open-source data ingestion platform. It is a powerful and reliable system used for processing and distributing data between different systems. It helps to manage and automate the flow of data between the systems. NiFi originally stands for **Niagara Files (NiFi)** that was developed by **NSA** but is now maintained by **Apache foundation** for further updates. It provides web-based User Interface (UI), which uses HTTPS protocol to run NiFi on a web browser that makes user interaction secure with NiFi.

The web-based UI platform has a main canvas, where we need to define the ***Source***, ***Processor***, and ***Destination***for the collection of data, transmission of data, and data storage, respectively. Each *Processor* in NiFi has some relationships (such as success, failed, retry, or invalid data, etc.), which are used while connecting one processor to another.

**Features of Nifi**

Apache NiFi helps to manage and automate the flow of data between the systems. It can easily manage the data transfer between source and destination systems. It can be described as data logistics. Apache NiFi helps to move and track data similar to the parcel services as how data move and track. It provides **web-based User Interface** (UI) to manage data in real-time.

Apache NiFi is an open-source, therefore it is freely available. It supports various data formats (such as logs, social feeds, and geographical location data, etc.) and protocols (such as KAFKA, SFTP, and HDFS, etc.). It Supports of a wide variety of protocols which make this platform more popular in IT industry.

Below are some reasons given that why use Apache NiFi:

* NiFi allows you to pull the data from various sources into NiFi and create flow files.
* It allows you to use the existing libraries and Java ecosystem functionality.
* It guarantees that the data must be delivered to the destination.
* NiFi helps to fetch, aggregate, split, transform, listen, route, and drag & drop dataflow.
* It visualizes the dataflow at an enterprise level.
* NiFi can easily install on AWS (Amazon Web Service).
* It allows us to start and stop components individually as well as group level.

Apache NiFi supports the directed graph of data routing, system mediation, and transformation. There are some reasons why NiFi came up is because of the data challenges we have. NiFi has a list of data challenges that are the features of NiFi. So, the various features of NiFi are described below:

**Web-based UI**

NiFi offers **web-based User Interface (UI)** that can run over HTTPS, which makes user interaction secure with NiFi. It also manages the data in real-time. NiFi provides experience with design, control, monitoring, and feedback.

**Guaranteed Delivery**

It is one of the most important and very powerful features of Apache NiFi that the delivery of data is guaranteed to be done. It can be achieved by the effective use of persistent write-ahead log and content repository. They both are designed together in such a way that allows for high transaction rate, copy-on-write, effective load spreading. NiFi is highly configurable.

**Data Provenance or Data Lineage**

NiFi provides a data provenance module for tracking and monitoring data flows from beginning to end. NiFi automatically records, indexes, and makes available the provenance data as objects flow through the system. For supporting compliance, optimization, troubleshooting, and many other scenarios, this information becomes very useful.

**Extensible**

This feature allows you to create your own processor. It enables fast development and effective testing. NiFi supports secure protocols such as SSH, SSL, HTTPS, encrypted content and also provides multi-tenant authorization as well as internal policy management. In NiFi, the number of different connectors is increasing.

* The user can build their custom processor according to the requirement.
* This feature of NiFi offers rapid development and effective testing.

**Visual Command and control**

Dataflows can be quite complex. NiFi has an interactive user interface for the user, capable of visualizing and expressing the dataflows. It allows the visual formation of dataflows and helps to express them visually to reduce the complexity of dataflow. NiFi not only enables the visual formation of dataflows but is performed in real-time. If you make any change in data flow or modify it, that change is immediately reflected. You don’t need to stop the entire flow to make any specific modification.

**Security**

Apache NiFi offers system to system, user to system, and multi-tenant authorization security feature. NiFi uses secure protocols such as SSL, SSH, and HTTPS for security reasons. It also uses other encryption to make data secure.

**Advantages of NiFi**

* Apache NiFi supports SFTP protocol using which it enables the data fetching from remote machines.
* Apache NiFi offers web-based User Interface (UI). So that NiFi can run on a web browser using localhost and port. On a web browser, it uses HTTPS protocol that makes user interaction with NiFi secure.
* It also provides security policies at user level, process group level as well as other modules.
* NiFi supports all the devices that run Java.
* A user can create custom plugins to support different types of data systems, although NiFi already supports around 188 processors.
* It provides real-time control that eases the movement of data between source and destination.

**Disadvantages of NiFi**

* In case of primary node switching, Apache NiFi has a state persistence issue. Because of this issue, sometimes it does not enable the processor to fetch the data from the source.
* While making any change by a user node gets disconnected from the NiFi cluster, and then flow.xml becomes invalid. A node cannot reconnect to the cluster until the administrator manually copies the xml file from the connected node.
* All data are not created similarly.
* It offers SSL and topic level authorization, which may not be sufficient.
* To work with Apache NiFi, you must have good underlying system knowledge.

**Key concepts of NiFi**

While discussing NiFi Architecture, the user must be familiar with the following key terms of Apache NiFi. So, we will discuss fundamental key concepts at a high level. They are related to the idea of ***Flow-Based Programming*** (FBP).

The key components of Apache NiFi architecture are discussed below in detail:

**Flow**

**Flow** is created by connecting two or more different processors. It is used to transfer the data from one data source to another destination data source. This data can be modified if required.

**Processor**

A **Processor** is a java module used to either fetch data from the source system or to be stored in the destination system. There are many other processors available that can be used to add attribute or alter/change the content in the FlowFile. Processor is responsible for sending, receiving, creating, splitting, merging, transforming, routing, and processing flowfiles.

**Connection**

**Connection** is known as a bounded buffer in FBP terms. It is a link between processors that connects the processors. It acts as a queue that holds the data in queue when needed. It allows several processes to interact at different rates.

**Process Group**

A **Process Group** is a set of NiFi flows. It helps the users to manage the flows and keep them in a hierarchical manner. Basically, it is a set of processes and their connections, which can receive data through the input ports and send through the output ports.

**FlowFile**

**FlowFile** is the original data with which meta-information is associated. It represents each object that is moving through the system. The NiFi processor changes to the FlowFile while object moves from source to destination processor. Basically, a FlowFile is created by parts that are **Content** and **Attributes**. Content is the user data and Attributes are the key-value pairs that are attached with user data.

**Data Provenance**

It is a repository, which allows the user to check information about FlowFile. It also enables the troubleshooting if any issue occurs while processing FlowFile.

**NiFi User Interface**

Once the NiFi has been started successfully, UI will bring up to you to create and monitor the dataflows. NiFi UI provides an interactive interface that can be accessed by the users on a web browser. Users can drag and drop components in NiFi. It provides various type of information about NiFi such as:

* Active threads
* Total queued data
* Transmitting Remote Process Groups
* Not Transmitting Remote Process Groups
* Running components
* Stopped components
* Disabled components
* Invalid components
* Stale Versioned process Groups
* Up to date Versioned Process Groups
* Sync failure Versioned Process Groups
* Locally modified Versioned Process Groups
* Stale and Locally modified Versioned Process Groups

The below screenshot is User Interface of Apache NiFi:

[A screenshot of a computer

AI-generated content may be incorrect.](http://localhost/wp-content/uploads/2024/04/Nifi-UI-Interface.png)

**Components of Apache NiFi**

There are following components of Apache NiFi, which are listed under the component section of the toolbar

[A screenshot of a computer

AI-generated content may be incorrect.](http://localhost/wp-content/uploads/2024/04/Nifi-UI-Toolbar.png)

**Processor**

Processors are basic blocks that are used for creating a data flow. Apache NiFi has several processors, where each processor has different functionality. Users can drag and drop the processor icon on the canvas to add the processor and then select the desired processor to create the dataflow.

[A logo with a circle in the middle

AI-generated content may be incorrect.](http://localhost/wp-content/uploads/2024/04/nifi-processor.png)

Drag the process icon on the canvas that will open an **Add Processor** window. Choose the desired processor you needed for the data flow in Apache NiFi.

[A screenshot of a computer

AI-generated content may be incorrect.](http://localhost/wp-content/uploads/2024/04/Capture-decran-2024-04-23-205206-1.png)

To know more about the processor, right-click on it and go to **Usage**. This will bring up the documentation of the processor. It provides information like what a processor does, properties that need to be configured, and the relationship for the p

**Input Port**

An Input Port is used for getting data from the processor, which is not present in that Process Group. The input port can be dragged on the canvas by clicking on the icon given below.

[A blue and black logo

AI-generated content may be incorrect.](http://localhost/wp-content/uploads/2024/04/nifi-input-port.png)

To add an input port to any data flow, drag the icon on canvas.

After dragging the icon onto the canvas, NiFi asks you to enter the name for the input port. Provide the name of the input port and click on the Add button.

[A screenshot of a computer

AI-generated content may be incorrect.](http://localhost/wp-content/uploads/2024/04/nifi-add-input-port.png)

**Output Port**

An Output Port is used for transferring data to the processor, which is not present in that process group. The output port can be dragged on the canvas by clicking on the icon given below.

To add the output port to any data flow, drag the icon on canvas.

[A blue square with a black arrow

AI-generated content may be incorrect.](http://localhost/wp-content/uploads/2024/04/nifi-output-port.png)

After dragging this icon onto the canvas, NiFi pops up a screen to enter the name for the output port. Provide the name of the output port and click on the Add button.

[A screenshot of a computer

AI-generated content may be incorrect.](http://localhost/wp-content/uploads/2024/04/nifi-add-output-port.png)

**Process Group**

The below icon is used to add the process group in NiFi canvas.

[A logo of a company

AI-generated content may be incorrect.](http://localhost/wp-content/uploads/2024/04/nifi-process-group.png)

After dragging this icon to the canvas, NiFi pops up a screen where enter the name of the Process Group and then add it to the NiFi canvas by clicking the Add button.

[A screenshot of a computer

AI-generated content may be incorrect.](http://localhost/wp-content/uploads/2024/04/nifi-add-process-group.png)

**Remote Process Group**

The below icon is used to add the Remote Process Group in NiFi canvas.

[A logo of a cloud

AI-generated content may be incorrect.](http://localhost/wp-content/uploads/2024/04/nifi-remote-process-group.png)

**Template**

Templates are used to export/import the data flow in same or different NiFi instances. The icon that is given below is used to add the template onto the NiFi canvas.

[A close-up of a diagram

AI-generated content may be incorrect.](http://localhost/wp-content/uploads/2024/04/nifi-template.png)

After dragging the Template icon, the user can choose the template already added in NiFi.

**Funnel**

Funnel helps to transfer the output of a processor to several other processors. With the help of below icon, a user can add the funnel to a data flow.

[A blue and black logo

AI-generated content may be incorrect.](http://localhost/wp-content/uploads/2024/04/nifi-funnel.png)

**Label**

Labels are used to add the text about any components that exist in NiFi. It also provides various color options. The developers can change label color as well as size of the text. They can use these colors to add an aesthetic sense.

[A pencil and writing symbol

AI-generated content may be incorrect.](http://localhost/wp-content/uploads/2024/04/nifi-label.png)

In the top menu of NiFi UI, first rightmost icon is used to add label in NiFi canvas.

**Summary**

In this tutorial we presented what Apache NiFi. It is an incredibly powerful tool for data integration and processing, allowing you to streamline and automate complex data flows with ease. In this tutorial, we covered the key features of Apache NiFi, the Nifi user interface and the Nifi components that you will need to get started and create your dataflows.